# Darknet mining for threat detection

Sapan Gandhi, Nishant Pounikar, Alwin Johns

The University of Texas at Dallas

Richardson ,Tx-75080, USA

***Abstract*—In this paper, we present a basic implementation of threat detection and classification of various threats like cyber threats, Drugs related threats, Weaponry threats, etc. from internet blogs, web pages and forums especially from Darknet market and forums. We have developed a basic operational system to crawl pages, parse them and classify the basic parsed page to different categories. This provides a significant service for automatic classification of internet Big Data and Classify each of them into different threat categories. Further with the help of sentiment analysis and threat seriousness we can further classify each page to relevant and irrelevant threats.**

## I. Introduction

Internet Data is becoming larger and larger day by day and with the more and more reliance on online data and communication the threat to individual and organizations increases more and more. With the advancement of technology people have find ways to get loopholes in the system and become a relevant threat to the society.

This paper provides a practical and basic approach to classify Big Data on the market especially Darknet Market and classify them for future use by the intelligence agency to get the relevant information from particular high risk domain and target the sites and threats offending the society. This paper mainly focuses on the data obtained from the Darknet market.

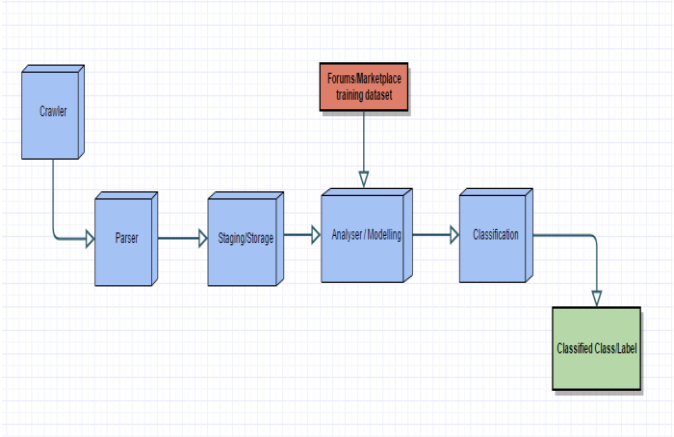
**Background:** Darknet and Deepnet sites are the sites not available for use by the general public. Basically these sites cannot be indexed by the search engines. Within the dark net, both web surfers and website publishers are entirely anonymous. Whilst large government agencies are theoretically able to track some people within this anonymous space, it is very difficult, requires a huge amount of resources, and isn't always successful [1]. Darknet can be used for underground communication between organizations or for anonymously posting recent glitches and loopholes in the different fields like security, Drugs and Weapons, fraud, counterfeit products, soft wares and malware.

**Accessing Darknet:** Darknet anonymity is usually achieved using an onion network. Normally, when accessing the pedestrian Internet, your computer directly accesses the server hosting the website you are visiting. In an onion network, this direct link is broken, and the data is instead bounced around a number of intermediaries before reaching its destination. The communication registers on the network, but the transport medium is prevented from knowing who is doing the communication [1]**.** The most commonly used software for such underground communication is "The Onion Router" most frequently called "TOR" Browser. From the tor browser we can easily access the Darknet markets and forums and buy things needed using bitcoin (cryptocurrency and a payment system developed by anonymous programmers).

## II. System Overview

**Dataset:** The Dataset for Darknet is not easily accessible via a simple crawler. For this system implementation we are using an available database on the torrent. Modelling of this system is done via "Alphabay" Darknet market and tested on "AlphaBay"(live pages via TOR) , "Silk Road" and "TOR Market" Reference link [4].

Current system for threat detection works in mainly 5 steps as shown in the below diagram.




**Crawler:** The basic task of a crawler is to crawl the content of a link or web page into basic html format. The crawler in the system is a modified version of the same for higher efficiency of the system. It takes two parameters into consideration, "K" the number of associated web links in the main link to be considered and "N" the number of depth or levels to be visited from the main link. This crawler crawls the important body of the web pages, i.e. it parses the html tags and other unnecessary items in the document and just keeps the main body and "K" links in the body for visiting deeper html pages from the main link. This creates a tree structure to the main parent link to the all the Crawled pages through depth of "N". After getting all

the associated pages from the main link, our next step is to parse the tree to a proper formatted text file for the final stage of classification.

**Parser:** The data tree created from the crawler stage is processed for the further simplification in the parser stage. In this stage we do stemming on the body of the html pages and remove the stop words to make the simplest data and save it to text file.

**Staging/Storage:** All the data processed in previous stages are processed into main memory but we cannot guarantee the length and size of the database and so this stage provides proper storage of the pages and in a formatted tree structure. For example, given link in the crawler will create a main parent folder and the links associated with it creates child nodes or text files in the parent folder as sibling nodes. This goes for the "N" depth as given in the crawler.

**Analyzer/Modelling:** This is the stage where the training data is being modelled and used for further classification. The method used for modelling is naïve bayes classification algorithm. All the common words that a document has are the stop words which we already removed in the parsing stage. Also all words have been converted to the root words using stemming so that each word with same meaning fall for the same root words and are given proper weightage in while classification. This document is then converted to featured vector according to the words and their probability weights and then modelled with the associated class.

**Classification:** This stage is where the test data set is given probability of being classified into each class/label and a label with highest probability is given to the test data set and is classified under that category.

## III. NAÏVE BAYES DOCUMENT CLASSIFICATION

**Bayes Theorem:** In probability theory and statistics, Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. For example, if cancer is related to age, then, using Bayes' theorem, a person's age (prior knowledge) can be used to more accurately assess the probability that they have cancer, compared to the assessment of the probability of cancer made without prior knowledge of the person's age [2].

The statement of theorem can be expressed as below equation,

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Here the events are A and B

P (A) & P (B) is corresponding probability of events A and B

P (A|B) is a conditional probability of event A given that event B has already occurred

P (B|A) is a conditional probability of event B given that event A has already occurred [2]

The purpose of this project is limited to classify the documents into five genres namely Cyber, Theft, Drugs, Weapons, Security threats. [3]

The inputs are

- A document d
- A fixed set of classes C = {c1, c2,..,cj}
- A training set of m hand labeled documents (d1, c1), (d2, c2), …., (dm, cm)

The outputs are

- A learned classifier
- Class prediction for test document[3]

A simple (naïve) classification method is used which is based on Bayes' rule to perform the designated task. The document is represented as a bag of words. In bag of words assumption, we are considering that the position of words doesn't matter [3].

We have to find out given a document what class it belongs to. When we apply the Bayes' rule to our document d and class c we can write

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

In plain English, using Bayesian probability terminology, the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Now,

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, P(c \mid d)$$

$$= \underset{c \in C}{\operatorname{argmax}}\, \frac{P(d \mid c)P(c)}{P(d)}$$

$$= \underset{c \in C}{\operatorname{argmax}}\, P(d \mid c)P(c)$$

MAP = Maximum a posteriori which is most likely class

The document d is represented as bag of words which do not depend on position (i.e. words independent of each other) so the document can be represented as feature vector i.e.

p (d|c) = P(x1, x2, x3,…., xn |c) where x1, x2, x3 are representing words in document. Now we can write,

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, P(x_1, x_2, \ldots, x_n \mid c)P(c)$$

Where using conditional independence we can say

$$p(x_1, x_2, \ldots, x_n \mid C) = p(x_1 \mid C) * p(x_2 \mid C) \ldots p(x_n \mid C)$$

We use Laplace's (add 1) smoothing to deal with zero probabilities. To calculate the probability of any word i.e. P ($x_i$ |c) in document we use below formula,

$P(x_i|c) = (n_i + 1)/(n + |v|)$

$n_i$ = Number of times word 'i' appears in document

n = total number of words in case of class c

|v| = total vocabulary size

Below steps are followed to implement the algorithm.

1) Use the training set to create the model.
2) Read each training set document labelled according to the class and convert the document in the feature vector form using bag of words representation.
3) Prior probability of each class is calculated using the count of documents present in each class and total number of documents.
4) Each word from training set and its count corresponding to each class is stored in a Hash table data structure which forms out Naïve Bayes' model.
5) While each document in training set is being processed the count of total number of words in each class is maintained in a separate array.
6) The total vocabulary size is nothing but the size of hash table which contains the unique words taken from all the documents in training set.
7) Now that the model is formed when a new document arrives for getting classified. That document is again converted in feature vector.
8) Probability for each word is calculated using Laplace's (add 1) smooth formula mentioned above.
9) For each class we do the following
   Multiply the probabilities of all the words calculated as mentioned in step (8)
   Multiply this with the prior probability of class
   Store this in an array
10) The class which has the maximum probability is the class of given document.

Eg.

Main Data

| Doc | Text | Class |
|---|---|---|
| 1 | I loved the movie | + |
| 2 | I hated the movie | - |
| 3 | a great movie. Good movie | + |
| 4 | poor acting | - |
| 5 | great acting | + |

Parsed data:

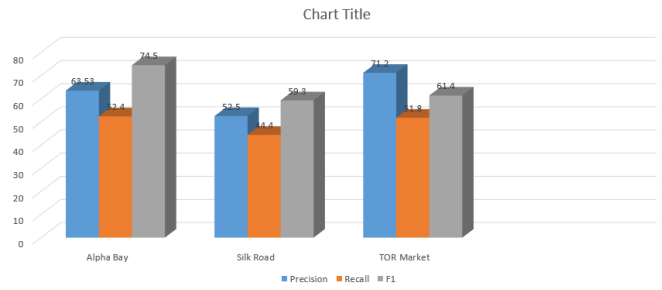
| Doc | Text | Class |
|---|---|---|
| 1 | love movie | + |
| 2 | hate movie | - |
| 3 | great movie good movie | + |
| 4 | poor act | - |
| 5 | great act | + |

Feature matrix:

| Doc | love | movie | hate | great | good | poor | act | Class |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | | | | | + |
| 2 | | 1 | 1 | | | | | - |
| 3 | | 2 | | 1 | 1 | | | + |
| 4 | | | | | | 1 | 1 | - |
| 5 | | | | 1 | | | 1 | + |

## IV. Results

Below are the statistics for the testing of dataset via live TOR browser and also data from the torrent link.[4]




Results on live crawling of links

```
Accuracy : 33.33 %
https://www.fireeye.com/blog/threat-research/2016/11/fireeye_respondsto.html : CyberThreat

        https://www.fireeye.com/solutions/small-and-midsize-business.html : UndefinedCategory

        https://www.fireeye.com/services/mergers-and-acquisitions-risk-assessment.html : CyberThreat

        https://www.fireeye.com/customers-threats.html : TheftThreat

        https://www.fireeye.com/current-threats.html : CyberThreat
```

## V. Conclusions

In this paper we implement a system to pre-process the online forums and web pages from the internet, Darknet and Deep web and process this data to classify these data into categories of threats. Thus with this approach we can classify the Big data of Darknet market and other forums to different categories. For future scope output of this system can be used as input to an intelligent system that identifies relevant and irrelevant threats to the current market scenario.

## VI. References


[1] https://turbofuture.com/internet/A-Beginners-Guide-to-Exploring-the-Darknet

[2] https://en.wikipedia.org/wiki/Bayes'_theorem

[3] http://www.ijcttjournal.org/2015/Volume30/number-4/IJCTT-V30P132.pdf

[4] https://www.gwern.net/Black-market%20archives