Spam Review Detection Using the Linguistic and Spammer Behavioral Methods

[1]Shaik Mohammed jabeer, [2]Shaik Reehana
[1]Assistant Professor, Department of CSE, Global College of Engineering and Technology, Kadapa, A.P – 516162.
[2]PG student, for CSE, Global College of Engineering and Technology, Kadapa, A.P – 516162.

ABSTRACT

Online reviews regarding different products or services have become the main source to determine public opinions. Consequently, manufacturers and sellers are extremely concerned with customer reviews as these have a direct impact on their businesses. Unfortunately, to gain profits or fame, spam reviews are written to promote or demote targeted products or services. This practice is known as review spamming. In recent years, the spam review detection problem has gained much attention from communities and researchers, but still there is a need to perform experiments on real-world large-scale review datasets. This can help to analyze the impact of widespread opinion spam in online reviews. In this work, two different spam review detection methods have been proposed: Spam Review Detection using Behavioral Method (SRD-BM) utilizes thirteen different spammer's behavioral features to calculate the review spam score which is then used to identify spammers and spam reviews, and Spam Review Detection using Linguistic Method (SRD-LM) works on the content of the reviews and utilizes transformation, feature selection and classification to identify the spam reviews. Experimental evaluations are conducted on a real-world Amazon review dataset which analyze 26.7 million reviews and 15.4 million reviewers. The evaluations show that both proposed models have significantly improved the detection process of spam reviews. Specifically, SRD-BM achieved 93.1% accuracy whereas SRD-LM achieved 88.5% accuracy in spam review detection. Comparatively, SRD-BM achieved better accuracy because it works on utilizing rich set of spammers behavioral features of review dataset which provides in-depth analysis of spammer behavior. Moreover, both proposed models outperformed existing approaches when compared in terms of accurate identification of spam reviews. To the best of our knowledge, this is the first study of its kind which uses large-scale review dataset to analyze different spammers'.

1. INTRODUCTION

In last few year social networking websites such as Facebook, Twitter and YouTube have made a dramatic growth in terms of popularity. A whooping 1billion unique users visit YouTube every month and they watch almost 4 billion hours of video content. Amidst this growth, shared video contents of these sites become a predominant part of user's daily lives on the web. The web has become the major channel for delivery of multimedia contents. Video is present and apparent throughout the internet. It helps and supports new types of interaction among users, including political debates, video chats, video mails and video blogs. A number of web services offer video based functions; alternative to text based ones, such as video reviews for products, video advertisements and video responses. Video spammers are motivated to spam in order to promote specific content, advertise to generate sales, spread pornography (often as advertisement) or to compromise the sites' reputation and also increase view count to make it more credible. A number of spam detection techniques exploit characteristics present in the text based sites (e.g., email body, commentaries in a blog) [6]. Moreover, users of such system scan quickly learn to identify some text spams (e.g., URLs to suspect Web sites), skipping or ignoring them. On the other hand, video spamming, particularly in social video sharing systems, can be much more challenging to detect and combat. Content-based detection techniques are not easily applied to non-textual video objects on the other hand exploiting characteristics of the traffic to specific videos, such as number of views and number of comments received can be useful to distinguish spams. We propose and evaluate a video spam detection mechanism that classifies a spam based on the video's attributes, the viewer's response to the video, and the amount of sharing by viewers. In summary, the main contributions of this research are:

Find out quantitative evidence of video spamming activity (as defined above) in social online video sharing systems, particularly in YouTube. The identification and characterization of a set of video attributes that can be used to distinguish video spammers from legitimate users.

A test collection of videos from YouTube, classified as spams or legitimate videos. A video spam detection mechanism based on a set of classification algorithms, for example, decision tree and Naïve Bayes, and clustering algorithm. Predict the spams from the data mining model.

2. SYSTEM ANALYSIS

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies.

2.1 Existing System

In existing system data mining techniques are used for detecting spam messages. Most of these methods work only after posting messages. There is need of system which can automate this process before posting message.

a) Disadvantages of Existing System

- Most of the existing works are based on data mining techniques which are not accurate and time taken for analysis is high.
- Detection process is analyzed after the review was given.

2.2 Proposed System

The prediction of the spam comments present in the comments section of YouTube videos using the concept called machine learning, it is also known as subset of artificial intelligence, is done. Supervised learning approach depends on a very large number of labeled datasets. . The proposed classification algorithm (Logistic Regression) is used in order to predict the spam comment.

The proposed classification algorithm (Logistic Regression) is used in order to predict the spam comment. The purpose of project is to introduce briefly the techniques of machine learning and to outline the prediction technique. Being much more superior to the conventional data analysis techniques, machine learning can open a new opportunity to explore and increase the prediction accuracy.

2.3 Architecture analysis:

Structured project management techniques (such as an SDLC) enhance management's control over projects by dividing complex tasks into manageable sections. A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed. But none of the SDLC models discuss the key issues like Change management, Incident management and Release management processes within the SDLC process, but, it is addressed in the overall project management. In the proposed hypothetical model, the concept of user-developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. In the proposed hypothetical model, the concept of user-developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. The ―one size fits all‖ approach to applying SDLC methodologies is no longer appropriate. We have made an attempt to address the above mentioned defects by using a new hypothetical model for SDLC described elsewhere. The drawback of addressing these management processes under the overall project management is missing of key technical issues pertaining to software development process that is, these issues are talked in the project management at the surface level but not at the ground level.

2.4 Data Preprocessing

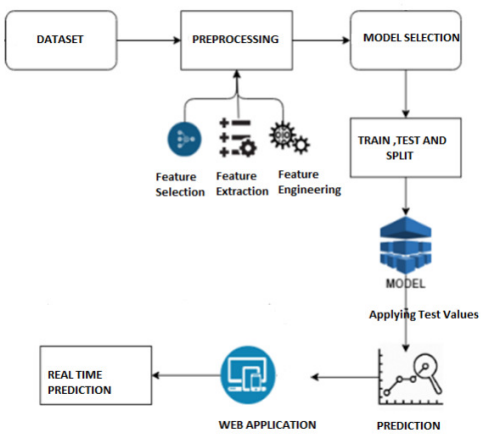
There are three symbolic data types in NSL-KDD data features: protocol type, flag and service. We use one-hot encoder mapping these features into binary vectors. One-Hot Processing: NSL-KDD dataset is processed by one-hot method to transform symbolic features into numerical features. For example, the second feature of the NSL-KDD data sample is protocol type. The protocol type has three values: tcp, udp, and icmp. One-hot method is processed into a binary code that can be recognized by a computer, where tcp is [1, 0, 0], udp is [0, 1, 0], and icmp is [0, 0, 1]

3. SYSTEM DESIGN

3.1 System architecture

The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It's a serious impact on the later part, notably testing and maintenance.

The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.



a) Dataset collection

In this module collection of datasets is taken from Kaggle website. Dataset consists of features and labels. Features are comments from YouTube channels and labels or spam or not spam.

b) Data preprocessing

In this module features and labels are extracted from dataset and stored in x and Y variables and as data is in string format it is converted using vectorizer technique.

c) Data set splitting

Data set is split in to testing and training sets. Training set has 80 percent of features and labels where as testing has 20 percent features and labels.

d) Initialize Algorithm

In this module machine learning algorithm is initialized and training set features and labels ate fitted to it which will train algorithm and them model is saved to system.

e) Prediction

In this stage user enters values in the web page and data is given to prediction model then result is shown as spam or not spam.

3.2 Input Design

The input design is a component of the overall system design. The following is the main goal of the input design:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

3.3 Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

3.4 Fundamental Concepts on (Domain)

To give credibility to the found results and in order to make the experiments reproducible, we present in this section the settings used for each classification method, as well as general information about datasets and experimental methodology. A datasets we have collected and created five databases composed by real, public and non-encoded data directly extracted from YouTube through. We have selected five of the ten most viewed YouTube videos during the collection period. Each sample represents a text comment posted in the comments section of each selected video. No preprocessing technique was performed. Subsequently, each sample was man-ally labeled as spam or legitimate (ham), using a collaborative tagging tool developed for this purpose, called Labeling8.The samples have associated a metadata information, such as the author's name and publication date, which have been preserved.

4. IMPLEMENTATION

a) Design

The software system design is produced from the results of the requirements phase. Architects have the ball in their court during this phase and this is the phase in which their focus lies. This is where the details on how the system will work is produced. Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.

b) Implementation

Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle. For a developer, this is the main focus of the life cycle because this is where the code is produced. Implementation my overlap with both the design and testing phases. Many tools exists (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

5. TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole.

So in a nutshell, that is a very basic overview of the general software development life cycle model. Now let's delve into some of the traditional and widely used variations.

5.1 System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

5.2 Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

5.3 Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

5.4 Acceptance Testing

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

6. OUTPUT SCREENS

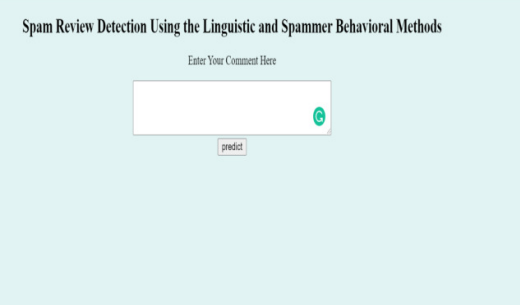


Figure: Main Screen

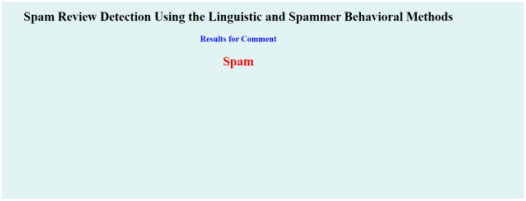


Figure: Detection

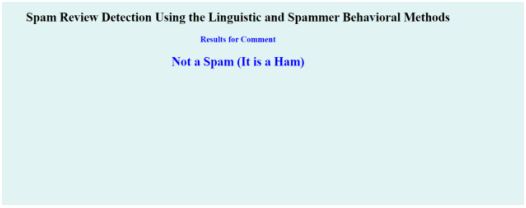


Figure: Not spam

7. CONCLUSION

By using three data mining models we find a similar result. We can generate a Lift Chart. A Lift Chart graphically represents the improvement that a mining model provides when compared against a random guess, and measures the change in terms of a lift score. By comparing the lift scores for various portions of our data set and for different models, we For example at 40% of the total population Naïve Bayes has a predict probability of 99.75%, decision tree 98.66% and clustering has 98.98%. But, at 85% of total population Naïve Bayes has a predict probability of 80.20%, decision tree has 82.11% and clustering has 65.79%. Therefore, we could conclude that, for higher number of test cases Naïve Bayes and Decision Tree models are more accurate for predicting spammers.

REFERENCES

[1] Peter Mel and Tim Grace, "The NIST Definition of Cloud Computing", NIST, 2010.

[2]Achill Buhl, "Rising Security Challenges in Cloud Computing", in Proc. of World Congress on Information and correspondence Technologies ,pp. 217-222, Dec. 2011.

[3]Srinivasarao D et al., "Breaking down the Superlative symmetric Cryptosystem Encryption Algorithm", Journal of Global Research in Computer Science, vol. 7, Jul. 2011

[4] Tingyuan Nye and Tang Zhang "An investigation of DES and Blowfish encryption algorithm‟ , in Proc. IEEE Region 10 Conference, pp. 1-4 ,Jan. 2009.

[5] Jitendra Singh Adam et al.," Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm" , International Journal of Advanced Research in Computer Science and Software Engineering ,vol. 2,Aug. 2012.

[6] Manikandan.G et al., "A changed cryptographic plan improving information", Journal of Theoretical and Applied Information Technology, vol. 35, no.2, Jan. 2012.

[7] Niles Maintain and Subhead Bhingarkar, " The examination and Judgment of Nimbus, Open Nebula and Eucalyptus", International Journal of Computational Biology , vol. 3, issue 1, pp 44-47, 2012.