Collaborative Anti Spam Technique to Detect Spam Mails in E-Mails

Mrs.Latha.K [*], Nivedha.P [**], Menagagandhi.G [**], Ramya.T [**]

[*] Assistant Professor,Dept of IT, Adhiyamaan College of Engineering, Hosur, Krishnagiri (Dist), Tamilnadu, 635109, India
[**] UG Scholar, Dept of IT, Adhiyamaan College of Engineering, Hosur, Krishnagiri (Dist), Tamilnadu, 635109, India

Abstract- In the field of collaborative spam filtering by near-duplicate detection, an e-mail abstraction scheme is required to more certainly catch the evolving nature of spams. Compared to the existing methods in prior research, in this work, we explore a more sophisticated and robust e-mail abstraction scheme, which considers e-mail layout structure to represent e-mails. The specific procedure SAG is proposed to generate the e-mail abstraction using HTML content in e-mail, and this newly-devised abstraction can more effectively capture the near-duplicate phenomenon of spams. The hash function is easily identified and it not an efficient. To overcome this drawback our proposed system using Feature-preserving fingerprint technique. To generate a TFset of a message M, we use a sliding window algorithm, in which a window of some predetermined length (W) slides through the message. At each step, the algorithm computes a Rabin fingerprint of W consecutive tokens that fall within the window. Each fingerprint is in the range (0, 2^K-1), where K is a configurable parameter. Feature-preserving fingerprint is just one level of privacy protection; the amount of information exchanged during collaboration can be further controlled for stronger privacy protection. In particular, we design the collaborative antispam system equipped with privacy-aware message exchange protocol based on the following spam/ham dichotomy that revealing the contents of a spam e-mail does not affect the privacy or confidentiality of the participants, whereas revealing information about a ham e-mail constitutes a privacy breach. Our proposed system to improving spam mail detection as well as provide privacy for message exchange.

Index Terms- antispam ,e-mail abstraction, , fingerprinting, privacy.

I. Introduction

Detecting spam mails in e-mail is the most emerging research in e-mails must have the following ways, First, software-based spam detection on MTAs(Mail Transfer Agent) is not capable of detecting spam at high throughput and will not cope with the increase in future e-mail traffic.MTAs are e-mail servers that run on general-purpose processors (GPPs). GPP-based systems could not scale with increases in link speed .Even for custom GPP-based spam detectors , their throughputs are limited to hundreds Mbps. Specialized hardware architectures, with improved processing power, are needed for fast spam detection and to support the network bandwidth growth. Second, spam detection at the application layer (layer 7)1 restricts where, when, and how fast spam detection can be performed. Detecting spam at layer 7 makes detection at intermediate nodes (between the sending and the receiving MTAs) infeasible due to the need for complex Transport Control Protocol/Internet Protocol (TCP/IP) processing at link speed. TCP, which requires reassembly, byte alignment, and state tracking requires large computation overhead. As a result, spam control is restricted to MTA implementation as an end-to-end spam control mechanism.An improved spam detection approach, at lower e-mail abstraction levels, is needed to lift the end-to-end implementation restriction and to allow fast spam detection, closer to spam sources.

Third, due to the lack of outbound spam control, e-mails are effectively classless upon reception at the receiving MTA. In the current spam control, e-mail classes are unknown until e-mails have been classified and detected for spam. Thus, all incoming e-mails are queued in a common queue and delivered (to recipients) with equal priority. For e-mail traffic that consists mainly of spam, non-spam e-mails are delayed due to the presence of spam in the queue. Furthermore, non-spam e-mails maybe lost during queuing. Spam wastes MTA processing and bandwidth resources. Any attack on the common queue could disrupt server operations. A scheme at receiving MTAs to maintain e-mail delivery is needed to reduce the non-spam delay and loss due to queuing. [1]so the identified mail must be correctly identified .A more effective spam detection using vendors reply helps to detects spam more accurately and privacy is achieved by rabin fingerprinting hash algorithm, as only the hash values with ranges are sent to different vendors privacy is achieved in a this algorithm and spams are detected with more effectively.

II. RELATED WORK

In the evaluation of highest probability of nearest neighbour classifier they evaluate the performance of the highest probability SVM nearest neighbour classifier,which is an improvement over the SVM nearest neighbour classifier on the task of spam filtering.In order to classify a sample x,it first selects k samples to train an SVM model which is used to make decisions.As such,the SVM nearest neighbour algorithm proposes no rule for selecting the parameter made an attempt to estimate k by internal training and testing on a training data,but this approach brought uncertain results.[2]

By privacy aware collaborative spam filtering a large privacy aware collaborative spam filtering technique ALPACAS was designed inorder to detect spam effectively .But they were based on the content in e-mail.the spammers tried to defeat this algorithm by inserting a random paragraph like structures into the e-mail messages which does not detect spam mails effectively in e-mail.[3]

Mail ranking method which considered e-mail address of senders by ranking priority if it is detected as a spam by its content with its two mail rank variants basic mail rank and personalized mail rank.Trust and reputation algorithms have become increasingly popular to rank .Building upon the e-mail network graph,a power iteration algorithm is used to compute the score of e-mail address.This process does not have much scalability and faster executable and is more complicated.[4]

Markav random chain process uses the incoming mails with its contents are only identified and uses its its weighting scheme for spam filtering .As there is a chance of inserting a random paragraph into itwhile only contents are taken. This has a drawback of storage utilization is high and not efficient to use.[5]

In false positive safe neural network an approach of online cumulative training is proposed .If a system would learn each time something new it arrives ,the phrases are rephrased after sometime if the features are not too good the system would correctly recognize as a spam .This can be done on client side which makes a lot of work for user.The newral network defines whether the patterns are important a high false positive rates are achieved.[6]

III. PROPOSED SYSTEM

The input e-mail messages are etracted with its HTML content in e-mail. By rabin fingerprint hash algorithm a feature set (SF) is generated by overlapping the sequences and are numberedby converting a sequences into binary values.A hashing operation is performed for these binary values and the range of hash values with their ranges are sent to different vendors .Each vendors performs the same operation for different kinds of spam mails and stores the hash values into the databases .while transferring the hash values a privacy is achieved as only their ranges are sent to different vendors.each vendors with a database a hash value ranges for ham mails and for spam mails and thus spam mails are accurately identified with a true positive rate.

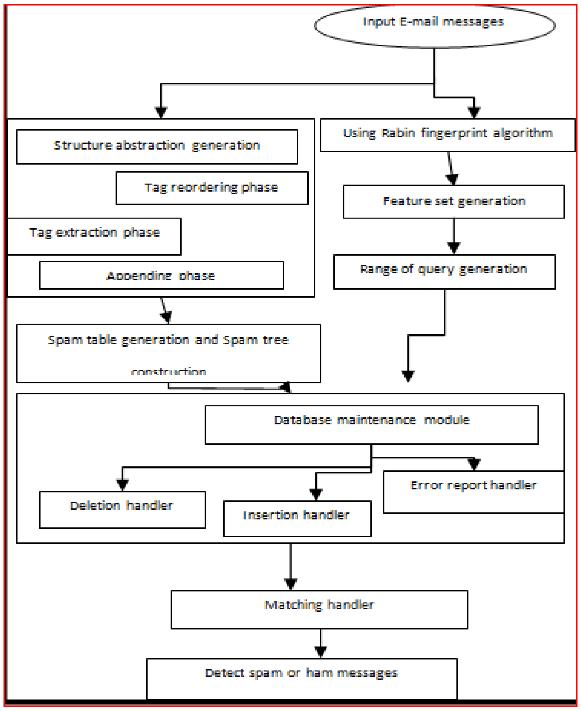

Input E-mail messages
Structure abstraction generation
Tag reordering phase
Tag extraction phase
Appending phase
Using Rabin fingerprint algorithm
Feature set generation
Range of query generation
Spam table generation and Spam tree construction
Database maintenance module
Deletion handler
Insertion handler
Error report handler
Matching handler
Detect spam or ham messages

Figure 1 SYSTEM ARCHITECTURE

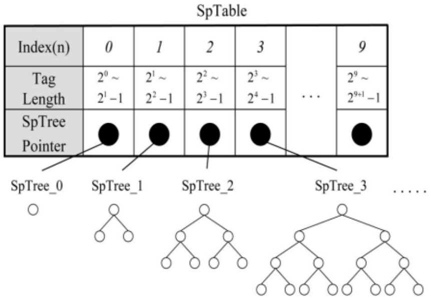
A. STRUCTURE ABSTRACTION GENERATION

In this module e-mail structure abstraction is to be generated. We propose the specific procedure SAG to generate the e-mail abstraction using HTML content in e-mail. SAG is elaborated with the example and the algorithmic form of SAG. Procedure SAG is composed of three major phases, Tag Extraction Phase, Tag Reordering Phase, and <anchor> Appending Phase. In Tag Extraction Phase, the name of each HTML tag is extracted, (i.e) the front and rare tags are eliminated ,mismatched tags, and mismatched positions are eliminated as the HTML tags in e-mail are abstracted. Tags attributes and attribute values are eliminated inorder to detect spam effectively. In addition, each paragraph of text without any tag embedded is transformed to <mytext=>. In lines 4-5, <anchor> tags are then inserted into AnchorSet, and the first 1,023 valid tags are concatenated to form the tentative e-mail abstraction.

i) SPAM TABLE AND SPAM TREE CONSTRUCTION

In this module SP table and SP tree are constructed for the above system fig2 .One major focus of this work is to design the innovative data structure to facilitate the process of near-duplicate matching. SpTable and SpTrees (sp stands for spam) are proposed to store large amounts of the e-mail abstractions of reported spams. As several SpTrees are the kernel of the database, and the e-mail abstractions of collected spams are maintained in the corresponding SpTrees. According to Definition 3, two e-mail abstractions are possible to be near-duplicate only when the numbers of their tags are identical. Thus, if we distribute e-mail abstractions with different tag lengths into diverse SpTrees, the quantity of spams required to be matched will decrease. However, if each SpTree is only mapped to one single tag length, it is too much of a burden for a server to maintain such thousands of SpTrees. In view of this concern, each SpTree is designed to take charge of e-mail abstractions within a range of tag lengths. As SpTable is created to record overall information of SpTrees.

The ith column of SpTable links to the root of SpTree_i by a pointer, and e-mail abstractions with tag lengths ranging from 2^i to 2^{i+1} -1 belong to SpTree_i. Regarding how an e-mail abstraction is stored in SpTree, it gives an example with the same e-mail abstraction derived from Figure. An e-mail abstraction is segmented into several subsequences, and these subsequences are consecutively put into the corresponding nodes from low levels to high levels. As such, an e-mail abstraction is stored in one path from the root node to a leaf node of SpTree, and hence the matching between a testing e-mail and known spams is processed from root to leaf.



SpTable

Index(n)	0	1	2	3		9
Tag Length	$2^0 \sim 2^1-1$	$2^1 \sim 2^2-1$	$2^2 \sim 2^3-1$	$2^3 \sim 2^4-1$	…	$2^9 \sim 2^{9+1}-1$
SpTree Pointer	●	●	●	●		●

Figure 2 SP TREE AND SP TABLE

B. SPAM DETECTION

In this module, Matching Handler in Spam Detection Module takes charge of determining results. There are three types of e-mails, reported spam, testing e-mail, and misclassified ham, required to be dealt with by Cosdes in fig3.When receiving a reported spam, Insertion Handler adds the e-mail abstraction of this spam into the database except that the reputation score of this reporter is too low. Whenever a new testing e-mail arrives, Matching Handler performs the near-duplicate detection with collected spams to do the judgment. Meanwhile, if a testing email is classified as a spam, this e-mail will be viewed as a reported spam and be added into the database. Moreover,Error Report Handler copes with feedback misclassified hams and adjusts Cosdes by degrading the reputation of related reporters to prevent malicious attacks. For every Td, Deletion Handler is triggered to delete obsolete spams which exist over time Tm. The main functionalities of deleting outdated spams are not only to alleviate the overhead of the server, but to reduce the risk of accidental deletion of hams. Due to the evolving nature of spams, it is inappropriate to utilize old spams to filter current ones. Overall, Cosdes is self-adjusting and retains the most up-to-date spams for near-duplicate detection.

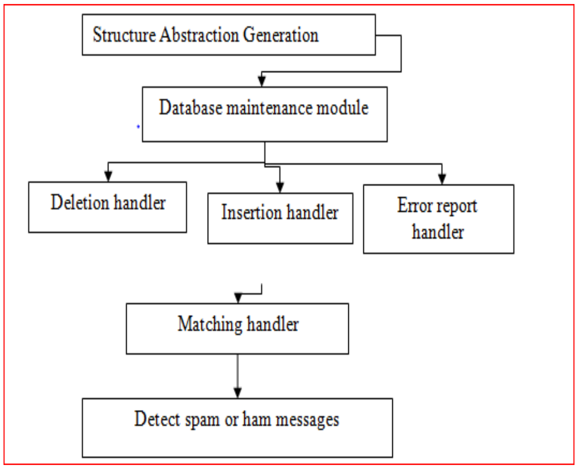

Structure Abstraction Generation
Database maintenance module
Deletion handler
Insertion handler
Error report handler
Matching handler
Detect spam or ham messages

Figure 3 SPAM DETECTION MECHANISM

i) DATABASE MAINTEINANCE

A database maintainence module is designed with three handlers as insertion handler,deletion handler,error report handler.Insertion handler is used to insert a newly detected spam mails.Deletion handler deletes the mails which are detected earlier,thus making the memory space to free to insert a newly detected identified mails.Error report handler reports a false positive rate of the misclassified ham mails and higher rate of true positive rate is achieved in the proposed system.Each of the handlers has a ham and a spam databases to identify the accurate spam mails at a higher rate.

ii) REPUTATION MECHANISM

In this consists of, to ensure the truthfulness of spam reports and to prevent malicious attacks, we propose the reputation mechanism to evaluate the credit of each reporter. The fundamental idea of the reputation mechanism is to utilize a reputation table to maintain a reputation score SR of each reporter according to the previous reliability record. Each inserted spam is given a suspicion score equal to SR of the reporter. In such a context, when doing near-duplicate detection, if the sum of suspicion scores of matched spams exceeds a predefined threshold, the testing e-mail will be classified as a spam. The reputation mechanism is described in detail as follows:

1. Each reporter is assigned an initial score S initial when he submits a reported spam at the first time.

2. If a reporter submits any feedback spam once more, the reputation score will be incremented by a smaller incremental score Sincre. The value of Sincre is set as Sinitial10 in the experiments.

3. If a reporter is charged that his previous feedback spam is mistaken, the reputation score will be halved.

To prevent malicious error reports and to attain a near-zero false positive rate, we cautiously increase the reputation score but drop it drastically while a false positive error is issued. On the other hand, when SR of a reporter is smaller than Sinitial, his subsequent feedback spams will not be added into the database until SR is equal to or larger than Sinitial. Regarding the parameter Sth, we simply use a fixed small value (set as three in the experiments) instead of determining the threshold according to the ratio of total users. The reason is that as long as there are certain trusty users reporting the e-mails with the same e-mail abstraction as spams, it is sufficiently reliable to classify the subsequent near-duplicate e-mails as spams.

C. PRIVACY PRESERVING COLLABORATIVE SPAM DETECTION

feature-preserving fingerprint hashing technique

In consideration of the privacy preservation, the message transformation uses a Rabin fingerprint algorithm, which is a one-way hash function such that it is computationally infeasible to generate the original e-mail from its TFset. However, it is possible to infer a word or a group of words from an individual feature value.

The transformation from message to feature set can be based on words rather than characters, i.e., the sliding window is over the W consecutive words rather than bytes. We choose to use character-based token selection because it is more general than word-based token selection and can be easily implemented. It considers important features, such as message layout symbols, rather than just the text. The character-based selection is also better suited for short messages, such as Picospam and comment spam.

To shuffle the e-mail content in an acceptable manner, our feature-preserving fingerprint scheme adopts a controlled shuffling strategy wherein the tokens are shuffled in a predetermined format. Further, the position of a token after shuffling is always within a fixed range of its original position. Specifically, the controlled shuffling scheme works as follows: The e-mail text is divided into consecutive chunks of tokens. Each chunk consists of z consecutive tokens of the email text, where z is a configurable parameter. The tokens in each chunk are shuffled in a predetermined manner, whereas the ordering of the chunks within the e-mail text remains unaltered. Concretely, each chunk is further divided into y subchunks (we assume that y is a factor of z). The tokens within an arbitrary chunk CK_h are shuffled such that the token at rth position in the sth subchunk is moved to (r *y +s)th position within CK_h.

i)PRIVACY-PRESERVING COLLABORATION PROTOCOL

Our protocol works as follows: When an agent EA_j receives a message M_a, EA_j computes its TFSet : TFSet(M_a). It then sends a query message to other e-mail agents in the system to check whether they can provide any information related to Ma. However, instead of sending the entire TFSet(M_a) as the query message to all agents, EA_j sends a small subset of TFSet(M_a) to a few other e-mail agents (the e-mail agents to which the query is sent is determined on the basis of the underlying structure;). The subsets of TFSet(M_a) included in the queries sent to various other e-mail agents need not be the same (our architecture optimizes the communication costs by sending nonoverlapping subsets to carefully chosen e-mail agents).

An e-mail agent that receives the query, say EA_k, checks its spam and ham knowledge bases looking for entries that include the feature subset that it has received. A feature set is said to match a query message if the set contains all the Fes included in the query. Observe that there could be any number of entries in both spam and ham knowledge bases matching the partial feature set. For each matching entry in the spam knowledge base, EA_k includes the complete TFSet of the entry in its response to EA_j. However, for any matching ham entries, EA_k sends back a small, randomly selected part of the TFset in fig4.

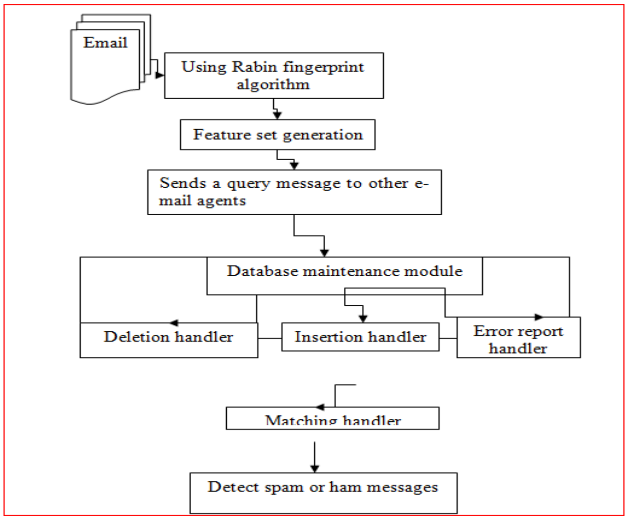

Email
Using Rabin fingerprint algorithm
Feature set generation
Sends a query message to other e-mail agents
Database maintenance module
Deletion handler
Insertion handler
Error report handler
Matching handler
Detect spam or ham messages

Figure 4PRIVACY PRESERVING MECHANISM

D.PERFORMANCE EVALUATION

In this module performance of proposed system is evaluated with the existing system. The performance evaluation consists of the following parameters such as, number of mails, execution time, Probability of matching, false negative, false positive, percentage of message trained. The execution time is also lesser than the time to execute in existing system.In fig5 the results show that the proposed system is more scalable than the existing system.The performance graph shows that the proposed system is more efficient than existing system.

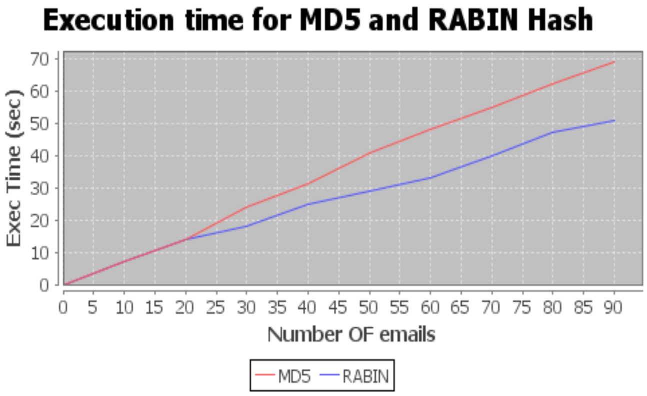


Figure 5 EXECUTION TIME FOR MD5 AND RABIN HASH

A high level of true positive rate is achieved as compared to existing system in fig 6.The spams are detected correctly and accurately by not misclassifying hams as a spams.

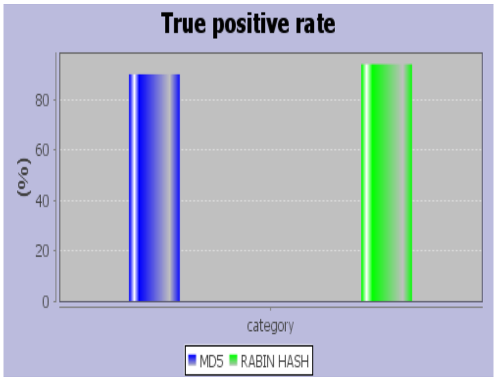


Figure 6 TRUE POSITIVE RATE OF SPAM DETECTION

IV. CONCLUSION

The project is implemented for detecting spam effectively in e-mail by using rabin hash fingerprinting algorithm and about 98% spams are detected efficiently.Privacy of emails is produced by rabin hash fingerprinting algorithm.By collaborative anti spam technique spam mails with their hash ranges are calculated in order to detect spams effectively .A new Algorithm must be proposed inorder to Detect Spams with much accuracy and with simple technique.

REFERENCES

[1] Personalized,"Collaborative Spam Filtering"Alan Gray and Mads Haahr Distributed System Group,Department of Computer cience,Trinity college Dublin,Ireland.

[2] E. Blanzieri and A. Bryl, "Evaluation of the Highest ProbabilitySVM Nearest Neighbor Classifier with Variable Relative ErrorCost," Proc. Fourth Conf. Email and Anti-Spam (CEAS), 2007.

[3] Kang Li, Zhenyu Zhong and Lakshmish Ramaswamy,privacy aware collaboration spam filtering"

[4] P.-A. Chirita, J. Diederich, and W. Nejdl, "Mailrank: UsingRanking for Spam Detection," Proc. 14th ACM Int'l Conf.Information and Knowledge Management (CIKM), pp. 373-380, 2005.

[5] S. Chhabra, W.S. Yerazunis, and C. Siefkes, "Spam Filtering Usinga Markov Random Field Model with Variable WeightingSchemas," Proc. Fourth IEEE Int'l Conf. Data Mining (ICDM),pp. 347-350, 2004.

[6] A.C. Cosoi, "A False Positive Safe Neural Network; The Followersof the Anatrim Waves," Proc. MIT Spam Conf., 2008.

[7] M.-T. Chang, W.-T. Yih, and C. Meek, "Partitioned LogisticRegression for Spam Filtering," Proc. 14th ACM SIGKDD Int'l Conf.Knowledge Discovery and Data mining (KDD), pp. 97-105, 2008.

[8] R. Clayton, "Email Traffic: A Quantitative Snapshot," Proc. of theFourth Conf. Email and Anti-Spam (CEAS), 2007.

[9] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati,"An Open Digest-Based Technique for Spam Detection," Proc. Int'lWorkshop Security in Parallel and Distributed Systems, pp. 559-564,2004.

[10] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati,"P2P-Based Collaborative Spam Detection and Filtering," Proc.Fourth IEEE Int'l Conf. Peer-to-Peer Computing, pp. 176-183, 2004.

[11] P. Desikan and J. Srivastava, "Analyzing Network Traffic toDetect E-Mail Spamming Machines," Proc. ICDM Workshop Privacy and Security Aspects of Data Mining, pp.67-76,2004.

[12] H.Drucker, D. Wu,and V.N. Vapnik, "Support VectorMachines for Spam Categorization," Proc. IEEE Trans. Neural Networks, pp. 1048-1054, 1999.

[13] D.Evett,"Spam Statistics," http://spam-filterreview.top tenreviews.com/spam-statistics.html,2006.

[14] A.Gray and M. Haahr, "Personalised, Collaborative SpamFiltering," Proc. First Conf. Email and Anti-Spam (CEAS),2004.

[15] S.Hershkop and S.J. Stolfo, "Combining Email Models for FalsePositive Reduction," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD),pp.98-107,2005.

[16] J. Hovold, "Naive Bayes Spam Filtering Using Word-Position-Based Attributes," Proc. Second Conf. Email and Anti-Spam (CEAS),2005.

AUTHORS

First Author – Mrs.Latha.K Assistant Professor,Department of IT,Adhiyamaan College Of Engineering,Affiliated to Anna University, Coimbatore,India., Email:klathak2002@gmail.com
Second Author – Nivedha.P,Department of Information Technology Adhiyamaan college of engineering Affiliated to Anna University ,Coimbatore, India., Email:nivedha332@gmail.com
Third Author – Menagagandhi.G,Department of Information Technology Adhiyamaan college of engineering Affiliated to Anna University,Coimbatore,India., Email:menaga90@gmail.com
Fourth Author – Ramya.T,Department of Information Technology Adhiyamaan college of engineering Affiliated to Anna University,Coimbatore,India., Email:ramyanadham@gmail.com