Machine Learning Framework To Analyze Against Spear Phishing

J. Vijaya Chandra, Narasimham Challa, Sai Kiran Pasupuletti

Abstract**: *The objective of this paper is to design and implement machine learning based ensemble algorithm on dataset to fit into the models that can be understood and executed by machines. In this paper we discussed different algorithms and machine learning concepts that can be implemented on the datasets, we taken email spam filter dataset for experiment and analysis, as the Advanced persistent threat the latest threat is intruded using the emails and major intrusion is done through spam emails. Machine learning uses different datamining techniques and mechanisms and accepts the input-data and gives the output as the statistical analysis. We implemented different email classification algorithms on the datasets based on spam and ham emails where spear phishing methods are identified and implemented different classification and regression methods to get the accurate results. In this paper for the better results in spite of existing algorithms we introduced the ensemble methods such as boosting, bagging, stacking and voting for much accuracy and higher level of classification and combining different algorithm. This paper will measure different machine learning algorithms performance on spam email filtering on the huge datasets. The framework provides implementation of learning algorithms that you can apply to larger datasets. An obvious approach to making decisions more reliable is to combine the output of different models. We even compared the existing algorithms and proposed algorithm; comparison tables are drawn along with the statistical analysis, data and graphical analysis is given.***

Keywords: Advanced Persistent Threat, Spear phishing, Email classification, Machine Learning, Data mining.

I. INTRODUCTION

The Email spam can be filtered based on the previous experiences and the list created for identification, the good senders list is known as the white list, bad senders list is known as the blacklist, a spam message list is known as the fingerprint lists. These lists are established and maintained commonly all over the world, the grey listing is apart from other methods it capturers some behaviour of the sender and assumes that a message is spam, based on the directions such as delay in delivery, network traffic, risk of message loss in the real time environment [1].

The attacks on email based on the content filtering are differentiated as tokenisation, obfuscation, weak statistical and strong statistical. The tokenisation is usually targets at feature selection step of spam classification where it modifies or splits a word for example such attack is “free” can be split as “f r e e”, the next method of attack is obfuscation which uses to hide features from the classifier by adding codes like HTML that is concealed from a user. For example: word “free” could be changed to “fr <! . .><..>ee or fr$#101xe or FR3E”. The attack that is next to the weak statistical attack is usuallyS known as passive attack because it is done without any direct feedback from the filter. It modifies the statistics of an email file for mismatching the statistics of a spam file. It is done by adding arbitrary words or whole chunk of text in the message. Finally, the Strong Statistical attack instead of just adding the words or random text in the message, it alters the information within the spam portion of the message. This attack accesses the direct feedback from the spam filter and known as an active attack. Statistical based attacks constitute dictionary attacks, frequent word attacks, and the frequency ratio attacks [2].

Machine learning research for email spam classifier is classified into binary classification, multi-class classification and multi-labelled classification. Where the binary classification is the problem of classifying observations into two possible classes that are spam and ham, one generally given example is email spam filtering, which identifies email messages. The most efficient algorithm to filter spam uses machine learning techniques, most spam filtering methods uses text techniques, therefore most of the problems are related to classification, the present study classifies rules to extract features from an email [3].

The aim is to identify, design and develop reliable classifier to get maximum accuracy in spam classification of electronic mail system. A brief review is done on existing email classification algorithms on datasets, the accuracy of the email classification is verified by using different parameters such as True Positive, False Positive, False Negative and True Negative. The accuracy is calculated by the total number of correct classifications either as the class of interest that is true positive and reverse of it that is true negative, where accuracy is measured based on formula the sum of True Positive and True Negative by sum of True Positive, True Negative, False Positive and False Negative. The different parameters based on evaluation metrices are time taken to build the model using the specified algorithm, correctly classified instances and in correctly classified instances, also their percentages also calculated [4].

Revised Manuscript Received on October 5, 2019.

***** Correspondence Author

J.Vijaya Chandra*, Research Scholar, Dept of CSE, KLEF (Deemed to be University), Guntur, Andhrapradesh, India,

Dr. Narasimham Challa, Dean IQAC and Professor, Dept of CSE, Vignan’s Institute of Technology and Science, Vishakapatnam, A.P., India.

Dr.Sai Kiran Pasupuleti, Professor, Dept of CSE, KLEF(Deemed to be University), Guntur, Andhrapradesh, India.

II. RELATED WORK

WEKA is a popular machine learning workbench, contains implementations of algorithms for classification, clustering, and association rule mining along with graphical user interface and visualization utilities for data exploration and algorithm evaluation. The major functionality of the WEKA is data processing, classification, clustering, Attribute selection and Data visualization. Data mining and machine learning are the concepts related to artificial intelligence that focus on pattern discovery, prediction and forecasting based on properties of collected data, WEKA is a tool that can be compactable with java programming, attribute-relation file format is the data storage and linking system with the applications in java where background process is done by the java virtual machine. One of WEKA's major strengths is that it is easily extended with customized or new classifiers, clusters, attribute selection methods, and other components.

Cloud computing provides convenient data and information storage systems, due to targeted cyber-attacks on cloud, Industrialists and Researchers all over the world concentrated on the cloud security, as cloud provides convenient data storage system for industry, government, researchers and academic users with cost-effective access to distributed services through internet, the main challenge is the data privacy and information security [5]. In 2015 Saranya Chandran et.al focused on the real-world problem that is advanced persistent threat and identification of an efficient classification model for detection of APT using different classifiers such as Random Forest, Naïve Bayes and J-48 [6].

According to the Radicati Group Investigation the total number of world-wide E-mail accounts is expected to increase from 4.3 billion accounts in 2016 to over 5.3 billion accounts in 2018. The unwanted (spam) email growth is increasing significantly because many companies and industries are using digital form of advertisement by sending bulk email messages. Emails are used to spread viruses, worms, Trojans, Spyware etc., through attachments. The spam Emails are categorizing different forms such as E-mail Frauds, Chain letter spam, Malicious mails and commercial Advertisements.

Spam is an un-solicited Junk mail by the companies to sell their product or services or exploit the organization intent to do fraud, spammer takes a database of list and sent emails built in thousands as bulk mails, few will in trap of spammer. Spam Mail wastes a great amount of bandwidth and space of both the ISP of senders and receivers. The most common precautions are Ignore Emails from unknown senders, be careful with delivery failure emails, don't give your primary email address to marketing agencies or in websites for any type of promotions.

Targeted Malicious Emails are concern designed to capture sensitive confidential data from victims. These attacks are not only misleading inexperienced end users but also technically aware end users, mainly targeted email service providers employers. The mails do not ask directly the email addresses in the servers but asks or provoke or insists to click on provided links in the emails that contains malware which infect not only single employee system but maps to spread all the systems of the company to elevate privileges and company resources.

They even use social engineering and psychological technologies such as humans will not read every letter in a word, they read sequentially and randomly, for example a victim may not find difference in between support@microsoft.com and support@mircosoft.com where the only two letters r and c are swapped. Thus, the attackers may benefit without the knowledge of the victims [7].

The spam filtering methods were experimented and investigated by users, practitioners, vendors and researchers and classified into three groups that is manual inspection, system-oriented approaches and content-based filtering. The content-based filtering is further classified as Ad-hoc Rule-based filters, Practical learning filters and machine learning research. The generally used test classifiers in research are Bayesian classifier, Naïve Bayes, Perceptron, Winnow, Support Vector Machine and the clustering methods such as decision tree and nearest neighbor mechanisms under Machine Learning Research.

spam filtering incorporates two approaches that is header-based and content-based features for filtering. The header-based filtering methods uses information available in the header of email, the solution is commonly used by verifying the sender email by adding the block listing and applied to block untrusted emails or spams. content-based filtering- uses features mined from the message contents. The mechanism used here is machine learning (ML) principles to extract knowledge based on artificial intelligence mechanisms from a set of messages supplied and use the obtained knowledge in the classification of newly received messages and classifies either the message is a Spam or Ham.

Unwanted emails that are sent anonymously, mass mailed and are unsolicited fall in the category of spam. Even though they do not affect directly the privacy, they end up occupying a large chunk of your inbox capacity. Some hackers use spam email to extract information from the user, this information may be financial, personal or could be professional. Sometimes the spam emails look so authentic that it's difficult for a novice to differentiate it from useful emails. Victims of spam emails have collectively lost more than $20 billion in the past year [8].

The spam emails are categorized as spam-vertised sites, phishing, 419 scams and Image spam, the spam-vertised sites are as the name suggests that these are kind of emails that advertise products. They contain URLs to other websites. Phishing is extracting financial information from the users by asking users to enter their legitimate bank information on the fake websites, 419 scams area kind of spam emails offer huge sum of money to the users in return for a small initial payment. Image spams are the kind of emails, the content or body of the message is displayed as a GIF or JPEG image.

For Spam detection, system approaches work on the information extrinsic to the message and the user. These approaches are applied before delivery of the message to the end user. Collaborative filtering is an approach which exploits the facts that similar email spam is sent to many end users. Captures email spam for identifying the redundancy over many systems. If the message is received from email address that have never used for legitimate emails, it will consider as spam. Due to the bulky nature of spam email, it is difficult to store all messages. As the number of messages will increase, decisions will be more complicated and time consuming.Challenge Response is another system for spam filtering is challenge response, which include a cost of sending an email. This cost is decided in such a fashion that it is easy for a legitimate sender but costly and time consuming for a spammer to send millions of spams. The cost includes asking the user to resend, clicking on a link and making a payment. In this system, an email is delivered with some instructions like how to respond, which add difficulty for legitimate user.Spear phishing is a targeted form of phishing in which fraudulent emails only target a small group of selected recipients. Sender sends bulk or a group of mails regarding business promotion or malicious mails intentionally sent to attack through Email Service Provider Servers. Email Firewalls filters the malicious mails based on previous feedback which is stored at spam databases and block-list based on the feedbacks at DNS web Servers. Emails are again filtered at the Email Servers and if found as spam they will send to the spam folder [9]. Email filtering is the process by which email is sorted by specific criteria. In the case of email spam filtering, it means filtering out the unsolicited and unwanted emails which clog up your inbox and email server. With the vast number of emails sent and received daily, email filtering has become vital for processing incoming and outgoing emails in accordance with anti-spam techniques.

III. EXPERIMENTAL ANALYSIS USING WEKA TOOL

The WEKA suite has three separate graphical interfaces the Explorer, Experimenter and the Knowledge Flow. WEKA uses its own file format called Attribute Relationship File Format (ARFF). The ARFF format presents the data according to the requirement of the applied algorithms during the modelling stage. Pre-processing stage saves memory space and training time. In it first, all the irrelevant features are filtered out and then stemming is applied to stem out some of the generic terms. ARFF file has three primary sections namely @relation, @attribute and @data [10].

In the ARFF file, the first line will be the relation declaration, where it simply defines the name of the data set, next attribute declarations where the declarations are an ordered sequence of statements in which each attribute must have its own exclusive @attribute statement. The statement will uniquely define the name of that attribute and the data type associated with that attribute. The different attributes that are used in WEKA are numeric attributes that are real and integer values, nominal attributes, string attributes and date attributes. Finally, data declarations the software expects that all that attributes values will be found in a comma delimited manner in the same order that they were defined in the attribute declaration section.

WEKA is an open source package mainly mean for the pre-processing, classifiers, clustering and association rule, created by the researchers at university of Waikato in New Zealand and it is compatible with java. The interface can link with email information to gather the information for pre-processing then generate the coaching and look at data sets then we tend to convert each set into rail format. We tend to pass coaching set to the rail library to coach the classifier then look at the effectiveness with look at set, The Spam Email Classification.

IV. KEY TERMS EVALUATING SPAM AND HAM

- **True Positive (TP) Rate: Based on the Algorithm for classification, the percentage of emails** of phishing are correctly classified, the number of phishing emails in the dataset is given by **p** and the **Np is the number of correctly classified phishing emails and the true positive rate TP** is given by **TP= Np/p**
- **True Negative (TN) Rate: legitimate emails percentage is calculated by using algorithm that** were correctly classified as legitimate. NL is denoted as the number of legitimate emails that were correctly classified as legitimate and L is the total number of legitimate emails, then finally **TN = NL/L**
- **False Positive (FP) Rate: It calculates the percentage of legitimate emails that were incorrectly** classified by the algorithm as phishing emails. If we denote the number of legitimate emails that were incorrectly classified as phishing by **Nf, and the total number of legitimate emails as L, then FP = Nf/L**
- **False Negative (FN) Rate: It calculates the number of incorrectly classified phishing emails** by the algorithm. If we donate the number of phishing emails that were classified as legitimate by the algorithm by **Npl and the total number of phishing emails in the data set is denoted by p, then FN = Npl / p**
- **Precision: is used to calculate the exactness of the classifier; which identifies the percentage** of emails that the classifier is labelled as phishing, where exactly the phishing emails are identified, and it is given by: **Precision = Tp/ Tp + Fp**
- **Recall: It calculates the total completeness of the classifier results; which includes what** percentage of phishing emails identified by the classifier label as phishing, and is given by: **Recall = TP/ TP + FN**
- **F-measure: It is used to identify the harmonic mean of Precision and Recall, and known as Fscore,** is defined as and given by**:**

F-measure = 2*Precision*Recall / Precision + Recall

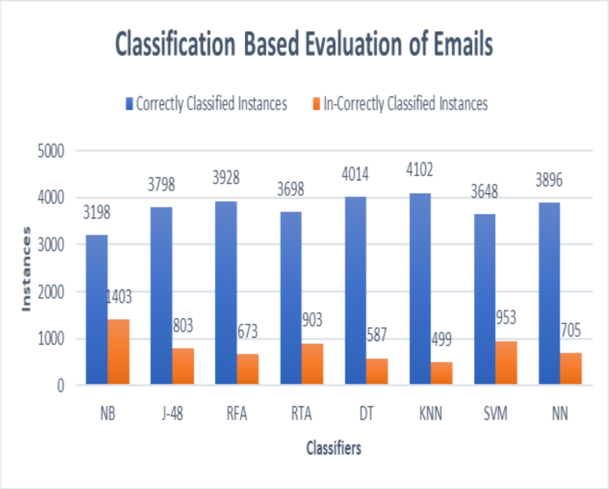
- **Receiver Operating Characteristic Area (ROC): it is a milestone that calculates to** demonstrates the accuracy of a binary classifier by plotting TP against FP at various threshold values

V. EVALUATION METRICES

The Evaluation Metrices with Naïve Bayes, J-48 classifier, Random Forest Algorithm, Random Tree Algorithm, Decision Tree, K-Nearest Neighbours, Support Vector Machines and Neural Network multi-layer Perception.

Table 3.2: Evaluation Matrices using different classifiers

Evaluation Metrices	NB	J-48	RFA	RTA	DT	KNN	NN	SVM
Time taken to build model	0.577	0.84 sec	2.15 sec	0.07 sec	0.03 sec	3.73 sec	0.13 sec	0.1 sec
Correctly classified Instances	3198	3798	3928	3698	4014	4102	3896	3648
Percentage of Correctly classified Instances	69.50663%	82.54727%	85.37275%	80.37383%	87.24190%	89.15454%	84.67725%	79.28711%
In Correctly classified Instances	1403	803	673	903	587	499	705	953
Percentage of In Correctly classified	30.49337%	17.45272%	14.62725%	19.62617%	12.75810%	10.84546%	15.32275%	20.71299%



Graph 3.3: Graphical Analysis of Classification Based Evaluation of different Classifiers

VI. BUILDING A CLASSIFIER AND PRACTICAL ANALYSIS OF JAVA-WEKA PROGRAM

Eclipse is a free and opensource Integrated Development Environment (IDE) that is an application that provides source code editor, compiler, debugger and many other features, it is significantly increase programmer productivity and simplifies many tasks. After creating a new java project, we must build path to access java and add the External JARs that supports WEKA [11].

Designing a new classifier we extend each classifier weka.classifers.classifier abstract class, as it imports enumeration, to define the capabilities of classifier we need to possess the weka.core.capabilities class and some constants from the same class, that is import weka.core.Capabilities and import weka.core.Capabilites.Capability. We also import the Instance and Instances class, the next step is creating a new class with a classifier name that extends the weka.classifiers.Classifier abstract class. If we want the classifier to be incremental, make sure that we implement the weka.classifiers.updatable classifier interface and the updateClassifier(Instance) method.

The major steps involved in this snippet are loading the dataset, executing the dataset and finally displaying the classification. The classification model displays, Correctly Classified Instances, Incorrectly Classified Instances, Kappa Statistic, Mean absolute error, Root mean squared error, Relative absolute error, Root relative squared error, Total Number of Instances.

The classification models can be combined for the better results, these methods are boosting, bagging, stacking and voting. Combining classifiers methods are possible for better performance and reduction of the error. Boosting is an ensemble technique that attempts to create a strong classifier from the number of weakclassifiers. AdaBoost ensemble method for machine learning provides more accurate and efficient results. We import and extend the classifiers as shown in the below part of java snippet.

```
import weka.classifiers.meta.AdaBoostM1;
import weka.classifiers.meta.Bagging;
import weka.classifiers.meta.Stacking;
import weka.classifiers.meta.Vote;
```

To denomistrate the recipe, we will use a dataset that contain the spam & ham data that is “spambase. Arff”. The part of next code involves with loading data set and getting instances object, finally declare and set the class index as follows.

```
String data = "/home/spambase.arff";
  DataSource source = new
DataSource(data);
  Instances tData = source.getDataSet();

if (tData.classIndex() == -1) {
  tData.setClassIndex((tData.numAttribut
es() -1);
}
```

First, we inherit all possible capabilities a classifier can handle, as classifier to be able to handle numeric and nominal attributes only, we Enable them by setting to required constants from the weka. core. capabilities. Capability class using enable (enum capability) method. Next enable the classifier to handle nominal class values. Specify that it needed training instances with the method and uses different additional methods such as AdaBoost, bagger, stacker and voter, finally return the capabilities object and gives result.

```
AdaBoostM1 m1 = new AdaBoostM1();
m1.setClassifier(new NaiveBayes());
m1.setNumIterations(20);
m1.buildclassifer(tData);
Bagging bagger = new Bagging();
bagger.setClassifier(new RandomTree());
Stacking stacker = new Stacking();
stacker.setMetaClassifier(new J48());
Classifier[] classifers = {
new J48[],
new NaiveBayes(),
new RandomForest()
};
```

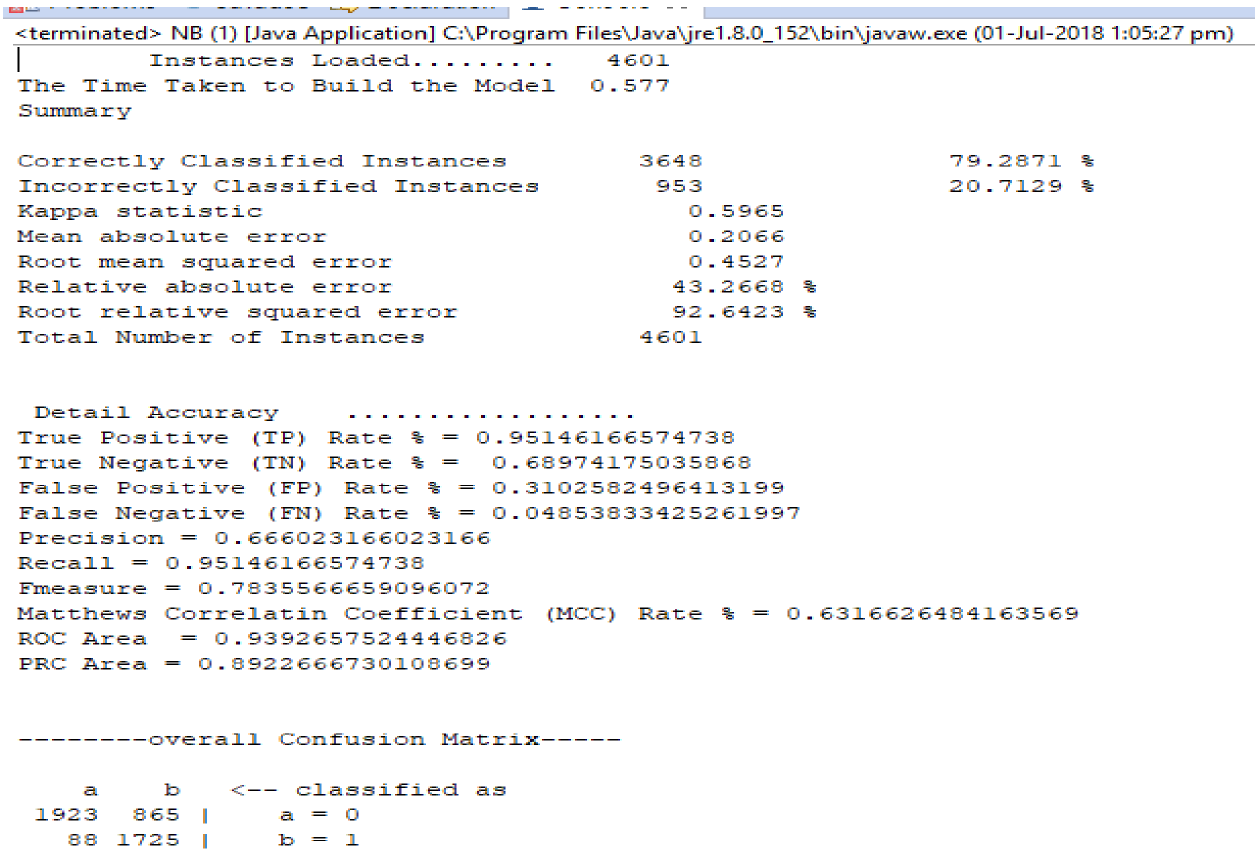
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

```
stacker.setclassifier(classifiers);
stacker.buildClassifier(tData);
vote voter = new Vote();
voter.setClassifier(classifiers);
voter.buildclassifer(tData);
}
}
```

The classification is a predictive modelling, the classification is one of the major task of the supervised learning, where responsibility of the classification model is to assign class label to the target feature based on the value of the predictor features, where the basic steps involved in machine learning process are accepting the data, preparing a model and at the next step learning then performance evaluation and finally performance improvement identification. We did compare analysis of different models available based on machine learning algorithms to develop accurate and reliable classifier we used different mechanisms such as booster, bagger, stacker and voter. We adopted ensemble learning mechanism where the process is done by which multiple models are combined based on the requirements and quality analysis, where a hybrid mechanism is designed to solve a problem based on conditional probability were precise computational intelligence-based mechanism is used. Based on the evaluation process done on different classification-based algorithms, we find the k neareast neighbors algorithms uses effective implementations and matches the new pattens during the prediction efiicient and found the better among the available algorithms [12].

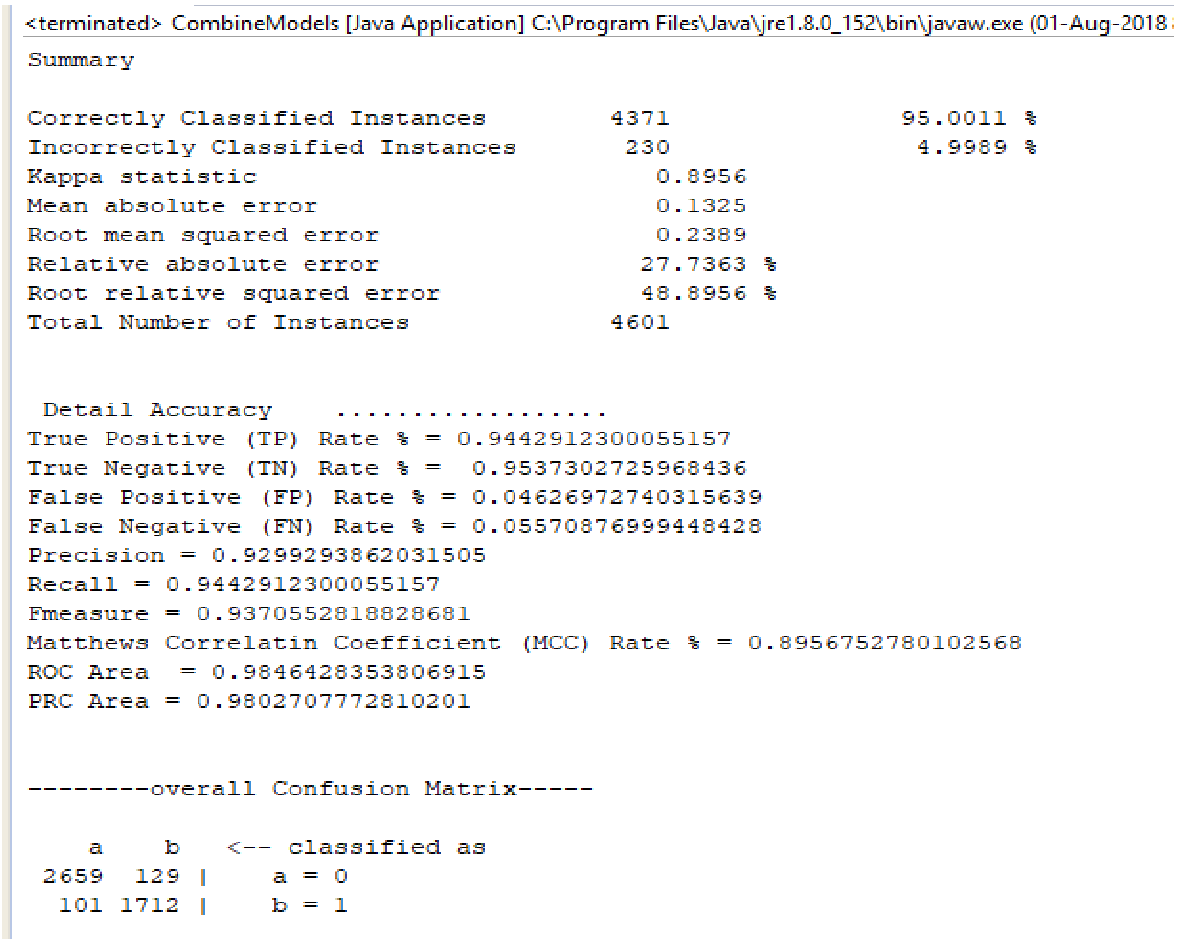
CLASSIFIER	**HAM**			**SPAM**			**ACCURACY %**
	Precision	Recall	F-Measure	Precision	Recall	F-Measure	
NAVIES BAYES	0.691	0.712	0.701	0.683	0.727	0.704	**69**
J-48	0.821	0.839	0.829	0.816	0.824	0.819	**82**
RANDOM FOREST	0.853	0.867	0.859	0.812	0.848	0.829	**85**
RANDOM TREE	0.801	0.812	0.806	0.799	0.811	0.804	**80**
K-NEAREST NEGIBOURS	0.873	0.884	0.878	0.889	0.861	0.874	**87**
SUPPORT VECTOR MACHINES	0.793	0.803	0.797	0.666	0.951	0.783	**79**
NEURAL NETWORKS	0.841	0.851	0.845	0.811	0.821	0.839	**84**
DESIGNED CLASSIFIER	0.953	0.963	0.957	0.944	0.929	0.936	**95**

Table 5.2: The Classification Results on Dataset in-terms of Precision, Recall and F-measure and Accuracy



```
<terminated> NB (1) [Java Application] C:\Program Files\Java\jre1.8.0_152\bin\javaw.exe (01-Jul-2018 1:05:27 pm)
        Instances Loaded.........    4601
The Time Taken to Build the Model   0.577
Summary

Correctly Classified Instances        3648               79.2871 %
Incorrectly Classified Instances       953               20.7129 %
Kappa statistic                          0.5965
Mean absolute error                      0.2066
Root mean squared error                  0.4527
Relative absolute error                 43.2668 %
Root relative squared error             92.6423 %
Total Number of Instances             4601


 Detail Accuracy    ..................
True Positive (TP) Rate % = 0.9514616657473 8
True Negative (TN) Rate % =  0.68974175035868
False Positive (FP) Rate % = 0.3102582496413199
False Negative (FN) Rate % = 0.04853833425261997
Precision = 0.666023166023166
Recall = 0.9514616657473 8
Fmeasure = 0.7835566659096072
Matthews Correlatin Coefficient (MCC) Rate % = 0.6316626484163569
ROC Area  = 0.9392657524446826
PRC Area = 0.8922666730108699


--------overall Confusion Matrix-----

    a    b   <-- classified as
 1923  865 |    a = 0
   88 1725 |    b = 1
```

Result 4.1: The Result of Evaluation of a Classifier



```
<terminated> CombineModels [Java Application] C:\Program Files\Java\jre1.8.0_152\bin\javaw.exe (01-Aug-2018
Summary

Correctly Classified Instances        4371               95.0011 %
Incorrectly Classified Instances       230                4.9989 %
Kappa statistic                          0.8956
Mean absolute error                      0.1325
Root mean squared error                  0.2389
Relative absolute error                 27.7363 %
Root relative squared error             48.8956 %
Total Number of Instances             4601


 Detail Accuracy    ..................
True Positive (TP) Rate % = 0.9442912300055157
True Negative (TN) Rate % =  0.9537302725968436
False Positive (FP) Rate % = 0.04626972740315639
False Negative (FN) Rate % = 0.05570876999448428
Precision = 0.9299293862031505
Recall = 0.9442912300055157
Fmeasure = 0.9370552818828681
Matthews Correlatin Coefficient (MCC) Rate % = 0.8956752780102568
ROC Area  = 0.9846428353806915
PRC Area = 0.9802707772810201


--------overall Confusion Matrix-----

    a    b   <-- classified as
 2659  129 |    a = 0
  101 1712 |    b = 1
```

Result 4.2: The Result of Evaluation of a Combined Classifier

VII. CONCLUSION

The classification results of the proposed combined classifier scheme with that of individual classifiers on Spambase dataset to calculate Precision, Recall, F-measure and finally Accuracy is calculated. As a Sample we taken the Accuracy related values for only three classifiers to do comparative study for the designed and implemented classifier [13].

The main objective of this paper is to identify the best classifiers sot that their individual decisions can be combined to get extreme classification output in spam classification domain. The aim of this research is to develop a robust, fast, accurate, sensitive and customizable content-based spam filtering model that could cater the basic need of the organisations and the internet service providers. As part of research identified the characteristics of classifiers and existing machine learning algorithms and combining methods that can be used for the better results based on the Accuracy [14].

REFERENCES

1. J. V. Chandra, N. Challa and S. K. Pasupuleti, "A practical approach to E-mail spam filters to protect data from advanced persistent threat," *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Nagercoil, 2016, pp. 1-5.
2. M. K. Chae, A. Alsadoon, P. W. C. Prasad and A. Elchouemi, "Spam filtering email classification (SFECM) using gain and graph mining algorithm," *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2017, pp. 1-7.
3. A. W. Wijayanto and Takdir, "Fighting cyber crime in email spamming: An evaluation of fuzzy clustering approach to classify spam messages," *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, Bandung, 2014, pp. 19-24.K. Chen, Linear Networks and Systems (Book style).Belmont, CA: Wadsworth, 1993, pp. 123–135.
4. J. Song, D. Inque, M. Eto, H. C. Kim and K. Nakao, "An Empirical Study of Spam: Analyzing Spam Sending Systems and Malicious Web Servers," *2010 10th IEEE/IPSJ International Symposium on Applications and the Internet*, Seoul, 2010, pp. 257-260.
5. J. V. Chandra, N. Challa and S. K. Pasupuleti, "Advanced Persistent Threat defense system using self-destructive mechanism for Cloud Security," *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, Coimbatore, 2016, pp. 7-11.
6. S. Chandran, Hrudya P and P. Poornachandran, "An efficient classification model for detecting advanced persistent threat," *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, 2015, pp. 2001-2009.
7. Chandra, J. Vijaya, Narasimham Challa and Sai Kiran Pasupuleti. "Intelligence based Defense System to Protect from Advanced Persistent Threat by means of Social Engineering on Social Cloud Platform." (2015).
8. A. K. Pandey, D. S. Rajpoot and D. S. Rajpoot, "A comparative study of classification techniques by utilizing WEKA," 2016 International Conference on Signal Processing and Communication (ICSC), Noida, 2016, pp. 219-224.
9. S. K. Trivedi and S. Dey, "A Combining Classifiers Approach for Detecting Email Spams," 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Crans-Montana, 2016, pp. 355-360.
10. H. I. Bulbul and Ö. Unsal, "Comparison of Classification Techniques used in Machine Learning as Applied on Vocational Guidance Data," 2011 10th International Conference on Machine Learning and Applications and Workshops, Honolulu, HI, 2011, pp. 298-301.
11. P. Chandrasekar, K. Qian, H. Shahriar and P. Bhattacharya, "Improving the Prediction Accuracy of Decision Tree Mining with Data Preprocessing," 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, 2017, pp. 481-484.
12. A. Gahlaut, Tushar and P. K. Singh, "Prediction analysis of risky credit using Data mining classification models," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-7.
13. H. Kadkhodaei and A. M. E. Moghadam, "An entropy-based approach to find the best combination of the base classifiers in ensemble classifiers based on stack generalization," *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, Qazvin, 2016, pp. 425-429.
14. A. J. Ibrahim, M. M. Siraj and M. M. Din, "Ensemble classifiers for spam review detection," *2017 IEEE Conference on Application, Information and Network Security (AINS)*, Miri, 2017, pp. 130-134.

AUTHORS PROFILE

J. Vijaya Chandra is a Research Scholar at KLEF (Deemed to be University); Koneru Lakshmaiah Education Foundation. Research areas are Cloud Security, Network Security, Intelligence Security and Data Security. Published 10 Research Papers for International Journals. He is Oracle Certified Associate and Member of IEEE and ACM

Dr. Narasimham Challa, Professor of Computer Science Engineering & Dean IQAC and Former Principal of Vignan's Institute of Information Technology have been working in the field of Teaching and Research for the last 24 years. The Author received Ph D in Computer Science in the year 2009. Guided three Ph D scholars and guiding four research scholars. Published 79 research articles in various National and International journals. At present, registered for Post-Doctoral work leading to D. Sc with UoS Panama and doing work in the area of verifiable encryption, cryptography and security.

Dr. Sai Kiran Pasupuleti, Ph.D., is Professor, Dept. of Computer Science and Engineering, KLEF (Deemed to be Univesity), Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur District, A.P., INDIA. He is having rich teaching and Research Experience. His research areas are Mobile Computing, Cloud Computing and Computer Networks. He published about 30 Research Papers in International Journals.