

Detection of Poisoning Attacks using Quasi Newton Back Propagation Algorithm

Nayan Kumawat¹; Ravi Khatri²

M.Tech Scholar (Cyber Security)¹, Professor and Head (Computer Science)², VITM Indore^{1,2}

nayan_kumawat555@yahoo.in¹; k.ravi@vitmindore.com²

ABSTRACT

With the increasing popularity of neural networks for classification, the chances of poisoning attacks have also increased. One major area of research that has emerged is the detection of poisoning attacks using neural networks due to the complexity of data set of attacks. Several approaches have been used so far for the effective classification of poisoning attacks. Sophisticated attackers have strong incentives to manipulate the results and models generated by machine learning algorithms to achieve their objectives. The proposed work utilizes the Scaled Conjugate Gradient based back propagation approach for detection of poisoning attacks. It has been found that the proposed approach achieves higher rate of accuracy compared to previously existing systems [1].

Keywords: Poisoning Attack, Scaled Conjugate Gradient (SCG), Mean Square Error (MSE),

1. INTRODUCTION

With the increasing popularity of neural networks for classification, the chances of poisoning attacks have also increased. One major area of research that has emerged is the detection of poisoning for android systems using neural networks due to the complexity of data set of attacks. Several approaches have been used so far for the effective classification of poisoning attacks. Sophisticated attackers have strong incentives to manipulate the results and models generated by machine learning algorithms to achieve their objectives. For instance, attackers can deliberately influence the training dataset to manipulate the results of a predictive model in poisoning attacks. It can be inferred that these attacks become easier to mount today as many machine learning models need to be updated regularly to account for continuously-generated data. Such scenarios require online training, in which machine learning models are updated based

on new incoming training data. The poisoning attack tries to trick machine learning approaches which are rapidly emerging as a vital tool in a variety of networking and large-scale system applications because they can infer hidden patterns in large complicated datasets, adapt to new behaviors, and provide statistical soundness to decision making processes. Application developers thus can employ learning to help solve big data problems and these include a number of security-related problems particularly focusing on identifying malicious or irregular behavior. In fact, learning approaches have already been used or proposed as solutions to a number of such security-sensitive tasks including spam, worm, intrusion and fraud detection.

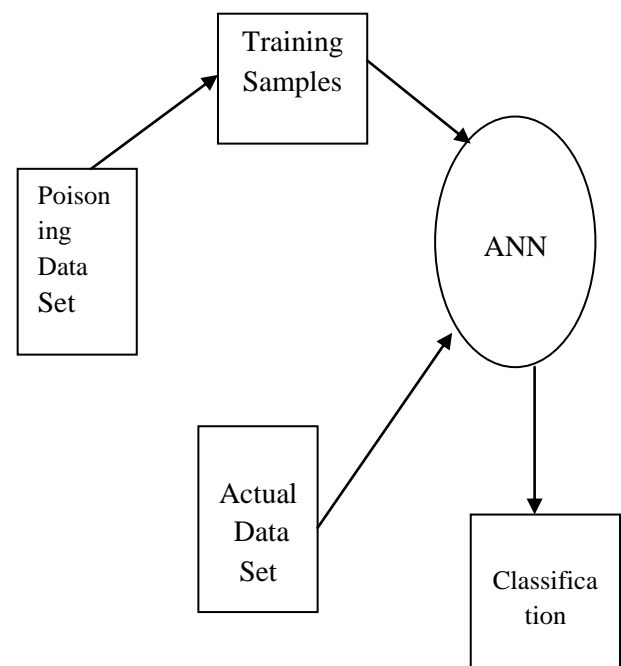


Fig 1: Concept of the poisoning attack

The poisoning attack can be mathematically summarized as:

Let the training vector be:

$$\text{Training Data} = X(i) \quad (1)$$

Manipulating the training vector is done using the poisoning vector given by:

$$X_v = V(i) \quad (2)$$

The weights of the system are governed the training vector and learning algorithm, given by:

$$w_k = f(X_v, k, f_a) \quad (3)$$

Here,

$X(i)$ is the real training input

$V(i)$ is the poisoning vector

k is the number of iteration

f_a is the activation function

w_k is the weight for iteration k .

This attack is especially important from the practical point of view, as an attacker usually cannot directly access an existing training database but may provide new training data; e.g., web-based repositories and honeypots often collect malware examples for training, which provides an opportunity for the adversary to poison the training data. Poisoning attacks have been previously studied only for simple anomaly detection methods.

2. RELATED WORK

In 2018 IEEE, Matthew Jagielski et al. in [1] showed that as machine learning becomes widely used for automated decisions, attackers have strong incentives to manipulate the results and models generated by machine learning algorithms. In this paper, authors perform the first systematic study of poisoning attacks and their countermeasures for linear regression models. In poisoning attacks, attackers deliberately influence the training data to manipulate the results of a predictive model. Authors propose a theoretically-grounded optimization framework specifically designed for linear regression and demonstrate its effectiveness on a range of datasets and models.

In 2018 Elsevier, Sen Chen et al. in [2] proposed the feasibility of constructing crafted malware samples; examine how machine-learning classifiers can be misled under three different models; then conclude that injecting carefully crafted data into training data can significantly reduce detection accuracy. To tackle the problem, the authors propose KUAFUDET, a two-

phase learning enhancing approach that learns mobile malware by adversarial detection. KUAFUDET includes an offline training phase that selects and extracts features from the training set, and an online detection phase that utilizes the classifier trained by the first phase. To further address the adversarial environment, these two phases are intertwined through a self-adaptive learning scheme, wherein an automated camouflage detector is introduced to filter the suspicious false negatives and feed them back into the training phase.

In 2017 IEEE, Chao Chen et al. in [3] authors address the non-iid setting of time series forecasting. Authors consider a forecaster, Bob, using a fixed, known model and a recursive forecasting method. An adversary, Alice, aims to pull Bob's forecasts toward her desired target series, and may exercise limited influence on the initial values fed into Bob's model. The authors consider the class of linear autoregressive models, and a flexible framework of encoding Alice's desires and constraints. They describe a method of calculating Alice's optimal attack that is computationally tractable, and empirically demonstrate its effectiveness compared to random and greedy baselines on synthetic and real world time series data.

In 2017 IEEE, Nida Mirza et al. in [4] proposed a technique for spam classification based on hybrid feature selection. The major advantage of this approach was the fact that the hybrid parameters can be an amalgamation of both textual features and non-textual features. The evaluation of the performance of the proposed system was done on the basis of mean square error, hit rate and the accuracy. The performance of hybrid feature selection was shown to be better than the average features computation algorithms.

In 2016 IEEE, Scott Alfeld et al. in [5] proposed a mechanism for the classification of bi-lingual tweets using machine learning algorithms. The methodology of the system was the use of natural language processing and thereafter the use of deep neural networks with multiple hidden layers. The learning rates were dependent on the differential changes in the architecture of the neural network used.

3. Artificial Neural Networks

The entire mathematical model of the neural network can be explained as:

$$y = \sum_{k=1}^{k=n} x_k w_k + \emptyset \quad (3)$$

x represents the parallel inputs
 y represents the output of the ANN
 \emptyset represents the bias
 W represents the weights associated with the inputs received the ANN.

In the recent years there has been much big technological advancement in terms of automation. The one area which has been showing tremendous growth and application potential is the field of Artificial Intelligence. Artificial Intelligence is the field where machines show the capability of doing many tasks that the humans can perform.

Usually computers are programmed to do work or function in a certain way by getting instructed through a set of instruction codes. But humans are different, in the sense that they possess a certain self learning ability which makes their thinking process unique. So use of artificial intelligence prudently in a broad spectrum of applications can be very useful in terms of performance and efficiency.

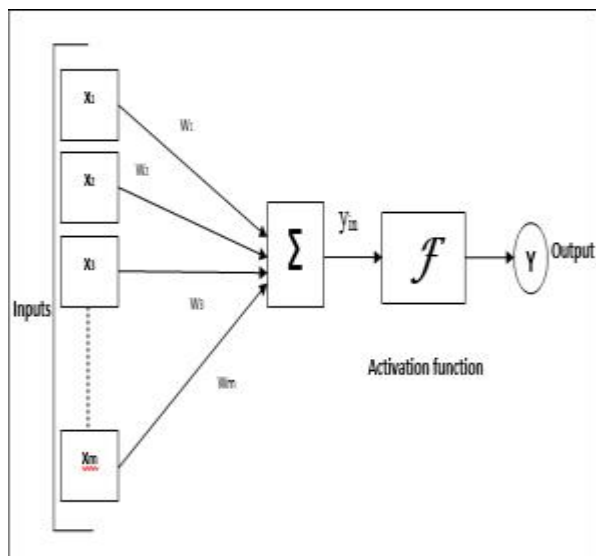


Fig 2: Mathematical Model of ANN

The implementation of neural network is defined in two phases' first training and second prediction: training method utilizes data and designs the data model. By this data model next phase prediction of values is performed [18].

Training:

1. Prepare two arrays, one is input and hidden unit and the second is output unit.
2. Here first is a two dimensional array W_{ij} is used and

output is a one dimensional array Y_i .

Testing:

In this phase the neural network is tested for the output rendered by the trained network. In this phase, the accuracy of the system is computed.

4. PROPOSED METHODOLOGY

The proposed approach uses the Quasi Newton back propagation approach which is relatively fast and accurate compared to other back propagation algorithms.

The mathematical modeling of the Quasi Newton approach is given in the subsequent section.

This method of backpropagation is an improved version of conjugate gradient method. It makes use of a Hessian matrix $[A_k]^{-1}$ consisting of the performance index for the current weights and bias parameters. The BFGS method is an alternative to the conjugate gradient methods for fast optimization. Newton's method often converges faster than conjugate gradient methods. The weight update for the Newton's method is:

$$W_{k+1} = W_k - A_k^{-1} g_k \tag{4}$$

Here,

k is the iteration number

w_k is weight of present iteration (k)

w_{k+1} is the weight of next iteration ($k+1$)

g_k is the gradient vector given by $\frac{\partial e}{\partial w}$

A_k^{-1} is the inverse of the Hessian Matrix, which is the second order derivative of errors with respect to weights

The Hessian Matrix is given by:

$$H = \begin{bmatrix} \frac{\partial^2 e}{\partial x_1 \partial w_1} & \dots & \frac{\partial^2 e}{\partial x_1 \partial w_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 e}{\partial x_n \partial w_1} & \dots & \frac{\partial^2 e}{\partial x_n \partial w_n} \end{bmatrix} \tag{5}$$

Here,

A_k is the Hessian matrix of the performance index at the current values

Where,

I_i is the enter after the i^{th} enter neuron,

$net_j = \sum_i W_{ij} \theta_i$ is realized local subject advanced at the contribution over the enactment work connected together with the hidden neuron (i), and

f_j is the initiation work about the neurons between the mystery layer.

When A_k is large, it is complex and time consuming to compute W_{k+1} .

Calculation of error for the back propagation algorithm is as follows:

Error Derivative (EA_j) is the modification among the real and desired target:

$$EA_j = \frac{\partial E}{\partial y_j} = y_j - d_j \quad (6)$$

Here,

E represents the error

y represents the Target vector

d represents the predicted output

Error Variations is total input received by an output changed

$$EI_j = \frac{\partial E}{\partial X_j} = \frac{\partial E}{\partial y_j} X \frac{dy_j}{dx_j} = EA_j y_j (1 - y_i) \quad (7)$$

Here,

E is the error vector

X is the input vector for training the neural network

In Error Fluctuations calculation connection into output unit is required:

$$EW_{ij} = \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial X_j} = \frac{\partial X_j}{\partial W_{ij}} = EI_j y_i \quad (8)$$

Here,

W represents the weights

I represents the Identity matrix

I and j represent the two dimensional weight vector indices

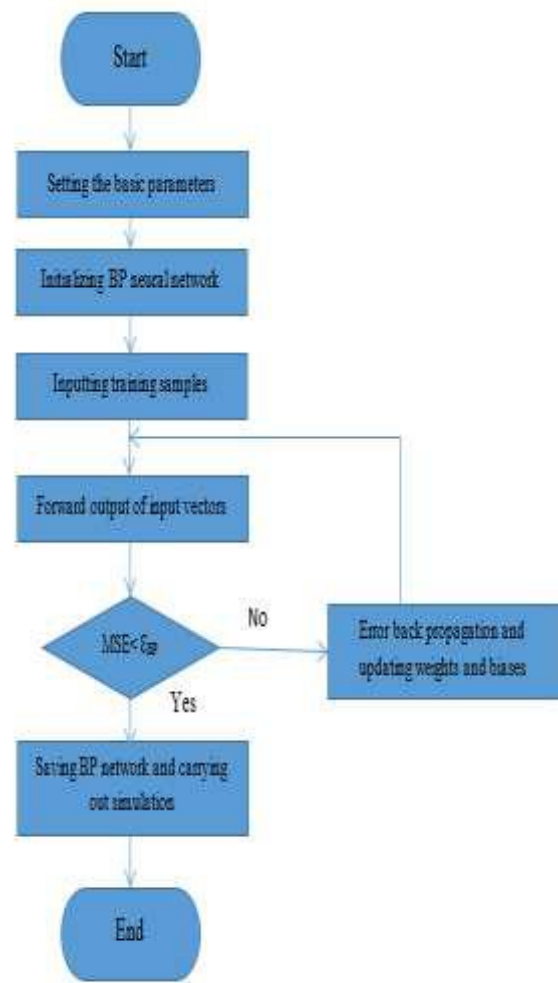


Fig 3: Flowchart for proposed approach

The proposed approach uses an **ada-boost** approach for android malware detection. In this approach, the output of one neural network is fed as the input to another neural network. The characteristic of such an approach is the fact that it can achieve higher effectiveness of classification accuracy compared to a single neural architecture for classification since the parameters which distinguish malwares and non-malwares are very similar and often makes the classification accuracy plummet.

5. RESULTS

The results obtained in the proposed work are enlisted here. A comparative analysis of the proposed and previous work is also given. The system simulation parameters are given below

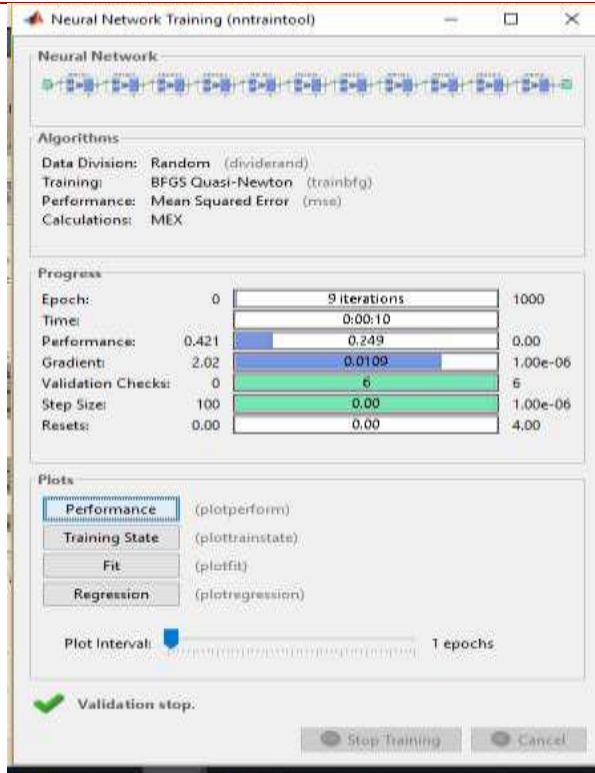


Fig 4: Designed Neural Network

The figure above depicts the designed neural network for the classification. The Quasi Newton Backpropagation approach is used to train the ANN.

The figure above depicts the training states as a function of the number of epochs while training the neural network.

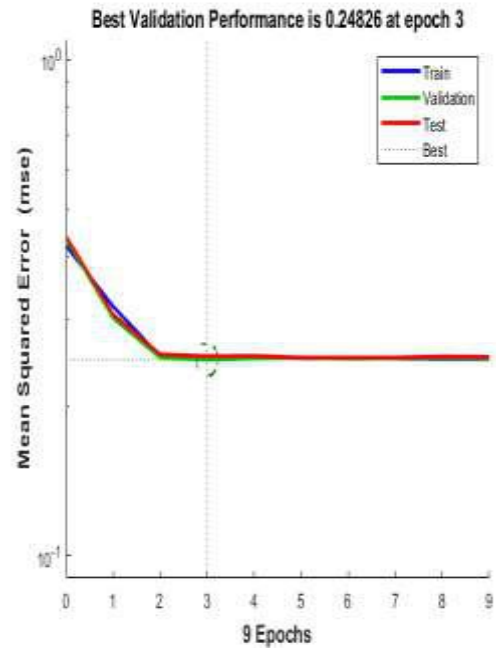


Fig 6: Variation of MSE while training the ANN

The figure above depicts the variation of MSE with the number of iterations.

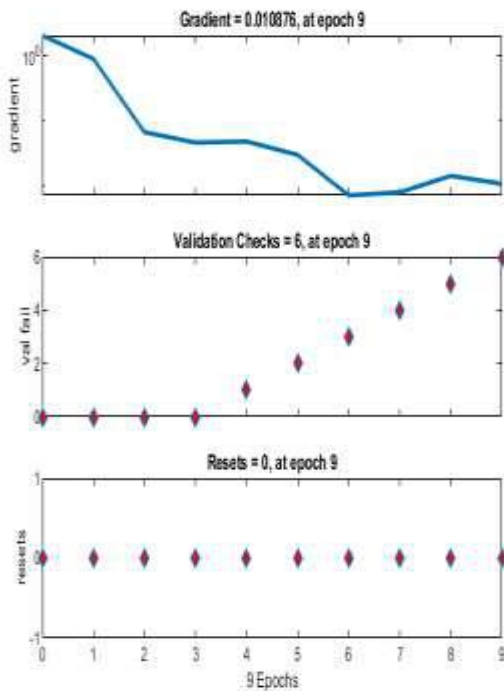


Fig 5: Neural Network Training States

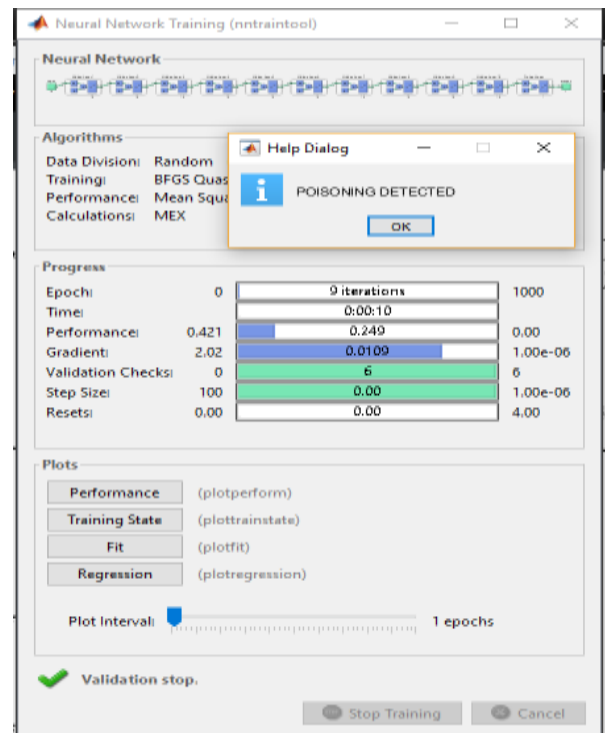


Fig 7: Detection of Poisoning

The figure above depicts the detection of poisoning.

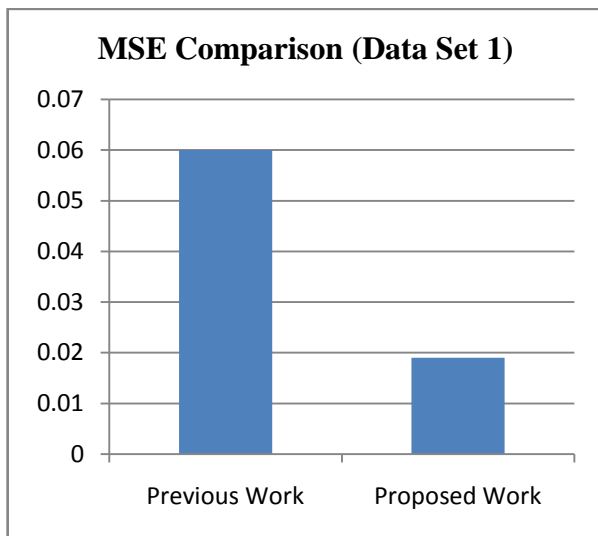


Fig 8: Comparative MSE for dataset 1

The figure above depicts the comparative MSE analysis for the previous and proposed work for data set 1.

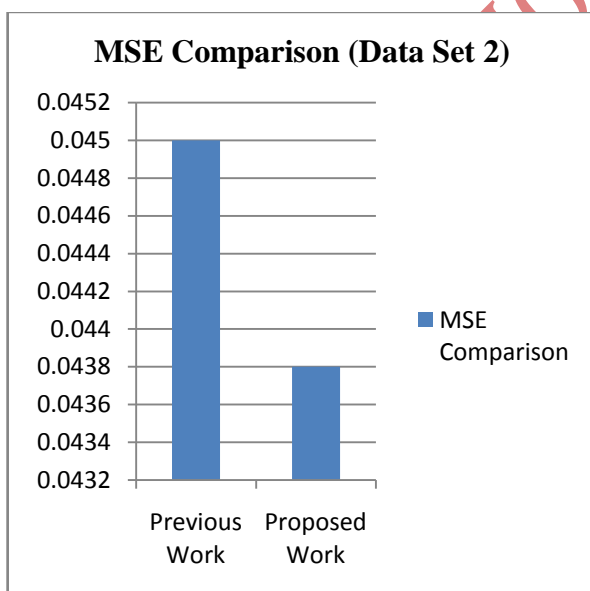


Fig 9: Comparative MSE for dataset 2

The figure above depicts the comparative MSE analysis for the previous and proposed work for data set 2.

6. CONCLUSION

It can be concluded that the increasing popularity of neural networks for classification, the chances of poisoning attacks have also increased. One major area of research that has emerged is the detection of poisoning for android systems using neural networks due to the complexity of data set of attacks. Several approaches have been used so far for the effective classification of poisoning attacks. Sophisticated attackers have strong incentives to manipulate the results and models generated by machine learning algorithms to achieve their objectives. The classification of poisoning is generally challenging keeping in mind the complexity and size of the data. The algorithm used is the Quasi Newton training algorithm which is based on the back propagation. The error obtained for two different classes of data sets are:

1) 0.019

2) 0.0438

This is much lesser than the mse of previous work of 0.06 and 0.0450

7. REFERENCES

- [1] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, Bo Li, "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning", IEEE 2018
- [2] Sen Chen, Minhui Xue, Lingling Fan, Shuang Hao, Lihua Xu, Haojin Zhu d, Bo Li, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach", Elsevier 2018
- [3] Chao Chen, Yu Wang, Jun Zhang, Yang Xiang, Wanlei Zhou, Geyong Min, "Statistical Features-Based Real-Time Detection of Drifted Twitter Spam", IEEE 2017
- [4] Nida Mirza, Balkrishna Patil ,Tabinda Mirza ,Rajesh Auti, "Evaluating efficiency of classifier for email spam detector using hybrid feature selection approaches",IEEE 2017
- [5] Scott Alfeld, Xiaojin Zhu, and Paul Barfor, "Data Poisoning Attacks against auto regressive models",IEEE 2016
- [6] Hailu Xu ,Weiqing Sun ,Ahmad Javaid," Efficient spam detection across Online Social Networks", IEEE 2016
- [7] Nadir Omer Fadl Elssied,Othman Ibrahim ,Ahmed Hamza Osman," Enhancement of spam detection mechanism based on hybrid kkkk-mean clustering and support vector machine",SPRINGER 2015

- [8] Tarjani Vyas , Payal Prajapati , Somil Gadhwal,” A survey and evaluation of supervised machine learning techniques for spam e-mail filtering”,IEEE 2015
- [9] Nishtha Jatana ,Kapil Sharma,” Bayesian spam classification: Time efficient radix encoded fragmented database approach”, IEEE 2014
- [10] Kamalanathan Kandasamy ,Preethi Koroth,” An integrated approach to spam classification on Twitter using URL analysis, natural language processing and machine learning techniques”, IEEE 2014
- [11] Navneel Prasad ,Rajeshni Singh ,Sunil Pranit Lal,” Comparison of Back Propagation and Resilient Propagation Algorithm for Spam Classification”,IEEE 2013
- [12] Wojciech IndykEmail author, Tomasz Kajdanowicz, Przemyslaw Kazienko,Slawomir Plamowski,” Web Spam Detection Using MapReduce Approach to Collective Classification”, SPRINGER 2013
- [13] Ashwin Rajadesingan, Anand Mahendran,” Comment Spam Classification in Blogs through Comment Analysis and Comment-Blog Post Relationships”, SPRINGER 2012
- [14] Alper Kursat Uysal ,Serkan Gunal ,Semih Ergin ,Efnan Sora Gunal, “A novel framework for SMS spam filtering”, IEEE 2012
- [15] D. Karthika Renuka , T. Hamsapriya ,M. Raja Chakkaravarthi, P. Lakshmi Surya, “Spam Classification Based on Supervised Learning Using Machine Learning Techniques”, IEEE 2011
- [16] Safvan Vahora ,Mosin Hasan ,Reshma Lakhani, “Novel approach: Naïve Bayes with Vector space model for spam classification”, IEEE 2011
- [17] Lourdes Araujo, Juan Martinez-Romo, “Web Spam Detection: New Classification Features Based on Qualified Link Analysis and Language Models”, IEEE 2010
- [18] Sang Min Lee, Dong Seong Kim , Ji Ho Kim ,Jong Sou Park, “Spam Detection Using Feature Selection and Parameters Optimization”, IEEE 2010
- [19] Chih-Hung Wu, Chiung-Hui Tsai, “Robust classification for spam filtering by back-propagation neural networks using behavior-based features”, SPRINGER 2009
- [20] Chi-Yao Tseng, Ming-Syan Chen, “Incremental SVM Model for Spam Detection on Dynamic Email Social Networks”, IEEE 2009