International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

# AI Based Energy Management System for Smart Grid with RES Integration

**Dr.S.Saravanan, Mrs.M.Selvakumari, Gokulnath V, Harikrishnan C, Manikandan A, Manojkumar P**

Department of Electrical and Electronics Engineering, Muthayammal Engineering College (Autonomous), Rasipuram, Tamil Nadu, India

**Publication History:** Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03. 2026; Published: 28.03.2026.

**ABSTRACT:** Network security applications generally require the ability to perform powerful pattern matching to protect against attacks such as viruses and spam. Traditional solutions are intended for firewall routers. However, the solutions in the literature for firewalls are not scalable, and they do not address the difficulty of an antivirus with an ever-larger pattern set. The goal of this work is to provide systematic virus detection processor for network security. Instead of placing entire matching patterns on a chip, our solution is a two-phase dictionary-based antivirus processor that works by condensing as much of the important filtering information as possible onto a chip and infrequently accessing off-chip data to make the matching mechanism scalable to large pattern sets. In the first stage, the filtering engine can filter out more than 90% of data as safe, using a merged shift table. Only 10% or less of potentially unsafe data must be precisely checked in the second stage by the exact-matching engine from off-chip memory. To reduce the impact of the memory gap, we also propose three enhancement algorithms to improve performance 1) a skipping algorithm 2) a cache method and 3) a prefetching mechanism.

**KEYWORDS:** Network security, memory gap, algorithmic attacks, virus detection.

## I. INTRODUCTION

Network security has always been important issue. End users are vulnerable to virus attacks, spam, and Trojan horses, for example. They may visit malicious websites or hackers may gain entry to their computers and use them as zombie computers to attack others. To ensure a secure network environment, firewalls were first introduced to block unauthorized Internet users from accessing resources in a private network by simply checking the packet head (MAC address/IP (address/port number). This method significantly reduces the probability of being attacked. However, attacks such as spam, spyware, worms, viruses, and phishing target the application layer rather than the network layer. Therefore, traditional firewalls no longer provide enough protection. Many solutions, such as virus scanners, spam-mail filters, instant messaging protectors, network shields, content filters, and peer-to-peer protectors, have been effectively implemented. Initially, these solutions were implemented at the end-user side but tend to be merged into routers/firewalls to provide multi-layered protection.

As a result, these routers stop threats on the network edge and keep them out of corporate networks. When a new connection is established, the firewall router scans the connection and.

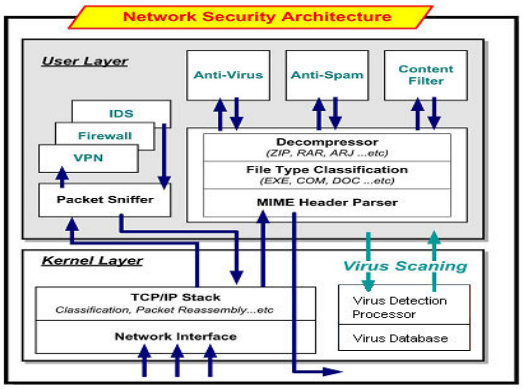



fig. 1Architecture of firewall router

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

forwards these packets to the host after confirming that the connection is secure. Because firewall routers focus on the application layer of the OSI model, they must reassemble incoming packets to restore the original connection and examine them through different application parsers to guarantee a secure network environment. For instance, suppose a user searches for information on web pages and then tries to download a compressed file from a web server. In this case, the firewall router might initially deny some connections from the firewall based on the target's IP address and the connection port. Then, the firewall router would monitor the content of the web pages to prevent the user from accessing any page that connects to malware links or inappropriate content, based on content filters.

When the user wants to download a compressed file, to ensure that the file is not infected, the firewall router must decompress this file and check it using anti-virus programs. In summary, firewall routers require several time-consuming steps to provide a secure connection. However, even under numerous security constrictions, firewall routers are still required to provide high-speed transmission. Fortunately, most security-guaranteed programs use rule-based designs. Therefore, we have tried to develop a pattern matchingprocessor to accelerate the detection speed. The purpose of pattern matching is to check whether a text contains at least one of a given set of patterns.

There are many algorithms [1-4] and accompanying hardware accelerators for fast pattern matching [8] [9]. One of the typical algorithms is the automation approach. This approach is based on Aho and Corasick's algorithm (AC) [1] which introduces a linear-time algorithm for multi-pattern search with a large finite-state machine. Its performance is not affected by the size of a given pattern set (the sum of all pattern lengths), but it requires a significant amount of memory due to state explosion.

## II. VIRUS DETECTION PROCESSOR

A two-phase pattern-matching architecture mostly comprising the filtering engine and the exact-matching engine. The filtering engine is a front-end module responsible for filtering out secure data efficiently and indicating to candidate positions that patterns possibly exist at the first stage. The exact-matching engine is a back-end module responsible for verifying the alarms caused by the filtering engine. Only a few unsaved data need to be checked precisely by the exact-matching engine in the second stage. Both engines have individual memories for storing significant information. In this case, we used a 32-kB on-chip memory for the ClamAV virus database, which contained more than 30 000 virus codes and localized most of the computing inside the chip. Conversely the exact-matching engine not only stores the entire pattern in external memory but also provides information to speed up the matching process. Our exact-matching engine is space-efficient and requires only four times the memory space of the original size pattern set. The size of a pattern set is the sum of the pattern length for each pattern in the given pattern set; in other words, it is the minimum size of the memory required to store the pattern set for the exact-matching engine.In this case, 8 MB of off-chip memory was required for the ClamAV virus database (2 MB). The proposed exact-matching engine also supports data prefetching and caching techniques to hide the access latency of the off-chip memory by allocating its data structure well. The other modules include a text buffer and a text pump that prefetches text in streaming method to overlap the matching progress and text reading. A load/store interface was used to support bandwidth sharing. This proposed architecture has six steps shown in Figure for finding patterns.The filtering engine fetches a piece of text from the text buffer according to the pattern pointer and checks it by a shift-signature table. If the position indicated by the pattern pointer is not a candidate position, then the filtering engine skips this piece of text and shifts the pattern pointer right multiple characters to continue to check the next position. The shift-signature table combines two data structures used by two different filtering algorithms, the Wu- Manber algorithm and the Bloom filter algorithm, and it provides two-layer filtering. If both layers are missing their

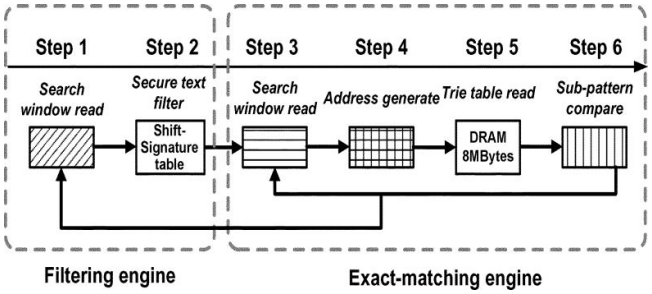



Fig.2 Flow of virus detection processor

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

filter, the processor enters the exact-matching phase. The next section has details about the shift-signature table.
After an alarm caused by the filtering engine, the exact-matching engine precisely verifies this alarm by retrieving a trie structure [5]. This structure divides a pattern into multiple sub-patterns and systematically verifies it. The exact-matching engine generally has four steps for each check. First, the exact-matching engine gets a slice of the text and hashes it to generate the trie address. Then, the exact-matching engine fetches the trie node from memory.

## III. FILTERING ENGINE (FE)

Filtering engine can filter out more than 90% of data as safe, using a merged shift table. It uses three algorithm Wu-Manber algorithm, Bloom filter algorithm, and Shift signature algorithm.

A. Wu-Manber Algorithm

The Wu-Manber algorithm is a high-performance, multi-pattern matching algorithm based on the Boyer-Moore algorithm. It builds three tables in the preprocessing stage: a shift table, a hash table and a prefix table. The Wu-Manber algorithm is an exact-matching algorithm, but its shift table is an efficient filtering structure. The shift table is an extension of the bad-character concept in the Boyer-Moore algorithm, but they are not identical. The matching flow is shown in Figure 3.3. The matching flow matches patterns from the tail of the minimum pattern in the pattern set, and it takes a block B of characters from the text instead of taking them one-by-one. The shift table gives a shift value that skips several characters without comparing after a mismatch. After the shift table finds a candidate position, the Wu-Manber algorithm enters the exact-matching phase and is accelerated by the hash table and the prefix table. The performance of the Wu-Manber algorithm is not proportional to the size of the pattern set directly, but it is strongly dependent on the minimum length of the pattern in the pattern set. The minimum length of the pattern dominates the maximum shift distance (m-B+1) in its shift table. For the pattern set {erst, ever, there} shown in Figure.3, the maximum shift value is three characters for B=2 and m=4. The related shift table, hash table and prefix are shown in Figure 3.

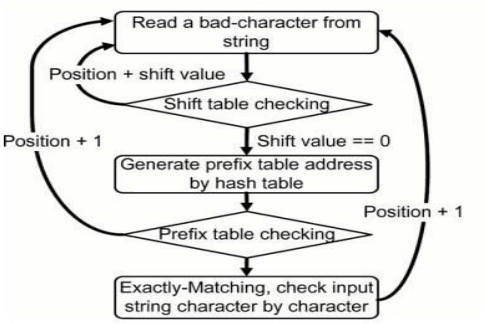



Fig.3 Flow of wu-manber

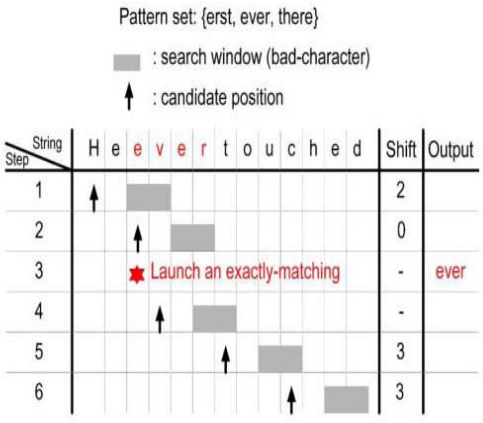



Fig.4 matching flow

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

The Wu-Manber algorithm scans patterns from the head of a text, but it compares the tails of the shortest patterns. In step 1, the arrow indicates to a candidate position that a wanted pattern probably exists, but the search window (gray bar) is actually the character it fetches for comparison. According to shift [ev] =2, the arrow and search window are shifted right by two characters. Then, the Wu-Manber algorithm finds a candidate position in step 2 due to shift[er] =0. Consequently, it checks the prefix table and hash table to perform an exact-matching and then outputs the 'ever' in step 3. After completing the exact match, the Wu-Manber algorithm returns to the shifting phase, and it shifts the search window to the right by one character to find the next candidate position in step 4. The algorithm keeps shifting the search window until touching the end of the string in step 6.

B. Bloom Filter Algorithm

A Bloom filter is a space-efficient data structure used to test whether an element exists in a given set. This algorithm is composed of different hash functions and a long vector of bits. Initially, all bits are set to 0 at the preprocessing stage. To add an element, the Bloom filter hashes the element by these hash functions and gets positions K of its vector. The Bloom filter then sets the bits at these positions to 1. The value of a vector.

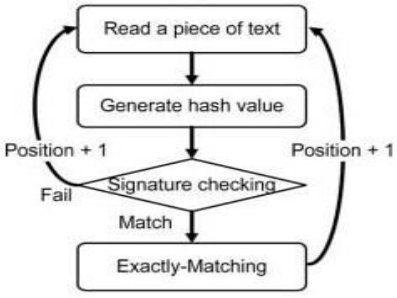



Fig.5 Flow of bloom filter

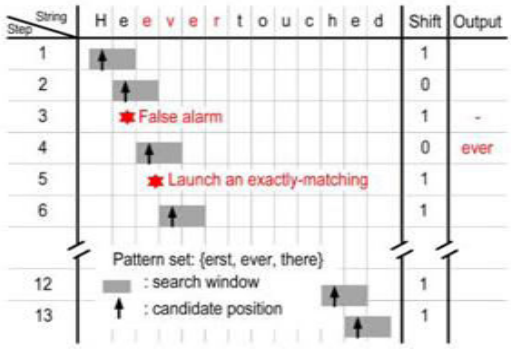



Fig.6 matching flow

that only contains an element is called the signature of an element To check the membership of a particular element, the Bloom filter hashes this element by the same hash functions at run time, and it also generates K positions of the vector. If all of these K bits are set to 1, this query is claimed to be positive, otherwise it is claimed to be negative. The output of the Bloom filter can be a false positive but never a false negative. Therefore, some pattern matching algorithms based on the Bloom filter must operate with an extra exact-matching algorithm. However, the Bloom filter still features the following advantages: 1) it is a space-efficient data structure; 2) the computing time of the Bloom filter is scaled linearly with the number of patterns; and 3) the Bloom filter is independent of its pattern length.

Then, this algorithm checks whether the signature exists in the bit vector. If the answer is yes, it shifts the search window to the right by one character for each comparison and repeats the above step to filter out safe data until it finds a candidate position and launches exact-matching. Figure 3.7 shows how a Bloom filter builds its bit vector for a pattern set {erst, ever, there} for two given hash functions. The filter only hashes all of the pattern prefixes at the preprocessing stage. Multiple patterns setting the same position of the bit vector are allowed. Figure 3.8 shows an example of the matching process. The arrows indicate the candidate positions.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

The gray bars represent the search window that the Bloom filter actually fetches for comparison. Both the candidate position and search window are aligned together.

Thus, the Bloom filter scans and compares patterns from the head rather than the tail, like the Wu-Manber algorithm. In step 1, the filter hashes 'He' and mismatches the signature with the bit vector. The filter then shifts right 1 character and finds the next candidate position. For the search window 'ee', the Bloom filter matches the signature and then causes a false alarm to perform an exact-matching in steps 2 and 3. The filter then returns to the filtering stage and shifts one character to the right in step 4, which launches a true alarm for the pattern 'ever'. Finally, the Bloom filter filters the rest of text and finds nothing.

C. Shift Signature Algorithm

The proposed algorithm re-encodes the shift table to merge the signature table into a new table named the shift-signature table. The shift-signature table has the same size as the original shift table, as its width and length are the same as the original shift table. There are two fields, S-flag and carry, in the shift signature table. The carry field has two types of data: a shift value and a signature. These two data types are used by two different algorithms. Thus, the S-flag is used to indicate the data type of a carry. The filtering engine can then filter the text using a different algorithm while providing a higher filter rate. The method used to merge these two tables is described as follows. First, the algorithm generates two tables, a shift table and signature table, at the preprocessing stage. The generation of the shift table is the same as in the Wu-Manber algorithm. The shift table is used as the primary filter. The signature table could be considered a set of the bit vector of the Bloom filter, and it is used for the second-level filtering. The signature table's generation is similar to the Bloom filter but is not identical; it hashes the tail characters of patterns to generate their signatures instead of the prefix. Generated signatures are mapped onto the signature table and indexed by bad-characters, which have shift values of zero in the shift table. In other words, a pattern is assigned a zero shift value in the shift table by its last characters, and it uses the same index to locate its signature in the signature table. After the shift table and signature table are generated, the algorithm re-encodes the shift value into two fields: an S-flag and a carry in the shift-signature table. The S-flag is a 1-bit field used to indicate the data type of the carry. Two data types, shift value or signature, are defined for a carry. The size and width of the shift-signature table are the same as those of the original shift table. To merge these two tables, the algorithm maps each entry in the shift table and signature table onto the shift-signature table. For the non-zero shift values, the S-flags are set, and their original shift values are cut out at 1-bit to fit their carries. Conversely, for the zero shift values, their S-flags are clear, and their carries are used to store their signatures. In this method, all of the entries in the shift-signature table contribute to the filtering rate at run time. Because of the address collision of bad-characters, most

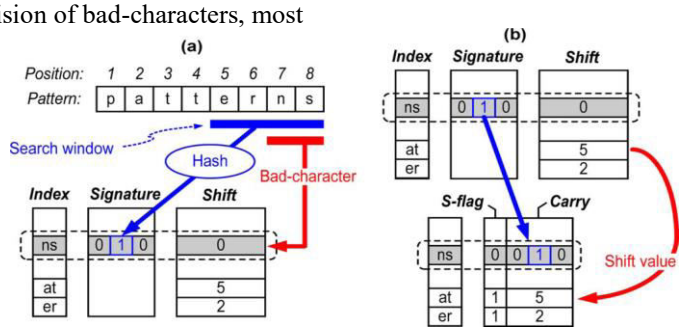



Fig.7 Shift signature algorithm

entries contain less than half of the maximum shift distance for a large pattern set. Therefore, although this method sacrifices the maximum shift distance, the filter rate is not reduced but rather improved.

Fig. 7(a) shows an example of generating the shift and signature tables. Suppose the length of the shortest pattern "patterns" in the pattern set is 8 characters. The size of the bad-character is 2 characters, thus the maximum shift distance is 8-2+1=7characters. Seven possible bad-characters ("pa", "at", "tt", "te", "er", "rn", "ns") are defined according to the Wu-Manber algorithm, and their shift values are 6, 5, 4, 3, 2, 1, and 0. Before replacement, the algorithm first builds the signature table. For each pattern, the algorithm hashes the tail characters of a pattern (blue bar) to generate its signature. The signature is then assigned to the signature table indexed by the bad-character "ns". For multiple signatures mapped to the same entry, the entry stores the results of the OR operation of these signatures. In this work, we only use one hash function because of the space limitation of the signature table. The method of merging the shift table and signature table is shown in Fig. 7(b). The shift[ns] is replaced by its signature ("010" in binary)

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

because its shift value is zero. In contrast, the shift [at] =5 and shift[er] =2 keep their shift values in the shift-signature table.

## IV. EXACT MATCHING ENGINE

The EME must verify the false positives when the filtering engine alerts. It also precisely identifies patterns for upper-layer applications. Most exact-match algorithms use the two kinds of trie structures shown in Fig. 8 loose and compact tries, to establish their pattern databases. Both trie structures have their merits. The AC algorithm uses loose tries, which check each input character in a constant amount of time because of their fan-out states for all possible input characters. Thus, the input data do not affect the AC-based algorithm's performance, but their memory requirements increase exponentially with pattern size. Unlike loose tries, compact tries construct pattern databases with two pointers, sibling and child, to reduce their memory requirements. However, this method has potential performance problems because it may redundantly search link lists formed by sibling pointers. Despite this limitation, compact tries are still highly practical because, in practice, attack texts are not easy to generate.

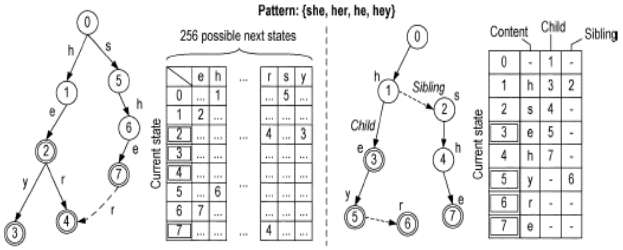
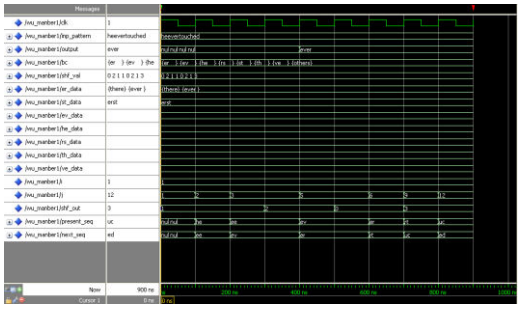



Fig.8 AC algorithm

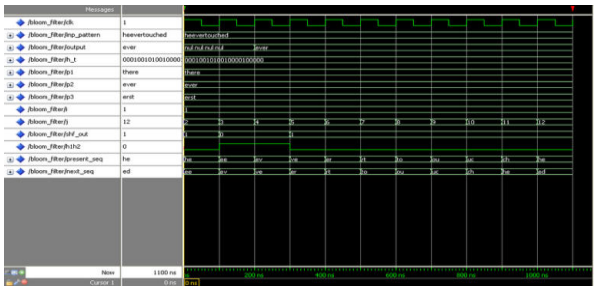
Attacks can be avoided by removing patterns that cause attacks before constructing the pattern database. For this reason, we use compact tries as our exact-matching engine's algorithm.

D. Simulation and Results

1) Wu-manber algorithm



2) Bloom filter algorithm



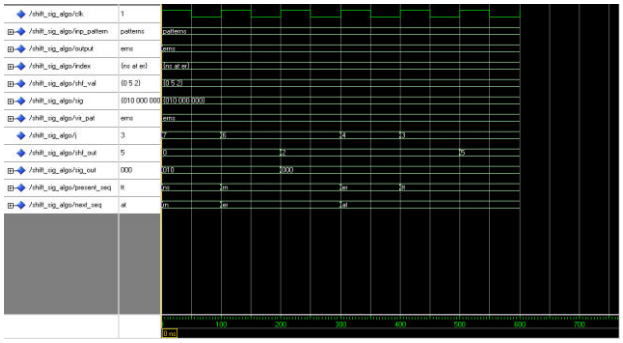
International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

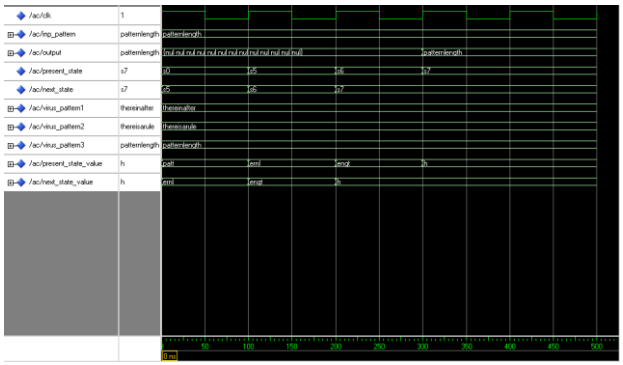
| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

3) Shift signature algorithm



4) AC algorithm



## V. DISCUSSIONS

The result from the shift signature algorithm is send into the exact matching engine algorithm [7]. It will filter the virus which contains alarm. Many previous designs have claimed to provide high performance, but the memory gap [6] created by using external memory decreases performance because of the increasing size of virus databases. Two-phase heuristic algorithms are a solution with a tradeoff between performance and cost due to an efficient filter table existing in internal memory; however, their performance is easily threatened by malicious attacks.

## VI. CONCLUSION

This work analyzes two scenarios of malicious attacks and provides two methods for keeping performance within a reasonable range. First, we re-encoded the shift table to make it provide a bad-character heuristic feature and high filter rates for large pattern sets at the same time. Second, the proposed skip mechanism cushions the blow to performance under algorithmic attacks. The proposed re-encoding stage removes 95.42% of the exact-matching events in the front-end for normal cases, and the skip mechanism eliminates 95.8% of the external memory accesses for attack cases with just 32 kB internal memory and 8 MB external memory.

## REFERENCES

1. A. V. Aho and M. J. Corasick, (1975), 'Efficient string matching: An aid to bibliographic search', Commun. ACM, vol. 18, pp. 333–340.
2. R. S. Boyer and J. S. Moore, (1977), 'A fast string searching algorithm', Commun. ACM, vol. 20, pp. 762–772.
3. B. H. Bloom, (1970), 'Space/time trade-offs in hash coding with allowable errors', Commun. ACM, vol. 13, pp. 422–426.
4. S. Dharmapurikar, P. Krishnamurthy, and T. S. Sproull, (2004), ' Deep packet inspection using parallel bloom filters', IEEE Micro, vol. 24, no. 1, pp.52–61.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

5. C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor &Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
6. S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
7. E. Fredkin, (1960), 'Trie memory', Commun. ACM, vol. 3, pp. 490–499.
8. Y. H. Cho and W. H. Mangione-Smith, (2005), 'A pattern matching coprocessor for network security', presented at the 42nd Annu. Des. Autom. Conf., Anaheim, CA
9. S. Wu and U. Manber, (1994) 'A fast algorithm for multi-pattern searching', Univ. Arizona, Tucson, Report TR-94-17.
10. K.Prakashraj, G.Vijayakumar, S.Saravanan and S.Saranraj, "IoT Based Energy Monitoring and Management System for Smart Home Using Renewable Energy Resources," International Research Journal of Engineering and Technology, Vol.7, Issue 2, pp.1790-1797, 2020.
11. J Mohammed siddi, A. Senthil kumar, S.Saravanan, M. Swathisriranjani, "Hybrid Renewable Energy Sources for Power Quality Improvement with Intelligent Controller," International Research Journal of Engineering and Technology, Vol.7, Issue 2, pp.1782-1789, 2020.
12. T.R. Vignesh, M.Swathisriranjani, R.Sundar, S.Saravanan, T.Thenmozhi," Controller for Charging Electric Vehicles Using Solar Energy", Journal of Engineering Research and Application, vol.10, Issue.01,pp.49-53, 2020.
13. G. Poovarasan, S. Susikumar, S. Naveen, N. Mohananthini, S. Saravanan," Study of Poultry Fodder Passing Through Trolley in Feeder Box," International Journal of Engineering Technology Research & Management, vol.4, Issue.1, pp.76-83, 2020.
14. M.Revathi, S.Saravanan, R.Raja, P.Manikandan," A Multiport System for A Battery Storage System Based on Modified Converter with MANFIS Algorithm," International Journal of Engineering Technology Research & Management, vol.4, issue 2, pp.217-222, 2020.
15. D Boopathi, S Saravanan, Kaliannan Jagatheesan, B Anand, "Performance estimation of frequency regulation for a micro-grid power system using PSO-PID controller", International Journal of Applied Evolutionary Computation (IJAEC), Vol.12, Issue.4, pp.36-49, 2021.
16. V Kumarakrishnan, G Vijayakumar, D Boopathi, K Jagatheesan, S Saravanan, B Anand," Frequency regulation of interconnected power generating system using ant colony optimization technique tuned PID controller", Control and Measurement Applications for Smart Grid: Select Proceedings of SGESC 2021, pp..129-141.
17. G Vijayakumar, M Sujith, S Saravanan, Dipesh B Pardeshi, MA Inayathullaa," An optimized MPPT method for PV system with fast convergence under rapidly changing of irradiation", 2022 International Virtual Conference on Power Engineering Computing and Control: Developments in Electric Vehicles and Energy Sector for Sustainable Future (PECCON), pp.1-4.
18. VM Geetha, S Saravanan, M Swathisriranjani, CS Satheesh, S Saranraj, "Partial Power Processing Based Bidirectional Converter for Electric Vehicle Fast Charging Stations", Journal of Physics: Conference Series, Vol.2325, Issue.1, pp.012028, 2022.
19. M Santhosh Kumar, G Dineshkumar, S Saravanan, M Swathisriranjani, M Selvakumari, "Converter Design and Control of Grid Connected Hybrid Renewable Energy System Using Neuro Fuzzy Logic Model", 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), pp.1-6, 2022.
20. C Gnanavel, A Johny Renoald, S Saravanan, K Vanchinathan, P Sathishkhanna, "An Experimental Investigation of Fuzzy-Based Voltage-Lift Multilevel Inverter Using Solar Photovoltaic Application", Smart Grids and Green Energy Systems, pp.59-74, 2022.
21. V Kumarakrishnan, G Vijayakumar, D Boopathi, K Jagatheesan, S Saravanan, B Anand, "Optimized PSO technique based PID controller for load frequency control of single area power system", Solid State Technology, Vol.63. Issue.5, pp.7979-7990, 2020.
22. G. Poovarasan, S. Susikumar, S. Naveen, N. Mohananthini, S. Saravanan, "Implementation of IoT Based Poultry Feeder Box", International Journal of Innovative Research In Technology, Vol.6, Issue.2, pp.33-38, 2020.
23. N.Gokulnath, B.Jasim Khan, S.Kumaravel, Dr.A.Senthil Kumar and Dr.S.Saravanan, "Soldier Health and Position Tracking System", International Journal of Innovative Research In Technology, Vol-6 Issues 12, pp.39-45, 2020.
24. P.Navaneetha, R.Ramiya Devi, S.Vennila, P.Manikandan and Dr.S.Saravanan, " IOT Based Crop Protection System against Birds and Wild Animal Attacks", International Journal of Innovative Research In Technology, Vol-6 Issues 11, pp.133-143, 2020.
25. K. Punitha, M. Rajkumar, S. Karthick and Dr. S. Saravanan, " Impact of Solar And Wind Integration on Frequency Control System", International Research Journal of Engineering and Technology, Vol 7 Issue 3, pp.1357-1362,2020.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

26. A.Arulkumar, S.Balaji, M.Balakrishnan, G.Dineshkumar and S.Saravanan, "Design And Implementation of Low Cost Automatic Wall Painting Machine" International Journal of Engineering Technology Research & Management, Vol-4 Issues 03, pp.170-176, 2020.
27. V.Periyasamy, S.Surya, K. Vasanth, Dr.G.Vijayakumar and Dr.S.Saravanan, "Design And Implementation of Iot Based Modern Weaving Loom Monitoring System" International Journal of Engineering Technology Research & Management, Vol-4 Issues 04, pp.11-18, 2020.
28. M.Yogheshwaran, D.Praveenkumar, S.Pravin, P.M.Manikandan and Dr.S.Saravanan, "IoT Based Intelligent Traffic Control System" International Journal of Engineering Technology Research & Management, Vol-4 Issues 04, pp.59-63, 2020.
29. R.Pradhap, R.Radhakrishnan, P.Vijayakumar, R.Raja and Dr.S.Saravanan, "Solar Powered Hybrid Charging Station For Electrical Vehicle" International Journal of Engineering Technology Research & Management, Vol-4 Issues 04, pp.19-27, 2020.
30. S.Shenbagavalli, T.Priyadharshini, S.Sowntharya, P.Manikandan and Dr.S.Saravanan, "Design and Implementation of Smart Traffic Controlling System" International Journal of Engineering Technology Research & Management, Vol-4 Issues 04, pp.28-36, 2020.
31. M.Pavithra, S.Pavithra, R.Rama Priya, M.Vaishnavee, M.Ranjitha and S.Saravanan, "Fingerprint Based Medical Information System Using IoT" International Journal of Engineering Technology Research & Management, Vol-4 Issues 04, pp.45-51, 2020.
32. A.Ananthan, A.M.Dhanesh, J.Gowtham, R.Dhinesh, G.Jeevitha and Dr.S.Saravanan, "IoT Based Clean Water Supply" International Journal of Engineering Technology Research & Management, Vol-4 Issues 03, pp.154-162, 2020.
33. R.Anbarsan, A.Arsathparvez, K.S.Arunachalam, M.Swathisriranjani and Dr.S.Saravanan, "Automatic Class Room Light Controlling Using Arduino" International Journal of Engineering Technology Research & Management, Vol-4 Issues 03, pp.192-201, 2020.
34. S.Karthikeyan, A.Krishnaraj, P.Magendran, T.Divya and Dr.S.Saravanan , "The Dairy Data Acquisition System" International Journal of Engineering Technology Research & Management, Vol-4 Issues 03, pp.163-169, 2020.
35. M.Amaran, S.Mannar Mannan, M.Madhu, Dr.R.Sagayaraj and Dr. S.Saravanan, "Design And Implementation of Low Cost Solar Based Meat Cutting Machine" International Journal of Engineering Technology Research & Management, Vol-4 Issues 03, pp.202-208, 2020.
36. N.Harish, R.Jayakumar, P.Kalaiyarasan, G.Vijayakumar and S. Saravanan, "IoT Based Smart Home Energy Meter" International Journal of Engineering Technology Research & Management, Vol-4 Issues 03, pp.177-183, 2020.
37. K.Subashchandrabose, G.Moulieshwaran, M.Raghul, V.Dhinesh and S.Saravanan, "Design of Portable Sanitary Napkin Vending Machine", International Journal of Engineering Technology Research & Management, Vol-4 Issues 03, pp.52-58, 2020.
38. D.Hemalatha, S.Indhumathi, V.Myvizhi and S.Saravanan, "Design and Implementation of Intelligent Controller for Domestic Applications", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.4-7, 2023.
39. S. Divyasri, E. Indhu, M. P. Keerthana, M. Selvakumari and S. Saravanan, "Gas Cylinder Monitoring System using IoT", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.67-71, 2023.
40. J.Arul, R.Balaji, S.Jeyamoorthy, M.Manipathra, R.Sundar and S.Saravanan, "IoT based Air Conditioner Control using ESP32", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.48-52, 2023.
41. Vundel Munireddy, J.Prahathesvaran, C.R.Thirunavukarasu, M.Santhosh Kumar and S.Saravanan, "IoT Based Charge Controller for Direct Fast Charging of Electric Vehicles Using Solar Panel", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.77-81, 2023.
42. D.Monish Kumaar, K.Akash, S.Aswinkumar, S.Saravanan and R. Sagayaraj, "IoT based Industry Surveillance and Air Pollution Monitoring using Drones", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.14-18, 2023.
43. T.Silambarasan, R.Surya, J.Pravinkumar, R.Sundar and S Saravanan, "IoT based Monitoring System For Sewage Sweeper", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.88-93, 2023.
44. R.Aravinthan, Alwin.Augustin, P.Divagaran, S.Saravanan and P.Manikandan, "IoT Based Power Consumption and Monitoring System", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.43-47, 2023.
45. S.Partheeban, S.Sundaravel, S.Umapathi, R.Sagayaraj and S.Saravanan, "IoT based Safety Helmet for Mining Workers", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.116-120, 2023.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

46. K.Eswaramoorthi, R.Manikandan, R.Balamurugan, C.Ramkumar and S.Saravanan, "Smart Parking System using IoT", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.53-57, 2023.
47. S.Nirmalraj, C.Pranavan, M.Prem and S.Saravanan, "Smart Trolley With IoT Based Billing System", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.111-115, 2023.
48. V.Gunasekaran, M.Gowtham , S. Anbubalaji, S.Saravanan and R.Prakash, "Solar based Electric Wheel Chair", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.8-13, 2023.
49. P Thava Prakash, P.Venketesan, D.Vignesh, S.Prakash, S.Saravanan, "Design of Low Cost E-Bicycle using Brushless DC Motor with Speed Regulator", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.148-153, 2023.
50. D.Tamilarasan, V.S.Vairamuthu, Y.Vasanth, K.Umadevi, S.Saravanan, "GSM based Agricultural Motor Control", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.172-177, 2023.
51. P. Vimal, S.Veerasigamani, R.Srihari, C.S.Satheesh, S.Saravanan, "IoT Based Optimal Power Management System For Smart Grid", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.160-165, 2023.
52. S.Abimanyu, P.Jagadheeswaran, S.Jaganath, K.Sanjay, R.Sivapranesh, K.Velmurugan, N.Mohananthini, C.S.Satheesh, S.Saravanan, "Portable Solar Tree", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.154-159, 2023.
53. M.Karthikeyan, S.Bilalahamad, V.A.Chandru, V.Deepika and S.Saravanan, "Design and Development of IoT based Motor Starter", International Journal of New Innovations in Engineering and Technology, Vol.22, Issue.3, pp.178-183, 2023.
54. R.Anbarsan, A.Arsathparvez, K.S.Arunachalam, M.Swathisriranjani and S.Saravanan, "Automatic Class Room Light Controlling Using Arduino" International Journal of Engineering Technology Research & Management (IJETRM), Vol-4 Issues 03, pp.192-201, 2020.
55. S.Karthikeyan, A.Krishnaraj, P.Magendran, T.Divya and S.Saravanan , "The Dairy Data Acquisition System" International Journal of Engineering Technology Research & Management (IJETRM), Vol-4 Issues 03, pp.163-169, 2020.
56. N.Harish, R.Jayakumar, P.Kalaiyarasan, G.Vijayakumar and S. Saravanan, "IoT Based Smart Home Energy Meter" International Journal of Engineering Technology Research & Management (IJETRM), Vol-4 Issues 03, pp.177-183, 2020.
57. G. Poovarasan, S. Susikumar, S. Naveen, N. Mohananthini, S. Saravanan," Study of Poultry Fodder Passing Through Trolley in Feeder Box," International Journal of Engineering Technology Research & Management, vol.4, Issue.1, pp.76-83, 2020.
58. A.Ananthan, A.M.Dhanesh, J.Gowtham, R.Dhinesh, G.Jeevitha and S.Saravanan, "IoT Based Clean Water Supply" International Journal of Engineering Technology Research & Management (IJETRM), Vol-4 Issues 03, pp.154-162, 2020.
59. Ram Kumar C, Saravanan S, and Nagarajan C,"Hybrid LSTM and Deep Reinforcement Learning for Autonomous Battery Health Optimization in Electric Vehicles", Electrical Power Systems Research, Vol-253 Issues 112535, ISSN No:0378-7796,2025.
60. Gopinathan, V. R. (2024). Real-Time Fault-Tolerant Multi-Cloud Database Architectures for High Availability Applications. International Journal of Future Innovative Science and Technology (IJFIST), 7(4), 13148.
61. Chandra, S., Rengarajan, A., Sahoo, G. S., & Sharma, S. (2023, December). Identifying Neuronal Damage and Plasticity by Analyzing Changes in Diffusion Tensor Imaging. In International Conference on Data Science, Machine Learning and Applications (pp. 433-438). Singapore: Springer Nature Singapore.
62. Sugumar, R. (2025). Federated AI in Offline-First Mobile Health Architectures for Privacy-Preserving Clinical Intelligence. International Journal of Science, Research and Technology, 8(4), 14589-14600.
63. Murugeshwari, B., Rajalakshmi, S., & Sudharson, K. (2023). Hybrid Approach for Privacy Enhancement in Data Mining Using Arbitrariness and Perturbation. Computer Systems Science & Engineering, 44(3).
64. Pandey, V. K., Mishra, S., Rengarajan, A., Savita, & Roomi, M. M. (2024, March). Enhancing Weather Forecasting with Machine Learning Techniques. In International Conference on Renewable Power (pp. 147-156). Singapore: Springer Nature Singapore.
65. Soundappan, S. J. (2025). Next Generation AI Enabled Holistic Cognitive Platform for Secure Cloud Network Intelligence Enterprise Systems and Digital Trust Optimization. International Journal of Computer Technology and Electronics Communication, 8(5), 11534-11542.
66. Mathew, A. (2022). Leveraging Big Data Analytics to Power AI and ML (Machine Learning) Automation. Educational Research (IJMCER), 4(5), 131-134.
67. Sugumar, R. (2024). AI-Augmented Quality Engineering for Performance Optimization and Test Orchestration in Distributed Systems. International Journal of Science, Research and Technology, 7(5), 12835-12846.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

68. Akila, R. (2024). A deep reinforcement learning approach for optimizing inventory management in the agri-food supply chain. J. Electrical Systems, 20(4s), 2238-2247.
69. Mahendran, M., Anbazhagan, K., Pavithran, G., Nivas, A., & Pandey, S. D. (2022). Earthquake Damage Prediction using Machine Learning. Grenze International Journal of Engineering & Technology (GIJET), 8(1).
70. Gopinathan, V. R. (2025). Enterprise AI Frameworks for Financial Data Engineering Behavioural Analytics and Intelligent Cloud Solutions. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 8(4), 12499-12506.
71. Kondalsamy, P., & Kaliappan, K. (2025). An Optimal Prediction of Leaf Disease Based on Hybrid Deep Learnings and Metaheuristic Technique. Traitement du Signal, 42(1), 363.
72. Deivendran, P., Babu, P. S., Malathi, G., Anbazhagan, K., & Kumar, R. S. (2023). Emotion Recognition for Challenged People Facial Appearance in Social using Neural Network. arXiv preprint arXiv:2305.06842.
73. Sugumar, R. (2025). Unified AI Framework for Predictive Data Engineering and Real Time Prescription and Billing Systems. International Journal of Advanced Engineering Science and Information Technology (IJAESIT), 8(5), 17261.
74. Vekariya, V., Kumar, S., & Rengarajan, A. (2024). A distinctive and smart agricultural knowledge-based framework using ontology. In Sustainability in Digital Transformation Era: Driving Innovative & Growth (pp. 207-213). CRC Press.
75. Gopinathan, V. R. (2025). Software engineering practices for AI-driven systems: From development to deployment (MLOps perspective). International Journal of Science, Research and Technology, 8(1), 13493-13500.
76. Mathew, A. R. (2022). Threats and protection on E-sim: a prospective study. Novel Perspectives of Engineering Research, 8, 76-81.
77. Naveena, S., & Kavitha, K. (2025). Gossypium herbaceum: Folium disease identification and classification using Efficient Net-Coordinate Convolutional Neural Network (EcoNet). Engineering Applications of Artificial Intelligence, 152, 110701.
78. Rengarajan, A., Mishra, A., Kulhar, K. S., Shrivastava, V. P., & Alawneh, Y. J. J. (2024, March). Role of Deep Reinforcement Learning in Mitigating Cyber Security Issues: A Review. In International Conference on Renewable Power (pp. 37-48). Singapore: Springer Nature Singapore.
79. Achari, A. P. S. K., & Sugumar, R. (2024, November). Performance analysis and determination of accuracy using machine learning techniques for naive bayes and random forest. In AIP Conference Proceedings (Vol. 3193, No. 1, p. 020199). AIP Publishing LLC.
80. Mathew, A., & Alex, H. (2022). Detect & protect-medical device cybersecurity. Curr. Overview Sci. Technol. Res, 1, 60-68.
81. Sammy, F., Chettier, T., Boyina, V., Shingne, H., Saluja, K., Mali, M., ... & Shobana, A. (2025). Deep Learning-Driven Visual Analytics Framework for Next-Generation Environmental Monitoring. Journal of Applied Science and Technology Trends, 114-122.
82. Anbazhagan, K. (2024). Trustworthy and Adaptive AI Systems for Enterprise Analytics Cybersecurity and Decision Optimization Using API-First and Cloud-Native Architectures. International Journal of Technology, Management and Humanities, 10(03), 65-74.
83. Mathew, A. (2021). Deep reinforcement learning for cybersecurity applications. Int J Comput Sci Mob Compu, 10(12), 32-38.
84. Dhinakaran, D. (2022). Joe Prathap P. M, Selvaraj D, Arul Kumar D and Murugeshwari B," Mining Privacy-Preserving Association Rules based on Parallel Processing in Cloud Computing,". International Journal of Engineering Trends and Technology, 70(3), 284-294.
85. Karthika, K., Anusha, K., Kavitha, K., Harshadha, R., Dharshini, D. S., & Sundhar, N. A. (2025, April). Frequency Reconfigurable Antenna using Advanced Materials: A Study. In 2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA) (pp. 1-6). IEEE.
86. Thavamani, C., & Rengarajan, A. (2024). Clustering related behaviour of users by the use of partitioning and parallel transaction reduction algorithm. International Journal of Advanced Intelligence Paradigms, 29(2-3), 122-132.
87. Sugumar, R. (2025). Unified AI Framework for Predictive Data Engineering and Real Time Prescription and Billing Systems. International Journal of Advanced Engineering Science and Information Technology (IJAESIT), 8(5), 17261.
88. Soundappan, S. J., & Sugumar, R. (2016). Optimal knowledge extraction technique based on hybridisation of improved artificial bee colony algorithm and cuckoo search algorithm. International Journal of Business Intelligence and Data Mining, 11(4), 338-356.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802345

89. SakthiPreetha, A., Kavitha, K., Karthika, K., & Manohari, R. G. (2025, April). A Novel Metasurface-Embedded Antenna for WBAN Communications. In 2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA) (pp. 1-4). IEEE.
90. Murugeshwari, B., Selvaraj, D., Sudharson, K., & Radhika, S. (2023). Data Mining with Privacy Protection Using Precise Elliptical Curve Cryptography. Intelligent Automation & Soft Computing, 35(1).
91. Gopinathan, V. R. (2025). Software engineering practices for AI-driven systems: From development to deployment (MLOps perspective). International Journal of Science, Research and Technology, 8(1), 13493-13500.
92. Anbazhagan, K., Kumar, R., Thilagavathy, R., & Anuradha, D. (2024, March). Shortest Job First with Gateway-based Resource Management Strategy for Fog Enabled Cloud Computing. In 2024 4th International Conference on Data Engineering and Communication Systems (ICDECS) (pp. 1-6). IEEE.
93. Kannadhasan, S., Vasuki, S., Kavitha, K., Karthikeyan, P., & Usha, S. G. A. (Eds.). (2025, April). Preface: Role of Artificial Intelligence and IoT in Engineering, Technology & Science [ICRAETS 2024]. In AIP Conference Proceedings (Vol. 3258, No. 1, p. 010001). AIP Publishing LLC.
94. Dhinakaran, D., Prathap, P. J., Selvaraj, D., Kumar, D. A., & Murugeshwari, B. (2022). Mining privacy-preserving association rules based on parallel processing in cloud computing. International Journal of Engineering Trends and Technology, 70(3), 284-294.