Ontology Driven NLP Based Classification of Malware Documents

Submitted in partial fulfillment of the requirements
of the degree of
Master of Technology
by

Mayukh Rath
153050025

Supervisor :

Prof.R.K.Shyamasundar

Computer Science & Engineering
Indian Institute of Technology, Bombay

June 27, 2017

Acknowledgements

Firstly, I would like to express my heartfelt gratitude to my supervisor **Prof. R.K. Shyamasundar** for the continuous support & encouragement in my MTech study and research.His guidance helped me all the time in my research and writing this thesis. I could not have imagined having a better supervisor and mentor for my MTech study.

Besides my advisor, I would like to thank **Ms. Shivali Agarwal** for bringing this project to me and also for the immense help and contribution to the project. Each time I was stuck at some point, her advice always helped me find the solution.

I would also like to thank the rest of my thesis defense committee: **Prof. G.Sivakumar**, **Prof. Virendra Singh** for examining my thesis defense and also for the insightful remarks & question which drives me to widen my research from various perspectives.

Last but not the least, I would like to thank my family for supporting & motivating me throughout my life.

Approval sheet

This dissertation entitled **Ontology Driven NLP Based Classification of Malware Documents** by **Mayukh Rath(Roll Number: 153050025)** is approved for the degree of Master of Technology in Computer Science and Engineering from IIT Bombay.

Examiners

(VIRENDRA SINGH)

Supervisor

RK Shyamasundar

Chairman

Date: 27.06.2017

Place: IIT Bombay

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Mayukh Rath.

(Signature)

Mayukh Rath

(Name of the student)

153050025

(Roll No.)

Date: 27.06.2017

Abstract

With the recent advancement of modern technologies, more complex Cyber attacks are being introduced day by day. Most of the Cyber attacks are much more complex and contain multistage processes. Description of such attacks is quite often discussed in Malware and Security related blogs, possibly with the proposed solutions to the attack.Sometimes these kinds of recent trends of attacks are published in Journals, Conferences and even in Newspapers also. Getting the proper/adequate information about such attacks is very important to analyze the causes, vulnerabilities & intents behind these attacks and to build up systems that are secure from these kinds of attacks. The major problem is the amount of data is huge and impossible for human being to analyze such large amount of data.Hence proper classification of the available data into viable Malware classes is highly important. After proper classification, different kinds of attacks fall into the different domain, so it becomes much easier to analyze the attack scenarios and find the appropriate solution.

A natural requirement is to use the analysis of texts via machine learning. But existing Machine learning based Document Classification techniques alone fail to generate the proper classification with good accuracy due to its inability to understand Malware related concepts that are inferred from the attack scenarios.
In this thesis, we describe an approach that enables us to get such a classification that will be very useful for security perspectives. Our method consists of following steps:-

(a) Extract concepts from texts using NLP & Text Analysis Tool- In our experiments we have used AlchemyApi(Watson Natural Language Understanding) Tool developed by IBM, to extract malware-related information from the text. AlchemyApi provides us with Concepts that are inferred from the text along with important Entities & Keywords from the text.
We generate a concept vector consisting of pre-decided malware classes from the output of AlchemyApi.

(b) Design a Semi-Supervised Clustering algorithm that starts with initial seeding and progresses based on the similarity between Malware documents and the centroid of the cluster.

(c) Design an Ontology of Malware Classes in a hierarchical manner considering Characteristics, Damages, Vulnerabilities, Categories, Removal Techniques etc.
We classify the documents to the malware class that has the highest matching of the characteristics of the document based the ontology model.

We have experimented our approach on large number of documents and found that it gives very good classification as compared to other analysis techniques for documents containing information about Malware.

Contents

1 Introduction **1**

1.1 **Introduction** . . . 1

1.2 **Thesis Organization** . . . 2

2 Survey of Traditional Classification Techniques **3**

2.1 **Survey of Traditional Classification Techniques** . . . 3

2.1.1 **Representing Documents as Vectors** . . . 3

2.1.2 **Supervised Algorithm** . . . 5

2.1.3 **Unsupervised Algorithm** . . . 5

3 Our Aim & Approach **8**

3.1 **Approach** . . . 8

3.1.1 **AlchemyAPI - Automated Concept Extraction** . . . 8

3.1.2 Similarity based classification . . . 9

3.1.3 **Ontology Based Classification** . . . 9

4 Ontology Based Approach **11**

4.1 **Design of Ontology** . . . 11

4.2 **Tool:-** . . . 12

5 Algorithm **18**

5.1 Similarity Based Classification . . . 18

5.1.1 **Representing Documents as Vectors** . . . 18

5.1.2 **Semi-Supervised Centroid Based Clustering** . . . 19

5.2 **Ontology Based Classification** . . . 21

6 Overall System Implementation **23**

6.1 **Overall System & Components** . . . 23

6.1.1 **Analyzing Plaintext Documents using NLP Tool** . . . 23

6.1.2 **Ontology Handler** . . . 24

7 Results & Inferences **25**

7.1 **Results** . . . 25

7.1.1 **Overall Accuracy** . . . 25

7.1.2 **Malware Concept based Accuracy** . . . 26

7.2 **Advantages of Ontology Model** . . . 30

8 Conclusion & Future Work **31**

8.1 **Conclusion** . . . 31

8.2 **Future Work** . . . 31

List of Figures

2.1 Backpropagation_Algorithm 6

3.1 Workflow_Diagram. 10

4.1 Spyware_Hierarchy_1. 12
4.2 Spyware_Hierarchy_2. 13
4.3 Spyware_Hierarchy_3. 14
4.4 Worm_Hierarchy. 15
4.5 Ransomware_Hierarchy_1. 16
4.6 Ransomware_Hierarchy_2. 17

6.1 Training_Data_Format. 24

List of Tables

7.1 Adware Classification 26
7.2 Spyware Classification 27
7.3 Botnet Classification 27
7.4 Virus Classification 27
7.5 Worm Classification 28
7.6 Trojan Classification 28
7.7 Rootkit Classification 28
7.8 Backdoor Classification 29
7.9 Ransomware Classification 29
7.10 Rogue Security Software - Classification 29

Chapter 1

Introduction

1.1 Introduction

New Cyber-Threat attacks are being introduced every day at increasingly rapid pace. These attacks are very much complex and involve multiple stages. Now there are Malware blogs, social sites, newspaper articles that discuss such recent trends of attacks. Analyzing these attack scenarios and figuring out what kinds of attacks these scenarios represent is highly important to understand the properties, vulnerabilities of these attacks and also make systems secure from such attacks. Now the amount of data available is huge, so proper classification of these documents into different malware classes is highly important to further proceed towards necessary steps.

Referring to the recent trends of attacks, we can mention about the Ransomware attack that took place in May 2017 that almost challenged all the security measurements and precautions taken in the windows based systems. It is biggest ransomware attack outbreak in history that spread over many countries including India. The intents behind the WannaCry Ransomware was to exploit money from the victims by encrypting system files on the targeted machine by generating secret key and assuring the victims that the system would be decrypted and released after successful payment of the demanded amount of money in Bitcoins. It actually required a good amount of time for the security experts to find the loophole and bring back the affected systems to the normal state.
So it is highly important to gather and understand all the information regarding such attacks available in blogs or social sites etc, to build a system that is secure from such attacks. Analyzing such textual information and classify them into specific malware classes helps us understand the properties of different such attacks without human intervention.

Some of the techniques that can be used to analyze and classify these documents are briefed below:-

(a) **Pure Text Analyzers considering keywords:-**

Here documents are considered as a bag of words. Tokenizer is utilized to extract most relevant keywords from the text.Now machine learning algorithm can be used to learn any model and use it to classify text data.

(b) **Concept Analyzer:-**
There are text analyzing tools available, that extracts important concepts inferred from the text. These concepts along with relevant weights can alone be used to classify the documents.

(c) **Pure Ontology Based Approach:-**
In this technique, we can structure characteristics, types, causes,vulnerabilities & removal techniques of different malware classes in a hierarchical manner and matches the properties of different subclasses to the document to figure out the malware related concepts from the document.

In the traditional classification based method, first documents are tokenized and then stop words removal technique & some filtering is applied to the data.Now remaining keywords in the whole document set are used to create tf-idf matrix that is used to learn any machine learning model.But this technique considers only important keywords within the text. But in our approach, we want to realize important patterns/phrases in the text to extract Malware related concepts from the text. Understanding the inferred Malware related information from the text requires Natural Language Processing techniques that can analyze the text and figure out what kind of Malware related information is inferred from the text.
In this thesis, we explored integrated approach of NLP Concept Analyzer with Clustering technique & NLP Concept Analyzer with Ontology model to classify malware related documents.

1.2 Thesis Organization

The Thesis represents how Ontology-driven NLP based technique can be used to classify documents containing Malware related information with high accuracy.
Chapter 2 includes the survey of traditional text analysis techniques based on individual domains like Keywords based Machine Learning approach including Supervised & Unsupervised Techniques. Chapter 3 provides the functionality of NLP tool used and also depicts briefly the internal working of the Tool. Chapter 4 includes The design of Ontology model & how it becomes useful for text classification. Chapter 5 describes the Algorithms in detail. Both the Document Clustering & Ontology Driven Classification approaches are described informally and also in pseudocode format. Chapter 6 describes the implementation details, including file formats, output format & Java Packages involved etc. Chapter 7 highlights the result obtained till now and also depicts the observations from the results. Chapter 8 contains the remaining work to be done to enhance the functionality of the system.

Chapter 2

Survey of Traditional Classification Techniques

2.1 Survey of Traditional Classification Techniques

There are various Machine Learning based techniques that can be applied to the Document Classification problem. Most of them start with tokenizing the document and removing insignificant keywords from the documents. Remaining keywords are used as features for the document classification learning algorithm.Learning a model using these features is the process of assigning weights to the set of features.Once the learning model converges these set of weights can be used for the classification of the training set of documents.

Document Clustering is also another technique that can be used in place of Document classification.Here documents with the similar properties clustered into the same bucket. So each of the documents in the same Cluster share similar kinds of properties with each other.

So based on whether to Classify the documents or cluster the documents, the techniques are broadly divided into two main categories. **Supervised Learning** and **Unsupervised Learning**. Approaches of Some of the techniques are briefed here.

2.1.1 Representing Documents as Vectors

The text documents need to be represented as a multi-dimensional vector in the vector space for applying Classification techniques. First, significant keywords are selected from the set of documents that represent the features in the classification or clustering techniques.only terms that relate important characteristics of different classes are taken into consideration.

Using the selected features, each document can now be represented as vectors in multi-dimensional vector space. Now distance or similarity between two vectors gives us useful measurements for classification of the documents. After feature selection & vector formation, the vector space model

for n documents can be represented in the matrix representation.

$$\begin{bmatrix} tw_{i,j}.....tw_{i,n} \\ ..... \\ ..... \\ tw_{m,j}.....tw_{m,n} \end{bmatrix}$$

Here term $tw_{i,j}$ represents weight of $term_i$ with respect to $Document_j$. These matrix is further utilised for feature selection, Classification and clustering of documents.

TF_IDF Matrix

The vector space model described above can be represented as tf-idf matrix format. Here tf represents **term-frequency** & idf stands for **inverse-domain-frequency**. Tf-Idf matrix represents normalized term frequency of various terms in a document. These matrix can be realized for the feature selection purposes. Each row of tf-idf matrix can be calculated as follows:-

$$tf_i = \log(term\_frequency_i + 1)$$

$$idf_i = \log(\frac{N}{document\_frequency_i + 1})$$

$$tw_i = tf_i * idf_i$$

Binary Representation

Binary represents provides information regarding whether a particular is present in a document or not. it does not assign weight to any feature like tf-idf matrix. Let total number of feature is m, so the size of the vector is m. So the vector can be formed as follows.

$$\forall t_i \quad \varepsilon \quad m, \quad tw_i = \begin{cases} 1, & \text{if term } t_i \text{ is present in Document} \\ 0, & \text{otherwise.} \end{cases}$$

2.1.2 Supervised Algorithm

- **Multi-Class SVM:-**

  Support Vector Machine is one of the most regarded Supervised Classification technique. Though it was originally invented for binary classification, but after recent enhancement made to the technique, SVM now can be used for multiclass Classification problem. The aim of the SVM technique is to train the model with training/labeled data points belonging to different classes and creating optimal separating hyperspace between all the classes.Optimal hyperspace is the hyperspace that differentiates between participating classes with maximum margin.The ultimate formulation of the SVM belongs to constrained optimization types of problems. Solving the problem with test data provides us the optimal hyperspace that can be used for the classification of documents.

- **Neural Network:-**

  Neural Network is another very popular Supervised Classification Technique. Neural Network consists of one input layer, one output layer, and one or more hidden layers. Weights are assigned to input to each layer and output from that layer to another layer. Total number of nodes in the input layer is equal to the size of the feature vector and number of nodes in output layer is equal to the number of output classes. The output is a vector of length equal to the number of participating classes with one bit set to 1, and other bits are zero.

  Backpropagation Learning algorithm is used on the training data set to learn the model and assign weights to the links between the nodes of the internal network. Once the system converges , we use it for the classification of the test data set. The learning process in the neural network is depicted in figure 2.1.

2.1.3 Unsupervised Algorithm

- **Centroid Based Clustering:-**

  Probably the simplest way of clustering documents is to use Centroid Based Clustering technique using tf-idf matrix. Here each of the participation classes is represented as a cluster, with the centroid representing the center of mass. Here the centroid represents the optimum point of being a document belongs to the particular cluster. As previously mentioned all the documents can be represented as a point in the n-dimensional space just like the centroid of the clusters by using Word tokenization and tf-idf matrix representation. When a new Document arrives we calculate the Cosine Similarity with the new document with each cluster centroid. Cosine similarity can be described between two vectors as follows:-

$$\boldsymbol{cos(\theta)} = \frac{\vec{\boldsymbol{V}} \cdot \vec{\boldsymbol{W}}}{|\vec{\boldsymbol{V}}| \cdot |\vec{\boldsymbol{W}}|} = \frac{\sum_{i=1}^{n} \boldsymbol{v_i w_i}}{\sqrt{\sum_{i=1}^{n} \boldsymbol{v_i}^2} \sqrt{\sum_{i=1}^{n} \boldsymbol{w_i}^2}}$$

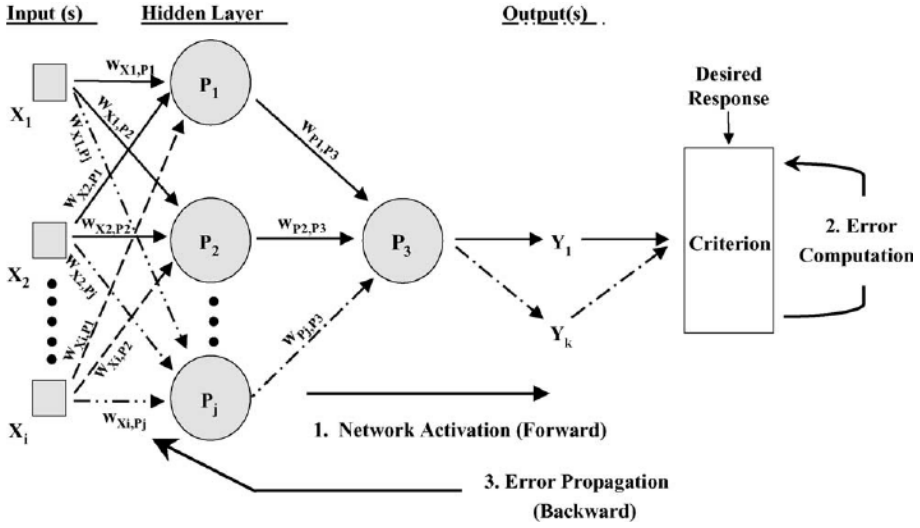


Figure 2.1: Backpropagation_Algorithm

Source:- *Neural net modeling of estuarine indicators: Hindcasting phytoplankton biomass and net ecosystem production in the Neuse (North Carolina)and Trout (Florida) Rivers, USA*. David F. Millie ,Gary R. Weckman,Hans W. Paerl,James L. Pinckney,Brian J. Bendis,Ryan J. Pigg,Gary L. Fahnenstiel

The document is assigned to the cluster with the highest similarity value. **Each time a Document is assigned to the cluster , the centroid is updated to as the average of all the documents in the Cluster**.

Source:- Patrick Trinkle , *An Introduction to Unsupervised Document Classification.*

- **Kmeans Clustering**

  The objective of the algorithm is to choose the optimal clustering that minimizes the following term:-

  $\boldsymbol{SSE = \sum_{i=1}^{k} \sum_{x_j \epsilon C_i} |x_j - \mu_i|^2}$

  After each of the iteration, this error term is re-calculated, when this term is less than certain threshold the algorithm converges.

  At each iteration , for each data point we calculate the minimum distance form that point with each cluster centroid.

  $\boldsymbol{dist_j = argmin_{i=1}^{k}\{|x_j - \mu_i|\}^2}$

  Each time a new point is added to the cluster the centroid is updated as the mean of all the point within the cluster.

  Iteration of these steps continue until the algorithm converges.

$K - MEANS(D, k, \varepsilon)$

initialization

Randomly initialize k centroids:- $\mu_1, \mu_2, \ldots\ldots\mu_k \; \epsilon R^d$

repeat

  $t \leftarrow t + 1$

  $C_j \leftarrow \Phi \quad \forall \quad j = 1, 2\ldots\ldots k$

  foreach $x_j \epsilon D$ **do**

    $j^* \leftarrow argmin_i \{|x_j - \mu_i^t|^2\}$

    $C_{j^*} \leftarrow C_{j^*} \cup (x_j)$

  end

  foreach *i=1 to K* **do**

    $\mu_i^t \leftarrow \frac{\sum_{x_j \epsilon C_i}}{|C_i|}$

  end

until $\sum_{i=1}^{k} |\mu_i^t - \mu_i^{t-1}|^2 \leq \varepsilon$;

Algorithm 1: K_Means Algorithm

Source:-Mohammed J. Zaki, Wagner Meira Jr ,*Data Mining and Analysis Fundamental Concepts and Algorithms,page-335.*

Chapter 3

Our Aim & Approach

Suppose there are large number of documents that contain malware-related information within it. Our aim is to provide the appropriate label to each of the documents based on the malware-related information present in the text. Proper labeling of the documents provides us different sets of documents where each of the set contains documents that infer the same type of malware-related information.As we are considering only plain text documents so the problem transforms to document classification problem with different malware classes as representative classes.

3.1 Approach

In our approach, instead of extracting important words using tf-idf matrix we take significant patterns/phrases into consideration for classification. As different positions of same words in a sentence conveys different meanings, so taking important phrases into consideration instead of only words, gives us the better understanding of the text. We use NLP tool that provides us possible malware related concepts that are inferred from the text and also extracts important patterns or phrases that represent significant properties of specific malware classes. Now we match these phrases with our predefined Ontology model to extract Cyber Threat-related information from the text and further classification or labeling of the document.

3.1.1 AlchemyAPI - Automated Concept Extraction

AlchemyApi is a text analyzing software that provides functionalities for inferring important concepts from the plaintext. Each of the functionalities uses NLP based techniques to analyze the plaintext and extract high-level semantic information from the text. The major functionalities involve: -

- Entity Extraction:
- Keyword Extraction:
- Sentiment Analysis

- Concept Extraction
- Relation Extraction
- Taxonomy Classification
- Author Extraction
- Language Detection

In our experiments, we have used Entity Extraction, Keyword Extraction & Concept Extraction API's only, to extract malware related concepts and features from the text. The output of AlcheyApi has been used for two purposes:-

1. Create concept vector to represent each document as a vector in multi-dimensional space using Alchemy Output.
   Each component of the vector represents different malware classes. The size and format of the concept vector is universal throughout the system implementation.
2. Create three hashmaps of concepts, keywords & entities consisting of the actual terms in the output with the corresponding relevance that is passed through the Ontology Handler for classification.

After representing each document as multidimensional vector , we apply two classification technique:-

3.1.2 Similarity based classification

Alchemy Based Classification

This is simplest and straightforward technique to classify the documents. We obtain vector representation of the document from Alchemy. We now compare each component of the vector to find the class with the largest weight and the document is labeled as that class.

Clustering of Documents

In this technique, we devise a clustering algorithm based on cosine similarity measurement between the centroid of a cluster and document vector. Each time a new document is assigned to a cluster, centroid is updated to the average of all the documents in the cluster. Here centroid is the center of mass of a particular cluster, is the optimal point of any point to be in the cluster. So at each step, we measure the similarity of the document with the centroid.

3.1.3 Ontology Based Classification

NLP tool provides us valuable information extracted from the text as Concepts,Keywords & Entities. But sometimes single Document contains information about Multiple Malware classes rather than one.Sometimes NLP analyzer fails to assign the largest weight to the Malware concept that is

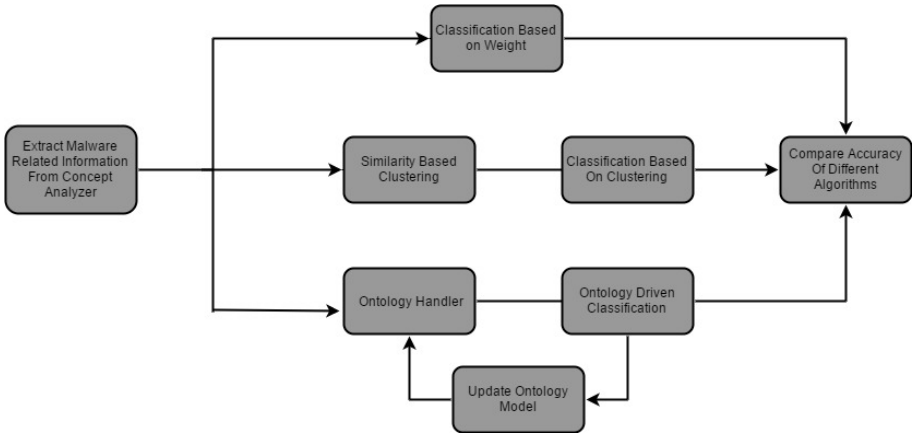


Figure 3.1: Workflow_Diagram.

most relevant to the document between all the Malware related concepts extracted from the text. To classify the documents accurately into specific Malware Classes, we create our own Ontology model. Ontology model consists of different Malware classes, properties, removal techniques etc in a hierarchical manner. We try to match different characteristics of each of the malware classes to the document with respect to the output of NLP Tool , to figure out the classification of the document. The overall functionality of the system is depicted in figure 3.1

Chapter 4

Ontology Based Approach

As discussed in earlier, we use combined approach of NLP-driven Concept Extraction along with Ontology-based Pattern matching to find the proper classification of the documents.

4.1 Design of Ontology

In our Ontology model, we have tried to represent Different Malware classes in a hierarchical manner. Malware Characteristics, Attack Intents, vulnerabilities and removal techniques can be captured by our proposed Ontology model. Malware is the top most class in the hierarchy that is divided into different subclasses like Adware, Spyware, Botnet.... etc. Now each of the classes is further divided into different subclasses. Example:- Malware is divided into the classes listed below:-

[Adware, Spyware, Botnet,Virus, Worm , Trojan,Rootkit, Backdoor,Keylogger, Rogue Security Software, Ransomware,Browser Hijack, Bitcoin]
Now each of the classes are further classified into different subclasses based on Causes,Vulnerabilities,Properties, Types, Removal Techniques.
Example:- For Spyware , we can create different subclasses like:-

- Spyware_causes
- Spyware_Properties
- Spyware_Attacks_&_Vulnerabilities
- Spyware Types
- Spyware Removal

Each of these classes contain properties that represent characteristics of each of the malware classes in a particular domain.

4.2 Tool:-

We have used tool, Protege to visualize the whole Ontology model in structured manner. Protege is opensource Java Ontology editor and it provides GUI based platform to visualize and update the Ontology model.

Protege also provides GUI based interface to query the Ontology model. It provides useful information about the Ontology model. We provide specific class name as the query to the Ontology model, the query returns the superclass, subclasses of the specific class.

In our implementation, we have stored the Ontology model in Malware_Ontolgy.owl file and used org.semanticweb.owlapi package to query the Ontology model. After the Document is passed through AlchemyApi, we obtain different malware related concepts that is inferred from the text. System makes automated query to the Ontology model to extract properties relevant to malware concept. Now it matches the properties of the malware class with the content of the document to figure out which one is the most relevant concept and based on the result it classifies the document. Detailed algorithm is described in Chapter 5.3.

Structure of some of the malware classes is shown below:-

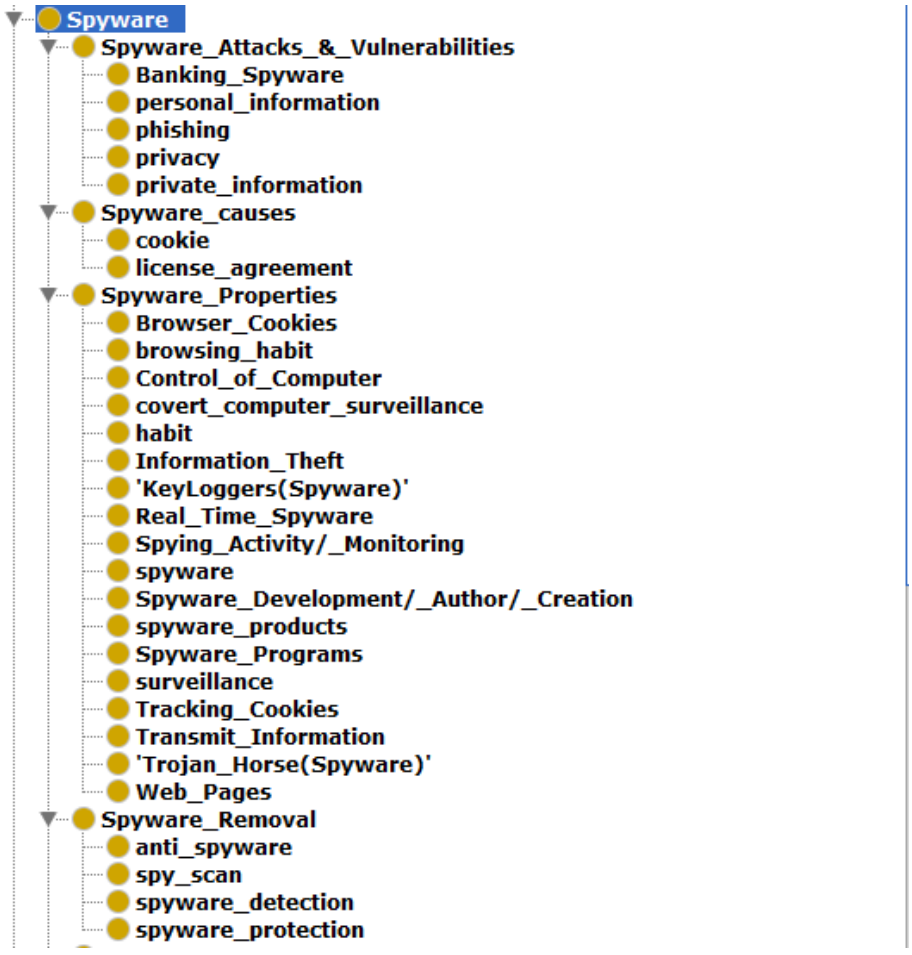


Figure 4.1: Spyware_Hierarchy_1.

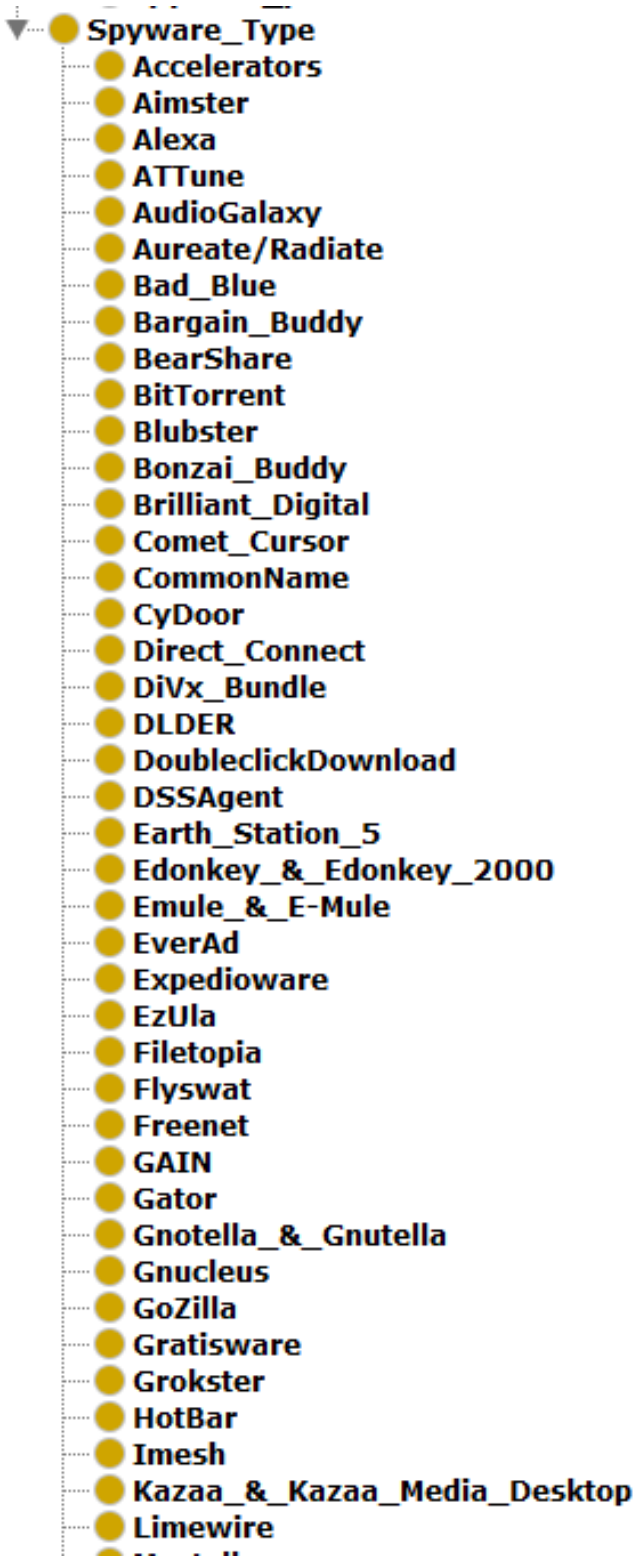


Figure 4.2: Spyware_Hierarchy_2.

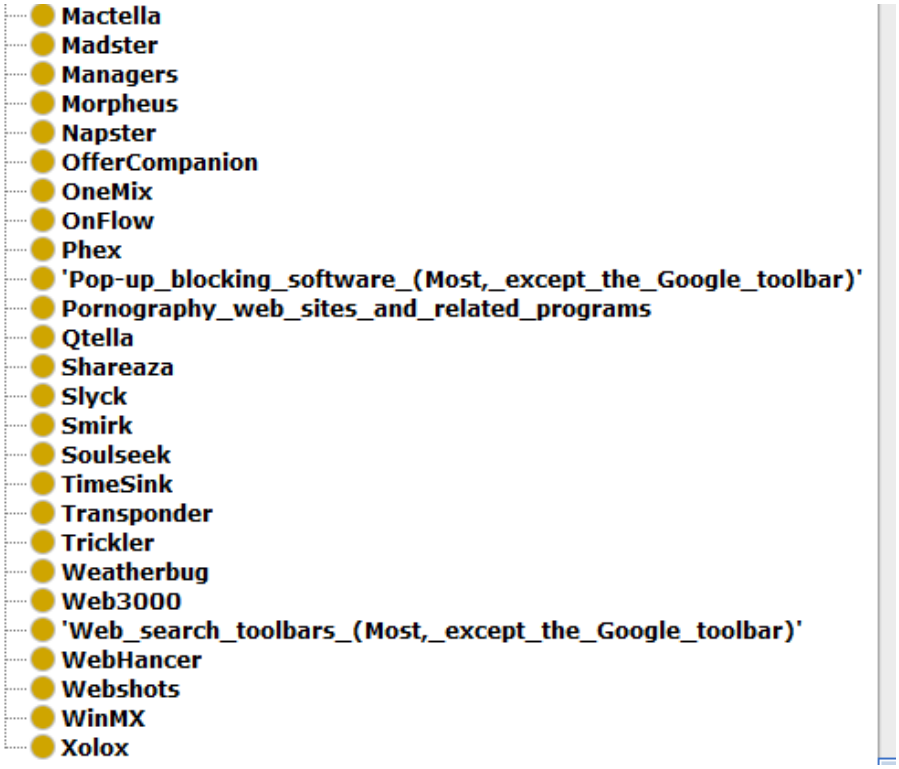


Figure 4.3: Spyware_Hierarchy_3.

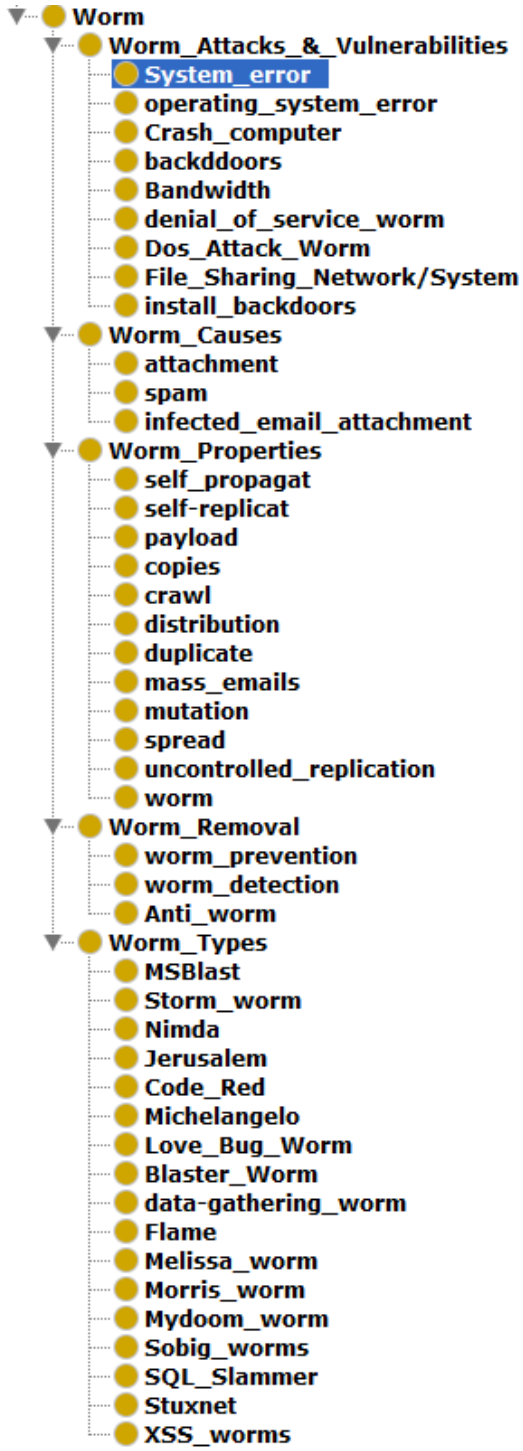


Figure 4.4: Worm_Hierarchy.

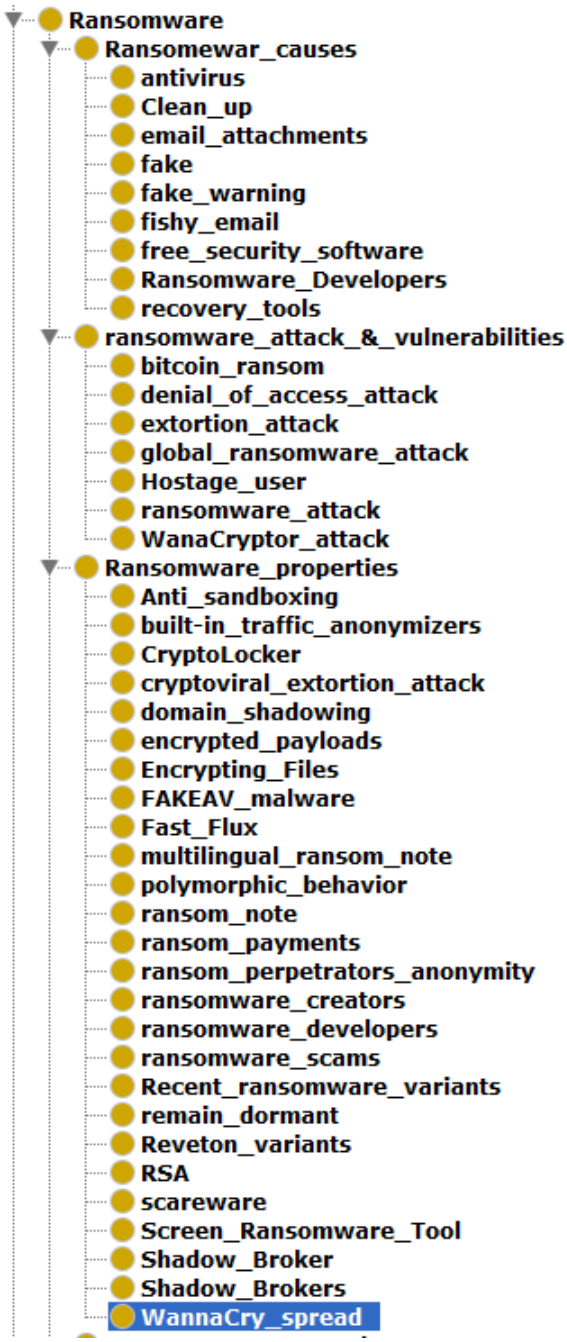


Figure 4.5: Ransomware_Hierarchy_1.

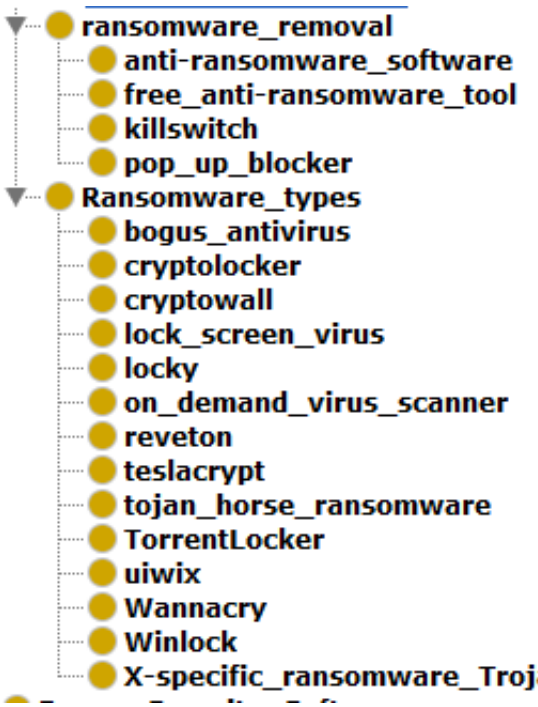


Figure 4.6: Ransomware_Hierarchy_2.

Chapter 5

Algorithm

The proposed algorithm requires three phases of computation and analysis.

5.1 Similarity Based Classification

5.1.1 Representing Documents as Vectors

1. First, choose our customized Concept vector. It contains all the malware concepts that we want to consider.

   - Ex –[Adware,Spyware,Botnet,Virus,Worm,Trojan,Rootkit,Backdoor,Keylogger,Rogue_Security_Software,I
     Browser_Hijacking, Bitcoins].
     It actually represents centroid of each cluster.
   - The vector size can be anything. Our aim is to take into account most popular malware Kinds of Attacks.

2. Run each training document in AlchemyApi, we get different XML output for concepts, keywords, entities etc.

3. Analyze XML output and create three HashMaps of Entities, Keywords & Concepts with the term as key and corresponding relevance as Value. We create such key/value pairs for all the terms present in the XML output of Concepts,Keywords & Entities and put it into respective hashmap.

4. Match each of the concepts of the original concept vector with the hashmaps consisting of Entities,Keywords & concepts.

5. New Vector is created with the concepts that are present in the hashmaps with the highest relevance value.

6. Ex- let we run document D and get following results –

   - Botnet – x1

- Ransomware – x2
- Trojan - x3
- Other Concepts are irrelevant. Now we can represent the document as.
  Document D - [0,0,x1,0,0,x2, 0,0,0,0, x3, 0 , 0]
- Each document can be represented in this format. So each document is represented as vector in n-dimensional space.

5.1.2 Semi-Supervised Centroid Based Clustering

1. **Initialization:-** First Manually select documents that represent a particular type of malware class. Put each document into the specific malware cluster. Now we repeat the same step for all the clusters. That provides us the initial seeding from where we can continue the similarity based clustering algorithm.
   Example- suppose analyzing a document in Alchemy we get the following concepts:-[Botnet, Ransomware, Bitcoin].
   Now suppose Bitcoin is the concept with the largest weight assigned among all the relevant concepts. So we assign the document to the cluster that represents Bitcoin.

2. Each time we initialize any cluster with a new document, the document vector becomes the cluster centroid.

3. Now any new document is first analyzed in Alchemy and we obtain a concept vector. The detailed process is described in section 5.1.
   Suppose analyzing a document in AlchemyAPi , we get the following concepts:-
   $\{0,x_{\mathbf{1}},0,x_{\mathbf{2}},0,0,x_{\mathbf{3}},0...0\}$, where $x_{\mathbf{1}}$,$x_{\mathbf{2}}$ and $x_{\mathbf{3}}$ represents the relevance value of those concepts that is inferred from the text.
   Now for each of the concepts extracted from the document, we calculate similarity with the corresponding cluster centroid and document vector.

4. We assign the document to the cluster with the highest similarity value. Also, we update the centroid value as the average of all the points inside the cluster.

5. **Similarity Algorithm :-** For new Documents , compute cosine similarity between each cluster centroid and the document vector.
   The Document is assigned to the cluster with the largest similarity. The cluster centroid is also updated accordingly.

Each Cluster centroid is represented as a vector.

$$\vec{C_i} = < c_1, c_2, c_3, ...c_n >$$

Suppose for any random Document D, after analyzing the document in AlchemyApi, We get the following concept vector.

$$\vec{D} =< d_1, d_2, d_3, ...d_n >$$

Now the similarity between the new Document and the cluster centroid Ci can be represented as

$$Similarity = \frac{\vec{C_i} \cdot \vec{D}}{|\vec{C_i}| \cdot |\vec{D}|}$$

6. Each time a new Document is assigned to a Cluster, The centroid of the cluster is updated. New centroid is calculated as the average of all the Document Vectors in the Cluster.

$$\vec{C_i} = \frac{\sum_{d \epsilon D_i} \vec{d}}{|D_i|}$$

Here, D_i represents subset of documents that belongs to the cluster C_i.

input : CSV file consisting of Training Documents
output: maximum flow from s to t

Initialize:- Choose a Document that belongs to a particular malware concept & put it in the corresponding Cluster.
Update each Cluster centroid accordingly.
for *every Document D in CSV file* **do**
    similarity=0;
    cluster= null; **for** *every Cluster Centroid* $C_i \in C$ **do**

$$new\_Similarity = \frac{\vec{C_i} \cdot \vec{D}}{|\vec{C_i}| \cdot |\vec{D}|}$$

        if *(new_similarity > similarity)* **then**
            similarity=new_similarity
            cluster=c_i
        end
    end
    if *(similarity>0)* **then**
        Assign Document D to Cluster c_i
        update the cluster centroid.
    end
end
for *every Cluster* $C_i \in C$ **do**
    Print all the Documents in Cluster c_i
end

Algorithm 2: Similarity Based Clustering

5.2 Ontology Based Classification

1. The design of the Ontology model is described in chapter 4. As described earlier different malware classes along with their properties, types, vulnerabilities are structured in a hierarchical manner. Example:- For Adware ...Different Subclasses are as follows:-

   - Adware_Causes
   - Adware_Attacks_&_Vulnerabilities
   - Adware_Properties
   - Adware_Types
   - Adware_Removal

     Similar subclasses have been created for each of the Malware concepts belong to the original concept vector.

2. Each of these subclasses also contains various subclasses that are leaf nodes.These nodes represent characteristics of each subclass in each malware domain.

3. First, any document is analyzed in Alchemy and we get a concept vector that represents probable malware concepts that are extracted from the text. Suppose analyzing a document in AlchemyApi, we obtain the following concept vector:-
   $\{0,x_1,0,0,x_2,0,0,0,x_3,...0\}$, where x_1, x_2,x_3 represents relevance value of the respective concepts that is extracted from the text.
   For each of the malware concept present in the document vector with confidence value greater than zero. We make Java query to the Malware_Ontology.owl file that contains the Ontology model and extract all the properties of the specific malware class.

4. We also generate three Hashmaps for Entities,Keywords and Concepts that is obtained from AlchemyApi output. Each of the Hashmap contains specific term as key and corresponding relevance as value.

5. Now all the properties of specific malware classes are matched with the three hashmaps obtained.We maintain variable similarity that measures the similarity of the document with each malware class. Whenever we find a match (specific property is present in any hashmap), we sum up corresponding relevance value to the similarity variable.

6. Instead of direct String matching, substring search method is applied. Each time we try to match any property(term) of a malware class, we find that term is a substring of any key in three hashmaps. If such match is found we add the corresponding relevance value to the similarity measurement.

7. after ontology based matching with all the malware class that is present in original document vector, we classify the document to the malware class that has highest similarity measurement

according to the above mentioned procedure.

```
input  : Any random Document D (Passed as url)
output: Classification of the Document based on Malware Ontology Model
AlchemyApi:-  Pass Document Url as argument to AlchemyApi.
Create Vector V of Malware concepts & store it in CSV File.
Extract output of AlchemyApi . Create three Hashmaps of Concepts,Keywords & Entities
for every Concept c_i in Vector v do
    class=null ;
    if (c_i > 0) then
        prev_weight = weight ;
        weight = 0 ;
        Query Ontology Model of Malware Concept c_i
        Create list L of all the properties of c_i
        for every key K ∈ Concept.map cmap do
            for every s ∈ L do
                if isSubString(K,s) then
                    Weight = weight+cmap.getValue(K);
                end
            end
        end
        for every key K ∈ Entity.map emap do
            for every s ∈ L do
                if isSubString(K,s) then
                    Weight = weight+emap.getValue(K);
                end
            end
        end
        for every key K ∈ Keyword.map kmap do
            for every s ∈ L do
                if isSubString(K,s) then
                    Weight = weight+kmap.getValue(K);
                end
            end
        end
        if weight > prev_weight then
            Class = c_i ;
        end
    end
end
return class
```

Algorithm 3: Ontology Based Classification

Chapter 6

Overall System Implementation

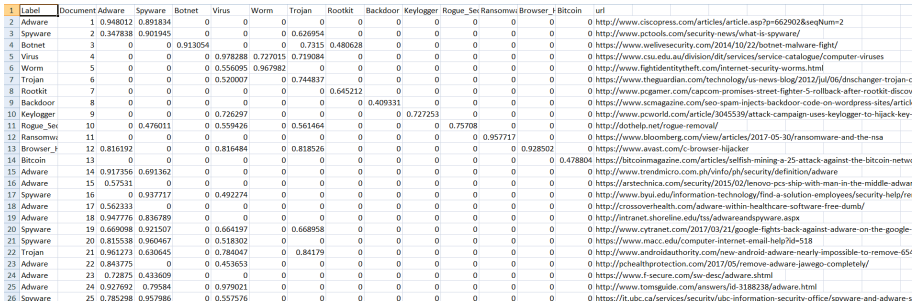
6.1 Overall System & Components

6.1.1 Analyzing Plaintext Documents using NLP Tool

Each of the document is passed through the NLP Tool that can extract Cyber Threat related information that can be infered from the text.

Analyze the Result

- The output obtained is in XML format. So Java packages have been used to extract information from the output files.
- The file Concept.java captures the output of AlchemyApi after running a particular Document in AlchemyApi.
- There lists have been generated that represents Concepts, Keywords & Entities respectively. Each of the items is assigned a weight(Relevance) that represents the probability of that particular phrase.
- It analyzes the Concepts , Entities and keywords list and creates a vector of the following format.
  Concept vector:- [Adware,Spyware,Botnet,Virus,Worm,Trojan,Rootkit,Backdoors,Keyloggers, Rogue Security Software, Ransomware, Browser Hijacker, Bitcoins] The detailed process is described in section 5.1.
- The result is stored in result.csv file. The format of the file is as follows:-
  - Label indicates actual/intended output of the document.
  - Specific Labels indicates specific components of the vector after analyzing the output of AlchemyApi.
  - URL - contains the url of the document.



	Label	Document	Adware	Spyware	Botnet	Virus	Worm	Trojan	Rootkit	Backdoor	Keylogger	Rogue_Se	Ransomwa	Browser_H	Bitcoin	url
2	Adware	1	0.948012	0.891834	0	0	0	0	0	0	0	0	0	0	0	http://www.ciscopress.com/articles/article.asp?p=662902&seqNum=2
3	Spyware	2	0.347838	0.901945	0	0	0	0.626954	0	0	0	0	0	0	0	http://www.pctools.com/security-news/what-is-spyware/
4	Botnet	3	0	0	0.913054	0	0	0.7315	0.480628	0	0	0	0	0	0	https://www.wellivesecurity.com/2014/10/22/botnet-malware-fight/
5	Virus	4	0	0	0	0.978288	0.727015	0.719084	0	0	0	0	0	0	0	https://www.csu.edu.au/division/dit/services/service-catalogue/computer-viruses
6	Worm	5	0	0	0	0.556095	0.967982	0	0	0	0	0	0	0	0	http://www.fightidentitytheft.com/internet-security-worms.html
7	Trojan	6	0	0	0	0.520007	0	0.744837	0	0	0	0	0	0	0	https://www.theguardian.com/technology/us-news-blog/2012/jul/06/dnschanger-trojan-c
8	Rootkit	7	0	0	0	0	0	0	0.645212	0	0	0	0	0	0	http://www.pcgamer.com/capcom-promises-street-fighter-5-rollback-after-rootkit-discov
9	Backdoor	8	0	0	0	0	0	0	0	0.409331	0	0	0	0	0	https://www.scmagazine.com/seo-spam-injects-backdoor-code-on-wordpress-sites/articl
10	Keylogger	9	0	0	0	0.726297	0	0	0	0	0.727253	0	0	0	0	http://www.pcworld.com/article/3045539/attack-campaign-uses-keylogger-to-hijack-key-
11	Rogue_Se	10	0	0.476011	0	0.559426	0	0.561464	0	0	0	0.75708	0	0	0	http://dothelp.net/rogue-removal/
12	Ransomwa	11	0	0	0	0	0	0	0	0	0	0	0.957717	0	0	https://www.bloomberg.com/view/articles/2017-05-30/ransomware-and-the-nsa
13	Browser_H	12	0.816192	0	0	0.816484	0	0.818526	0	0	0	0	0	0.928502	0	https://www.avast.com/c-browser-hijacker
14	Bitcoin	13	0	0	0	0	0	0	0	0	0	0	0	0	0.478804	https://bitcoinmagazine.com/articles/selfish-mining-a-25-attack-against-the-bitcoin-netwo
15	Adware	14	0.917356	0.691362	0	0	0	0	0	0	0	0	0	0	0	http://www.trendmicro.com.ph/vinfo/ph/security/definition/adware
16	Adware	15	0.57531	0	0	0	0	0	0	0	0	0	0	0	0	https://arstechnica.com/security/2015/02/lenovo-pcs-ship-with-man-in-the-middle-adwa
17	Spyware	16	0	0.937717	0	0.492274	0	0	0	0	0	0	0	0	0	http://www.byui.edu/information-technology/find-a-solution-employees/security-help/re
18	Adware	17	0.562333	0	0	0	0	0	0	0	0	0	0	0	0	http://crossoverhealth.com/adware-within-healthcare-software-free-dumb/
19	Adware	18	0.947776	0.836789	0	0	0	0	0	0	0	0	0	0	0	http://intranet.shoreline.edu/tss/adwareandspyware.aspx
20	Spyware	19	0.669098	0.921507	0	0.664197	0	0.668958	0	0	0	0	0	0	0	http://www.cytranet.com/2017/03/21/google-fights-back-against-adware-on-the-google-
21	Spyware	20	0.815538	0.960467	0	0.518302	0	0	0	0	0	0	0	0	0	https://www.macc.edu/computer-internet-email-help?id=518
22	Trojan	21	0.961273	0.630645	0	0.784047	0	0.84179	0	0	0	0	0	0	0	http://www.androidauthority.com/new-android-adware-nearly-impossible-to-remove-654
23	Adware	22	0.843775	0	0	0.453653	0	0	0	0	0	0	0	0	0	http://pchealthprotection.com/2017/05/remove-adware-jawego-completely/
24	Adware	23	0.72875	0.433609	0	0	0	0	0	0	0	0	0	0	0	https://www.f-secure.com/sw-desc/adware.shtml
25	Adware	24	0.927692	0.79584	0	0.979021	0	0	0	0	0	0	0	0	0	http://www.tomsguide.com/answers/id-3188238/adware.html
26	Spyware	25	0.785298	0.957986	0	0.557576	0	0	0	0	0	0	0	0	0	https://it.ubc.ca/services/security/ubc-information-security-office/spyware-and-adware-s

Figure 6.1: Training_Data_Format.

– **Packages Involved:-** AlchemyAPi java, javax.Xml Package & org.xml.sax package(Handling AlchemyAPi Output), org.w3c.dom package(Analyzing the output and automatic updating csv file).

6.1.2 Ontology Handler

- Ontology Handler takes the output vector generated from AlchemyAPi output, make queries to the Ontology model and matches the properties of Malware classes to the AlchemyApi Output.

- AlchemyApi uses a predefined Ontology format in the background to extract **hot phrases** from the plaintext to understand the inferred concept from the text.

In our approach, we also define an Ontology model to represent different Malware classes, it's causes, vulnerabilities , intents in hierarchical manner.

- Based on the AlchemyApi output, Ontology handler make Queries to the Ontology model to extract properties of specific Malware classes.
  Java API has been used to make the query. Extracted set of terms are kept in an ArrayList and further matched to AlchemyApi output to make the classification.

- ontology model is upgraded periodically to capture more specific Malware class related properties.

- **Packages Involved:-** org.semanticweb.owlapi(read the owl file,create owl object,extract information from owl object), org.semanticweb.Hermit packages(Query the .owl file) ,javax.xml package(Handle AlchemyApi output)*, org.w3c.dom package.

Chapter 7

Results & Inferences

7.1 Results

7.1.1 Overall Accuracy

Different methods that is described above is tested on 250 documents. The accuracy of different methods is described below:-

Alchemy Based Classification

- **Accuracy:-** Out of 260 documents 205 documents are classified properly. Remaining 55 documents are classified wrongly.Accuracy of Classification on Malware Related Documents is 78.8%.
- **Analyze The Result:-** AlchemyApi provides us possible Malware related concepts that are inferred from the text.It tries to find out important patterns/phrases from the document using NLP & Text Analyzing techniques and match these phrases with predefined Ontology model to extract proper classification of malware related documents. With each malware related concept it tries to assign a weight that represents the relevance of the document infer that concept.

  But when more than one malware related concept is inferred from the text, in some cases AlchemyAPi fails to assign highest weight to the most significant concept in the text. In our technique, we improve the accuracy of AlchemyApi using our Similarity based Clustering & Ontology driven Classification Technique.

Similarity Based Clustering

Out of 260 documents 226 documents are classified properly.Remaining 34 documents give wrong classification.

- **Accuracy : -** The accuracy of the classification by using Similarity Based Classification technique is - 86.9%.

- **Analyze the Result:-** AlchemyApi analyzes the text within the document and try to extract malware related concepts within the text. The relevance value alongwith the concept provides the probability of any document to infer specific Malware related concepts. But when more than malware related concept is inferred from the text , AlchemyApi fails to figure out which one of the corresponding malware concepts is the most accurate one. So sometimes the actual Malware concept that is inferred from a document is not the highest relevant Malware concept in the AlchemyApi output. Utilizing Clustering technique our aim is to find the optimal centroid for each of the malware clusters. Computing similarity with respect to the optimal centroid improves the accuracy of classification.
  To further improve the accuracy of the classification, Ontology based search technique has been introduced. Ontology based Classification technique tries to figure out specific malware related properties that is present in the document. So for each of the malware concept that is present in the AlchemyApi output , corresponding malware properties are searched in the **Malware_Ontology.owl** file. The corresponding output is matched the Entities,Keywords & concepts in the AlchemyApi output. Whenever a match is found the corresponding relevance value is added to the result.
  The Document is classified as the malware concept that has the highest matched values between Ontology model & AlchemyApi Output.

Ontology Based Classification

- **Accuracy:-** Out of 260 documents 251 documents are classified properly.Remaining 9 documents give wrong classification.

- The accuracy of the classification by using Ontology Based Classification technique is - 96.5%.

7.1.2 Malware Concept based Accuracy

Different Malware Concept based Classification result is as follows:-

- **Adware**

Table 7.1: Adware Classification

Adware Classification						
Label	Alchemy		Similarity		Ontology	
Adware	Correct	Wrong	Correct	Wrong	correct	wrong
Total - 20	14	6	18	2	20	0
Accuracy	0.7		0.9		1	

- **Spyware**

Table 7.2: Spyware Classification

Spyware Classification						
Label	AlchemyApi		Similarity		Ontology	
Spyware	Correct	Wrong	Correct	Wrong	correct	wrong
Total - 17	11	6	12	5	16	1
Accuracy	0.64		0.7		0.94	

- **Botnet**

Table 7.3: Botnet Classification

Botnet Classification						
Label	Alchemy		Similarity		Ontology	
Botnet	correct	wrong	Correct	Wrong	Correct	Wrong
Total - 50	47	3	48	2	50	0
Accuracy	0.94		0.96		1	

- **Virus**

Table 7.4: Virus Classification

Virus Classification						
Label	AlchemyApi		Similarity		Ontology	
Virus	Correct	wrong	Correct	Wrong	Correct	Wrong
Total - 20	19	1	19	1	18	2
Accuracy	0.95		0.95		0.90	

- **Worm**

Table 7.5: Worm Classification

Worm Classification						
Label	Alchemy		Similarity		Ontology	
Worm	Correct	Wrong	Correct	Wrong	Correct	Wrong
Total - 17	12	5	15	2	17	0
Accuracy	0.70		0.88		1	

- **Trojan Horse**

Table 7.6: Trojan Classification

Trojan Classification						
Label	Alchemy		Similarity		Ontology	
Trojan	Correct	Wrong	Correct	Wrong	Correct	Wrong
Total - 24	16	8	22	2	22	2
Accuracy	0.66		0.92		0.92	

- **Rootkit**

Table 7.7: Rootkit Classification

Rootkit Classification						
Label	Alchemy		Similarity		Ontology	
Rootkit	Correct	Wrong	Correct	Wrong	Correct	Wrong
Total - 45	37	8	44	1	44	0
Accuracy	0.82		0.98		1	

- **Backdoors**

Table 7.8: Backdoor Classification

Backdoor Classification						
Label	AlchemyApi		Similarity		Ontology	
Backdoors	Correct	Wrong	Correct	Wrong	Correct	Wrong
Total - 11	7	4	10	1	11	0
Accuracy	0.63		0.90		1	

- **Ransomware**

Table 7.9: Ransomware Classification

Ransomware Classification						
Label	AlchemyApi		Similarity		Ontology	
Ransomware	Correct	Wrong	Correct	Wrong	Correct	Wrong
Total - 20	15	5	16	4	20	0
Accuracy	0.75		0.80		1	

- **Rogue Security Software**

Table 7.10: Rogue Security Software - Classification

Rogue Security Software - Classification						
Label	AlchemyApi		Similarity		Ontology	
Rogue Software	Correct	Wrong	Correct	Wrong	Correct	Wrong
Total - 34	24	10	29	5	31	3
Accuracy	0.70		0.85		0.91	

7.2 Advantages of Ontology Model

There are certain advantages of using Ontology based classification technique over Similarity Based Clustering:-

1. Ontology Based Classification provides better accuracy than Similarity based technique. Both techniques use NLP in the Background, but Ontology model tries to match the significant patterns present in the text that represents properties of different malware classes.

2. Learning is required in both the models but Ontology model requires less amount of time to merge than similarity based technique. Learning with huge number of documents is required in Similarity based technique to achieve high accuracy. On the other hand, Ontology model tends to merge very quickly.

3. Ontology Model provides more information about the text than classification. As the Malware classes are structured in hierarchical manner , so each layer provides significant information about the document.
   Hierarchical Classification of documents can be obtained from the Ontology model.

Chapter 8

Conclusion & Future Work

8.1 Conclusion

In this thesis, we have devised an approach to classify Malware related documents based on NLP analyzer & Ontology matching. According to the experimented result that is depicted earlier this techniques provide high accuracy of classification of Malware related documents with very few bad classifications overall. As the classification is highly dependent on the design of the Ontology model, so improvement of the Ontology model can further improve the classification. Besides classification this technique can also provide other important characteristics regarding the document, it can be information about Malware removal , vulnerabilities and intents. It can be very helpful for security experts, organizations to classify huge number of Malware related documents into appropriate Malware Classes in a completely automated way.

8.2 Future Work

Future work stands at designing more scalable and optimized version of the whole system.

(a) Optimize the design of Ontology model , so that it can capture Malware related information more accurately.

(b) Till now we are getting only the classification of a document to a Malware class. Aim is to devise some kind of learning algorithm , so that besides classification it can provide probability of a document infer the certain Malware concepts.

(c) Use NLP technique to capture more features from the text. Suppose in the ontology model , we are capturing the term "Financial Scam". but in the text the term present "Banking Fraud". But these two term are related conceptually. So NLP techniques can be used to enhance the system to capture features from the text based on the similarity between related patterns / phrases.

Bibliography

[1] Patrick Trinkle. *An Introduction to Unsupervised Document Classification* . University of Maryland Baltimore County,May 12th, 2009.

[2] Rahul Potharaju,Navendu Jain and Cristina Nita-Rotaru. *Juggling the Jigsaw: Towards Automated Problem Inference from Network Trouble Tickets* . April, 2013.

[3] Deepshikha Kalita,Gauhati University; *Supervised and Unsupervised Document Classification-A survey* . 2015.

[4] Vandana Korde and C Namrata Mahender. *Text Classification And Classifiers: A Survey* . 2015.

[5] Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, May 2014. ISBN: 9780521766333.

[6] **AlchemyApi - IBM Watson Natural Language Understanding** , IBM. [Software]
Available from https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/

[7] **Protege 5.1.0- Ontology Editor**.[Software]
Available at -http://protege.stanford.edu/support.php#documentationSupport