**A Gini’s Index Based Approach and Back Propagation in Machine Learning for Mobile Spam Classification**

1Priyanka Patidar, 2Prof. Nilesh Joshi

1Mtech scholar, 2Assistant Professor

1,2Department of Computer Science and Engineering

1,2LNCT(Bhopal) Indore Campus, Indore

*Abstract*—With increased internet usage, one of the most prevalent problems faced is constant spamming. While web applications and mailing services are heavily spammed, the upsurge of handheld mobile devices has led to an outburst of heavy mobile spamming. The matter is more severe in mobile devices due to lesser sophisticated filtering mechanisms in built in mobile operating systems. Spam detection is challenging due to the need for semantic analysis of the mobile spam messages, which generally tend to have overlapping polarities. In this paper, a mobile spam classification technique is developed based on Gini’s index and Back-propagation in machine learning. The approach uses the Gini’s splitting criteria for the data sets and backpropagation based neural network as the machine learning classifier. The probabilistic classifier is well suited for datasets of texts messages with overlapping boundaries. The evaluation of the proposed system is based on the accuracy of classification and number of iterations. The results obtained in the proposed work are compared with existing techniques and it is shown that the proposed technique outperforms them in terms of accuracy of classification.

Keywords—Mobile Spam Classification, Gini’s Index, Neural Networks, Back Propagation, Training Iterations, Classification Accuracy.

1. INTRODUCTION

Mobile spamming has become one of the most common techniques for promotions, customer churning and potential attacks targeting the frequently used handheld mobile devices which are more prone to such attacks. The ease of collecting mobile contacts, connected data bases and relatively lesser sophisticated filtering mechanisms for the mobile spam filtering makes its extremely challenging to thwart spamming attacks. A numeric estimation of the rising spamming attacks has been depicted in figure 1, for the 3rd quarter of 2020 citing an increasing trend.



***Fig.1 Number of Mobile Spams for Quarter-3, 2020***

***Courtesy: Kaspersky Labs Security Report***

Some of the spamming attacks may be benign while others may be malignant trying to redirect mobile users to malicious websites where user security may be compromised. Since the amount of data is staggering large and complex, off late machine learning based approaches are becoming common to filter out spams. One of the challenges which machine learning based approaches face for mobile spamming platforms is the limited computational and processing capabilities of hand held mobile devices. This makes is necessary to design and test algorithms which are compatible with various versions of mobile operating systems and also supported by limited memory and processing hardware as there exists a lot of diversity in the mobile hardware of different devices. This paper is organized as:

Section I introduces the basic concepts pertaining to mobile spam classification and its necessities.Section II briefly summarizes the work done in the domain.Section III discusses the proposed approach.Section IV illustrates the obtained results. The findings of the paper are concluded in the conclusion.

1. **RELATED WORK**

Various approaches have been devised for mobile spam classification.

A spam classification mechanism based on text normalization and back propagation based neural network has been proposed by Jain et al. in [1]. The approach also compares the findings of the work with conventionally existing algorithms. Adewole et al. in [2] proposed a bio-inspired evolutionary machine learning based approach for the detection of spam. Different spamming features and sources are analyzed using standard machine learning algorithms using the Weka machine learning tools. The approaches used are the support vector machine (SVM), Ada-Boost, Random Forests, Multi-Layer Perceptron (MLP) Bayes’ net. Barushka et al. in [3] proposed the use of regularized deep neural nets for the classification of spams. Sedhai et al. in [4] proposed a semi-supervise approach for spam classification. The data set used was that of Twitter. A similar approach was used by Chao et al. in [5] which aimed at analyzing drifted Twitter spam. Mirza et al. in [6] analyzed the effect of feature selection on spam filtering. A comparative analysis of addition and removal of features from the training data set was done. Afzal et al. in [7] proposed a techniques based on machine learning for bi-lingual data classification. Xu et al. in [8] developed an efficient machine learning based classifier for classification of multiple data sets corresponding to different social media platforms. Elssied et al. in [9] used a dual approach comprising of K-means clustering and support vector machine for spam classification. The clustering based approach was primarily used for data preparation and structuring while the multi-dimensional hyperplane based Support Vector Machine was used for the final classification. Vyas et al. in [10] presented a comprehensive review on the various supervised machine learning based approaches for email spam classification. The concepts of feed forward nets, convolutional nets, back propagation and recurrent nets were discussed in the context of spam filtering. Jatana et al. in [11] proposed a Bayesian classifier based approach for spam filtering. Kandasamy et al. in [12] proposed a natural language processing (NLP) based approach for spam classification using social media data. Prasad el al. in [13] compared the performance of Back Propagation and Resilient Propagation based machine learning approaches for spam classification. Indyk et al. in [14] proposed a Map Reduce based approach for collective spam classification.

1. **PROPOSED SYSTEM MODEL**
2. ***Data processing and normalization:***

Since neural nets directly process numeric data sets, the processing of data is done prior to training a neural network. The texts are first split into training and testing data samples in the ratio of 70:30 for training and testing. Further, a data vector containing known and commonly repeated spam and ham words is prepared. The SMS spam collection v.1 dataset is used as a dataset for the proposed work. Text normalization is followed by removal of special characters and punctuation marks.

Subsequently the data set structuring and preparation is performed based on the feature selection. The features selected are:

1. Spam words
2. Ham Words
3. URLs in the message
4. Lengthy numerical strings which can be contact numbers
5. Character length
6. Special symbols
7. Presence of currency values
8. Self-answering texts

The feature vectors along with the list of commonly accepted spam and ham lists of words comprises of the training vector. A similar process is done for both the training and testing datasets.

1. ***Neural Network Training***

The neural network training model used in the proposed work is the Gini’s Index and Back Propagation based neural net. The Gini’s index is especially useful for overlapping data sets since it can split data sets with overlapping classes based on conditional probability. The Gini’s index for splitting is defined as:

$GI=1-\sum\_{i=1}^{n}p\_{i}^{2}$ **(1)**

Here,

GI represents the Gini’s Index

P is the probability of a class

The next step is the design of the neural network for classification. The neural network has the property of being able to process large data streams in parallel and adapting as per obtained outcomes. The fundamental model of a neural net is depicted in figure 2.



***Fig.2 Mathematical Model of ANN***

Here,

X represents the parallel input vector.

Y represents the output

Ɵ represents the bias.

W represents the weights.

The prepared data vector for training is used for training wherein the weights are initialized randomly. A stepwise implementation is done as:

1. Prepare two arrays, one is input and hidden unit and the second is output unit.

Here, a two dimensional array $W\_{ij}$ is used as the weigt updating vector andoutput is a one dimensional array Yi.

3. Original weights are random values put inside the arrays after that the output**.**

$ x\_{j}=\sum\_{i=0}^{}y\_{i}W\_{ij}$ **(2)**

Where,

yi is the activity level of the jth unit in the previous layer and

$W\_{ij}$ is the weightof the connection between the ith and the jth unit.

4. Next, activation is invoked by the sigmoid function applied to the total weighted input.

$y\_{i}=\left[\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}\right]$ **(3)**

Summing all the output units have been determined, the network calculates the error (E)**.**

$E=\frac{1}{2}\sum\_{i}^{}\left(y\_{i}-d\_{i}\right)^{2}$ **(4)**

Where, yi is the event level of the jth unit in the top layer and di is the preferred output of the ji unit.

### **Implementing Back Prop:**

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

Error Derivative ($EA\_{j}) $is the modification among the real and desired target:

$EA\_{j}=\frac{∂E}{∂y\_{j}}=y\_{j}-d\_{j}$ **(5)**

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 given by:

$EI\_{j}=\frac{∂E}{∂X\_{j}}=\frac{∂E}{∂y\_{j}}X\frac{dy\_{j}}{dx\_{j}}=EA\_{j}y\_{j}(1-y\_{i})$ **(6)**

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 computed as:

$EW\_{ij}=\frac{∂E}{∂W\_{ij}}=\frac{∂E}{∂X\_{j}}=\frac{∂X\_{j}}{∂W\_{ij}}=EI\_{j}y\_{i}$ **(7)**

Here,

W represents the weights

I represents the Identity matrix

I and j represent the two dimensional weight vector indices

Overall Influence of the error:

$EA\_{i}=\frac{∂E}{∂y\_{i}}=\sum\_{j}^{}\frac{∂E}{∂x\_{j}}X\frac{∂x\_{j}}{∂y\_{i}}=\sum\_{j}^{}EI\_{j}W\_{ij}$ **(8)**

The partial derivative of the Error with respect to the weight represents the error swing for the system while training. The gradient is computed as:

$g=\frac{∂e}{∂w}$ **(9)**

Here,

g represents the gradient

e represents the error of each iteration

w represents the weights.

The gradient is considered as the objective function to be reduced in each iteration. A probabilistic classification using the Bayes theorem of conditional probability is given by:

$P\left(\frac{H}{X}\right)=\frac{P\left(\frac{X}{H}\right)P\left(H\right)}{P\left(X\right)}$ **(10)**

Here,

Posterior Probability [P (H/X)] is the probability of occurrence of event H when X has already occurred

Prior Probability [P (H)] is the individual probability of event H

X is termed as the tuple and H is is termed as the hypothesis.

Here, [P (H/X)] denotes the probability of occurrence of event X when H has already occurred.

The final classification accuracy is computed as:

$Ac=\frac{TP+TN}{TP+TN+FP+FN}$ **(11)**

Here.

**TP** represents true positive

**TN** represents true negative

**FP** represents false positive

**FN** represents false negative

1. **RESULTS**

The system is implemented on Matlab. The results obtained on implementing the proposed system is discussed in this section.



***Fig.3 Raw data samples***

The raw data samples are collected after which it is imported to the Matlab workspace.



***Fig.4 Conversion of Data into string***

Subsequently, the data is converted into strings for ease of analysis of textural data. The data is split into training and testing data samples in the ratio of 70:30. While other data division ratios could have been uses, but in this work, the standard 70:30 ratio is adhered to.

The next process is invoking the Gini’s split criterion.

****

***Fig.5 Invoking the Gini’s Split criteria***

The Gini’s split criteria is the precursor to the training of the system.

****

***Fig.6 Training Performance***

Figure 6 depicts the training parameters of the proposed system which consumes 5 seconds to run 378 iterations of the back propagation algorithm. A 20 neuron hidden layer is designed for the system.

****

***Fig.7 Training Error Histogram***

The training error histogram is depicted in figure 7 which is an indicator of the errors occurring during the training process.

****

***Fig.8 GUI for detection***

Figure 8 depicts the GUI for spam detection. A similar GUI represents that of the non-spam or ham case. A comparative analysis with existing approaches is tabulated in table I.

Table I: Comparative Accuracy Analysis of Proposed and Existing Algorithms

|  |  |  |
| --- | --- | --- |
| S.No. | Technique | Accuracy (%) |
| 1. | SNAP | 83.9 |
| 2. | AIR SENTI | 80.5 |
| 3. | Naïve Baye’s | 64 |
| 4. | Random Forests | 63 |
| 5. | ANN with BackProp | 95.81 |
| **6.** | **Proposed Approach: Gradient Descent with BackProp and Gini-Index** | **99.75** |

It ca be observed from the tabulated results, that the proposed work outperforms the existing algorithms such as SNAPM AIR SENTI, Naïve Gayes’, Random Forests and ANN with Back Prop.

**CONCLUSION: It can be concluded from the aforesaid arguments that mobile spam classification is extremely challenging and non-trivial due to the constraints of computational power and memory at our disposal. Moreover, easier access to handheld devices makes systems more prone to spamming attacks. Text spam classification is non trivial in the sense that it generally belongs to non-clear or fuzzy boundary datasets. The proposed approach presents a mobile spam classification mechanism Using Gini’s Index and Back Propagation. It has been shown that the proposed approach outperforms the existing techniques in terms of classification accuracy. Additionally, the technique consumes moderate number of iterations and low execution time which are critical considerations for mobile devices.**

**References**

[1] AK Jain, D Goel, S Agarwal, Y Singh, G.Bajaj, “Predicting Spam Messages Using Back Propagation Neural Network”, Journal of Wireless Personal Communications, Springer 2020, vol. 110, pp. 403-422.

[2] KS Adewole, NB Anuar, A Kamsin, “SMSAD: a framework for spam message and spam account detection”, Journal of Multimedia Tools and Applications, Springer 2019, vol. 78, pp. 78, 3925–3960.

[3] Aliaksandr Barushka, Petr Hajek, “Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks”, Springer 2018

[4] Surendra Sedhai, Aixin Sun, “Semi-Supervised Spam Detection in Twitter Stream”, IEEE 2018

[5] Chao Chen, Yu Wang, Jun Zhang, Yang Xiang, Wanlei Zhou, Geyong Min, “Statistical Features-Based Real-Time Detection of Drifted Twitter Spam”, IEEE 2017

[6] Nida Mirza, Balkrishna Patil ,Tabinda Mirza ,Rajesh Auti, “Evaluating efficiency of classifier for email spam detector using hybrid feature selection approaches”,IEEE 2017

[7] Hammad Afzal ,Kashif Mehmood, “Spam filtering of bi-lingual tweets using machine learning”,IEEE 2016

[8] Hailu Xu ,Weiqing Sun ,Ahmad Javaid,” Efficient spam detection across Online Social Networks”, IEEE 2016

[9] Nadir Omer, Fadl Elssied,Othman Ibrahim ,Ahmed Hamza Osman,” Enhancement of spam detection mechanism based on hybrid k-mean clustering and support vector machine”,SPRINGER 2015

[10] Tarjani Vyas , Payal Prajapati , Somil Gadhwal,” A survey and evaluation of supervised machine learning techniques for spam e-mail filtering”,IEEE 2015

[11] Nishtha Jatana ,Kapil Sharma,” Bayesian spam classification: Time efficient radix encoded fragmented database approach”, IEEE 2014

[12] Kamalanathan Kandasamy ,Preethi Koroth,” An integrated approach to spam classification on Twitter using URL analysis, natural language processing and machine learning techniques”, IEEE 2014

[13] Navneel Prasad ,Rajeshni Singh ,Sunil Pranit Lal,” Comparison of Back Propagation and Resilient Propagation Algorithm for Spam Classification”,IEEE 2013

[14] Wojciech Indyk, Tomasz Kajdanowicz, Przemyslaw Kazienko,Slawomir Plamowski,” Web Spam Detection Using MapReduce Approach to Collective Classification”, SPRINGER 2013