iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** VI **Month of publication:** June 2019

DOI: http://doi.org/10.22214/ijraset.2019.6431

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

Extraction of Subset-Count in Data Streams using EMDMICA Algorithm

P. Logeshwari
Assistant Professor, Department of Computer Science, SNMV College, Malumachampatti, India

Abstract: ***In the EMDMICA method, the main stream Data Multiple Imputation is divided into m equal-sized segments of s transactions, and processes the Imputation/update of data stream incrementally in a segment-based manner. The current main stream Data Multiple Imputation in a segment-based fashion too. The proposed method would approximate the counts of Data and discover FIs over the main stream Data Multiple Imputation of a data stream. This method now processes data stream Imputation in a segment-based fashion. The proposed data-stream mining algorithm, namely the Efficient Main stream Data Multiple Imputation with Combinatorial Approximation (EMDMICA) algorithm.***
Keywords: Main Stream, Multiple Imputation, segments, transactions, EMDMIC Aalgorithm.

I. INRODUCTION

In the mining result of the data stream DS (with respect to some ms) returned by a mining method, a true answer is an Data who is frequent-or-not with respect to ms correctly decided, while a false answer is a misjudged item set, by the mining method. A true positive Data and a false positive Data are respectively the frequent and in Missing Data returned as frequent in the mining result. On the other hand, true negative and false negative means then the infrequent and Missing Data, respectively, returned as infrequent by the mining method. The true-positive rate of the mining result is the rate of true positives of all of the actually frequent item- sets, whereas the true-negative rate of the mining result represents the rate of true negatives of all the actually in Missing Data.

Many of the existing data-stream mining methods work therefore with a basic hypothesis that they know the user-specified ms in advance, and this parameter will stay unchanged all the time before the data stream actually terminates. This hypothesis may be somewhat unreasonable, since in general, a user may wish to tune the value of the ms each time he/ she makes a mining request for the purpose of obtaining a more preferable mining result. Practically an unchangeable ms leads to a serious limitation and may be impractical for most real-life applications. As a result, the constraint relaxed in this problem then allows a user to change the value of ms during the process of stream transit. Given a transactional data stream DS in which every incoming transaction has its items arranged in order, the problem of mining FIs over the main stream Data Multiple Imputation W of DS is to find out the set of Data whose counts over the W are above the threshold determined by some freely specified ms at different Main Stream Data.

II. PROBLEM DEFINITIONS OF EMDMICA ALGORITHM

Let I = $\{x_1, x_2,\ldots x_z\}$ be the set of items (or attributes) which may occur in a source of data stream. An Data (or a pattern) X is a subset of I and written as $X = x_i,x_j,\ldots x_m$. The length (i.e., number of items) of a Data X is denoted by |X|. And a transaction T is a set of items, and T supports a Data X if $X \subseteq T$. A transactional data stream DS is a sequence of continuously incoming transactions, in which every transaction is one basic element of DS. A data stream in the data stream is a time interval which covers a set of successive w transactions. A main stream Data Multiple Imputation W in the data stream is a data stream of most recent w transactions which Main Stream Data forward for transactions, where w denotes the size of W. The notation I_l used to denote the set of all possible Data of length l (that is, l- item sets) together with their respective counts in a set of transactions. In addition, $T_n$use to denote the latest transaction in the current data stream. Thus, the current data stream is $W = \{T_{n-w+1},\ldots T_n\}$.

In this research, a prefix tree is organized under the lexicographic order as the data structure, and also processes the growth of Data in a lexicographic-ordered fashion. As a result, an Data is treated a little bit like a sequence (while it is indeed an item set). A superset of a Data X is the one then whose length is larger than |X| and has X as its prefix. The terms count and count-value are now used interchangeably to represent the occurrence of an Data supported by transactions. For one Data X, the symbol cnt(X) is to represent its occurrences in a set of transactions. The count of X over W, denoted as $cnt_W(X)$, is the number of transactions in W that support X. Given a user randomly specified minimum-support threshold (ms), where 0 <ms≤ 1, X is a Missing Data (FI) over W if $cnt_w(X) \geq ms \cdot w$; otherwise X is an in Missing Data (IFI).

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

III. COMBINATORIAL APPROXIMATION AND SUBSET-COUNT LIMITATION

In this subsection we describe the basis of our mining method, which is an approximation-based approach. Consider that there are m sets, A_1, A_2……., and A_m. The following two equations Eqs. (3) And (4) are respectively the formula of Principle of Inclusion and Exclusion (24) and that of Approximate Inclusion-Exclusion (32), where the latter is derived from the former

$$|A1 \cup A2 \cup \ldots\ldots\ldots \cup Am| = \sum_{i}|Ai| - \sum_{i<j}|Ai \cap Aj| \sum_{i<j<k}|Ai \cap Aj \cap Ak| + \cdots +$$

$$(-1)^{n+1}|A_1 \cap A_2 \cap \ldots\ldots\ldots \cup A_m| \ldots\ldots\ldots (3)$$

$$|A1 \cup A2 \cup \cdots\cdots \cup Am| = \sum_{|s|\le k} \alpha_{|S|}^{k,m} |\bigcap_{i\in S} A_i| x \ldots\ldots\ldots (4)$$

Eq. (3) states that the size (i.e., number of elements) of the union of m terms (m sets, A_1, A_2, ... ,A_m) is equal to several sums of sizes of set intersections of distinct lengths below m. According to Eq. (4), the value of the m-union term can be approximated even if only the sizes of intersection of k partial terms are in a set of m terms. By combining Eqs. (3) And (4) can be possibly applied to calculate the counts of Data in data-mining domain due to the similarity between sets and items. The resulting equation is shown as follows:

$$\sum_i|Ai| - \sum_{i<j}|Ai \cap Aj| + \sum_{i<j<k}|Ai \cap Aj \cap Ak| + \cdots + (-1)^{n+1}|A_1 \cap A_2 \ldots\ldots \cap A_m| = \sum_{|S|\le K} \alpha_{|S|}^{k,m} |\bigcap_{i\in S} Ai|$$

…… (5)

By considering each set as an attribute (i.e., item) of the data stream, and its corresponding size as the number of occurrences (i.e. count) in stream transactions, the intersection of the (several) sets can be viewed as an item set. As a result, Eq. (5) can be applied to approximate the intersection size of m terms in Eq. (3) from the sums of their sub-terms of different lengths, and this value just corresponds to the count of an m-item set. This then is called as Combinatorial Approximation (CA).

IV. SEGMENT-BASED DATA STREAM IMPUTATION

The technique of CA (Combinatorial Approximation) has to approximate the counts of item sets, from some kept information called the base summary. In this method the base-summary size, i.e., s_{base}, is 2. For each incoming transaction T, all subsets of the first-two orders contained in T enumerate and record their occurrences in our data structure. As a result, the base summary (i.e., I_1 and I_2) of the data stream is maintained.

The base summary, which concerns the Imputation of data stream in a data stream, should be updated. In the (transaction-sensitive) Imputation-data stream model of data stream, as a new transaction arrives and is inserted into the current data stream, the oldest transaction is then dropped out, which is just one Imputation of the data stream. The insertion of a new transaction however is simple, while the deletion of an old transaction is more troublesome since one's knowledge requires which transaction is going to be dropped out each time the data stream Main Stream Data.

The data stream-Imputation problem has been handled by processing the Imputation of data stream in a segment- oriented manner. Although in many existing methods the data stream Imputation is handled transaction by transaction, we believe this may just not be exactly suitable for several reasons. First, in the Imputation-data stream model, unlike the landmark model, transactions will be both inserted into and dropped out from the data stream. The transaction-by-transaction Imputation of data stream leads to a

Huge amount of processing because the update of the transactions is excessively frequent. Besides, since the transit of a data stream is usually at a high speed, and the impact resulting from one single transaction to the entire set of transactions (in the current data stream) is negligible, it is reasonable therefore to handle the data stream Imputation in a wider magnitude. As a result, the segment-oriented data stream Imputation has been proposed

The main stream Data Multiple Imputation has been conceptually divided into several, m, segments, where a segment S is a sequence of the fixed number of transactions. Each of the m segments contains a set of successive transactions and is of the same size s_{seg} (i.e., contains the equal number of s_{seg} transactions). In addition, S_n is used to denote the latest segment in the current data stream. Thus, the current data stream is expressed as W = $\{S_{n-m+1},\ldots\ldots,S_n\}$.

The term segment is not only a quantity of the unit of transactions, and also in each segment the base summary belonging to that segment is also recorded. Besides this, the partial information of I_3 is kept for the upcoming transactions. The Imputation of segment-oriented data stream called as ''segment in-out'', is defined as follows.

Definition (Segment in-out): Let S_c denotes the current segment which is going to be inserted into the data stream next (after it is full of s_{seg} transactions). A segment in-out operation (of the data stream) first insert S_c into and then extract S_{n-m+1} from the original data stream, where n denotes the ID of latest segment in the original data stream. Therefore, the data streams before and after a Imputation are W = $\{S_{n-m+1},\ldots,S_n\}$ and W = $\{S_{n-m+2},\ldots S_n, S_c\}$, respectively.

The figure simply illustrates the concept of the segment-oriented main stream Data Multiple Imputation. It represents the data streams before and after a Imputation, respectively, while it also shows the process of a segment in-out operation, which further takes the data stream Imputation to its next stage. Now by this segment-based manner of Imputation, at each Imputation we insert the new segment and delete (or drop out) the earliest segment, which respectively contain the base summaries (i.e., I_1 and I_2, 1-items and 2-Data together with their respective counts) of data stream belonging to both segments. With this approach, there is no need to maintain the whole transactions within the current data stream in memory all along, while the Imputation of (segment-based) data stream is still feasible. In addition, the parameter m directly affects the consumption of memory and is now remarked. A larger value of m means the data stream will Main Stream Data/update more frequently (since each segment on an average contains fewer transactions), while the increasing overhead of the memory space is also considerable. In our opinion, an adequate size of m that falls in the range between 5 and 20 may be suitable for the general data streams.

The proposed method would approximate the counts of Data and discover FIs over the main stream Data Multiple Imputation of a data stream. This method now processes data stream Imputation in a segment-based fashion. The proposed data-stream mining algorithm, namely the Efficient Main stream Data Multiple Imputation with Combinatorial Approximation (EMDMICA) algorithm is described as follows.

A. AlgorithmEMDMICA

Input: A transactional data stream (DS), a minimum-support threshold (ms), and a Imputation-data stream size (w)
Output: A list of Missing Data (F)
Method:

```
Divide the data stream conceptually into m segments (where 5 ≤ m ≥ 20);
Set up a segment in-out pointer sp;
while data of DS is still streaming in do begin
        set F to be empty;
while no request from the user do begin
          Fetch the next incoming transaction T from DS;
          Enumerate all subsets of T and record increase their counts in  Sc;
          if the length of T is over 2 then begin
                Enumerate all 3-subsets of T and record their counts in Sc;
end if
          if Sc is full of sseg transactions then begin
Discard all kept 3-Data in Sc;
                Insert Sc as the latest segment Sn into L according to sp;
 if the number of segments in L is greater than m then begin
                Delete the oldest segment from L according to sp;
              end if
              Create a new Sc for the next round;
Set sp circularly to point to the next filed;
           end if
    end while
    Merge the m segments (in L) to obtain the count- values for Data in the
    current  data stream;
    Find all large 1-items and 2-Data and insert them into F;
 foreach frequent 2-Data X in F do begin
          foreach segment S in the data stream do begin
                Aggregate the count of each 3-superset of X with its sum of counts  so far
                respectively;
          end foreach
          Insert every frequent 3-Data into F;
end foreach
    foreach frequent 3-Data Y in F do begin
      repeat
          n ⟶ 4;
          Calculate the counts of n-Data with m = n and k = 3;
          Insert every frequent n-Data into F;
          n ⟵ n + 1;
until there is no frequent n-Data generated
end foreach
Output F as the mining result;
end while
```

Fig. 1.EMDMICA Algorithm

In the EMDMICA method, the main stream Data Multiple Imputation is divided into m equal-sized segments of s transactions, and processes the Imputation/update of data stream incrementally in a segment-based manner. The data structure used to keep the base summary of data stream is now a lexicographic-ordered prefix tree modified. This data structure maintains the base summary, i.e., I_1 and I_2 (in this research), over the current main stream Data Multiple Imputation in a segment-based fashion too. Besides, for the current segment of transactions, i.e., S_c, its base summary is further kept separately in an array. In the array, we also maintain the whole I_3 of S_c for the purpose of calculating and finding the Missing Data.

The EMDMICA algorithm processes on an on-line transactional data stream. As long as there is no mining request from the user, the EMDMICA continues receiving and processing the incoming transactions one by one, and handles the data stream Imputation in a segment- based manner.

For each incoming transaction T in the current segment S_c, the EMDMICA enumerates and records the first- three orders of subsets contained in T. When S_c is full of s_{seg} transactions, for each 2-Data X which is not in S_c, EMDMICA calculates FI. After the chosen I_3 is obtained totally, a segment in-out operation is finally performed. To insert S_c into the data stream, only I_1, I_2, and the chosen I_3 are updated into the tree, while the information of original I_3 over $S_c$is discarded.

For each Data X belonging to the base summary, the corresponding node in the tree includes then a circular array of size m which corresponds to the m segments of the main stream Data Multiple Imputation, and X's count over the current data stream is recorded respectively in these m fields of the array. If we combine the counts of all the segments, the count of X over the current data stream is obtained.

There is a (global) pointer to indicate in which field the count of X over S_c will be stored in. When the current segment S_c is full and a segment in-out operation is going to be performed, the count of X over S_c is then stored into the field of array indicated by the pointer, and that which is over the earliest segment is then dropped out naturally since it is replaced by the newly stored value as regards S_c.

In the rest of the portion of the approximation task, the EMDMICA uniformly employs I_1 and I_2 (i.e., the base summary) plus the approximated I_3 to approximate for the Data with longer length. The process of approximation proceeds in both depth-first order and lexicographic order. T

hat is, for any two 3-Data Y_1 and Y_2, if then Y_2 comes after Y_1 in lexicographic order, then before EMDMICA starts processing Y_2, all Data having Y_1 as their prefix have then been processed already.

As the data-stream mining method should work with a changeable value of ms, which is a requirement that our proposed method needs to satisfy.

Now EMDMICA meets this requirement adequately. As mentioned before, in the data structure of the proposed method, we maintain no more than I_1 and I_2 (and the chosen I_3) over the current data stream, and the mining process proceeds by approximating the counts of Data from such a base-summary data structure. From the viewpoint of ESWCA, an Data is (determined as) frequent when its approximate count is above ms. For a different ms set by a user, EMDMICA just employs Eq. (5) and its embedded techniques to approximate itemsets' counts according to its base-summary data structure, and then selects the frequent ones in terms of the respective ms. As a result, the usability of EMDMICA is not affected by one variable ms. A user may set then different ms at each time he/she makes a mining request, whilst the EMDMICA has to work for defining the sizes of the data streams.

V. CONCLUSION

As the data-stream mining method should work with a changeable value of ms, which is a requirement that our proposed method needs to satisfy. Now EMDMICA meets this requirement adequately. As mentioned before, in the data structure of the proposed method, we maintain no more than I_1 and I_2 (and the chosen I_3) over the current data stream, and the mining process proceeds by approximating the counts of Data from such a base-summary data structure. From the viewpoint of ESWCA, an Data is (determined as) frequent when its approximate count is above ms. For a different ms set by a user, EMDMICA just employs Eq. (5) and its embedded techniques to approximate itemsets' counts according to its base-summary data structure, and then selects the frequent ones in terms of the respective ms. As a result, the usability of EMDMICA is not affected by one variable ms. A user may set then different ms at each time he/she makes a mining request, whilst the EMDMICA has to work for defining the sizes of the data streams.

REFERENCE

[1] ERTE PAN, (Student Member, IEEE), MIAO PAN, (Member, IEEE), AND ZHU HAN, (Fellow, IEEE)Tensor Voting Techniques and Applicationsin Mobile Trace Inference, IEEE Access SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE ENABLE NETWORKING,VOLUME 3, 2015 Received October 30, 2015, accepted November 16, 2015, date of publication December 24, 2015, date of current version January 7, 2016.

[2] Ms.R.Malarvizhi and Dr.Antony Selvadoss Thanamani Cluster Based Mean Imputation International Journal of Research and Reviews in Applicable Mathematics& Computer Science.Vol 2.No.1,2012,

[3] Bayesian Learning of Noisy Markov Decision Processes,ACM Transactions on Modeling and Computer Simulation Vol. 23, No. 1, Article 4, Publication date: January 2013.SUMEETPAL S. SINGH, University of Cambridge

[4] Yosio Edemir Shimabukuro, Jukka Miettinen, René Beuchle, Rosana Cristina Grecchi,Dario Simonetti, and Frédéric Achard Estimating Burned Area in Mato Grosso, Brazil,Using an Object-Based Classification Method on a Systematic Sample of Medium Resolution Satellite Images,IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, VOL. 8, NO. 9, SEPTEMBER 2015,

[5] Xuanyu Zhao, Huafeng Zhou, Di Shi, Huashi Zhao, Chaoyang Jing, Chris Jones On-Line PMU-Based Transmission Line Parameter Identification,CSEE JOURNAL OF POWER AND ENERGY SYSTEMS,VOL. 1, NO. 2, JUNE 2015,

[6] Ms.R.Malarvizhi and Dr.Antony Selvadoss Thanamani Cluster Based Mean Imputation,International Journal of Research and Reviews in Applicable Mathematics& Computer Science.Vol 2.No.1,2012,

[7] Ms.R.Malarvizhi and Dr.Antony Selvadoss Thanamani.K-NN Classifier Performs Better Than K-Means Clustering in Missing Value Imputation,International Journal for Research in Science & Advanced Technologies,Vol 1.Issue-2,2013,

[8] S.Kanchana and Dr.Antony Selvadoss Thanamani.Classification of Efficient Imputation Method for Analyzing Missing Values,International Journal of Computer Trends and Technology(IJCTT),Vol 12.No.4-Jun 2014 ,

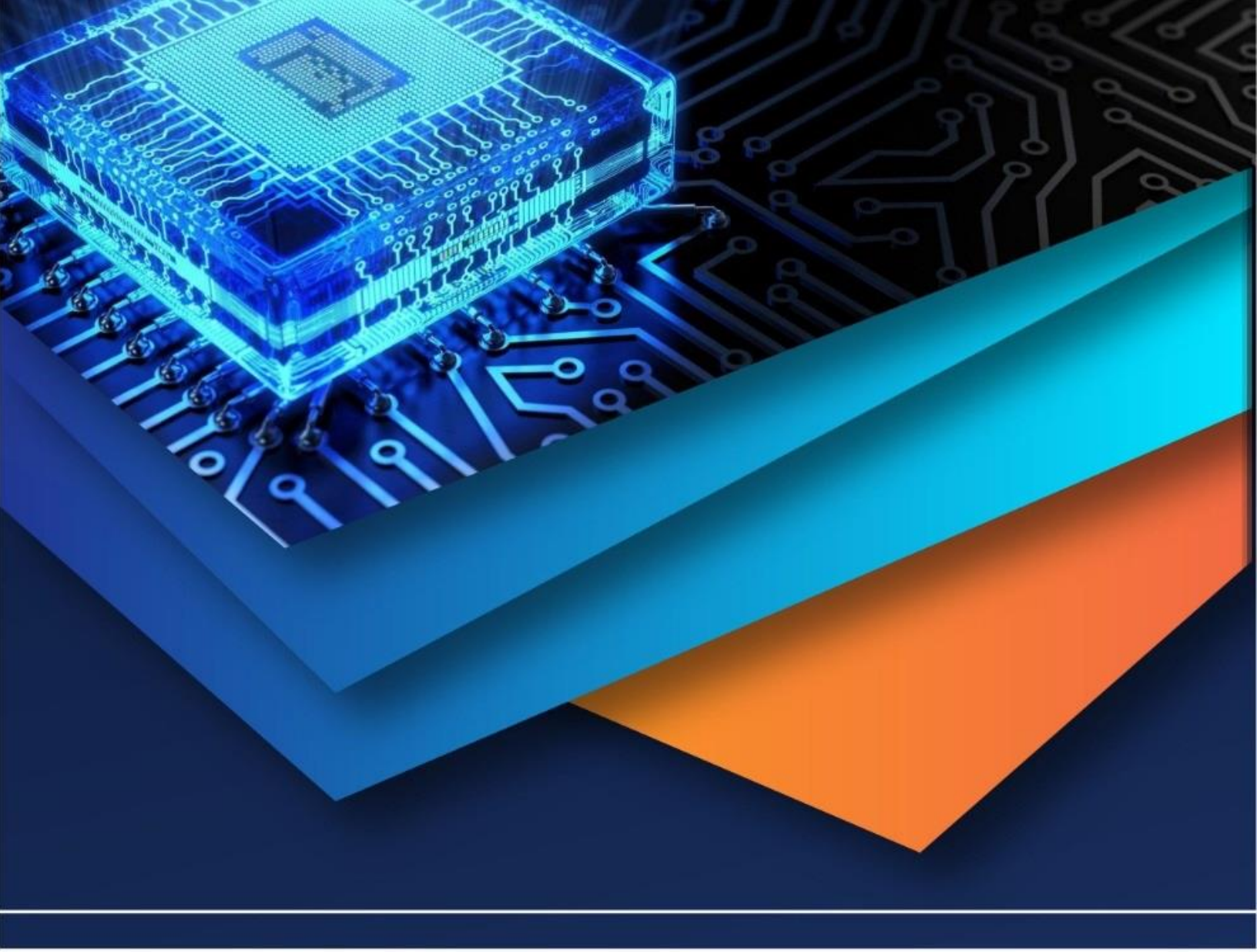
[9] S.Kanchana and Dr.Antony Selvadoss Thanamani Multiple Imputation of Missing Data Using Efficient Machine Learing Approach,International Journal of Applied Engineering Research,Vol 1.No.1 ,2015,

[10] Ms.R.Malarvizhi and Dr.Antony Selvadoss Thanamani..K-NN Classifier Performs Better Than K-Means Clustering in MissingValue Imputation Journal: International ournal for Research in Science & Advanced Technologies,Vol 1.Issue-2,2013,

[11] Mrs. P.Logeshwari and Dr.Antony Selvadoss Thanamani.A Survey On Missing Data And Methods To Find TheMissing Values International Journal For research In Science And Technology Volume 1, 2015,

[12] Mrs. P.Logeshwari and Dr.Antony Selvadoss Thanamani.Assignable Algorithms Available forMissing Data for Finding MV, International Journal Of Advanced Networking and Applications (IJANA), Special Issue,2015,

Mrs. P. Logeswari received her MCA., degree in computer Science from Sree Saraswathi Thiyagaraja College of arts and science, Pollachi, India in 2010. She completed her M.Phil., degree in computer Science from Sree Saraswathi Thiyagaraja College of arts and science, Pollachi, India on 2012 . She completed her PhD (Full Time) degree in Computer Science in NGM College (Autonomous), Pollachi under Bharathiar University, Coimbatore. She served as a Faculty of Computer Science at Government Arts College Udumalpet, from 2012 to 2013 and she served as a Faculty of Computer Science at Sree Ramu College of Arts and Science, NM Sunggam,Pollachi,India.from April 2013 to August 2014.Currently she is working as a Assistant Professor in SNMV college of arts and science, Coimbatore. She has presented papers in International/National conferences and published two papers in International journal. Her research focuses on Data Mining.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY