NTNU
Norwegian University of
Science and Technology

Extracting Cyber Threat Intelligence From Hacker Forums

Isuf Deliu

Master in Information Security

Submission date:	June 2017
Supervisor:	Katrin Franke, IIK
Co-supervisor:	Carl Leichter, IIK Hai Thanh Nguyen, IDI & Telenor

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Preface

This thesis concludes two years of Master's studies at the Norwegian University of Science and Technology (NTNU). The motivation for this study comes from the joint research interest between Telenor Group and Department of Information Security and Communication Technology at NTNU. Its intended audience includes security practitioners and enthusiasts who want to learn more about proactive cyber security controls.

01-06-2017

Acknowledgment

I would like to express my sincerest gratitude to my supervisor, Professor Katrin Franke, for the support, advice, and help she has provided to me during my two years of Master's studies. I am also grateful to my co-supervisor, Dr. Carl Leichter, who always found time for insightful discussions and suggestions . I would also like to thank Professor Hai Thanh Nguyen for encouraging me to work on this topic, and for his guidance towards achieving the objective.

A special thanks to my friends and fellow colleagues for the time we spent together, having fun and complaining, telling jokes, and having serious conversations.

Finally, I am profoundly indebted to my family. This thesis and everything else in my life would not have been meaningful without their endless love, support, and encouragement. *Faleminderit Shumë!*

I.D.

Abstract

The use of more sophisticated tools and methods from cyber criminals has urged the cyber security community to look for enhancements to traditional security controls. Cyber Threat Intelligence represents one such proactive approach and includes the collection and analysis of information for potential threats from multiple diverse sources of data. The objective is to understand the methodology that different threat actors are using to launch their campaigns, and proactively adapt security controls to detect and prevent such activity. In addition to proprietary feeds, open sources such as social networks, news, online blogs, etc. represent valuable sources of such information. Among them, hacker forums and other platforms used as means of communication between hackers may contain vital information about security threats. The amount of data in such platforms, however, is enormous. Furthermore, their contents are not necessarily related to cyber security. Consequently, the discovery of relevant information using manual analysis is time consuming, ineffective, and requires a significant amount of resources.

In this thesis, we explore the capabilities of Machine Learning methods in the task of locating relevant threat intelligence from hacker forums. We propose the combination of supervised and unsupervised learning in a two-phase process for this purpose. In the first phase, the recent developments in Deep Learning are compared against more traditional methods for text classification. The second phase involves the application of unsupervised topic models to discover the latent themes of the information deemed as relevant from the first phase. An optional third phase which includes the combination of manual analysis with other (semi)automated methods for exploring text data is applied to validate the results and get more details from the data.

We tested these methods on a real hacker forum. The results of the experiments performed on manually labeled datasets show that even simple traditional methods such as Support Vector Machines with n-grams as features yield high performance on the task of classifying the contents of hacker posts. In addition, the experiments support our assumption that a considerable amount of data in such platforms is of general purpose and not relevant to cyber security. The findings from the security related data however include zero-day exploits, leaked credentials, IP addresses of malicious proxy servers, etc. Therefore, the hacker community should be considered an important source of threat intelligence.

Contents

List of Figures

List of Tables

Abbreviations

IDS Intrusion Detection System

IPS Intrusion Prevention System

CTI Cyber Threat Intelligence

ANN Artificial Neural Networks

SVM Support Vector Machines

ML Machine Learning

AI Artificial Intelligence

PCA Principal Component Analysis

ICA Independent Component Analysis

SOM Self-Organizing Maps

LDA Latent Dirichlet Allocation

CFS Correlation Feature Selection

SL Supervised Learning

UL Unsupervised Learning

MCMC Markov Chain Monte Carlo

KL Kullback-Leibler Divergence

CBOW Continuous-Bag-of-Words

GloVe Global Vectors for Word Representations

GPU Graphical Processing Unit

ConvNN Convolutional Neural Networks

LRF Local Receptive Field

TTP Tools Techniques and Procedures

IOC Indicators of Compromise

RAT Remote Administration Tool

NLP Natural Language Processing

TF-IDF Term Frequency-Inverse Document Frequency

IRC Internet Relay Chats

POS Part-of-Speech

PV-DBOW Distributed Bag-of-Words Paragraph Vectors

PV-DM Distributed Memory Paragraph Vectors

RecursiveNN Recursive Neural Networks

RecurrentNN Recurrent Neural Networks

DOS Denial of Service

ReLU Rectifier Linear Unit

k-NN k-Nearest Neighbours

1 Introduction

This introductory chapter provides a general overview of the topic covered in the thesis. The problem that we address in this thesis is identified together with the main contributions.

1.1 Topic covered by the project

Digitization of services has increased the amount of data organizations possess and process on a daily basis. Being one of the greatest assets, protection of data security and privacy should be one of the main priorities for each organization. Traditional security controls, both on host and network level, are able to detect and prevent malicious activities, but are struggling to keep up with the pace and sophistication of cyber criminal tools and methods. Cyber criminals (also called hackers), are spending more time and resources on preparing advanced and targeted attacks, which are often able to circumvent conventional security controls such as firewalls, intrusion detection and prevention (IDS/IPS) systems, etc. In addition, the existing controls are mainly reactive; that is, they are usually updated with information from the analysis performed after a successful attack. Due to the sensitivity and importance of the data, more proactive approaches are necessary to increase the effectiveness of cyber security protection.

In the recent years, one such approach called Cyber Threat Intelligence (CTI) has gained the focus of the security community. The main idea of CTI is the enrichment of traditional security controls with information collected from multiple diverse sources, both in-house and external. In other words, organizations are building specialized teams to do research on potential threats, their intention, tools and methods they use, in order to anticipate future attacks. This allows updating the security controls in a timely manner and therefore increases the chance of detecting and preventing malicious activities.

A variety of threat intelligence sources exist, including proprietary vendor feeds and open sources. According to a SANS survey [1] sharing within the community remains the main source of intelligence, whereas open sources were regarded as an important part of CTI by half of the respondents. Open sources in this context represent platforms which are freely available on the internet and accessible by everyone with the adequate expertise and tools. Typical examples include online blogs, forums, social networks, news, Dark Web, etc. In this thesis, we focus on the last source, Dark Web, which is defined as the part of the internet accessed only through special software such as TOR[1]. More concretely, we study hacker forums as they represent a considerable part of Dark Web and have great potential for valuable threat intelligence. Only 20% of respondents from the SANS survey [1] declared the use of intelligence from vendors which monitor the Dark Web as an integrated part of their CTI platforms. Therefore, we believe that this project will demonstrate the relevance of knowledge which can be extracted from such sources in the protection of an organization's assets.

1.2 Keywords

Cyber Threat Intelligence, Open-Source Threat Intelligence (OSINT), Dark Web, Hacker Forums, Machine Learning, Deep Learning, Text Classification, Topic Modeling

1.3 Problem description

Despite all the security counter-measures implemented by security practitioners, the protection of data and other assets' security is an ongoing process with no winners. Continuous advances in technology have many obvious benefits, but at the same time they open new attack vectors that should be handled properly. As technology advances, so do the cyber criminals. In addition to the use of more sophisticated tools and techniques, the access to exploits is easier than ever before. Launching cyber attacks targeting individuals or organizations does not require the original creation of the attack payload. The emergence of so-called hacking-as-a-service has enabled even those without proper skills to easily launch cyber attacks. Different communication channels such as online forums are being used by cyber criminals to exchange these services. Unfortunately, even though the existence of such platforms is known, little has been done to leverage their contents to enhance the security controls. One of the main challenges with analyzing such sources is the presence of large amounts of data, not necessary related to cyber security. Cyber security related posts cover only a part of all illegal "goods" that can be found on such platforms. Additionally, their identification is just the first step in the process of discovering information relevant to different organizations. Based on the services they provide and assets they have to protect organizations are exposed to different cyber threats. Therefore, manual analysis of large collections of data is like looking for a needle in a haystack: time-consuming, resource demanding, and inefficient.

[1] https://www.torproject.org/

1.4 Justification, motivation and benefits

This research takes its motivation from the mutual interest in its findings by the industry and academic research at Norwegian University of Science and Technology. In particular, enterprises such as Telenor[2] are continuously looking for automated or semi-automated methods to enrich their cyber security controls. Analysis of data from hacker forums is an effective approach to get more insights on potential attackers, their motivation, tools, and techniques [2, 3, 4, 5]. This allows security practitioners to discover trends of these communities, expand their knowledge base of potential threats to the organization, and take the necessary counter-measures to deal with such threats.

The method proposed in this thesis is not limited to cyber security related data. Other institutions, organization, and agencies such as Law Enforcement Agencies (LEA) can adapt this method when processing large corpora of text documents such as emails, text messages, social network posts, etc.

1.5 Research questions

This thesis seek to answer the following research questions:

1. *How can different Machine Learning methods be used to classify the contents of hacker forums?*

2. *What are the main topics related to cyber security on such forums, and how does filtering through classification affect the discovered topics?*

1.6 Contributions

The primary contribution of this thesis is the exploration of potential of hacker forums as a source of cyber threat intelligence. This is achieved by using an automated process which consists of two main phases: (i) classification of the contents of forums and (ii) discovery of the main topics pertaining to cyber security. The classification performance of recent Deep Learning algorithms is compared with more traditional Machine Learning methods in a both a binary- and multi-class dataset constructed from a real hacker forum. To the best of our knowledge, this comparison is a novel contribution to the cyber security community. Additionally, we show the effect of filtering data irrelevant to security in the topics discovered using unsupervised machine learning. This effect is measured in the quality of topics (i.e. coherence) and the time required to run the algorithms.

[2] https://www.telenor.com

1.7 Thesis Structure

This remainder of this thesis is structured as follows:

- Chapter 2 provides an overview of the background knowledge required to understand the work done in this project. We discuss the main concepts of Machine Learning, including its phases and algorithms. A special focus is centered on Deep Learning, a sub-field of Machine Learning, and its usage for document classification.

- Chapter 3 gives a short introduction to Cyber Threat Intelligence and its benefits.

- Chapter 4 presents the state of the art on Dark Web research. It provides a brief introduction to different research areas on hacker forums, followed by a more detailed discussion on work related to content analysis.

- Chapter 5 describes the method we propose to tackle the problem identified by this thesis.

- Chapter 6 presents the results of the proposed method in real hacker data. The settings of the experiments including computation resources and tools are described first, followed by a detailed discussion on the results of the methods.

- Chapter 7 discusses the main findings of the thesis, and their theoretical and practical implications.

- Chapter 8 provides a summary of the thesis, and identifies ideas for improvements as part of future research.

2 Theoretical Background

The purpose of this chapter is to provide the reader with an overview of the main theoretical concepts required for understanding this thesis. We begin with an introduction to the main concepts of the field of Machine Learning with a focus on the algorithms that we use on this thesis: unsupervised Topic Models, supervised Artificial Neural Networks (ANN) and Support Vector Machines (SVM). Further, we provide an overview of the methods that are used to represent text as feature vectors, including distributional and distributed representations. Finally, a brief introduction to Deep Learning is given.

2.1 Machine Learning

With the ever increasing of the number of devices connected to the internet, the volume of data produced in daily basis by an average size company surpasses Gigabytes or even Terabytes. This explosion of the data has made the manual analysis of such data simply impractical. This obstacle demands automated algorithms to get insights into the data. One such approach is Machine Learning (ML), a suite of automated algorithms used to discover patterns from the data. These patterns can be used to anticipate the hidden structure of unseen data and take the necessary action accordingly.

Machine Learning is a subfield of Artificial Intelligence (AI) [6] and the two are often used synonymously. Even so, though the objective of all fields of AI is to introduce intelligence to computer programs, the manner on how this is achieved in ML is different from other subfields of AI . More concretely, the difference is in the learning process; ML is a data driven approach which aims to solve problems by utilizing the experience from the past examples [7]. That is, the algorithms are not specifically designed to solve a particular problem, which is often the case with other forms of AI where special algorithms are designed to play Chess, solve Sudoku, etc. We believe that all the readers of this thesis have already used at least one application of machine learning. A typical example are social networks such as Facebook which use machine learning for tagging objects and persons in photographs, tailoring the news feed such that pages that user interacts with the most appear on the top, as well as friend and page suggestions, etc.

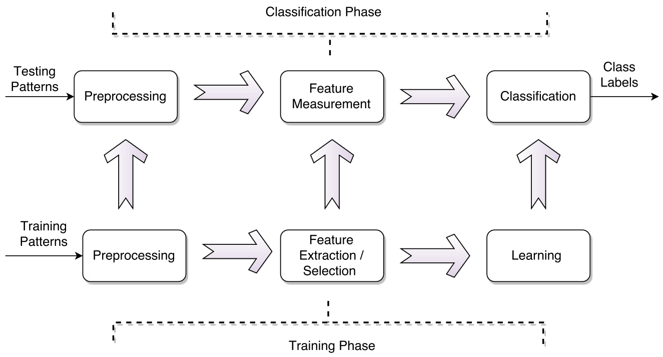


Figure 1: Machine Learning process [8]

Figure 1 shows the main building blocks of the machine learning process consisting of two phases: training and classification. The role of the first phase, training, is to build a model which is able to predict or discover some hidden aspects of the data. It begins by processing, cleaning, and representing the raw data as vectors, where each of the values of the vectors represent some salient feature of the raw data. The quality of these feature vectors is enhanced by creating new features as a combination of existing ones (feature extraction) or by selecting only a subset of the features to reduce the dimensionality of the vectors (feature selection). The first phase is concluded by training or learning the model itself, which includes learning and tuning specific parameters to the given problem. The second phase, classification, is the phase when the model built during the training is used to perform a given task (e.g. clustering) on new data samples. Prior to that, the new data samples are preprocessed and represented with the same features as during the training. The output of this phase depends on the type of algorithm and can be class labels, data clusters, real values for regression, etc. The following is a more detailed discussion on each of these steps.

Preprocessing

The "*No Free Lunch*" theorem in machine learning states that no algorithm is suitable to solve every problem in an effective and efficient manner. This is mainly due to the fact that each algorithm has been designed differently to work on certain problems, data, and configuration. Since changing the algorithm and tailoring it to new combinations is not a practical solution, preprocessing methods plays a crucial role in the entire machine learning process.

There are several preprocessing methods that can be applied, but at the very

least, the data should be represented in a format that can be handled by the learning algorithm. The data in real scenarios is usually in raw format such as text from email conversations, images from social media, network traffic, source code, etc. Unfortunately, most learning algorithms are capable of handling data only when available as feature vectors. The methods to achieve such representation include automated and hand-crafted features as well as a combination of both approaches. In this thesis, we make use of methods which can automatically learn features. Not only may algorithms be designed to work on feature vectors rather on raw data, but also they can be designed to work for certain types of features or problems. Therefore, methods such as binarization of the problem and/or features, discretization of continuous features, transformation of discrete features to continuous, handling of categorical data, etc. [6] are often performed as a preprocessing step.
Noise and outliers are often present on the data, either due to errors in measurement or tools used to capture it. Regardless of the source, their presence can result in a sub-optimal solution for the given problem. Therefore, appropriate methods for identification and removal of noise and handling of outliers should be in place before further analysis.

Having data in different units (e.g. meters, kilograms) and range can also have a negative impact in the overall results of the analysis. Typical examples are algorithms that rely on distances between data samples (e.g k-Nearest Neighbors). They calculate the distance (e.g. Euclidean) between each two feature vectors where each feature gives its contribution to the distance. Having one feature in the range of tens and another in the range of thousands would mean that the impact of low-value feature on the overall distance would be negligible. Thus, the use of data normalization and standardization is preferable. In order to handle data in different units, oftentimes z-transformations are applied.

Feature extraction and/or selection

The second step in the process entails two different methods, namely feature extraction and selection. Though often used interchangeably, they are different processes which can be run independently of each other. Application of both approaches is also possible; for example, feature selection can be used after the feature extraction process.

Feature Extraction is the process of transforming the existing feature space to a new space with lower dimensions. In other words, it is the construction of new features from existing ones but in a new dimensional space. The functions for doing so can be linear or non-linear and examples include Principal Component Analysis (PCA), Independent Component Analysis (ICA), Self-Organizing

Maps (SOM), Latent Dirichlet Allocation (LDA), etc. Formally, given a set of existing features $A_1, A_2, ..., A_n$, a mapping function F is applied to obtain the new features $B_1, B_2, ..., B_m$ where each feature is a mapping $B_i = F_i(A_1, A_2, ..., A_n)$

On the other hand, the objective of **Feature Selection** is to find the best subset of *m* features from all of the existing features *n* where (m<n). Not all the features contribute to the differentiation of data to different classes and redundant features may also exist. There are three main feature selection methods: wrappers, filters, and embedded methods. The main difference between wrappers and filters is the evaluation criterion [9]. More specifically, the difference is that wrappers involve learning in the evaluation criteria while filters do not. Nguyen et al. [10] identified the size of the data as one of the factors to consider between wrappers and filters. This is due to the fact that wrapper methods involve learning; if the dataset size becomes enormously large due to computational requirement, one may want to consider filters. On the other hand, embedded methods perform feature selection as an integrated part of modeling building(learning). A typical example are decision trees. To summarize, feature selection uses ranking or evaluation of feature subsets to find an optimal subset of features which have similar performance with a smaller number of features. The most common methods include Correlation Feature Selection (CFS), Mutual Information based Feature Selection, and a generalized method proposed by Nguyen [11].

Several factors need to be taken into account when deciding which one to use: feature extraction, selection, or both. Jain et al. in [8] argue that as the feature selection does not change the existing features they retain their physical interpretation, and as the number of features is smaller, the cost (time and space) of further analysis becomes more optimal. Alternatively, as feature extraction includes applying linear or non-linear mapping functions to the existing features, the feature space topology is not preserved, and therefore it is very likely that the physical interpretation of features is lost. However, the main advantage of feature extraction is the construction of new features which can be more discriminating than any other subset of existing features.

Learning

One of the properties of machine learning that distinguishes it from other automated methods is the ability to learn. Learning or training in this sense is defined as the process which allows the algorithms to improve their own performance on a given task. This is the core functionality of machine learning and its role is to build a generalized model which can be used to the discover latent structure of unseen data. There are the three main types of learning: supervised, unsupervised, and reinforcement learning. A fourth type called semi-supervised learning also exists, and represents an ensemble method of supervised and unsupervised learning.

2.1.1 Reinforcement Learning

Reinforcement Learning is different from other approaches as it involves an interactive model where an agent (learner) learns to perform a certain task through interaction with the environment and trial-and-error. More concretely, in each interaction the agent receives some form of feedback about the current state of the environment and takes an action based on that. That action is then evaluated and a reward or punishment is given to the agent depending on the effect that action had towards the achievement of the immediate goal. As the goal is to maximize the reward over the long term, the agent has to explore the environment in conjunction to what it knows (exploitation), constituting one of its major differences with other forms of classification [12].

2.1.2 Supervised Learning

Due to the availability of data samples with corresponding output labels, Supervised Learning (SL) is analogous to learning with the teacher. A considerable number of (data_samples, output_class) pairs serve as the "teacher" for the classifier, which uses the information from those samples to adapt its internal structure and therefore build a generalized model which is able to model new and unseen data samples. Let each sample X_i be represented by a feature vector $\{x_1, x_2, ..., x_n\}$ and $Y_i = \{y_1, y_2, ..., y_m\}$ be the set of all m-possible outputs. The objective is to find a function F such that $Y_i = F(X_i)$, for each sample X_i. Regarding the type of output class, two tasks of supervised learning exists: *Classification* and *Regression*.

Regression is one of the simplest learning models, as it represents the output (y) as a simple linear combination of inputs (x). Formally, the objective of (linear) regression is to learn a function such that $y = \alpha x + \beta$. Since the value of the input can be continuous the output values can be continuous as well. On the other hand, classification can be described as doing regression using binary logic. That is, the output value of classification is no longer a continuous but a discrete value.

Support Vector Machines

The main principle of the Support Vector Machine (SVM) classifier lies in the separation of data belonging to different classes by learning to place a separation hyperplane between classes in the feature space. Should the data be linearly separable, the hyperplane is placed in the original feature space. Otherwise, special functions are used to transform the data and map to a higher dimensional feature space [13]. These functions are called Kernels and enable SVM to solve non-linear problems, making SVM one of the most commonly used supervised classifiers.

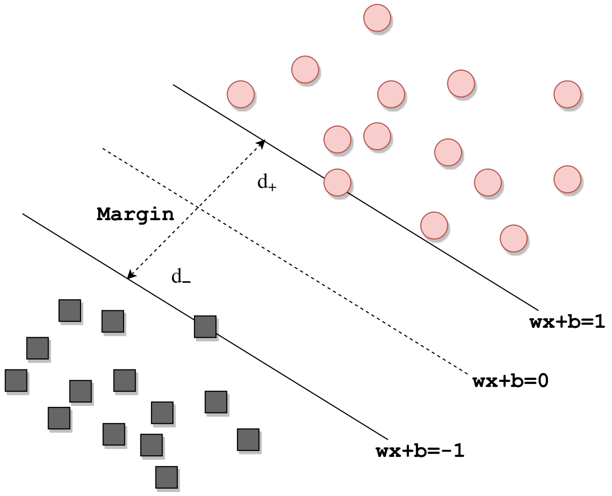


Figure 2: Linearly Separable Data for SVM

To demonstrate the working principle of SVMs, let x denote data samples and y their corresponding classes such that $y \in [-1, 1]$. For linearly separable data the following equations hold:

$$\begin{aligned} wx_i + b \geq 1 \;\; \text{if} \;\; y_i = 1 \\ wx_i + b \leq -1 \;\; \text{if} \;\; y_i = -1 \end{aligned} \tag{2.1}$$

where w is a vector and b is a scalar (bias). During the learning process, SVM tries to find an optimal hyperplane which maximizes the margin, defined as the distance between the hyperplane and the feature vectors closest to the hyperplane. These vectors are called support vectors, whereas the optimal hyperplane satisfies the following equation:

$$w_o x_i + b_o = 0 \tag{2.2}$$

The optimal hyperplane can be found by expressing it as an optimization maximization problem [6, 14] :

$$W(\alpha) = \sum_{j=1}^{n} \alpha_j - \frac{1}{2} \sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j (x_i, x_j); \tag{2.3}$$

where alphas represent positive Lagrange multipliers (variables of a strategy to find the minima of a function) with the following properties:

$$\begin{gathered} \alpha_j \geq 1; \forall j = 1, 2, ..., n \\ \sum_{j=1}^{n} \alpha_j y_j \geq 0; \end{gathered} \tag{2.4}$$

Artificial Neural Networks

Despite great advances in technology, there is still a considerable difference in performance between humans and computers when it comes to tasks such as object recognition. We, as humans, can easily understand an audio speech or recognize a cat in a picture; these however are not so trivial for computer programs. Artificial Neural Networks (ANN) are computational models which aim to achieve comparable performance on pattern recognition tasks by mimicking the information processing principle of human brain [15].

Analogous to the human brain, the basic processing unit of ANN architecture are *neurons*, linked to each other through direct connection called as *weights*. The role of a neuron is to compute the sum of all its inputs with the corresponding weights and pass it to an activation function which then calculates the output value of the neuron. The first neuron architecture proposed by McCulloch and Pitts [16] in 1943 is depicted in the figure 3. That model used a simple threshold activation function, whereas other functions including sigmoid, linear, tangent, Gaussian, etc. are available to use nowadays.

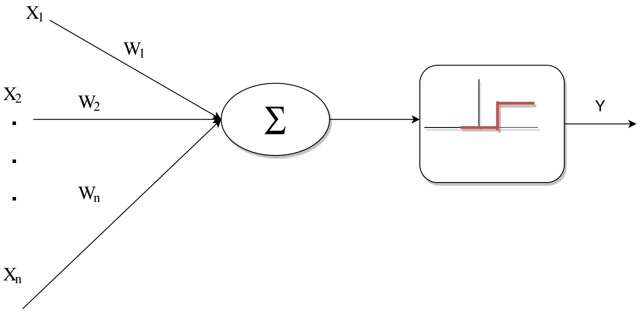


Figure 3: The neuron model proposed by McCulloch and Pitts [16]

An ANN is composed of several layers of neurons. At the very least, a simple ANN consists of two layers; the input layer, which is responsible for accepting the data in form of feature vectors, and the output layer which shows the result of the problem being solved. For classification, the number of neurons in the first layer is equal to the number of features extracted/selected from the data, whilst the number of classes determines the number of neurons in the output layer. This simple two-layered architecture can only solve problems with linearly separable data [15, 17, 6]. In order to learn to model more complex data topologies, one or more hidden layers are introduced between these two layers. The number of hidden layers and neurons per layer is a parameter to be decided by the designer/user of the ANN as a trade-off between computational complexity and performance.

Learning in ANN is defined as the process of creating a model which reflects data topology and can be used to perform a given task. This involves the updating of weights in each layer in order to increase performance. The four main learning rules used so far in the literature are: Error-Correlation rule, Boltzmann learning, Hebbian rule, and Competitive learning [6, 15, 17]. In the following, we briefly describe the main idea behind the error-correlation rule, one of the most commonly used learning methods. The ANN architectures we use in this thesis also use this method, which in the literature can be found under different names: the error-correlation rule for learning a two-layered feedforward ANN is often called Delta Learning rule, whilst the rule for learning a feedforward ANN consisting of multiple layers (including hidden layers) is denoted as Back-Propagation [6].

To illustrate the learning process in ANN, we consider the architecture shown in figure 4: a simple architecture with standard input, output, and a hidden layer. Generally, the goal of learning is to minimize the error defined as the difference between the desired output (d) and the output anticipated by the ANN (y). This

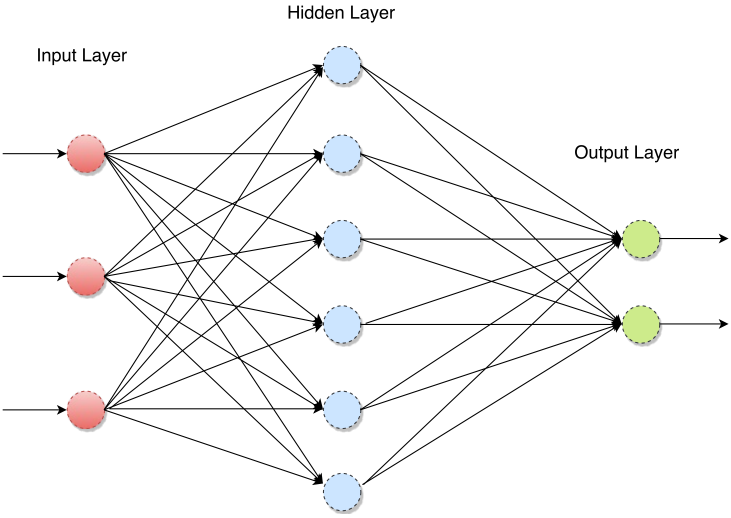


Figure 4: Artificial Neural Networks

is achieved in two phases: in the first phase, the algorithm computes the value of the output by going forward in each layer of the network. After the random initialization of weights, the input to the first hidden layer is calculated as follows:

$$u_i(s) = \sum_{j=1}^{j=N_x} W_{ji}^{(1)} * X_j(s) \tag{2.5}$$

where X(s) represents the learning sample, W the corresponding weights, and N_x the number of neurons in the input layer. Often an extra input X_0 called bias is introduced to avoid the situation where all the weights are randomly initialized with zero. The output for each neuron is computed using the following expression:

$$h_i(s) = \phi(u_i(s)) \tag{2.6}$$

where ϕ is the activation function. In an architecture with multiple hidden layers the output of the first hidden layer serves as input to the second hidden layers and so on. This continues until the last layer in the architecture, the input and output values of which are calculated as follows:

$$\begin{gathered} u_i'(s) = \sum_{j=1}^{j=N_h} W_{ji}' * h_j(s) \\ Y_i(s) = \phi(u_i'(s)) \end{gathered} \tag{2.7}$$

where N_h is the number of neurons on the hidden layer. This concludes the forward phase. Now, we turn to the second phase which includes minimization of the error, formally defined as :

$$E(s) = \frac{1}{2}\sum_{i=1}^{N_y} e_i(s)^2 = \frac{1}{2}\sum_{i=1}^{N_m}(D_i(s) - Y_i(s))^2 \tag{2.8}$$

where Y_i is the output as predicted by the algorithm, D_i the real (desired) output, and N_y the total number of predictions. Gradient Descent is among the most commonly used method for minimization of error. Its main idea is the update of weights in the direction of the negative descent. Formally, this is expressed as:

$$W_{updated} = W_{old} - \eta\frac{\partial E(s)}{\partial W_{old}} \tag{2.9}$$

where η is defined as Learning Rate and represents a positive value in the interval $0 \leq \eta \leq 1$. The formula for updating weights in the architectures like the one in figure 4 is the following:

$$\begin{aligned} W_{ji}^{updated(2)} &= W_{ji}^{old(2)} - \eta\frac{\partial E(s)}{\partial u_i'(s)}h_i(s) \\ W_{ji}^{updated(1)} &= W_{ji}^{old(1)} - \eta\frac{\partial E(s)}{\partial u_i(s)}X_i(s) \end{aligned} \tag{2.10}$$

where the first equation holds for weights between the output layer and hidden layer, while the weights between the hidden layer and input layer are updated according to the second equation. This concludes a single epochs of learning with ANN; it is repeated until the maximum number of epochs is reached, or the error has dropped below a threshold.

2.1.3 Unsupervised Learning

Unsupervised Learning (UL) is the process of inferring properties of the data without having any feedback from "the teacher" as in supervised learning nor from the environment as in reinforcement learning. The most popular form of unsupervised learning is clustering, often mistakenly used as synonym of UL. The main idea of clustering is to group the data so that data within the group are more similar to each other than to data belonging to other groups (i.e. clusters) [18]. This is achieved by using dissimilarity measures such as Euclidean or Manhattan Distance, Cosine dissimilarity, etc. and algorithms such as k-Means which use these metrics to perform the actual clustering.

Other important unsupervised algorithms also exist; for example, PCA, ICA, and SOM are unsupervised algorithms which are used to reduce the dimensionality of the input data by mapping it to another feature space [19]. Algorithms for approximating posterior probability such as Markov chain Monte Carlo (MCMC), Laplace approximation, Variational approximations, etc. also belong to the class of unsupervised learning. In this thesis, we use unsupervised Probabilistic Topic models, and a brief discussion on their main properties follows.

Probabilistic Topic Models

Probabilistic topic models are unsupervised machine learning methods used to discover the main topics or themes of a large collection of (unstructured) corpora of documents. The annotation with thematic tags contributes to a better organization of documents and an effective search based on the topics rather than keywords. This is extensively useful when no prior information is known about the content of the documents. For example, Griffiths and Steyvers [20] explain the role of topic models as analogous to the role of abstract for a scientific paper. They are able to extract meaningful information about the documents which can serve as a guide to analysts/investigators when considering certain documents for further analysis. That is, they provide general information about large collection of documents, but further analysis are required to learn the details of each document.

Several topic modeling algorithms exit, and in this thesis we only consider Latent Dirichlet Allocation (LDA). This is one of the simplest models, used extensively in the existing literature, and also serves the objective of the thesis. The main idea of LDA is the assumption that documents come from a generative process, where the topics are selected first, and words for each document are chosen with respect to those topics. A topic in the LDA terminology is defined as a distribution of words used in a similar context. Another assumption of LDA is that documents exhibit several topics but in different probabilistic distribution [21, 20, 22, 23]. In order to illustrate this process, let us suppose that we are asked to write a document on machine learning algorithms. According to LDA, the topics of the documents should be known prior to any word in the document. Let our document exhibit three different topics: supervised, unsupervised, and reinforcement learning. Since documents exhibit topics in different distribution let these value be 50% for supervised learning topic, and an equal distribution of 25% for the other two topics. Once the topics and their distribution are known, the generation of all the words in a document is performed first by selecting a topic (e.g. unsupervised) from three possible topics in our case, and then choosing a random word from that topic (e.g.

clustering). Formally, this process is formulated as follows [21, 20, 22, 23]:

1. Choose a multinominal topic distribution β_k for each topic k;

   For each document d:

   - Choose a document-topic distribution $\Theta_d \approx \mathrm{Dir}(\alpha)$
   - For each of the n words $w_{d,n}$ in a document d:
     1. Choose a topic $z_{d,n}$ from document-topic distribution Θ_d
     2. Choose a word from that topic $w_{d,n} \approx \mathrm{Multi}(\beta_{z_{d,n}})$

where Dir is the Dirichlet distribution parametrized by α and β which determine the smoothness of topic and word distribution respectively [22].

Posterior Inference

Given a corpora of unlabeled documents, LDA produces two matrices as outputs: the first one is the distribution of topics per document , while the other is the distribution of words per topic. The general idea of LDA can also be illustrated using the graphical representation of models that include hidden variables. This representation is shown in figure 5, where α and β are usually considered constants per corpus, the shadowed variable w represent the observed words, whereas the latent variables, the values of which we want to learn, are the document-topic distribution Θ_d and the word-topic assignment z. The plates represent the documents (M) and the words in each document (N) respectively.

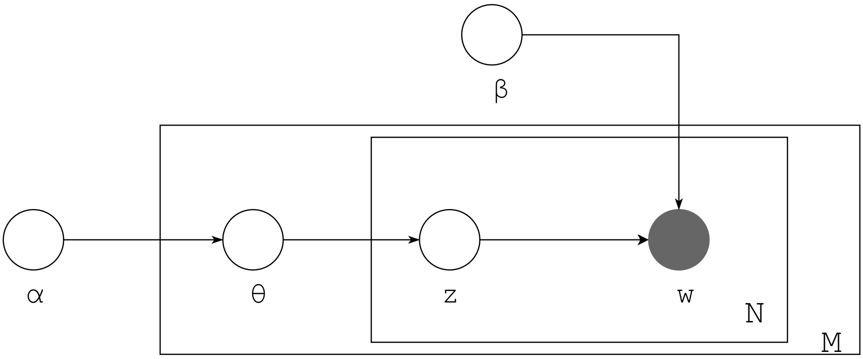


Figure 5: Graphical Representation of Latent Dirichlet Allocation

Given this interpretation, the main computational problem of probabilistic topic models is to compute the conditional probability (posterior) of latent variables given observables (words). Since the documents to be collected may contain millions of words, the exact computation of this probability becomes infeasible, and therefore approximation methods are used. The two main groups of approximation methods include *sampling* and *variational* methods.

Sampling methods work by sampling a tractable distribution from a high dimensional distribution such as LDA posterior [22]. They make use of MCMCs which are modeled to have the posterior distribution as their target [20]. One of the most commonly used algorithms is Gibbs sampling. The first step in this algorithm is to randomly initialize topic assignment for each word in the given document. Further, these assignments are updated by computing the probability of assigning the current word to each of the topics, given the topic assignment of all other words [22]. The update is performed using the expression proposed by [20]:

$$P(z_i = j | z_{-i}, w) \propto \frac{n^{wi}_{-i,j} + \beta}{n^{(*)}_{-i,j} + W\beta} \frac{n^{di}_{-i,j} + \alpha}{n^{(di)}_{-i,*} + W\alpha} \tag{2.11}$$

where $n^{wi}_{-i,j}$ represents the number of times the word w is assigned to topic j, excluding the current assignment. On the other hand, $n^{di}_{-i,j}$ counts the number of times a word from document d is drawn from the topic j. These values are for a single word topic, while the values for all of them are represented with $n^{(*)}_{-i,j}$.

Variational Inference reformulates the inference approximation as an optimization problem. Let X represent the observed variables and Z the latent variables. The goal is then to find a distribution $q(Z|V)$ where V represents a variational distribution, which is as close to the true posterior distribution $p(Z|X)$ as possible. The closeness (or difference) between two distributions is measured in terms of Kullback-Leibler Divergence(KL) :

$$\mathrm{KL}\left[p(Z|X) || q(Z|V)\right] = E_q \left[\log \frac{q(Z)}{p(Z|X)}\right] \tag{2.12}$$

Therefore, the objective is to find hidden variables so that the (KL) is minimized:

$$V* = \arg\min_{v} \mathrm{KL}\left[p(Z|X) || q(Z|V)\right] \tag{2.13}$$

Posterior Inference can also be approximated by maximizing a function called Evidence Lower Bound. Several algorithms for solving this optimization problem exist. In this thesis, we make use of the Online Learning algorithms proposed in by Hoffman et al. in [24]. According to the authors, this algorithm is as accurate as any other related algorithm, but several times faster. The source code of this algorithm has been freely published by its original author[1].

Evaluation

Similar to other unsupervised learning algorithms, the evaluation of topic models performance remains one of the challenges of using algorithms such as LDA. One of the most common methods used for this purpose is the Perplexity of the documents that are deliberately hidden (held-out) when running LDA. Perplexity is formally defined as [21]:

$$\mathrm{Perplexity}(\mathrm{D}_{\mathrm{held-out}}) = e^{-\frac{\sum_{d=1}^{M} \log(p(W_d))}{\sum_{d=1}^{M} N_d}} \tag{2.14}$$

where w_d represents words in document d, and N_d the length of the document d. In other words, perplexity computes the inverse likelihood of unobserved documents, and therefore better models have a lower value. Topic modeling algorithm allows users to label topics, and so need measures that account for the interpretability of the topics. For this reason, Chang et al. [25] proposed two methods, namely *word* and *topic intrusion*. These methods use human inputs for the task of finding which word/topic does not belong to a list of given words/topics. Word intrusion measures the level of agreement between the model and human for word-topic distribution. The respondents are presented with a list of n (e.g. 6) words from a random topic, where n-1 words have high probability in the respective topic, while the sixth word has low probability on that topic and a high probability in another topic. These words are mixed and presented to the respondents, and their high score on finding the word which does not belong indicate the power of the model. For example, suppose we have a list of following words {*trojan, virus, inject, attack, infect, and help*}. The easier it is to spot that the word "*help*" does not belong on this list, the more coherent and natural the topic is. Similarly, topic intrusion measures the level of agreements for topic-document distribution. The work principle is the same, with respondents asked to select the topic which does not belong to a given document from a list of potential topics.

[1] https://github.com/blei-lab/onlineldavb

2.2 Text Representation for Machine Learning

Most of the existing machine learning algorithms are able to build predictive models or infer intrinsic structural information only when the data is presented in the form of feature vectors. Consequently, (semi)automated methods or manual handcrafting of features is performed prior to the learning phase. The data we consider in this thesis is mainly in textual format, and distributional and distributed representations are the main approaches to represent textual documents as feature vectors. In this section, we summarize the main principles and differences between these two categories.

2.2.1 Distributional Representations

The distributional representations utilize information from a co-occurrence matrix to obtain semantic representations of words or documents [26, 27, 28]. This is a matrix *M* of dimensions $V \times C$ [26], where V represents the size of the vocabulary, and C the context. The rows of the matrix correspond to words of the vocabulary, columns represents the context, while the value of each cell represents some form of the frequency of word w_i in the given context. The context can be modeled with a window of size n (n=1,2,3,...) and can also represent the entire document.

$$M = \begin{array}{cc} & \begin{array}{ccccc} D_1 & D_2 & D_3 & & D_C \end{array} \\ \begin{array}{c} w_1 \\ w_2 \\ w_3 \\ \\ w_V \end{array} & \begin{pmatrix} f_{11} & f_{12} & f_{13} & \dots & f_{1C} \\ f_{21} & f_{21} & f_{23} & \dots & f_{2C} \\ f_{31} & f_{31} & f_{32} & \dots & f_{3C} \\ \vdots & & & & \vdots \\ f_{V1} & f_{V2} & f_{V3} & \dots & f_{VC} \end{pmatrix} \end{array}$$

One of the simplest representation models on this category is the so-called one-hot vector. This represents each word as a vector of size V (vocabulary size), the entries of which are set to 1 only on the location of the given word, and 0 otherwise. So, given that the word to be represented is the character "B" and the vocabulary consists of characters from English alphabet, then one-hot representation is a 26 dimensional vector equal to V["B"]=[0,1,0,0,...,0,0]. The documents are then represented using concatenation or some form of averaging of the word vectors.

The aforementioned methods suffer from high dimensionality and sparsity; each word in the vocabulary has an entry in the matrix making the representation very long when modeling documents with large number of words. Additionally, the distribution of words in documents is not uniform, resulting in a sparse matrix where a lot of elements are zero-valued. To tackle these issues, Deerwester et al. [29] proposed a method called *Latent Semantic Analysis* (can also be found as *Latent Semantic Indexing* in literature). This is a dimensionality reduction technique that works by applying Singular Value Decomposition to a co-occurrence matrix. That is, new features are generated by linear combination of the raw features resulting in a representation with lower dimensionality and less sparsity.

2.2.2 Distributed Representation

Distributed Representation (also known as Word Embeddings) represents models which map words onto real-valued vectors, whose dimensionality is low compared to the co-occurrence matrix, and the columns of which contain information about the semantic meaning of the word [26]. Contrary to distributional representations, word embeddings are complex models which include training through an ANN architecture.

One of the first attempts to generate such embeddings was proposed by Bengio et al. [30], who used a neural network with four layers (input, projection, hidden, output) for this purpose. It is important to emphasize the non-linearity introduced through a hidden layer (e.g. hyperbolic tangent), which actually increases the computation complexity of the model [31]. The model has served as the base for a simpler model proposed by Mikolov et al. [31], the details of which we present in the following.

word2vec embeddings

The continuous vector representation model proposed by Mikolov et al. [31, 32] consist of a simple feed-forward neural network with a standard input and output layer, and with a special hidden layer, which in the original paper is denoted as "projection layer". Two different architectures of this model are available, namely Continuous-Bag-of-Words (CBOW) and Skip-Gram. They are opposite to each other, but the working principles are the same, so we explain the CBOW model here, and emphasize the difference with the Skip-Gram model later.

The main principle behind CBOW is the learning of word representations via a predictive model which tries to predict the next word (target) given a sequence of words (context). The context words are encoded using one-hot vector representation and fed to the input layer, while the output layer represents that conditional probability of each word in the vocabulary to be the target word given the context. The advantage of word embeddings relies on the hidden layer, the activation function of which is a simple linear copy of one the input products. The weights between respective layers are trained using gradient descent, and the word representation is obtained from the matrix between input layer and hidden layer [33].

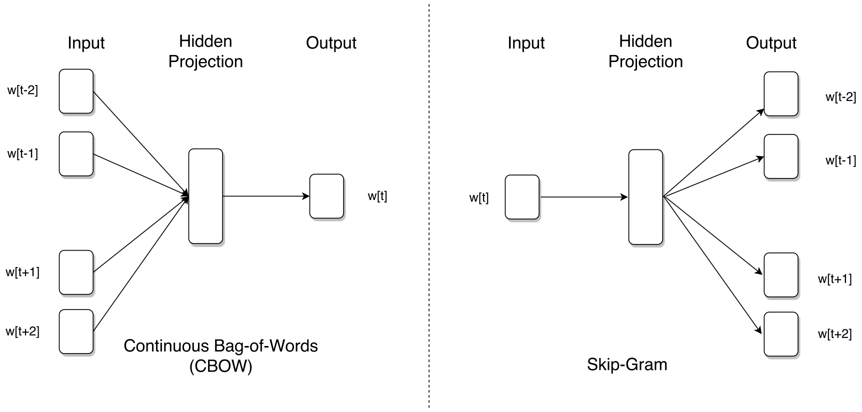


Figure 6: Distributed Representations: Left: CBOW, Right:Skip-Gram

In the following we provide more details on the mathematical expressions of the model, which are explained more in depth in [33]. For reasons of simplicity, let the size of the context be one word, V the size of vocabulary and D the size of hidden layer. The input layer is the one-hot encoding of all available words where only the context word is activated (set to 1). The input layer is fully connected to the hidden layer with a matrix of dimension V × D. Similar to all neural networks, the net input to the hidden layer is the summation of all products of the inputs and the corresponding weights :

$$u = W^T * x \tag{2.15}$$

As was already mentioned, the activation function of the "hidden" layer is a simple function that copies one of the input products to the next layer. This is due to one-hot encoding of the input where only one product value will be non-zero. Due to this simplicity, this layer is called the projection layer in the original paper. Formally, the output value of the hidden layer is just $h = u = W^T * x$. The hidden layer and the output layer are also fully connected with a weight matrix W' of size D × V. The net input to the output layer is computed using this expression:

$$u' = W'^T * h \tag{2.16}$$

The value of the output layer is the conditional probability of target word w_t given the context w_c, which can be computed using the softmax function:

$$y_t = P(w_t|w_c) = \frac{e_t^{u'}}{\sum_{j=1}^{V} e_j^{u'}} \tag{2.17}$$

The objective of this model is to maximize the log-likelihood of seeing the target word w_t given the context w_c. Thus, stochastic gradient descent and backpropagation are used to maximize the following funtion :

$$maxlog(P(w_t|w_c)) = u_t - \log(\sum_{j=1}^{V} e_j^{u'}) \tag{2.18}$$

The objective function in 2.18 is equivalent to the negative value of loss function (-E), and the problem can also be represented as a minimization problem.

In real applications, the size of the context is usually greater than 1, and may include words before and after the target word. For example, in the original paper [31] authors used a window size of 8 (4 words before and 4 after the target word). The difference from the model we explained so far is only in the way the input of hidden layer is calculated. When several words are considered as context, their corresponding vectors are aggregated to obtain that vector value:

$$u = \frac{1}{C} W^T * (x_1 + x_2 + ... + x_C) \tag{2.19}$$

where C is the size of the context window.

The second architecture, called the Skip-Gram model and shown in the right side of figure 6, is the opposite of CBOW model. Contrary to the first model, the input layer of Skip-Gram represents the target word, and its objective is to find its most likely context words. Formally, the objective function is to maximize the log-likelihood of the probability of context words given the target(current) word [32]:

$$\arg\max \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} \log P(w_{t+j}|w_t) \tag{2.20}$$

where w_t is the target word given as input, w_{t+j} are the context words to be predicted, T is the number of training words, and c the window size.

For datasets with large vocabularies, the calculation of softmax(e.g. hierarchical) probabilities can be very expensive due to the enumerated products of equation 2.17. An extension of the word embeddings was proposed by Mikolov et al. [32] to tackle this issue. Negative sampling can be used instead of softmax, and it only uses a sample (subset) of context words to update the weights matrix. Even though negative sampling is said to be an extension of softmax, the objective function to be optimized is completely different [34], resulting in significant speedup.

Another extension to the model which leads to better representation is a subsampling of frequent words (stop-words). In subsampling, the training words are discarded with a probability that in the original paper[32] is computed as follows:

$$P(w_i) = \sqrt{\frac{t}{f(w_i)}} \tag{2.21}$$

where t represent a threshold chosen by the user (e.g. 10^-5) while $f(w_i)$ is the frequency of word w_i.

Global Vectors for Word Representations : GloVe

According to Pennington et al. [35], models such as Skip-gram do not utilize the statistical properties of word co-occurrences. For this reason, they proposed a model they called as Global Vector (GloVe) representation model. The main idea behind this model is that the ratio between co-occurrence probabilities contain information which should be used when learning word vectors. In order to illustrate what the authors meant by that, we explain the example that was used in the original paper [35]. The example considers the conditional probability of a word $x \in \{solid, gas, water, random\}$ given the context word $y \in \{ice, gas\}$. Due to similarities or relatedness between words, we can anticipate that the probability of $P(x = water|ice)$ is large, whereas $P(X = water|gas)$ is small. The same conclusion can be inferred for $P(X = gas|steam)$ and $P(X = gas|ice)$. Thus, should we compute the ratio between these probabilities, the value will be either small or large if the word x is related to one of the context words. On the other hand, should x be related to both context words (e.g. x=water) or to none of them (e.g. x=random) the value of this ratio is approximately 1.

Table 1: The ratio of co-occurrence probabilities [35]

Probability and Ratio	**x=solid**	**x=gas**	**x=water**	**x=random**
$P(x\|ice)$	Large	Small	Large	Small
$P(x\|steam)$	Small	Large	Large	Small
$P(x\|ice)/P(x\|steam)$	Large	Small	~1	~1

Based on this logic, Pennington et al. [35] emphasized that the learning of word vectors should have the ratio between co-occurrence probabilities as a starting point, and not raw probabilities (like in skip-gram model). This is translated into a log-bilinear model, which satisfies the following equation:

$$w_i \cdot w_j = \log P(i|j) \tag{2.22}$$

where w_i, w_j are the vectors to be learned, while the value of their dot product is equal to the probability of word j being in the context of word i. With a simple difference of two such products, we see that the following equation is satisfied:

$$w_x \cdot w_a - w_x \cdot w_b = \log P(x|a) - \log P(x|b) = \log(\frac{P(x|a)}{P(x|b)}) \tag{2.23}$$

Note that the expression in the right side of the equation 2.23 is the same as the ratio of co-occurrence probabilities from table 1.

Formally, the objective function of such model expressed as weighted least squares regression model is as follows:

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \cdot (w_i^T \widetilde{w} + b_i + \widetilde{b_j} - logX_{ij})^2 \tag{2.24}$$

The parameters marked with a tilde in the equation 2.24 represent the vector and bias for context words while those without it represent the same values for target words. Note that the sum is weighted by a function $f(X_{ij})$, parametrized by the co-occurrence matrix, and whose role is to eliminate the undesired effect of stop-words. The authors used the following function in their paper:

$$f(x) = \begin{cases} (x/x_{max})^{\alpha} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases} \tag{2.25}$$

where x_{max} is a cutoff value (e.g $x_{max} = 100$), and alpha is a constant (e.g. $\alpha = 0.75$). For comparison of this model to the one proposed by Mikolov et al. [31] on a benchmark dataset, we suggest the reader to examine the original paper. However, since we use this model we show the results for our experiment in Chapter 6.

2.3 Deep Learning

Since machine learning algorithms are able to build predictive models only from the features extracted from raw data, their success is closely related to the quality of the features [36, 37, 38]. For a long time, these features have been manually crafted by experts, making this process highly manual and human dependent. Regardless, this has proven effective for solving some particular problems. Extracting salient features from complex data structures however is not straightforward. For example, imagine how difficult it is to extract a feature that recognizes the presence of the broken bus windows when detecting violent behavior from pictures. For this reason, a machine learning research field known as Representation Learning has gained the focus of researchers lately. The relation of representation learning to other forms of AI is shown in figure 7. Its main idea is to build algorithms which are able to automatically learn qualitative data representations that can be directly used from machine learning models. A good representation learner should be able to learn the representation of complex structures within a short period of time (up to several hours), and this representation should also be able to be used for different tasks. This not only saves time and human effort, but also enables the reuse of features for different tasks, which is not the case for usually task-specific manually engineered features.

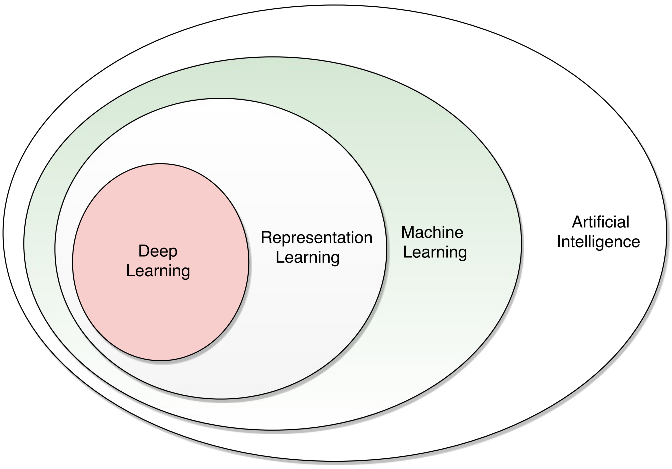


Figure 7: The relation of Deep Learning to Machine Learning [38]

Several approaches for learning features exist including manifold learning, autoencoders, probabilistic models, and deep neural networks [37] . In this thesis, we focus on deep neural network models, as we believe to be the dominant machine learning models in the future based on the attention they have gained recently.

Deep neural networks (or deep learning) have been around since the last century, but have gained the attention of the research community only recently. This is mainly due to the computation resources required to run deep learning algorithms [39]. Only recently, have we had access to affordable Graphical Processing Units (GPUs), the processing efficiency of which allows training deep models several times faster than their predecessors. Their advantage can be found mainly in the optimization of matrix calculations, which is the main computational burden of deep learning methods.

2.3.1 Convolutional Neural Networks

In general, deep learning models consist of several layers of neurons (thus the qualifier deep) where each layer processes the data from the previous layer and forwards to the subsequent layer. Higher levels of feature representations are learned at each layer. Different deep architectures and models exist in the literature, but an overview of all of them is beyond the scope of this thesis. We only discuss the details of Convolutional Neural Networks (ConvNNs), which we use to answer the first research question.

A typical architecture of a ConvNN consists of at least three different layers: convolutional, pooling, and a fully connected layer. Their number and order are specified by the designer of the model to best account for both the complexity of the problem and the need for algorithm efficiency. In the following, we explain each of these layers, and emphasize three main characteristics of a ConvNN: local receptive field, shared weights, and subsampling [40].

Convolutional Layer represents the core of a ConvNN, and is one of the architectural components which distinguishes it from traditional neural networks. In the latter, each two adjacent layers are fully connected to each other, meaning that there is a connection between each two neurons in these layers. With the increased number of hidden layers and neurons per layer, the calculation of the number of total weights gradually becomes computationally less and less feasible. To tackle this issue, ConvNN makes use of a *local receptive field* (LRF), a fixed-size subset of neurons. Thus, neurons of a layer are only connected to a subset of neurons from the previous layer. This dramatically reduces the number of parameters to be learned. The value of a neuron is obtained by computing the dot product between its weights and local receptive field. Similarly, the values of all neurons in a layer is computed by simply shifting the LRF by a fixed-size stride. This is equivalent to the convolution operator in mathematics:

$$f(x) * g(x) = \int_{i=1}^{n} f(\tau) g(x - \tau) d\tau \tag{2.26}$$

where f and g are functions representing weights and the value of LRF from previous layer respectively, whereas τ is the shifting stride. Since ConvNNs were initially designed for object recognition, convolution represents a manner to detect objects which are scaled, shifted, or distorted in variance [40]. Analogous to the feature extraction phase in the machine learning process, the role of the convolutional layer is to *detect and extract features directly from the input data*. Since the object in real data may not necessary positioned in a fixed area, there resides a need to take into account the distortions on the input. To achieve this, a ConvNN makes use of *shared weights.* That is, even though the value of LRF is shifted, the weights are reused. This not only reduces the number of weights to be learned, but also shares the same features learned in this case (shared weights), though positioned in different locations (LRF). The set of all values for a feature in a different location is called a **Feature Map**. Since more than one feature is desired, many feature maps(e.g. 100) are usually necessary.

Pooling Layer The convolutional layer is usually followed by a layer whose role is to reduce the dimensionality of the features extracted from the first layer. This is equivalent to the feature selection phase in the machine learning process. Even though in the literature this layer can be also found under the name of subsampling or downsampling, we use the term Pooling as it i the most often used in recent research. Not only does this reduce the size of features, it also makes the output less sensitive to shifts and distortions of input [40]. The common approach for pooling include Max-Pooling, Average-Pooling, etc. As it can be derived from the name, Max-Pooling returns the highest values within a feature map, while Average-Pooling computes the arithmetic mean of all the elements.

Fully Connected Layer is usually the last layer of a Deep Neural Network. This layer serves as a classifier (e.g. softmax), and its role is to find the most probable category for the given sample, represented with a feature vector learned through multi-layers as explained above.

3 Cyber Threat Intelligence

Advances in technology have not only enabled corporations to provide better services to their clients, but has also exposed them to new security risks. In order to properly handle these ever-evolving cyber threats, more proactive security controls are demanded. We believe that anticipation of these threats through analyzing data from multiple diverse sources will soon be an integrated part of cyber security of each organization. This assumption is based on the growing pace of a proactive approach called Cyber Threat Intelligence (CTI). Gartner defines CTI as :

> "Evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard [41]."

In other words, it identifies any indicator that can inform in advance about potential cyber threats, including their intent, resources, and methods [42]. In the cyber world, sometimes even raw data is considered intelligence, but in general a categorical separation should be made between raw data, preprocessed and organized information, and intelligence. According to Friedman and Bouchard [43], information can be deemed as intelligence only after it has been validated and prioritized, associated with specific threat actors, and customized for different consumers both inside and outside the organization. In other words, intelligence is information put into context. This can be achieved in several manners such as by adding risk scores to indicate the severity of the threat, annotation with tags such as "*Chinese malware targeting banks*", etc. This demands a process known as an intelligence cycle which involves planning, collection, analysis, evaluation and dissemination. The definition of intelligence in cyber space is in line with definition of military intelligence given by US Department of State [44]. According to this definition, there are two characteristics of intelligence: (i) it allows anticipation of future situations, and (ii) it informs decisions by illuminating the differences in available courses of action.

Though, a relatively young discipline, CTI has emerged as an important proactive approach in the cyber security community. In its latest report (2016), SANS [1] shows the results of surveying representatives from different industry companies. The results show a decrease in the number of respondents who do not have a CTI platform implemented to 6% from 15% as it was in 2015.

A variety of intelligence providers exist, both open source and proprietary, so it is therefore important to make a distinction between what is good intelligence and what is not. The quality of intelligence can be measured in terms of the following attributes: **relevance**, **timeliness**, **accuracy**, and **diversity of sources** [42, 43].

First of all, good intelligence should be relevant to the assets of a particular organization. Therefore, an identification of the assets which may be exposed to cyber threats should be done prior to the collection of information (intelligence) from different providers. Knowing that a hacker group is exploiting a vulnerability on Microsoft Internet Information Server is not relevant to an organization whose web servers are all Linux-based. The second attribute, has to do with the persistence of intelligence in responding to changes in the threat actor's Tools, Techniques,and Procedures (TTPs). Since threat actors are smart enough to change their TTPs to avoid detection, good intelligence remains valuable even after these changes. Next, threat intelligence should be accurate, which means that the number of false alerts should be kept as low as possible. Sometimes, having inaccurate intelligence may be worse than not having any at all, as it may push one's endeavors toward threat actors that do not pose a real threat to their organization, and therefore leave one vulnerable to real threats. Finally, threat intelligence should be collected from diverse sources. Different sources may have different levels of detail about a threat actor and their campaigns, and the corroboration of such information can provide a more complete picture of the potential threats, and more reliable intelligence at the same time.

3.1 Benefits of using Threat Intelligence

We have already mentioned that CTI is expected to be the next generation of proactive security; we now briefly summarize some of the benefits of having a CTI platform established.

Security Planning. Threat Intelligence aids in the identification of threat actors that pose a risk to an organization. First of all, this positively affects the communication between high level executives and the ones reporting to them. Having sufficient information about particular threats enables security staff to communicate known risks and their impact in an easier manner. Furthermore, this helps decision makers plan the necessary security defense by allocating appropriate resources in the budget (e.g. tools, people) to the relevant risks.

Detection and Prevention. Threat intelligence data in the form of Indicators of Compromise(IOC) could be easily translated into alerts of malicious activity (detection), network/host signatures, firewall rules (prevention), etc. This proactive update of security controls increases the detection and prevention rate of malicious activities. Additionally, these indicators can be used by investigators to rank the alerts from IDS so that the most severe ones are considered first.

Incident Response. Threat Intelligence can also serve as a guide to incident responders, as it may lead the investigators to the next place to look for evidence of a particular incident. For example, an incident responder may discover an indicator during investigation that enables attribution to a particular threat. With this information, it is possible to have a look at all the information available for that threat, and focus on the TTPs that they use.

Mitigation and Remediation. Threat intelligence also helps to identify and prioritize services that need to be patched in order to prevent future attacks.

3.2 Threat Intelligence Subtypes

Different security companies are utilizing different classification of threat intelligence. One detailed approach proposed by MWR Security [45] categorizes threat intelligence into the following subtypes: Strategic, Operational, Tactical, and Technical. On the other hand, companies such as ThreatConnect [42] and iSIGHT Partners [43] make use of a simple classification schema with only two subtypes:Strategic and Operational. Even though there is some level of agreement between these approaches, we follow the approach which categorizes threat intelligence into: high-level intelligence (Strategic) and low-level intelligence (Operational). The reason for this decision has to do with the challenges in placing boundaries between different subtypes when having multiple such subtypes. This classification is mainly based on the level of technical details provided by the information and the primary consumers. In the following we summarize the characteristics of these two subtypes.

Strategic Intelligence is information dedicated mainly to the highest level of decision makers within an organization (e.g. Board, C-level managers). As the audience mostly does not have a technical background, this type of intelligence is focused in the risk and impact of different cyber threats, and usually does not contain technical details about the threat. The aim is to inform decision makers about the trends of cyber threats, the likelihood of their occurrence, and the impact such threats have with respect to an organization's finances, reputation and business continuity [42]. Typical examples of strategic intelligence include information about a threat's motives (e.g. political, economical, fun), past campaigns, and targeted industries. Having the information that a certain threat group is stealing personal data from banks with the purpose of financial gain should raise an alarm to all related companies to take all the necessary measures to be prepared to defeat such threats. Since the primary consumers of strategic threat intelligence are those who plan the budget and are responsible for approving investments on certain security controls, it is vital that the intelligence is collected from reliable sources. Most of these sources are high-level as well and typical examples include geopolitical assessments, white papers from the industry, and human contacts [45], among others. Open sources such as blogs, news, social networks etc. may also contain relevant information as often financial impact and trends of cyber threats are made publicly available.

As the motives and intent of threats changes less frequently as opposed to its TTPs, strategic intelligence usually has a longer lifetime. Therefore, its final product does not have to be delivered in daily or weekly basis. Not only should the content of the reports not be technical, but less details are needed and shorter reports with couple of lines are sufficient to help the strategist understand the impact of certain cyber threats, and assist in decision making.

Operational Intelligence is produced and consumed by staff that deals with information security on a the daily basis. Our definition of operational intelligence is however different from the one given by Chismon and Ruks from MWR Security [45], who define it as information about the nature of an incoming attack, and the identity and capabilities of the attacker. As was also stated by the authors, this kind of information is difficult to collect for non-government organizations such as private security firms, as its collection often demands legal permission. In our definition, operational intelligence includes very technical information, and not only does it cover the operational intelligence as defined by [45], but also two other types: Technical and Tactical threat intelligence. Detailed and specific information is vital when detailing how potential threat actors can attack the organization. For example, it is crucial to know the vulnerabilities that different threat actors are exploiting, the methods and tools they use in each phase of the cyber Kill Chain, the communication manners between different threat actors, etc. Operational intelligence can be from a simple IP address, hash value (e.g. MD5) of a malicious file, domain name or URL, to more complex indicators extracted from thorough investigations.

Operational threat intelligence can be collected from various sources and following is a list of some of them:

- Malware analysis reports and feeds
- Incident reports
- Vendors reports
- Open sources (e.g. social media, news, blogs)

The time period when operational intelligence is expected to be relevant and timely is short compared to that of strategic threat intelligence. This is due to the fact that threat actors often change their TTPs to avoid detection. This depends on the threat actor of course, as some tend to change methods more often, such as for each campaign, or each target.

A summary of differences between strategic and operational intelligence is given in table 3.2.

Table 2: Strategic vs. Operational threat intelligence

	Strategic	**Operational**
Level of Information	High-Level	Low-Level
Consumers	The Board and other decision makers	Security staff (SOC, IR)
Expected lifespan	Long	Short
Main Focus	Risk and the impact to the business	Daily detection and prevention of threats
Collection Sources	Carefully selected sources	Various sources: open and proprietary

4 Related Work

In this chapter, we present a literature review on hacker community research. The literature is obtained from scientific research databases using combinations of keywords such as "*hacker community*", "*Dark Web*", "*Threat Intelligence*", "*hacker forum*" ,etc. and following the references from the obtained results.

The existing research on hacker communities can be grouped into four streams: (i) social structure of hacker platforms, (ii) identification of key hackers, (iii) understanding of hacker language, and (iv) identification of emerging threats. The work done in this thesis belongs to the last stream, and therefore more details are provided on the methods from this group.

4.1 Social Structure

There exists a perception that so-called hackers have profound skills with computers in general, and cyber security in particular (for those who have seen Mr. Robot series, this may seem familiar). Beyond the phantasies in TV shows however, several studies on these communities have shown that most of their members have little to no knowledge about cyber security [46, 47]. Holt et al. [46] categorized hacker forum members in several groups: according to this categorization, the majority of forum members represent students passionate about technology, but lacking the necessary background knowledge to successfully perform cyber attacks. Thus, fortunately, most of the members of these platforms are there to learn and advance their knowledge of certain technologies. The second category, which are significantly less in number than the first group, represent individuals with the sufficient skill to understand and use the information shared in these platforms. The most important group, the so-called "*elite*"constitute a third category who not only can understand and use the existing tools and techniques, but are also able to create new methods that can be exchanged with others. This group represents the highly skilled hackers and their percentage on such forums is usually limited to single-digit numbers.

The technical skills of hackers in their communities are reflected through two metrics: group (level) and reputation [46, 47, 48]. Mostly, hacker communities use a hierarchical structure where users are part of different tiers based on their involvement in the platform. The number of levels and rules assigned to each of them depends on the platform itself, but there are basic groups such as newbies (recently joined), normal users, administrators, and banned users that are part of the majority of platforms [46, 47, 48]. On the other hand, reputation is usually measured by other users' feedback.

The respect for reputation measures more than just the skills of the members, it reflects its use as a method to establish trust. It functions as a kind of assurance that information shared in these platforms does not infect the members themselves. Allodi et al. [48] presented the reasons why a stolen cards marketplace such as Carder.de has failed by comparing it with another well known marketplace whose identity was purposely not shown. Among them, the lack of proper reputation scores and enforcement of rules were identified as main contributors. According to authors, this is one of the reasons why hackers have moved from chat based towards more structured platforms such as forums and marketplaces.

The social network analysis has been used to understand the social structure of these platforms [47]. In general, the social structure of these platforms resembles other social networks such as Facebook and online marketplaces such as Ebay. However, the difference is in the intent of the members and type of information being exchanged. Contrary to platforms such as Facebook and Ebay, the intention of hacker platform members are usually malicious and the content of the post may contain information which not only is malicious but can also be unlawful. Similar to social networks such as Facebook, most hacker social networks support the creation of profiles, adding friends, posting and replying to public posts, and conversation through private messages. The difference is that a user's number of friends is typically rather small, as the the goal is not socialization [47]. The results from [47] show that less than 10% of all members have issued a "Friend Request". This is supported by the rarity of cases where hackers know each other personally, or have physically met in person. Another property of these platforms is the banning of users who do not act according to the rules. The reasons for banning may be different for different platforms, but duplicate accounts, spamming, and malicious activity [47, 48] are some of the most common.

These platforms tend to have a considerable number of members, and some of them are members of more than one platform [46, 47]. These users are identified by either analyzing the social network of the platform, or by simply checking the number of overlapping users (usernames, emails). However, not all the members of these communities are active and only a small number of users account for a large amount of activity. This has been shown by Motoyama et al. [47]. who found that around half of the trades in the marketplaces they considered were covered by top 10% of the members.

4.2 Identifying Key Hackers

Since only a small percentage of hacker community members are highly skilled, computational approaches such as social network analysis [49], ranking score [50, 51, 52, 53], and topic models [54] have been used to identify them in the existing literature.

One approach is the use of social network analysis and the calculation of metrics such as centrality. For example, Samtani and Chen [49] proposed a method for identification of key hackers for keylogger tools based on Bipartite social networks. The forum used in this study was modeled as a bipartite graph where members were modeled as type 1 node, while threads related to logger tools as type 2 nodes. Results of this method once again supported the claim that hackers usually interact with only a small number of other hackers.

Ranking of hackers is also performed by computing a reputation score as a combination of different fields. Huang and Chen [53] used topic-based social networks and clustering to extract fields such as number of posts, number of feedbacks, posts in last month, number of forums engaged, post in last session , and the year of first post. In an earlier work, Benjamin and Chen [50] used the weighted sum of the following parameters: average message length, number of replies, number of threads involved, tenure, sum of attachments, and total messages for this purpose. The results have shown that features related to the quality of discussion such as the average length of message does not play a significant role when ranking hackers based on the reputation. Put in other words, length of the message alone is not a good feature to assess the quality of the posts.

A similar approach was followed by Abbasi et al. [51], who applied Interaction Coherence and Content Feature Extraction analysis to extract four types of features for each hacker : topology, cybercriminal assets, specialty lexicon, and forum involvement. Hackers were clustered into four categories and the results show that around 86% of them were average users with low involvement, and only 12% of them were considered as technical enthusiast who often embed code and use technical language in their posts. This supports the assumption that only a small percentage of users are senior members.

Hackers have also been ranked based on the quality of the feedback on their posts. In a method proposed by Li and Chen [52] a deep Recursive Neural Network was used to calculate the sentiment score of the feedback. The greater the score, the higher the ranking of the hacker. Prior to applying deep learning, Maximum Entropy was applied to classify threads into different groups, however no information is given in the groups, and the features used are unclear. The results show that deep learning outperformed shallow classifiers in the task of sentiment analysis.

Fang et al. [54] used Author Topic Models to identify key hackers. After extraction of topics from the data, the weights of the topic distribution of the authors per each topic were calculated and used to rank the hackers, and extract top N for each topic. In an another study, Grisham et al. [55] claimed to have used topic modeling in the identification of key listers in a marketplace. Strictly speaking however, this does not stand up to close scrutiny. In their method, they only use LDA to learn what the listers are selling, while the ranking of users was done simply by counting the number of posts. In this experiment, top 3 listers were identified and topic models showed that only one of them was selling malware related products. However, the manner of how listers are ranked can be criticized. Having more posts does not automatically make a marketplace member more interesting to analyze, as quantity does not necessary mean that the products offered are relevant. In the same manner, only three listers were considered, which omits much of the the various products sold in the marketplace.

4.3 Understanding Hackers' Language

One of the challenges researchers face when studying hacker communities is in the terminology hackers use. Often, hackers make use of tools and language terms that may be unfamiliar to the researcher. Being able to understand such language is crucial to discovering relevant intelligence, so researchers need computational methods that are able to express hackers' jargon in a more accessible language. The existing research mainly focuses on the use of Lexical Semantic analysis, which aims to find the similarity between different lexical units such as words or sentences. Benjamin et al. [3] tested different variations of word embedding (CBOW,Skip-gram, Negative sampling, hierarchical softmax) in the task of understanding words similar to a given list of commonly used terms by hackers. More specifically, the ten words most related to security terms such as the following are computed: rat, card, logger, crypter, rootkit, salt, binder, dork, and vulnerability. For illustration, words "*adwind*","*xanity*", and "*spygate*" were returned as the most similar to the word RAT (Remote Administration/Access Tools). These are all types of RATs and researchers may not necessarily be familiar with this fact. Despite all the challenges that Chinese language poses to computational methods, Zhao et al. [56] showed that understanding of hackers' jargon through word similarities works for Chinese data as well.

Using a slightly different method, Benjamin and Chen [57] used the similarities in document level to group hackers' data in two language groups: English and Russian. An extended version of word embedding (chapter 5) called Paragraph Vectors was used to represent data with fixed size vectors. According to Benjamin and Chen [57] paragraph vectors outperform other methods such as n-grams for this task and suggest its use on other natural language processing (NLP) and ML tasks, either in combination with other handcrafted features or just as stand-alone features.

4.4 Understanding hacker's assets and emerging threats

The identification of the trends in cyber security from hacker communities demands the analysis of the contents of such platforms. The most common content analysis methods used in the related literature are comprised of classification and topic modeling using machine learning, information retrieval, and natural language processing.

Classification with Machine Learning

Closely related to the work done in this thesis, in particular to the first research question, is the model proposed by Nunes et al. [5]. This method seeks to assist security experts in the task of threat analysis by using binary classification to categorize the hacker forums posts and marketplace products into two classes: (a) relevant and (b) irrelevant to cyber security. Due to lack of labeled datasets, authors apply semi-supervised learning algorithms such as Label Propagation and Co-Training in conjunction with traditional supervised algorithms (Naive Bayes, SVM, Random Forests, etc.) The authors manually labeled 25% of the data, consisting of 1840 posts from forums. This number is significantly smaller than the number of posts labeled in this thesis at 16,000. Character n-grams were used to generate feature vectors, which are constructed by concatenation of the respective vectors from the title and the contents of the post/product. After reviewing some of the posts from our dataset, we believe that the title of the thread may lead to more false positives, and therefore use only the contents of the post to generate features. Besides the number of posts/products, no other information is given on the identify of the forums/marketplaces, making it impossible to make a comparison of the results. Additionally, the authors claim to have discovered 16 zero-day exploits from the marketplace data, but without giving any further information on the analysis of the data classified as relevant nor what tools or services were consulted to confirm that exploits were indeed zero-days.

On the other hand, Marin et al. [58] used unsupervised classification to explore products offered in different marketplaces in Dark Web. Similar to the work of Nunes et al. [5], no information is given on the identity of these marketplaces. Unsupervised k-Means clustering was used to categorize products into 34 different categories. Even though several fields from each product post were extracted (title, description, price, rating, posting data, vendor, and marketplace), only the product title was used to generate the features. More concretely, word and character level n-grams of product titles were used to generate feature vectors, and Term Frequency-Inverse Document Frequency (TF-IDF) was used as the value for each feature. From all the available data (17 marketplaces) only 500 samples(around 5% of the total data) were manually labeled and used for training and evaluation. The method was evaluated using 100 samples not shown to the algorithm during training, and cluster entropy and rand-index were used as measures of the quality of the results. Different combinations of parameters were tested, and the configuration of character n-grams in range from 3 to 6, cosine similarity as a distance metric, and fixed centroids showed the best results compared to other combinations. Additionally, the cluster entropy was used to find the dominant vendors and marketplaces for each category. A lower value indicates a more frequent presence of a vendor in that cluster.

We argue against the choice of the authors to use only the title of the product to construct the feature vectors. Usually, the title is rather short and the number of n-gram features that can be constructed may not be sufficient to successfully classify the posts. Moreover, should a vendor use a template to describe different products, then these products will be grouped in the same cluster regardless of their contents. This limitation is also recognized by the authors [58], as half of the products in a cluster denoted as Hacking Tools were advertised from the same vendor, which is unusual for these kind of platforms.

Topic Modeling

Our second research question is related to topic modeling, which is one of the most common methods used to explore large collections of documents, especially when there is a lack of labeled documents. One such approach was followed by Samtani et al. in [2], who applied topic modeling in hacker platforms. Prior to that, the authors categorized the contents of data into three groups: source code, attachments, and tutorials. Due to differences between groups, they were stored into separate relational databases and preprocessed in different manners. Actually, the preprocessing was identical for tutorials and attachments, and included methods such as stemming and stop-word removal. On the other hand, since source code contains terms not part of natural languages, additional preprocessing was performed for data belonging to this group. For example, all the identifiers were split into meaningful sub-words. That is, should the code contain a variable called "*sql_connection_string*", then, after preprocessing, three words are generated: "*sql*", "*connection*", and "*string*". In addition, aiming for a better organization of posts containing source code, a supervised classifier was used to classify code into one of the 10 programming languages (Java, Python, SQL etc.). Besides the fact that SVMs was used as a classifier, no other details have been revealed on the features used for classification. The experiments for this study were performed on five different hacker forums: *hackfive, hackhound, icode, Exploit.in, zloy,* and the results show some interesting findings. First, tutorials were considered the most common method used for information exchange between hackers. On the other hand, most of attachments and source code were considered general purpose; an exception were around 10-20% of attachments' topics and a source code topic related to banking vulnerabilities.

After reviewing 11 existing cyber threat intelligence portals and 14 malware analysis portals, Samanti et al.[4] found that none of them were collecting data from hacker communities. Motivated by that, they extended their work in [2] by developing a portal that provides a graphical interface which can be used to search, browse, download and visualize hackers' assets. This time they used only two hacker forums (English and Russian) suggested by security experts and known to contain malicious assets. Surprisingly, assets were split into only two groups, source code and attachments, contradicting the results of their previous work identifying tutorials as the main means of exchanging information on these communities. Topic models were applied to extract the topics, which were labeled manually and validated by six graduate cyber security students. The exploits from the discovered topics were categorized in one of the categories: Web, System, or Network. The results showed system exploits such as RATs, Crypters, Keyloggers, Dynamic Linked Libraries injections, and shell code as the most common hacker assets. Source code was also classified into programming languages using a supervised classifier, and the portal gives the opportunity to calculate the similarity between different posts by calculating the cosine similarity.

Topic modeling has also been used to analyze cyber threats and key hackers in Chinese communities. Fang et al. [54] explored the most common topics in Chinese forums, their evolution, and key hackers by using LDA and its variants. Special attention was paid to the number of topics as a LDA parameter, since too small value could lead to sub-optimal solution while too many topics would cause overfitting. After some empirical tests, this number was set to 10 topics. In addition to extracting topics, Dynamic Topic Models , an extension of traditional topic models (LDA) was used to track evaluation of the topics, splitting data into different time spans, and monitoring changes in the topic keywords weights. Topic keywords whose weights increased over time were considered trending threats. Without revealing any details on how the topics were labeled or selected, five out of ten topics were selected as more popular: *trading, fraud identification and prevention, contact for cooperation, causal chat, interception* and *monetizing*. These topics covered around 60% of all discussions. Regarding topic evolution, the weights of keywords such "CVV" and "101", used in stolen credit card jargon were reported as increasing over time, indicating their presence as an emerging trend.

Information Retrieval and Natural Language Processing

The data from hacker platforms can also be analyzed from an information retrieval perspective, as proposed by Banjamin et al. in [59]. According to this model, a list of known security keywords constructed by experts or extracted from the data acts as a query to an information retrieval algorithm, and posts are ranked based on the product of hacker reputation and the weights of the keywords. More precisely, TF-IDF value of the predefined keywords was used as the weight for the keywords. This method reflects the assumption that the severity of the cyber threats depends on the skills of the hackers, and that highly skilled hackers are those responsible for creating and disseminating the more severe attacks. The reputation of a hacker was obtained through another framework proposed by the same authors [52] which was explained in the previous section. The higher the value of this product, the more likely that the post contains relevant information.

This study is one of few that considers data also from Internet Relay Chats (IRC) and Carding shops. According to the authors, IRC chats pose extra challenges to researchers; first, they are usually not accessible through search engines, and require special client programs to get access to the data. Furthermore, messages exchanged are likely not stored for a later access, so data should be collected in real time. Carding shops are also relevant to threat intelligence since they contain information about stolen credit cards which, if when identified on time, can prevent significant financial loss. The experiments performed on these platforms whose identity was deliberately shown only partly revealed cyber threats which can be of a great interest for security. For illustration, we present two examples from forum data; the first one shows a tutorial on how to bypass a security feature on Bank of America accounts, while the other is an advertisement to rent a server which hosts several point-of-sale software which can be used to further scan for their vulnerabilities. Cyber threats were identified in IRC data as well; examples include a campaign against the French Ministry of Defense and a release of personal data stolen from financial institutions. Similarly, thousands of stolen cards were reported as being available in the carding shops.

In an another method, Macdonald et al. [60] used a combination of Part-of-Speech(POS) tagger and sentiment analysis to identify cyber threats against critical infrastructure. The POS tagger was used to annotate data with tags such as noun, adjective, verb, etc. After manual revisions, only 16 nouns were considered. These remaining nouns were used to assign sentiment scores to the data. More concretely, posts containing keyword pairs (critical infrastructure, hacking tool/jargon) were assigned a sentiment score, while the absence of such pairs was scored with a sentiment 0 and discarded from further analysis. As the list of nouns (keywords) was short, only 2.7% of all posts were assigned a sentiment score and thus considered further. From the manual analysis, authors were able to conclude that high sentiment posts related to banks contained the most relevant threats. More precisely, in 82 of the posts which contained the keyword pair (bank, botnet) were advertisements to sell malware such as Zeus or Spyeye. In addition to the limitations identified by the authors, we also see problems with the keywords used to retrieve the data. The number of keywords was small (5 for hacking tools, 6 for hacker jargon, and 5 for critical infrastructure), and only contained general words such as botnet, inject, malware, virus, exploit, breach, vulnerability etc. The method could decrease the likelihood of missing important posts should it consider a richer list of keywords.

A summary of the aforementioned methods is given in table 3.

Table 3: A summary of content analysis research on hacker community

Source	**Methodology**	**Data Source**	**Main findings/ contributions**
Nunes et al. [5]	Supervised and Semi-supervised learning	Marketplaces(10) and Forums(2)	Binary classification of hacker posts. 16 zero-day exploits discovered over 4 weeks.
Marin et al. [58]	Unsupervised learning	Marketplaces (17)	The products were grouped into 34 different clusters labeled manually.

Samtani et al. [2]	Topic Modeling (LDA)	Forums:
		English(3)
		Russian(2)
			Tutorials the primary methods of resource sharing;
			Only 10-20% of attachments useful;
			A source code topic related to banking vulnerabilities.
			Common topics: bots, crypters, keyloggers, password extractors, web and browser exploits, SQL injection.

Samanti et al.[4]	Topic Modeling (LDA)	Forums:
		English(1)
		Russian(1)
			An interactive graphic interface portal which allows users to search, browse, download, and visualize hacker assets.
			The majority of assets target systems.

Fang et al. [54]	Topic Modeling(LDA), Dynamic-Topic Modeling(DTM), Authorship-Topic Modeling(ATM)	Forums: Chinese(19)	The most popular topics: trading, fraud, contact for cooperation, causal chat, and monetization. Keywords changing over time: "WeChat", "Pinguing", "CVV","101"
Benjamin et al. in [59]	Information retrieval	Forums (10) IRC chats (8) Carding shops (4)	Highly relevant threats were discovered in each platform: examples include stolen personal and credit card data, tutorials how to bypass security controls, campaigns against institutions such as French Ministry of Defense etc.
Macdonald et al. [60]	POS tagger and Sentiment analysis	Forums	Most of the post containing (bank, botnet) keywords pertained to selling of malware.

5 Methodology

The method we use to get the answers to our research questions consists of two automated methods used to extract the relevant data, followed by optional manual or (semi)automated methods to validate the results and get more details from the extracted data. An overview of this methodology is shown in figure 8.

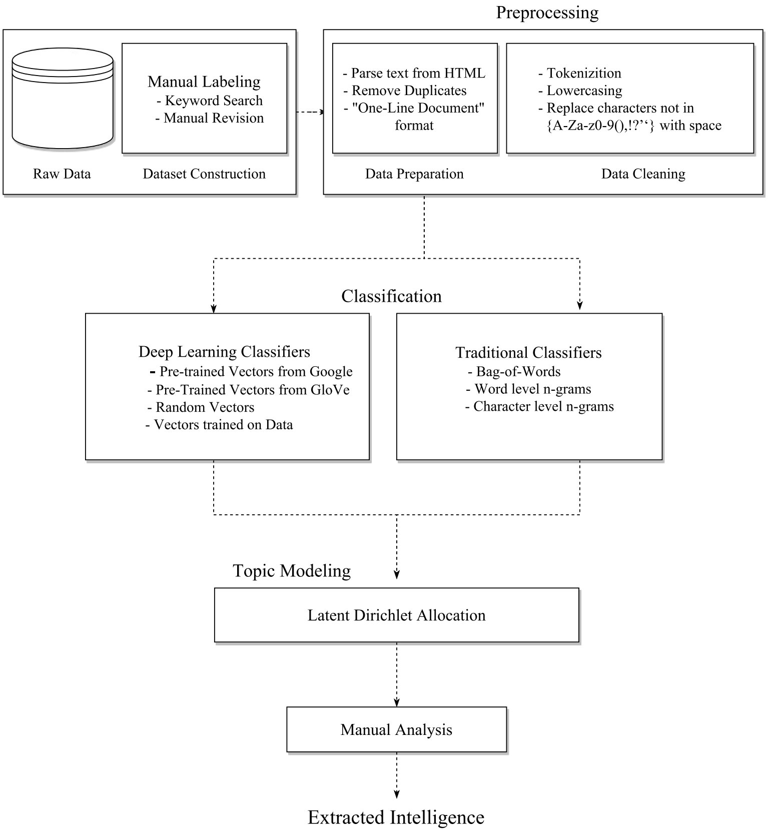


Figure 8: Methodology

As was discussed in the previous chapters, the hacker forums contain enormous number of posts, not necessarily related to security. To tackle this big data issue, we first filter out irrelevant posts by using supervised machine learning algorithms. Since these algorithms demand annotated posts to build a classification model, we constructed a binary and a multi-class dataset by manually labeling the posts. Secondly, unsupervised topic modeling algorithms are applied to discover the main themes of the posts related to security. While classification reduces the search space of the relevant posts, topic modeling provides a summary (in the form of topics) of their contents. Applying classification before topic modeling besides the efficiency of the algorithms also impacts the quality of the topics. Finally, we use an optional phase that we call manual analysis, to validate the results and get more insights from the posts. In the remainder of this chapter, we explain these methods in more details.

5.1 Dataset Construction

The first phase of analyzing open sources for threat intelligence is the identification and collection of the data. The identification of the data sources in the literature is performed mainly through keyword searches or by seeking experts' opinions [52, 3, 2]. Additional sources are further identified by using snowball sampling in the already identified data sources. Snowball sampling is a technique used to increase the number of samples (like a snowball) from the existing samples and it is used in situations where "recruiting" of samples is difficult. On the other hand, data collection demands special web crawlers tailored to each source. We must highlight that the collection from hacker forums is challenging; this is due to all the counter-measures enforced by the administrators of such platforms, including: CAPTCHAs, blacklists, registration, invitation only access, etc. Unfortunately, the number of available forums that have already been the subject of scientific research is limited. To the best of our knowledge, the only available English forum used in the existing literature is Hackhound[1]. However, due to the relatively small number of posts (4,242), we considered other sources. With the help of a security expert (PhD candidate at NTNU), we identified a forum which has been leaked and is publicly available. The data of this forum already exists in a mysql format and no special web crawlers are needed to collect it. The details of this forum called *Nulled.IO* are discussed in the following chapter.

In order to build a model which is able to filter out posts which are not relevant to security, the presence of labeled samples is required. Since to the best of our knowledge there is no such dataset available on the internet, we constructed a dataset by manually labeling some of the posts in the identified forum. The presence of common security keywords validated through skimming the text allowed us to categorize a document in the class *relevant*. This list of these keywords depicted

[1] http://www.azsecure-data.org/dark-web-forums.html

in table 4 represent general terms in cyber security and was constructed based on the author's experience and a review of related work.

Table 4: The list of keywords used to label the class Relevant

Common Cyber Security Keywords
adware, antivirus (kaspersky, avast, avira, etc.), backdoor, botnet, chargeware, crack, crimeware, crypter, cve, cyberweapon, ddos, downloader, dropper , exploit, firewall , hijack , infect, keylogger, logic bomb, malware, monitizer , password , payload, ransomware, reverse shell, riskware , rootkit, scanner , security, shell code, spam, spoof, spyware, stealware, trojan, virus, vulnerability, worm, zero-day, zeus

The obvious choice would be to categorize all the remaining data in the *irrelevant* class. However, since the list of common security keywords is not comprehensive, the same approach was followed to label the other class as well. Thus, data was labeled as irrelevant should it satisfy the following two conditions: (a) none of the security keywords from table 4 was present in the text and (b) non-security related keywords were present in the text. The list of general keywords consist of terms related to sport (football, basketball, etc.), music (song, rap, pop, etc.), movies (series, film, episode, etc.), drugs (marijuana, heroin, etc.), etc.

Additionally, we extended the binary dataset to a multi-class dataset by considering the irrelevant class as just one of the categories. Several other categories were inferred from security-related data. We must emphasize that this kind of labeling is much more demanding. First, the number and type of the categories must be decided. In our experiments, we chose categories based on two criteria: (a) coverage of different aspects of information security, and (b) relevance to practical scenarios. Additionally, we were also constrained by the type of data present in the forum we used. Similarly, though labeling was performed based on a keyword search, additional posts have been carefully reviewed to ensure comprehensiveness. For example, a post may have contained keywords such as *username, passwords, login, email, etc.* and after reviewing it was labeled as class credential leakage. The categories that we use in this thesis and illustrative examples for each of them are described in the next chapter.

The labeling was validated by 5 fellow students, who were asked to manually label 50 randomly selected posts. The labeling of respondents was consistent with our labeling in approximately 90% of the posts. The differences were mainly in the posts we classified as spam, which were often assigned to "Not related to security" class by the respondents. We could have obtained better results should we told the respondents more details about our strategy of labeling. However, deliberately the only information given to the respondents was the description of the task.

5.2 Preprocessing

The preprocessing consists of two sub-phases when analyzing hacker forums (figure 8): data preparation and cleaning. The objective of data preparation is the extraction of only relevant fields from the raw data and their storage in a adequate format (e.g. relational database). This is especially needed when data is collected using web crawlers which store the data in a raw HTML format, and special text parsers are used to extract fields such as the content of the posts, title, author, date, etc. The forum used in this thesis was already stored in a relational database, so no parser was needed to extract certain fields from raw text. However, the contents of the posts was stored together with the HTML tags so their removal was performed prior to any other preprocessing methods. We only selected the title and contents of each post for further analysis. After removal of duplicates, the posts are stored in a "one-line per document" and passed to subsequent methods.

On the other hand, the aim of data cleaning is to remove parts of the data which do not contribute to the goal (e.g. act as noise), and have a negative effect in the performance of given algorithms. Since we compare different classification models, the cleaning method used in this thesis was taken by Kim[2], the author of one of the methods used for comparison. A summary of these methods is shown in table 5.

Preprocessing method
Document Tokenization
Remove all leading and trailing white-space characters from the document
Lowercase text
Replace characters not in {A-Za-z0-9(),!?”‘} with space
Replace two or more consecutive white-space characters with a single white-space

Table 5: Data cleaning methods

5.3 Classification Methods

Supervised machine learning algorithms are used to classify the contents of forum posts, aiming to filter out posts which are not relevant to cyber security. In order to achieve this, the contents of each post is considered as a document. While the combination of posts title and content is also a possibility, we decided to use only the contents. This is mainly due the large number of posts with the same title (topic) which could bias the results. The first original scientific contribution of

[2] https://github.com/yoonkim/CNN_sentence/blob/master/process_data.py

this thesis is an empirical comparison of the classification performance of different machine learning algorithms on data from hacker forums. More concretely, the recent deep learning algorithms are compared to conventional methods such as SVM with different feature generation methods. Based on the success deep learning has shown in other application domains, we build off the following hypothesis regarding the performance of classifiers:

H1: Deep Learning methods will outperform traditional classifiers on the task of classifying data from hacker forums.

The following is a brief overview of supervised classification algorithms used to answer the first research question and support or refute our hypothesis.

5.3.1 Traditional Classifiers with Bag-of-Words features

One of the conventional methods for generating text features is through the so-called Bag-of-Words. This belongs to the group of distributional representations (chapter 2), and features are some sort of word frequency within each document of a corpora. That is, the dimensionality of the feature vectors is equal to the number of unique words in all documents in the dataset. The value of the features can be one of the three following frequency counts. *Binary:* should a word be present in a document, then the value is 1, otherwise 0; *Raw Frequency:* the frequency of each word within the document; *Normalized Frequency (TF-IDF):* the frequency of each word in a given document is normalized by the sum of occurrences of that word in all documents.

For illustration, let the dataset consist of three documents: S1="*The attacker exploited a vulnerability on the server*", S2="*The server was quickly compromised*", and S3="*It was a Trojan virus*". It total, there are 13 unique words, which represents the size of the feature vectors. Should the value of features be the raw frequency of the words then the feature vectors for the three documents will be: $S_1 = [1, 1, 0, 1, 0, 1, 0, 1, 2, 0, 0, 1, 0]$, $S_2 = [0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1]$, and $S_3 = [1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1]$, where $S_n = [v_1, v_2, ...v_{13}]$ and words are alphabetically sorted. For binary values, all non-zero features will be changed to 1, whereas these values are divided by their corresponding inverse document frequencies for TF-IDF features.

It is important to note that Bag-of-Words ignores the word order [61, 62]. That is, the representation of words that contain the same words but in different order will be the same. However, due to simplicity and often good performance Bag-of-Words remains one of the most common methods for text classification.

5.3.2 Traditional Classifiers with n-gram features

The word order is considered in a local context by n-gram features, which are defined as a sequences of *n* consecutive words/characters. In this thesis, we consider both word and character n-grams. For character level n-grams, features are constructed by enumerating all the possible characters of length *n* from a text document. For example, 2-gram and 3-gram characters of the word *virus* are the following :

*bigrams(n=2) : vi, ir, ru, us, s_*

trigrams(n=3): vir, iru, rus, us_, s__ ;

where the underline character represent the white space. Similarly, word level n-grams are sequences of n-words. In the following we show an example of word bigrams and trigrams for the sentence "I prefer python":

bi-grams(n=2) : "I prefer", "prefer python", "python _"

tri-grams(n=3): "I prefer python", "prefer python _", "python _ _"

5.3.3 Paragraph Vectors

The distributed word representation introduced in chapter 2 maps individual words in fixed length vectors so that similar words are close to each other in the new vector space. Le and Mikolov in [63] extended that model to be able to represent texts of any length. This model, called Paragraph Vectors, follows the same principles as distributed representation for words (word embeddings). That is, context words are used to predict the next word in the paragraph. The only difference is the presence of an additional vector which represents the entire text (paragraph) and is combined with the vectors of context words to predict the next word in the paragraph. Mathematically, the difference is that in the word embedding model the hidden layer is parametrized only by the context words, so $h = F(W)$; in paragraph vectors, the paragraph matrix D is added as an additional parameter, $h = F(W, D)$.

Paragraph vector representation inherits the properties of word embeddings. That is, similar variable length texts are close to each other in vector space. Analogous to word embedding, two architectures of this model are available: Distributed Bag-of-Words Paragraph Vectors (PV-DBOW) and Distributed Memory Paragraph Vectors (PV-DM). In the latter (see figure 9), the paragraph vector acts as a memory which stores the missing word in the context and hence the name. The value of paragraph vectors is shared among all the words of that paragraph, whereas the word vectors are shared among all paragraphs. On the other hand, the PV-DBOW architecture is similar to Skip-Gram model of word embeddings, where the para-

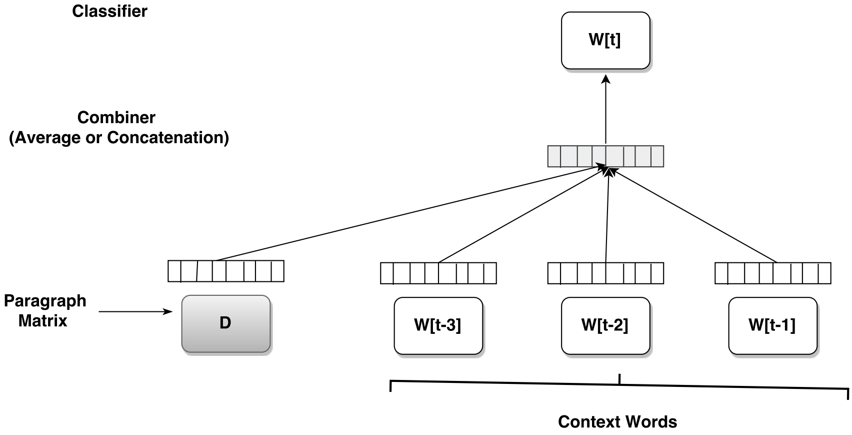


Figure 9: Paragraph Vectors [63]

graph vectors serve as inputs and the task is to predict the context words. The original paper [63] concatenates both architectures (DPV-DBOW, PV-DM) to generate fixed-size representations of variable length text. This model applies logistic regression on top of learned representation to categorize the documents into different categories.

5.3.4 Convolutional Neural Networks

With the increase of computational power deep learning algorithms have found application and set the state-of-the-art performance in many machine learning application domains. They have initially been used for computer vision, with an excellent performance in image recognition[64]. However, the increase of number of platforms which process textual data and their relevance to many investigation/analytic processes has motivated the research community to develop methods which work for text documents as well. Even though there are different deep learning architectures that can be used for this purpose, including *Recursive Neural Networks(RecursiveNN)* and *Recurrent Neural Networks(RecurrentNN)*, the scope of this thesis is limited to feed-forward deep *Convolutional Neural Networks (ConvNN)*. Lai et al. in [65] explained why ConvNN are more suited than other architectures for document classification. According to their explanation, the use of a tree structure from RecursiveNN make them very ineffective when handling large documents due to memory requirement. On the other hand, they consider RecurrentNN as biased, since words towards the end of the document have more impact than words appearing earlier. Even within feed-forward architecture there are several models that has shown remarkable success for text categorization [66, 67, 68, 69, 70]. In this thesis, we use a ConvNN model proposed by Kim in [69] mainly for two reasons: (i) the simplicity of the architecture and (ii) its high performance in natural language processing tasks such as sentiment analysis. The architecture of this model (figure 10) consists of a single convolutional layer, followed by a pooling layer and a fully-connected layer as output.

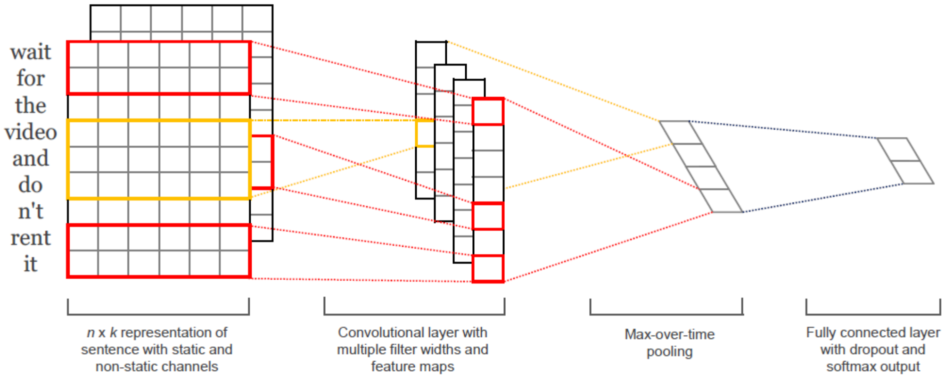

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 10: Convolutional Neural Network for text classification [69]

ConvNNs are not capable of working in variable length inputs; therefore, the inputs to the first layer should first be represented as fixed-size vectors. This is achieved by considering sentences/documents as sequences of words and obtaining their representation by concatenating the corresponding representations for each of the respective words. The variability of the length is solved by zero-padding all the sentences/documents to the length of longest (number of words) document on the dataset. For illustration, consider a sentence "*Machine Learning and Pattern Recognition*". Let us denote with v_1, v_2, v_3, v_4, and v_5 the vector representation for each of the words in the sentence, learned for example by Google's word embedding. The vector representation for this sentence takes the form of:

$$s = v1 \otimes v2 \otimes v3 \otimes v4 \otimes v5 \tag{5.1}$$

where $\otimes$ is the concatenation operator. Four different types of word vectors are used in this thesis:

- Pre-trained vectors from Google (word2vec)
- Pre-trained vectors from Stanford University (GloVe)
- Vectors trained on the data
- Random Vectors

As was explained in chapter 2, the role of the convolutional layer is the construction of features from the vector representation. The value of these features is obtained by applying a non-linear transformation (i.e. activation function) to the results of the convolution operator between the so-called filters and the respective weights. There are several activation functions that can be used for this purpose: Rectifier Linear Unit (ReLU), sigmoid function, hyperbolic tangent, etc. Their formal definition is given as follows:

$$\begin{aligned} \mathrm{ReLU} : f(x) &= \max(0, x) \\ \mathrm{sigmoid} : f(x) &= \frac{1}{(1 + e^{-x})} \\ \mathrm{tangent} : f(x) &= \frac{1 - e^{-2x}}{1 + e^{-2x}} \end{aligned} \tag{5.2}$$

These filters are shifted for a fixed height and width to obtain the value of the feature on different locations. Due to specifics of the texts, the weight of the filter is kept constant and equal to the dimensionality of word vectors (d) [71]. On the other hand, the height of the filter is varied. Multiple such filters (also known as feature maps) are applied to get multiple features. For consistency with literature we are going to refer to filter width as filter **region size**. The convolutional architecture used in this thesis support multiple region sizes(e.g. [3,4,5]), and the result of respective convolution is averaged to obtain a single output value.

The convolutional layer is followed by a pooling layer whose role is analogous to feature selection in machine learning process. Aiming to select only important features which preserve the relationship of the data, pooling takes the average or maximum value of the feature, and reduces the number of features to the number of feature maps. Following the original paper [69] and the practitioner's guide [71], we use the max-pooling strategy.

Regularization

Due to the complexity and the large number of parameters to be learned, deep neural networks are susceptible to overfitting [72]. One of the common methods used to tackle such issues is the combination of results from multiple networks. However, this is not suitable for deep networks for two reasons: (i) the time required to run multiple networks and (ii) such architecture would demand large number of training samples [72]. Srivastava et al.[72] proposed the **Dropout** model, which simulates the desired behavior by dropping out a number of neurons and their corresponding weights during training. In each training step, a new (random) set of neurons is dropped, leading to less parameters to learn and hence different results which can then be considered as having different neural networks. The number of retained neurons is not fixed; instead, they are dropped with a fixed probability p. In our model, we use $\mathrm{p} = 0.5$ as the value that has been commonly used by different authors [69, 71, 72]. The dropper can be applied to both input layer and the hidden layers. The dropping only takes place during training, whereas all the neurons are retained in testing. An additional regularization method to prevent overfitting is to constraint the l2-norms of the weight vectors, so their value will not exceed a scalar value s.

5.4 Topic Modeling

Topic models are unsupervised probabilistic models which assume that documents come from a generative process, where their topics are selected first, and then words are added for each of the topic. LDA, the topic modeling algorithm used in this thesis, and other similar algorithms, try to reverse this process and find the topics given the documents. In other words, the model accepts the documents as inputs and outputs the best topics that describe their contents. The details of LDA are explained in chapter 2, and here we just discuss about its generative process, which is actually the main principle of LDA.

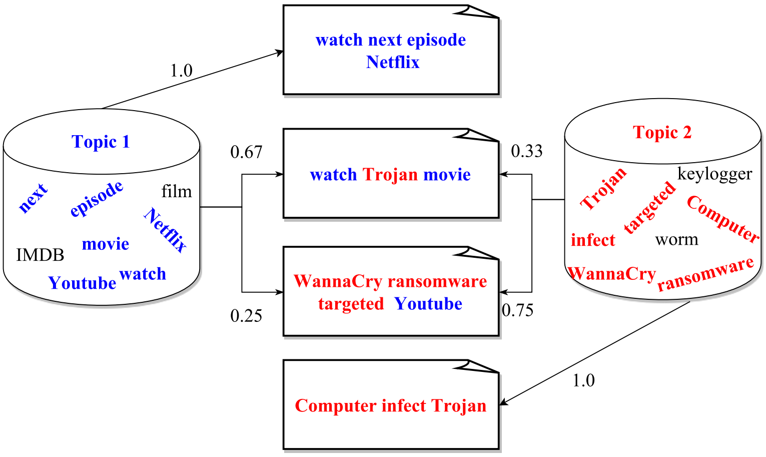


Figure 11: The generative process of topic modeling

For illustration, we have shown the generative process of LDA in figure 11. It consists of 2 topics (topic 1 and topic 2), defined as collection of words, and 4 documents. LDA regards documents as bag-of-words, where each document is created by using different distribution of the respective topics. For example, all the words of the first document "*watch next episode Netflix*" come from the first topic (marked in blue), and therefore the distribution of this topic in this document is 1.0 (100%). Similarly, the last document "*Computer infect Trojan*" is produced completely from the second topic (marked in red). On the other hand, the remaining documents are produced by mixing the words from both topics. For example, the topic distribution for the document "*watch Trojan movie*" for the first topics is 0.67(67%), while for the second topic 0.33 (33%).

From the topic keywords shown in the figure it is clear that the first topic is about movies while the second about malware. Indeed, this is the objective of LDA: to summarize the contents of large collection of documents using thematic information. We make use of this property of LDA to analyze posts from hacker forums posts, which are considered as documents. Additionally, we use this information to also group the documents based on the topics, which assists on finding relevant contents more efficiently.

Topic coherence is the measure we use to asses the quality of the topics. We define it as the information provided by the topic keywords which enables its users to easily determine the subject of the topic and differentiate it from other topics. There are several automated methods to measure a topic's coherence based on the word-similarities between each pair of words in the topic. However, since these measures may not always be consistent with the human interpretations, we use human judgment to asses the quality of the topic.

5.5 Manual Analysis

The last phase of our methodology (figure 8), manual analysis, is optional depending on the objective of analysis; should this be to just understand the security trends and asses their impact to an organization's assets (strategic intelligence), then manual analysis are not necessary as topic modeling provides enough information about the emerging trends on these forums. Furthermore, the organization and searching of posts through topics rather than keywords allows an effective discovery of the threats relevant to an organization. However, should the objective be to obtain more technical details from the posts (operational intelligence) further (manual) analysis may be necessary. Let us recall that the role of topic modeling is analogous to the role of abstract to a research paper, and further analysis analysis may be necessary to (a) validate the results and (b) get more details from the relevant posts.

In order to illustrate the role of this phase, let us suppose that the proposed methodology is used to extract malicious IP addresses. The application of the first two phases assists to efficiently locate posts that contain such addresses. One can simply extract all the available IP addresses and create rules in the security controls to block any incoming traffic from these addresses. In this scenario, no additional analysis are necessary. In order to obtain more details about these addresses however, one might have to carefully read the posts and corroborate the results with other online open sources(e.g. check the reputation of that address). Similarly, the first phases provide enough information about the trending malware and their location (for download). However, in order to convert this into actionable intelligence one may have to synchronize this information with sources such as Virustotal or even downland the malware files and perform a thorough reverse engineering.

6 Experiments and Results

This chapter presents the details of the experiment we performed to find the answers to our research questions. We provide the details of the entire experimental process including the environment and software used, the data, and the results.

6.1 Experimental Environment

The deep learning algorithms used in this thesis are computationally demanding. Actually, one of the reasons why deep learning has been an attractive research field only recently is the release of computers with GPU power and an affordable price. The computer system used for running the experiments has the following properties:

- Name: LUMPY
- Processor: Intel Quad Core i7-3820 3.60GHz
- Memory: 64GB RAM
- Hard Drive: 3.5TB
- Operating System: Windows 10 Pro 64-bit
- GPU: NVIDIA GeForce GTX 1060, 1280 CUDA cores
- GPU Memory: 6GB

All the tools and software used in this thesis are open source and freely available on the internet. The following is a list of requirements to run the code for the experiment.

- Python 3.6.0
- Sklearn 0.18.1
- Theano 0.8.2
- MySQL Community Server (GPL) 5.7.17-log

6.2 Data Overview

Even though the objective of this thesis was to explore forums from Dark Web, the data used in this thesis was obtained from *Nulled.IO* forum which is found in the surface web (i.e. Clearnet). There are several reasons behind this decision: first, the forum is publicly available on the internet eliminating any potential ethical issue with publication of the results. Furthermore, the complete relational database of the forum has been leaked which simplifies the process of extracting the fields of interest from the forum. Let us recall that the objective of this thesis is in the analytic part and not on methods to obtain the data. Finally, the forum is popular within the hacker community [73, 74], and we believe that this make it a good representative of similar forums, and at the same time increases the chance of finding relevant information.

The original forum as leaked can be obtained from `http://leakforums.net/thread-719337`. It is organized in topics (threads) and each topic has multiple posts. These two fields can be found in respective tables with the same name in the leaked mysql database. The original forum contains 121.499 threads and 3.495.596 posts with the most recent posts from the year 2016 (1 year old).

6.2.1 Binary Dataset

The first dataset constructed from the forum consists of 16,000 posts divided into 2 categories: 50% of the posts belong to the class *relevant* which cover cyber security related posts, while the other 50% consists of posts with different topics such as music, movies, sport, drugs, etc. Illustrative examples for both classes are shown in table 6.

Category	Example
Irrelevant	Hello dear nulled.io community. This is a simple question, what are your favourite movies? ; ; Mines? Idk. Probably Jackie chan movies and/or taken series
Relevant	NEW UPDATE: CVE-2015-1770 + CVE-2015-1650 This SILENT Exploit works on all Operating systems, works on all versions of Word EXCEPT Office 2013. it is a dual exploit in one builder - a combination of two different CVE's, it doesn't require any ftp or c...

Table 6: Example posts from binary dataset

6.2.2 Multi-class dataset

Motivated by the good performance on the binary dataset which will be discussed in the remainder of this chapter, we also constructed a multi-class dataset. We believe that for practical use multi-class classification is the desired solution, however, the construction of the dataset is much more demanding. Following is a description of each of the categories covered by this dataset:

Leaked Credentials. It is not unlikely for employees to use their work emails to register on different websites. Moreover, chances are that they use identical or similar passwords to the ones they use for work. Should these duplicate credentials be leaked to black markets for some reason, hackers will have access to both the data which has been leaked and, to corporate assets. One famous data breach which included credentials was the case with LinkedIn, where around 167 million of its user accounts have been leaked in a Russian Dark Web forum[1].

Denial of Service (DOS) attack. One of the requirements of cyber security is the availability of data and services to authorized users. A direct attack on this element of security is the Denial of Service attack (DOS) and its distributed version DDOS. The attack works by flooding the target with network requests until it exhausts its resources and becomes unavailable. It is very likely that the IP addresses of targets are published on hacker forums.

Keyloggers. Keystroke loggers (or keyloggers as they are known in the community) are special software designed to monitor and log the typing activity in the system when they are installed. Even though benign cases of the use of keyloggers exit, they are usually used for malicious purposes such as to obtain credentials, bank accounts, etc.

Crypters are software programs that use encryption to hide the presence of malicious code. This aims to deceive the anti-virus related solutions which are mainly based on string search or signature match.

Trojans. In Greek mythology, a Trojan Horse is the mock-wooden horse gifted to Troy that was filled with Greeks and allowed them to secretly enter the city and win the Trojan war. Similarly, in computer sciences, a Trojan is a malicious program which deceives users into installing and running by hiding its true intent. In other words, a Trojan is a malware hidden inside a legitimate program.

Remote Administration Tools (RATs) are software which allow user to access a system remotely. RATs can be used for legitimate purpose such as accessing a workstation when going on vacation, but are often used by hackers to control a victim's system.

Spamming involves posting or sending unwanted messages to a large number of recipients. The intention can be malicious (e.g. infecting other users) or in the form of advertisement. Spam posts including both those intended to infect other users and those used for advertisement, and are often present in hacker forums.

[1] http://fortune.com/2016/05/18/linkedin-data-breach-email-password/?iid=leftrail

SQL Injection is a web attack which allows the attackers to inject malicious SQL commands in the victim's system. It is caused by the lack of user input validation and its effects varies from access to unauthorized data to the complete deletion of a database. Even though it is one of the oldest web vulnerabilities, it still remains one of the most prevalent.

Not related to security category consists of data from the irrelevant class from the binary dataset. It contains data related to anything else but security: sports, drugs, music, movies, etc.

The distribution of labeled samples per category is shown in figure 12. This distribution is not uniform as credentials and posts not related to security cover 25% of samples each, spamming cover 15% of the posts, RATs 10%, while the renaming categories have an equal distribution of 5%.

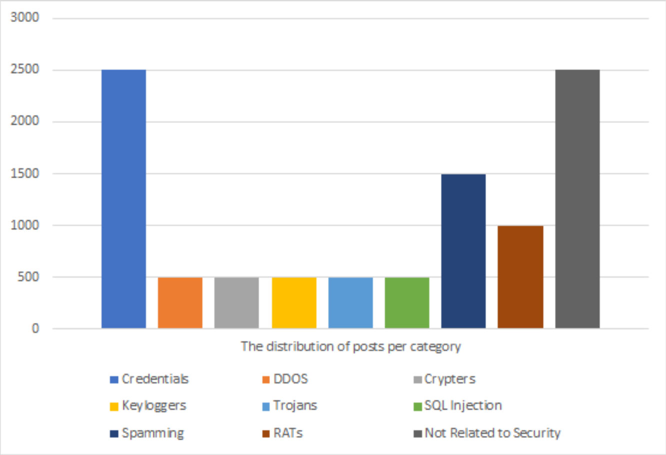


Figure 12: The distribution of data samples

Illustrative forum posts for each of the aforementioned categories are shown in table 7.

Table 7: Examples posts from multi-class dataset

Category	Example
Not Related to Security	Hello dear nulled.io community. This is a simple question, what are your favourite movies? ; ; Mines? Idk. Probably Jackie chan movies and/or taken series
Credentials	Hello Guys, just cracked these spotify accounts. It's premium. ; Enjoy. ; ; ; [hide] seand.bertran@gmail.com:Daniel81588 niru660@gmail.com:airbusa380 pskinner63@gmail.com:este11a ellis.nathan@gmail.com:unit24 jano761012@gmail.com:324657 sammy.pierce@gmail.com:kre8tv12 zelus.et.radix@gmail.com:ocelote csoto1251@gmail.com:blah123123 jcarrara24@gmail.com:maryjane yourwifealex@gmail.com:tita2165 ; [/hide] ;
DDOS attack	If i want to ddos someone, which Port should i use? and which method is best? UDP? ; 80 UDP is ok .
Keyloggers	Source code of a simple, litle Keylogger for Windows developed in C++ for Windows only. A good starting point to devlope a private FUD keylogger. ; ; Reply and upvote to unlock the source code link: it
Crypters	Hey guys! ; Here is the cracked Codelux version 3.6.6 It is a crypter to make files undetected by AV. Deobfuscated by li0nsar3c00l, cracked by Meth ; Here is the product description : ; ; ; Download ; [hide]https://mega.co.nz/#!K1wjjLDJ!YuTqbnlPVhjo6ivT-oCKX6G1G3CdyLfE3FuYmua1J6w[/hide]
Trojans	Enjoy ; ; http://www12.zippyshare.com/v/5AQ1WVsB/file.html TROJAN DETECTED!
SQL Injection	20 SQL injectable's. ; [hide] http//allaces.ru/p/episode.php?id=1' http//www.keswick2barrow.co.uk/faqs.asp?ID=1' http//www.alicantegolf.org/principal.php?idp=1' http//www.bulsu.edu.ph/news.php?id=999999.9 union all select 1,2,3,4,5,6,7777777,8,9,10 http//www.wollin.de/w3.php?nodeId=3'3 http//www.abramat.org.br/site/lista.php?secao=999999.9 union all select 1 ; ; [/hide] ; Have fun!
Spamming	User has 4 posts and spammed VIP section with 3 different fake scripts. Asking for a ban.
RATs	I think, personally, that this RAT is one of the best for its compatibility with all windows machines. ; ; Download Here: [hide] https://megabitload.com/download/index/96758211/ [/hide]

6.3 Classification of forums posts

The first phase of our methodology is the classification of hacker posts using machine learning algorithms. In this section, we show the experimental results of the baseline algorithms in the constructed datasets. However, before going into greater details of the results, we should restate that deep learning methods used in this thesis are computationally demanding. Since running the experiments in CPU takes a lot of time [71], and we only had 6GB of GPU memory in our disposal, we truncated each sentence to a maximum length of 250 words. We should emphasize that this value has not been randomly selected, but it is based in two arguments; first, 250 words is approximately the average number of words that can be written in a A4 format page, and we believe that a significant part of the information necessary for classification can be found within this size. Second, after checking the length (number of words) of the posts in our datasets we found that approximately 93% of the posts have an equal or smaller length than 250 words. Therefore, we strongly believe that our choice is reasonable considering the limitations. In table 8 we show the effect of truncation on the number of words per post.

	Original Dataset	Truncated Dataset
Maximum Length	9413	250
Average Length	90	54
Vocabulary Size	298819	158865
Documents <= 250 words	93%	100%

Table 8: The effects of post truncation

For development of traditional classifiers we use the *scikit-learn*[2] python library. In order to avoid incorrect results due to programming mistakes, we first replicate the results of other public datasets from the existing literature. The following three datasets are used to test the quality of the code:

1. **Sentence Polarity** [3]: sentiment classification dataset with 10662 samples categorized into two classes with equal sample distribution [75]
2. **Subjectivity** [4]: a binary sentence subjectivity dataset [76]
3. **Opinosis** [5]: multi-class dataset for opinion summarization [77]

[2] http://scikit-learn.org/stable/
[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[4] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[5] http://kavita-ganesan.com/opinosis

The reason for the choice of these datasets is that Zhang and Wallace in [71] report the performance of traditional classifiers using the same libraries as used in this thesis. The original paper reports the results of classification using SVM classifier with word unigrams and bigrams as features. In addition to these results, in table 9 we show the results of using other combinations of features, including word and character n-grams, and bag-of-words. The reported results are obtained by taking the average accuracy of running 10-fold cross validation for 100 times. Any potential difference in the performance of classifiers in benchmark datasets and in the datasets constructed for this thesis are due to the use of randomness and should be considered normal. The results that are highlighted in the following table and in the rest of the chapter (whenever present) indicate a better performance of that classifier (or configuration) with respect to whatever it is being compared. For example, the best performance for the Movie Reviews and Subjectivity datasets is achieved by SVM with word unigrams and bigrams, whereas for the Opinosis dataset by character unigrams and trigrams.

Features	Movie Reviews	Subjectivity	Opinosis
word bigrams	71.40	86.93	53.03
word trigrams	62.16	77.04	36.35
word (uni+bi)-grams	**78.47**	**91.61**	62.37
word (uni+tri)-grams	77.72	91.01	61.68
word (bi+tri)-grams	71.29	86.54	51.93
character unigrams	58.12	70.41	31.14
character bigrams	67.78	82.53	60.59
character trigrams	74.29	89.01	62.61
character (uni+bi)-grams	67.76	82.62	61.59
character (uni+tri)-grams	74.75	89.24	**62.80**
character (bi+tri)-grams	74.39	89.17	62.54
bag-of-words	76.87	90.92	62.41

Table 9: Classification performance of SVM classifier on three benchmark datasets. The combination of three types of features is reported: (i) word level n-grams, (ii) character level n-grams, and (iii) bag-of-words

Classification of Binary Dataset

In the following, we measure the performance of traditional classifiers in the binary dataset. The results are obtained by averaging the results of running 10-fold cross validation for 10 times. In each iteration (fold), a random sample of approximately 10% is hidden from the training algorithm and used as test data.

Features	Binary Dataset
word bigrams	91.87
word trigrams	83.16
word (uni+bi)-grams	98.09
word (uni+tri)-grams	97.58
word (bi+tri)-grams	91.23
character unigrams	75.59
character bigrams	94.21
character trigrams	**98.60**
character (uni+bi)-grams	93.99
character (uni+tri)-grams	98.45
character (bi+tri)-grams	98.55
bag-of-words	98.40

Table 10: Classification performance of SVM classifier on binary dataset

The results of SVM classifier using different features are shown in table 10. The performance of the classifier degrades when n increases for word level n-grams, and increases for character level n-grams. Contrary to the performance on baseline datasets, using character level n-grams yields better performance. More concretely, the best performance is obtained when using character trigrams.

Even though SVM is the most reported (traditional) classifier in the existing literature, we compared its performance to other classifiers such as Decision Trees and k-Nearest Neighbors (k-NN), and the results are shown in table 11.

Features	k-NN	Decision Trees	SVM
word (uni+bi)-grams	58.52	97.95	**98.09**
character trigrams	60.75	97.61	**98.60**
character(bi+tri)- grams	68.30	97.43	**98.55**
bag-of-words	61.22	98.11	**98.40**

Table 11: Classification accuracy of three classifiers on the binary dataset: k-Nearest Neighbors, Decision Trees, and Support Vector Machines

The results clearly indicate that SVM outperform k-NN and Decision Trees for the given task. The differences between accuracy values of k-NN and SVM are more significant, while the results are closer with Decision Trees. For this reason, we also measured the training and testing time for 10-fold cross validation. From the results shown in figure 13, we can infer that SVM has better classification performance, and also takes less time to run.

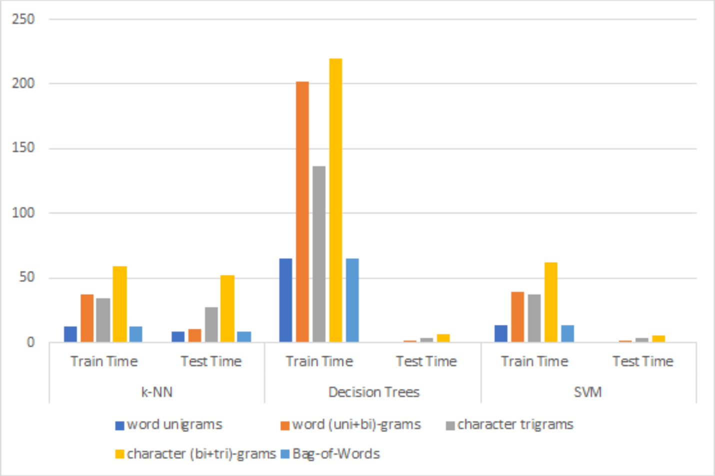


Figure 13: Training and testing time for the traditional classifiers

The value of the features can be either binary, raw frequency, or TF-IDF. Depending on the problem and the dataset, the value of the features can have a significant effect on the results. For our dataset, using binary values yields better performance in general. However, as shown in table 12, the differences in accuracy are small.

Features	Binary	Frequency	TF-IDF
word uni+bi grams	**98.19**	97.90	98.09
character trigrams	**98.82**	98.52	98.60
character bi+tri grams	**98.71**	98.50	98.55
Bag-of-Words	**98.45**	98.24	98.40

Table 12: The effect of feature normalization

So far, we have reported on the performance of more traditional (not deep-learning) classifiers such as SVM, k-NN, and Decision Trees, and identified the effects of different parameters on the performance. Now, we turn our focus to the performance of Deep Neural Network classifiers. The code for the Convolutional Neural Network used in this thesis was obtained from the original author[6] and adapted to our data. Similar to the more conventional classifiers, we begin the discussions of the results by showing their performance on the baseline datasets. In table 13 we show the results using the same configurations as in the original paper [69]. In addition to Google vectors as inputs **(w2v-ConvNN)**, we also report the performance of using Glove vectors **(Glove-ConvNN)**, random vectors **(Rand-ConvNN)**, and vectors trained internally of the data **(w2vInternal-ConvNN)**. We use these abbreviation for the rest of the thesis.

Algorithm	Movie Reviews	Subjectivity	Opinosis
w2v-ConvNN	81.52	93.36	65.79
Glove-ConvNN	79.48	93.19	64.96
Random-ConvNN	76.80	90.10	63.29
w2vInternal-ConvNN	-	-	-
Paragraph Vectors	75.12	90.83	62.22

Table 13: Deep Learning performance on baseline datasets

[6] https://github.com/yoonkim/CNN_sentence

The default configuration uses feature maps = 100, activation function = ReLU, max-pooling, dropout =0.5, l2 normalization constraint=3, and filter region size =[3,4,5]. Initially, we train for 25 epochs using the non-static model. The non-static model [69] adjusts the input vectors to the given tasks. On the other hand, the performance of Paragraph Vectors is obtained by concatenating the corresponding 300 size vectors of PV-DBOW and PV-DM models respectively.

The performance of deep learning classifiers in the binary dataset using this configuration is given in table 20. The model with vectors trained internally in the data shows better performance than the others. The vectors are obtained by running word2vec in one million posts from the Nulled dataset.

Algorithm	Accuracy(%)
w2v-ConvNN	98.22
Glove-ConvNN D=50	95.67
Glove-ConvNN D=100	97.04
Glove-ConvNN D=200	97.64
Glove-ConvNN D=300	97.65
Random-ConvNN D=50	95.23
Random-ConvNN D=100	96.69
Random-ConvNN D=200	96.91
Random-ConvNN D=300	97.23
w2vInternal-ConvNNs D=50	98.73
w2vInternal-ConvNN D=100	98.67
w2vInternal-ConvNN D=200	98.75
w2vInternal-ConvNN D=300	**98.79**
Paragraph Vectors	91.74

Table 14: Deep Learning performance on binary dataset

In a practitioners' guide, Zhange et al. in [71] identified the filter region size, the number of filters, and the activation function as parameter that have an important effect in the performance of the deep ConvNN classifier. We tested different values for these parameters, but the difference in accuracy from the default values was not significant.

The results for different filter region size(s) are shown in table 15, with all the other parameters kept constant as in the default configuration.

Filter size	Accuracy(%)	Filter size	Accuracy(%)	Filter size	Accuracy(%)
1	98.21	1,2	98.51	1,2,3	98.66
2	98.42	1,3	98.57	2,3,4	98.70
3	98.56	2,3	98.53	**3,4,5**	**98.79**
4	98.60	3,4	98.65	4,5,6	98.68
5	98.52	4,5	98.77	2,2,2	98.55
6	98.60	3,5	98.73	3,3,3	98.74
7	98.60	5,7	98.75	4,4,4	98.75

Table 15: The effect of filter region size

Similarly, the results of the four different activation functions are shown in table 16.

Activation Function	Accuracy(%)
ReLU	**98.79**
Tanh	98.66
Sigmoid	98.67
Iden	98.70

Table 16: The effect of the activation function

The results of using different number of filters are shown in table 17.

Number of filters	Accuracy(%)
10	98.55
50	98.60
100	**98.79**
200	98.68

Table 17: The effect of the number of filters

Classification of Multi-Class Dataset

We run all of the experiments reported so far in the multi-class dataset as well. In order to measure the possibility of false positives and false negatives, we also report three other performance measures in conjunction to accuracy: precision, recall, and F-measure (F1). Precision tells the ratio of forums posts which are indeed relevant from those predicted as relevant. On the other hand, recall measure the ratio of relevant posts "retrieved". F-measure is a combination(harmonic mean) of precision and recall using the following formula:

$$F = 2 * \frac{precision * recall}{precision + recall} \tag{6.1}$$

A low precision value means a greater probability for having more false positives, whereas a low recall increases the chance of having more false negatives. On the other hand, the F-measure (F1) represent a balance between precision and recall.

Features	Accuracy	Precision	Recall	F1
word (uni+bi)grams	96.93	97.69	95.48	96.51
character trigrams	**98.62**	**98.43**	**98.10**	**98.24**
character (bi+tri)grams	98.59	98.41	98.17	98.28
Bag-of-Words	97.27	97.76	96.07	96.86

Table 18: Classification performance on multi-class dataset using SVM classifier

The classification performance of SVM with different features is shown in figure 18, while its comparison to k-NN and Decision Trees is depicted in figure 19. Similar to binary dataset, SVM with character n-grams outperform other classifiers, and this is also supported by the three introduced measures.

Features	k-NN	Decision Trees	SVM
word (uni+bi)-grams	37.48	96.41	**96.93**
character trigrams	68.07	95.96	**98.62**
character(bi+tri)- grams	81.36	95.98	**98.59**
bag-of-words	66.76	96.45	**97.27**

Table 19: Classification accuracy of three classifiers on the multi-class dataset: k-Nearest Neighbors, Decision Trees, and Support Vector Machines

All of the above experiments are run using features with binary values, and this is the reason of the low performance by k-NN, which is more sensitive to the range of the features. Changing feature to TF-IDF values improve the performance of k-NN, but the two other classifier perform better with binary values.

Similar to the binary dataset, we also report the performance of deep learning classifiers. All the experiments are run using the same parameters as in the binary dataset.

Algorithm	Accuracy	Precision	Recall	F1
w2v-ConvNN D=300	97.74	98.28	96.27	97.22
Glove-ConvNN D= 50	96.78	96.99	95.33	96.09
Glove-ConvNN D=100	97.52	97.92	95.98	96.89
Glove-ConvNN D=200	97.39	97.48	95.95	96.67
Glove-ConvNN D=300	97.12	97.39	95.31	96.30
Random-ConvNN D= 50	97.23	97.90	95.70	96.74
Random-ConvNN D=100	97.41	97.94	95.76	96.77
Random-ConvNN D=200	97.45	98.27	95.75	96.94
Random-ConvNN D=300	97.17	98.22	95.24	96.63
w2vInternal-ConvNN D= 50	97.92	98.08	96.67	97.33
w2vInternal-ConvNN D=100	97.98	98.07	96.65	97.30
w2vInternal-ConvNN D=200	98.03	98.19	96.91	97.50
w2vInternal-ConvNN D=300	**98.10**	**98.24**	**97.02**	**97.60**
Paragraph Vectors	92.78	91.05	91.26	91.11

Table 20: Deep Learning performance on multi-class dataset

In general, the results are consistent with binary classification when it comes to internally trained vectors showing better performance than pre-trained and random vectors. This is supported by both F1 measure as combination of precision and recall, and the accuracy.

The best of both worlds

A summary of the classification results for both traditional and deep learning classifiers is shown in table 21. The high classification accuracy clearly indicates that filtering of irrelevant posts from hacker forums using machine learning classifiers is possible and effective.

Classifier(Features)	Binary Dataset	Multi-class dataset
k-NN(character (bi+tri)-grams)	68.30	81.36
Decision Tree(bag-of-words)	98.11	96.45
SVM(word (uni+bi)-grams)	98.19	96.93
SVM(character trigrams)	**98.82**	**98.62**
SVM(character (bi+tri)-grams)	98.71	98.59
SVM(bag-of-words)	98.45	97.27
w2v-ConvNN	98.22	97.74
Glove-ConvNN	97.65	97.12
Random-ConvNN	97.23	97.17
w2vInternal-ConvNN	98.79	98.10
Paragraph Vectors	91.74	92.78

Table 21: A summary of classification performance

What is maybe surprising from these results is the fact that the performance of SVM with trigram features is at least as good as the one of deep learning classifiers. The difference is so small (e.g 0.03%) so we cannot make a statement on which classifier performs better than the other in terms of accuracy. We can, however, claim that deep learning is much more computationally demanding than more traditional classifiers. The training may take several hours even when using GPU capabilities. This is an important issue for practical considerations of these methods.

6.4 Topic Modeling

In the second phase of our methodology we use topic modeling algorithms to discover the topics of the hacker forums posts. Additionally, we use these topics to organize and search the documents based on the topics rather than keywords. The topic modeling algorithm used in this thesis is Latent Dirichlet Allocation, and the development is done using scikit-learn[7] python library.

Since we seek to understand the effect of classification in the quality of the topics, we first report LDA results in the complete binary dataset, and then compare them to the topics of dataset after filtering irrelevant posts. LDA is not computationally demanding in terms of memory consumption and it is usually run using CPU, and therefore no truncation is applied to limit the length of the posts. For preprocessing, we remove all the non-ascii characters and lowercase the posts. Additionally, stop-words, word occurring less than 5 times, and words occurring in more than 85% of documents (tf-idf>0.85) are discarded.

Table 22: Topics discovered from running LDA on complete Binary Dataset

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
love	https	lt	**com**	favorite	music	play	anime	best	songs
wub	www	kappa	**gmail**	**spam**	game	like	**keylogger**	song	listen
movies	watch	gt	hide	**spamming**	like	just	**crypter**	food	**123456a**
thanks	youtube	league	**hotmail**	**banned**	pizza	thank	good	xd	**123456789a**
man	com	hide	**yahoo**	know	love	try	awesome	better	hide
nice	movie	**qwerty123**	**net**	series	games	good	love	love	house
favourite	http	legends	**aol**	**post**	haha	playing	cool	ty	liked
bro	hide	**abc123**	**uk**	don	wow	really	great	sharing	rep
really	use	**qwerty1**	**accounts**	like	dead	think	hope	thanks	football
sqli	**rat**	**1q2w3e4r**	naruto	**ban**	good	**trojan**	think	god	**qwerty123456**

Topic keywords shown in table 22 reflect the mixed nature of the original binary dataset. Clearly, the top keywords of the topics number 6 and 9 indicate that these topics are not related to cyber security. Similarly, the first, second, and seventh topics contains only a single security related keyword highlighted with **bold** in the table: *sqli*, *rat*, and *trojan* respectively. Slightly different, the remaining topics (3,4,5,8,10) contain a number of keywords that clearly indicate the presence of security related posts. However, the incoherence of the topics caused by the presence of non-security related keywords is evident. For example, the certainty to interpret the topic number 10 as security related indicated by the presence of common passwords *(123456a,123456789a,qwerty123456)* is weaken by the presence of words related to music (listen, song), sport (football), etc.

[7] `http://scikit-learn.org/stable/`

On the other hand, the results of applying topic modeling after filtering out non-security related data are shown in table 23. The keywords in **bold** are a clear indication of the main topics of the data. Thus, the main topics pertaining to the security in our dataset are **SQL injection** (topic 1) , **RATs**(topics 2 and 9), **Credentials Leakage** (topics 3,4, and 10), **Spamming/Banning** (topics 5 and 6), **Keyloggers** (topic 7), and **Crypters/Trojans** (topic 8).

Table 23: Topics after filtering security irrelevant data

Topic #1	**Topic #2**	**Topic #3**	**Topic #4**	**Topic #5**	**Topic #6**	**Topic #7**	**Topic #8**	**Topic #9**	**Topic #10**
sqli	**darkcomet**	**qwerty123**	**com**	**spam**	**banned**	**keylogger**	**crypter**	**rat**	hide
dumper	better	**abc123**	**gmail**	**spamming**	**spamming**	**ardamax**	**trojan**	**http**	**123456a**
sql	**njrat**	hide	**hotmail**	just	**ban**	doge	thanks	**download**	**123456789a**
thanks	mate	**qwerty1**	hide	**post**	**rules**	support	**rat**	use	**qwerty123456**
man	spoiler	**1q2w3e4r**	**yahoo**	kappa	read	37	**keylogger**	**https**	**accounts**
dorks	**nanocore**	lt	**net**	**stop**	leeching	55	hope	**www**	euw
ddos	**rat**	**lol123**	**aol**	chat	**member**	string	good	hide	upvote
injection	hard	**1qaz2wsx**	**uk**	people	**reason**	50	**fud**	**file**	spammer
good	**rats**	**password1**	**accounts**	like	know	balance	best	just	**123123a**
sharing	cracking	gt	**password**	**posts**	**username**	return	work	ddos	eune

What is maybe surprising is that certain passwords are returned as topic keywords (see topic 3 and 10). This means that they are very common among the leaked credentials. In order to validate this assumption, we located all the posts containing credentials using mutli-class classification, and computed the frequency of occurrences for each of the passwords. The list of top 10 passwords is depicted in figure 14. Their frequency values are in the range of thousands and therefore almost all of them (8/10) are returned as topic keywords from LDA.

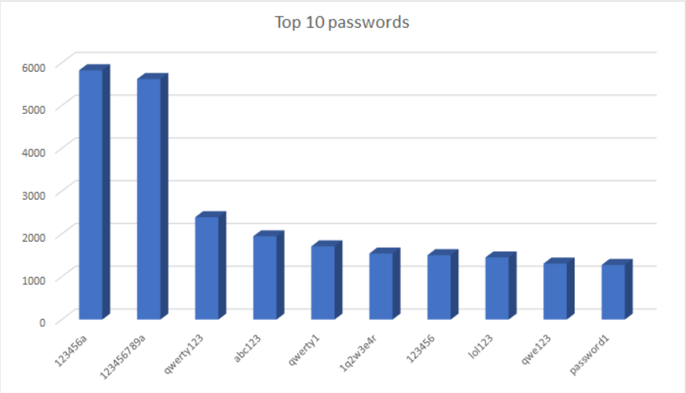


Figure 14: The frequency of top 10 passwords

It is also important to discuss the phenomenon of having similar topics expressed with different keywords. For example, the topics number 3, 4, and 10 can all be labeled as user credentials. But, since both the passwords and their context words are different, three such different topics are obtained as a result. These topic keywords can be used as indicators of the types of credentials shared in certain posts. For example, the keywords of topic number 4 are e-mail domains which indicates that the credentials of this topic contain e-mails as username. An example post from each of these three topics is shown in table 24.

Table 24: Illustration posts for topics pertaining to leaked credentials

Topic	Example
3	don't be a leacher and support the crackers ;) ; [hide]–[12/8/2015 12:54:02 am-euw]– z317:**qwerty123** derphunterz:**1qaz2wsx** meisterente:test123 aa13:123456789a sybreed1:**1q2w3e** lejew:**1q2w3e4r** trickish:1234qwer hawk33eye:a123456 keliopetit:keliopetit11 psychosimple:**abc123** blothgram:blothgram12 marcopola:1234qwer nicnax:pokemon1 dabaj:**qwerty123** fourmi72:fourmi72 jdizzle522:**password1** fiona69:fiona69 bogoss06:bogoss06 –[12/8/2015 3:05:24 pm-euw]-
4	psn accounts ;10/07/2015 ;[freshly cracked and ;checked] ;- ;please ;+rep ; ;if you want more accounts ! id:password - ; ajor. https://account.sonyentertainmentnetwork.com/login.action ; [hide] soskev6501@**gmail**.com:soskev6337 sven_siermans@**hotmail**.com:hitemup0 manutejo@**yahoo**.es:diamondfish219 [/hide]
10	i didn't had time to check them so there must be something (or not) there you go [hide] –[24-jul-15 1:40:00 pm-eune]– malenkas:malenkas1 filip1925:filip1925 scooby8911brown:scooby8911brown1 yoricfortop10:yoricfortop10 lolozas5:lolozas5 hazuref012:hazuref012 gridgg1:**123456789a** maxpejn789:maxpejn789 emil150:emil150 dubgabiezga227:dubgabiezga227 robikaa11:robikaa11 bluecilver:bluecilver12345 mogyika001:mogyika001 sima4kata:**qwerty123456** one2004:one2004 hazemhezo:hazemhezo123 czokolubie:czokolubie123 ...

For the sake of completeness, in table 24 we show the topics of data classified as irrelevant to security. The topics validate classification results as they are mainly related to sport, music, movies, food, games, etc.

Table 25: Topics from Non-Security Data

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
favorite	love	xd	cool	love	youtube	wub	league	movie	music
game	thank	year	nice	like	https	lt	champion	favourite	like
best	bad	old	op	food	www	love	favorite	http	song
anime	series	better	im	pizza	com	great	legends	awesome	naruto
think	man	ty	guy	just	watch	guys	riven	love	gt
vayne	breaking	liked	rap	know	amp	really	mid	lol	kappa
support	dead	looks	like	want	music	beer	love	team	good
skin	loved	played	coffee	movie	playlist	kind	adc	haha	love
clash	thrones	guess	pretty	thanks	list	girl	jinx	play	listen
love	season	2015	eminem	movies	songs	think	vayne	like	favorite

While topic modeling discovers the main themes pertaining the binary dataset, with multi-class dataset we have already specified the main topics (classes). Running LDA on the data from individual classes can be used as a form of validation of the labeling process, and also to organize and search the documents. For illustration, in table 26 we show topics inferred from running LDA in data classified in the SQL injection class.

Table 26: Topics from SQL injection documents

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
sqli	man	ty	program	explanation	thanks	interested	sentry	php	sqli
sql	thanks	sql	sqli	website	sqli	love	mba	58	dumper
use	injectable	injection	dumper	vulnerability	sql	hits	vulnerabilities	http	hide
dorks	dorks	xss	does	based	injection	lol	idk	www	thanks
dumper	sql	possible	kind	explain	tutorial	hq	mysql	id	need
injection	sqli	error	league	crack	site	sqli	database	select	http
learn	hide	injections	net	help	work	cracking	sqli	91	com
know	working	use	source	understand	hack	gyazo	rdp	union	got
want	urls	site	lol	experienced	learning	dorks	proxies	999999	dorks
good	dumper	products	error	hacking	lot	dumping	errors	com	download

First of all, the topic keywords shown in the table indicate that the classifier has done a remarkable job in locating posts related to SQL injection. For example, words such as *sqli, injection, urls, site, vulnerability, tutorial, dorks, php, id,* etc. are commonly used when discussing about SQL injection. Additionally, this decomposition of SQL injection posts makes the process of discovering relevant intelligence more efficient. For example, we can interpret the contents of the posts belonging to topic number 9 as follows: they contain vulnerable sites written in PHP, and the vulnerability is caused by a missing validation of the user input in parameter *ID*. This vulnerability allows the attacker to inject SQL commands such as *Select* or *Union*. After reviewing some of the posts from this topic an example of which is shown in table 27, we supported our assumption about the contents of the posts. This sort of summary allows the analysts/investigators to quickly decide whether these posts are relevant to the corporation’s assets. That is, the posts belonging to this topic may be irrelevant to a company that does not have websites written using PHP. Similarly, the posts from topic number 6 are tutorials in SQL injection attack, and we believe they have little relevance for security experts.

Table 27: Example post from SQL injection topics

Topic Keywords: php, 58, http, www, id, select, 91, union, 999999, com
fresh and working, 15 sqli’s. ; [hide] **http**//**www**.comellisrl.**com**/en/pagina.**php**?**id**=**999999**.9 **union** all **select** 1,2,3,4,5,6,7,8,9 **http**//**www**.rubenolivero.**com**.ar/prod_detail.**php**?**id**=**999999**.9 **union** all **select** 1,2,3,4,5,6,7,8,9 **www**.shapingtomorrowsworld.org/category.**php**?c=39999999.9’ **union** all **select** 1,2,3 and ’0’=’0 **www**.gl.ntu.edu.tw/joomla/teacher-detail.**php**?**id**=**999999**.9 **union** all **select** 1,2,3,4,5

An interesting effect that can be inferred from table 26 is the repetition of some words in most of the topics. We believe that this is mainly due to the length of the posts which is short in general, and the fact that some words such as sqli or inject are unavoidable when talking about this attack.

All of the results shown so far are obtained by running our methodology in a dataset which does not exceed 16.000 posts for binary dataset, and 10.000 posts for multi-class dataset. In order to understand the scalability of this methodology we applied it in a larger dataset the details of which are discussed in the following section.

6.5 Study Case

In order to explore the full potential of hacker forums as a source of threat intelligence, we applied the proposed method to a larger (unlabeled) dataset from Nulled.IO forum. The dataset consisted of 1 million randomly chosen posts. After classification using SVM with character trigrams as features, approximately 90% of the posts were classified as irrelevant to security. We should clarify that this does not mean that 90% of the topics in the forum are irrelevant; this high number of irrelevant posts is mainly due to the presence of "acknowledgment posts" where members thank each other for sharing certain data. We believe that these posts reveal little information about CTI and therefore classify them as irrelevant. For illustration, some of these posts are shown in table 28.

Table 28: Acknowledgment posts

Thanks Work!
Nice :) Thanks!
Thanks m8, time to try it out.
Thanks ell try
thanks,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Thanks mate, let's see this method :)
:-) thanks!111111111111111111111111111111111111111111

This filtering out of irrelevant posts has an significant impact on the execution time of topic modeling algorithms (e.g. LDA). While running LDA in the complete dataset (1 million posts) using our simple implementation, with no advanced optimization applied, takes approximately 17 minutes, doing the same in the relevant posts only take around 2.5 minutes. This is a significant reduction especially when analyzing data from multiple forums with each of them containing millions of posts.

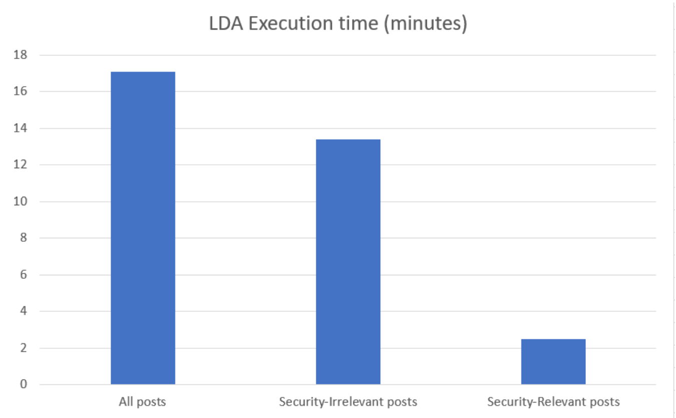


Figure 15: LDA execution time

While the time effect on the execution time is evident (figure 15), the presence of mixed data (relevant and irrelevant) and the frequency of "acknowledgment posts" are reflected in the topics shown in table 29. The topic keywords are incoherent and is very difficult to makes any generalization about the contents of the posts given only these topic-keywords. Note that increasing the number of topics will reveal more details; however, we wanted to simulate a practical scenario where the sufficient resources (time and personnel) are not always available to run the algorithm several time for each data sources in order to find the optimal number of topics.

Table 29: Topics from 1 million unfiltered posts

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
want	ty	thanks	bro	just	lets	thanks	thank	lt	com
account	thx	work	20	like	awesome	try	nice	love	rep
cool	dude	test	15	use	job	good	man	time	hide
pm	nulled	let	diamond	watch	god	works	thanks	gt	https
accounts	bol	check	18	youtube	amazing	hope	share	thanks	link
got	crack	sharing	2015	know	haha	best	kappa	update	www
im	bot	working	11	help	good	lot	wow	new	wub
buy	topic	great	30	people	game	help	mate	just	http
guy	tks	like	14	need	plz	oh	need	vouch	upvote
banned	io	xd	17	don	sounds	use	really	checking	download

Similarly, in table 30 we show the topics from posts classified as irrelevant. The topics just confirm the large number of "acknowledgment posts", and assuming that we have no prior information about their contents, no indicator of posts relevant to security can be depicted from these topic keywords.

Table 30: Topics from posts classified as irrelevant

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
thx	thank	good	rep	ty	lol	sharing	lt	test	thanks
want	kappa	bro	hide	works	wow	need	best	share	nice
use	working	love	help	hope	mate	thanks	dude	check	man
dont	awesome	thanks	http	let	edit	xd	accounts	lets	try
know	like	job	bol	great	does	post	hi	ll	work
just	testing	looking	com	look	work	lot	update	thanks	really
trying	sorry	looks	nulled	wub	just	m8	thanks	cool	gonna
problem	pretty	yes	link	friend	guys	guy	ok	upvote	god
don	server	watch	download	going	game	time	new	welcome	omg
stuff	better	youtube	oh	try	version	pm	op	skype	ill

Contrary, running LDA in the data classified as relevant, results in more coherent topics. The topics shown in table 31 are an indicator that the topics inferred from our sample dataset could be generalized to the entire forum. In other words, the main topics pertaining to security in the Nulled forum are : leaked credentials, SQL injection, spamming, malware, etc.

Table 31: Topics from posts classified as Relevant

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
hide	accounts	rat	account	nulled	bot	password	hide	spam	com
pastebin	account	thanks	just	banned	bol	123456a	com	ty	gmail
crypter	email	cracked	like	account	download	123456789a	http	cracker	hotmail
http	password	script	use	kappa	version	hide	www	gt	yahoo
enjoy	passwords	ddos	amp	topic	legends	username	https	na	hide
hope	thanks	scripts	need	io	use	user	file	spamming	euw
com	cracking	bol	want	key	exe	pass	download	crack	account
upvote	cracked	good	know	im	run	abc123	virustotal	positive	accounts
rate	sqli	nice	don	bol	error	qwerty123	analysis	just	unverified
rep	combo	man	make	ban	login	1q2w3e4r	html	lol	30

In the remainder of this chapter, we report the findings of our analysis on the security-relevant posts, which are of high importance for cyber threat intelligence platforms.

6.5.1 0days

Zero-days (i.e. 0days) are undisclosed exploits which utilize vulnerabilities on computer systems to perform some sort of malicious activities. Since they are believed to be undisclosed, there is a great chance they will go undetected by security controls. Similar to the work reported by [5], 0days are present in the Nulled forum as well. An assessment of their validity is beyond the scope of this thesis, and also almost impossible since the posts are from the last year (the most recent). However, in table 32 we show some of the posts claiming to share 0days. We should emphasize that they were extracted using keyword searches on security-relevant posts; this can be regarded as belonging to the last phase of our methodology (manual analysis). Illustrative examples of word similarities as returned from word2vec model are shown in appendix (section A.3).

Table 32: Examples of posts sharing 0days

Description	Post
OS X 0day	[hide]http://pastebin.com/i9KSpnRb[/hide] ; Not sure how this works lol ; "OS X 0day - works on latest version as of 4/30/15 BO exploitation @ fontd, allows payload to run code with fontd privileges." ; It's C syntax

OpenSSH 0day	/* * * Priv8! Priv8! Priv8! Priv8! Priv8! Priv8! Priv8! * * OpenSSH &lt;= 5.3 remote root 0day exploit (32-bit x86) * Priv8! Priv8! Priv8! Priv8! Priv8! Priv8! Priv8! * * */ void usage(char argv) ; ; printf(" HATSUNEMIKU");
	; ; printf("OpenSSH &lt;= 5.3p1 remote root 0day exploit");
	; ; printf(By: n3xus");
	; ; printf("Greetz to hackforums.net");
	; ; printf("Keep this 0day priv8!");
	; ; printf("usage: ; ; exit(1); ...

.doc 0day	Hello everybody... Me and my friend recoded and refuded 0day silent .doc exploit. We plan to sell it and ask you guys what do you think, ;how much this is worth and what should be the price for this exploit?https://www.youtube.com/watch?v=IqILJ2gfkgM ; Scan output 0/35:;scan.majyx.net/scans/result/b37c6e8be34752c3cfed82c27edcf927b85ce6b2
Joomla 0day	Hi everyone, Today , I'll share a private flaw, which is no longer seen as 2 person is already shared on the net it has paid to the deep 6months ago it is a Joomla based 0day hd flv player plugin so the flaw allows the recovery of the config file and with her you can exploit the DB seen that there are in the password; [hide]Please type in Google you inurl: /component / hdflvplayer / or inurl: com_hdflvplayer the feat you take a URL that contains the flaw. and paste the text you like thatlesitequicontienlafaille//components/com_hdflvplayer/hdflvplayer/download.php?f=../../../configuration.php /components/com_hdflvplayer/hdflvplayer/-download.php?f=../../../configuration.php ; Please look for PHPMYADMIN only by analyzing the site etc, for example if it is an OVH website you go to google you tapper phpmyadmin OVH and you put the logs that you download via the flaw.
Firefox 0day	. [hide] http://0day.today/exploit/24128[/hide]decent imo. ;this metasploit module gains remote code execution on firefox 35-36 by abusing a privilege escalation bug in resource:// uris. pdf.js is used to exploit the bug. this exploit requires the user to click anywhere on the page to trigger the vulnerability

6.5.2 Credentials, Credentials, and Credentials

One of the main topics of the Nulled forum is sharing of user credentials. This is supported by both the results of LDA and the number of posts classified in this class when using multi-class classification. In general, credentials present in this forum belong to different services such as **music/movie streaming** (Netflix, Spotify, Hulu, Tidal, Deezer, etc.), **social networks** (Facebook, Instagram, Twitter, etc.), **games/sports** (Minecraft, Runescape, Playstation, etc.), **pornography**, etc. In order to discover these types, we used similarities between word embeddings learned by applying word2vec in the data from the forum, in conjunction to topic modeling.

The analysis of these posts revealed an important phenomenon: the use of official(e.g. work) emails to register in other non-work (personal) related services. This is illustrated in table 33, which shows some of the government and educational domains which has been used for this purpose, and are present in the analyzed forum. The table should be interpreted as follows: there is at least one employee of Department of Homeland Security(@dhs.gov), a government institution in United States, that has used the official email to register in Netflix, and the corresponding credential has been leaked in the Nulled forum. A more complete lists of the leaked government and educational domains is shown in appendix (see table 35 and 36).

Table 33: Examples of using work emails for non-work related services

Service	Accounts From
Netflix	Bureau of Labor Statistics (@bls.gov), Department of Homeland Security(@dhs.gov), Environmental Protection Agency (@epa.gov), etc
Facebook	New York State Office of Parks(@parks.ny.go), Iowa State University(@iastate.edu), Monterey Peninsula College(@mpc.edu), etc
Fitbit	MIT Alumni Association (@alum.mit.edu), University of Nevada (@unr.edu), University of Oklahoma (@ouhsc.edu), University of Adelaide (@student.adelaide.edu.au), University of Utah (@nurs.utah.edu), University of California(@ucsf.edu), etc.
Shopping Stores	City of Campinas-Brazil (@campinas.sp.gov.br)
Origin	City of Bradford-United Kingdom (@bradford.gov.uk)

Regardless of the source of vulnerability (e.g. Neftflix or the educational institution), the presence of these credentials in hacker forums represent a threat to the organization's security. This is mainly due to the tendency to reuse the credentials. This is not uncommon since users tend to not remember a lot of passwords and therefore use the same (or similar) password in several services.

6.5.3 SQL injection vulnerable sites

Despite being around for a long time, SQL injection remains one of the most common cyber security attacks. Posts related to SQL injection in the analyzed forum include mainly tutorials on the attack, dorks to find vulnerable sites, and addresses of vulnerable sites including the parameter which is not validated. A large number of sites have been reported as vulnerable to SQL injection in Nulled forum; we used topics such as **{hide http com pastebin www php file html download 58}** to efficiently locate some of these sites. A list of (potentially) vulnerable sites including government institution, educational institutions, commercial sites, etc. is shown in table 34. Note that the topic keywords does not clearly indicate the presence of such sites (e.g. government), unless they are very common.

Table 34: A list of sites reported as vulnerable to SQL injection

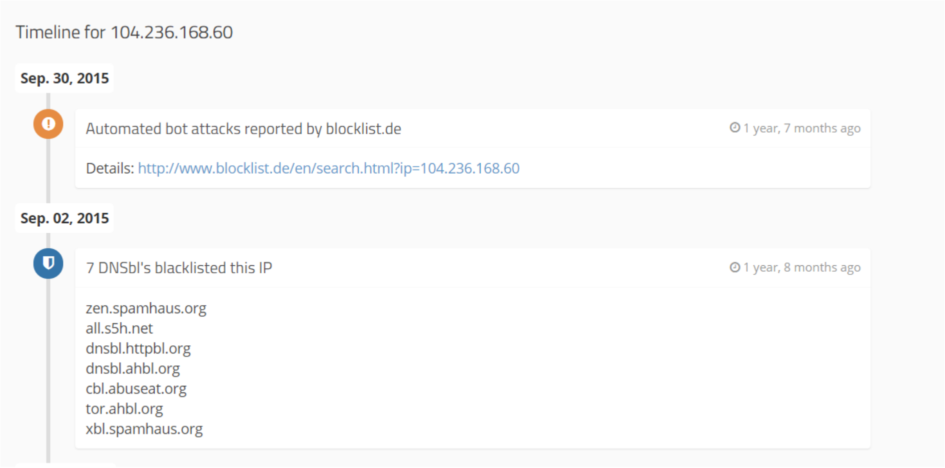
Sites reported as vulnerable
http://auto.kmart.com/product.php?brand=27
http://www.club.it.porsche.com/home.php?id=27
http://www.computerhistory.org/brochures/full_record.php?iid=27
http://www.exploratorium.edu/webcasts/archive.php?cmd=browse&project=27
http://cxc.harvard.edu/vguide/details.php?agascid=27
http://interfly.med.harvard.edu/pulldown.php?search_id=27
http://www.dsld.nlm.nih.gov/dsld/prdDSF.jsp?db=adsld&id=24180
http://scripts.mit.edu/2̃.670/schedule.php
http://list.shellypalmer.com/inc/rdr.php?r=27
http://www.andrethegiant.com/about/viewheadline.php?id=3948
http://www.ellafitzgerald.com/viewheadline.php?id=4120
http://www.jackierobinson.com/viewheadline.php?id=4181
http://www.bettedavis.com/about/viewheadline.php?id=4132
http://www.jeanharlow.com/about/viewheadline.php?id=1877
http://www.cmgww.com/baseball/munson/viewheadline.php?id=2669
http://alaska.usgs.gov/products/pubs/info.php?pubid=2410
http://calendar.ics.uci.edu/event.php?date=224
http://www.ncmc.edu/pressreleasedetail.asp?ID=96
http://www.police.gov.bd/content.php?id=27
http://www.cob.niu.edu/personnel/PersonnelDetails.asp?id=a1561246
http://un.org.np/3w/view.php?id=92
http://chechnya.gov.ru/page.php?r=180&id=1
http://ceas.stanford.edu/events/event_detail.php?id=3993

6.5.4 Malicious Proxies

Cyber criminals use anonymization methods to hide their true identity. In conjunction with the use of TOR, they launch their campaigns using proxy servers. A large number of IP addresses of such proxies are present in the analyzed forum. They were discovered by running LDA with 20 topics and following a topic with the keywords:**{80 8080 3128 2015 120 12 117 37 8123 11}**. These keywords represent port numbers of the proxy servers and a sample posts from this topic follows.

> some fresh proxy list with elite and anon proxies. i want to share them with you. ; [hide] 59.124.82.182:3128 217.23.68.70:3128 128.199.194.152:80 187.44.169.74:8080 212.47.230.183:3129 103.25.202.228:3128 219.93.183.94:80 201.249.88.202:3128 212.47.235.33:3129 46.191.69.30.239.2:80 125.227.63.200:3128 104.236.168.60:3128 119.28.14.97:80 201.207.103.10:3128 [/hide]

We checked the reputation of some of them in open source intelligence services and they were identified as malicious by several IP blacklist providers. For illustration, in figure 16 we show the output of analysis performed by https://cymon.io for one of the IP addresses chosen randomly. As it can be seen from the figure, this IP address has been marked as malicious since bot attacks has been reported as coming from this address.



Timeline for 104.236.168.60

Sep. 30, 2015

Automated bot attacks reported by blocklist.de — 1 year, 7 months ago

Details: http://www.blocklist.de/en/search.html?ip=104.236.168.60

Sep. 02, 2015

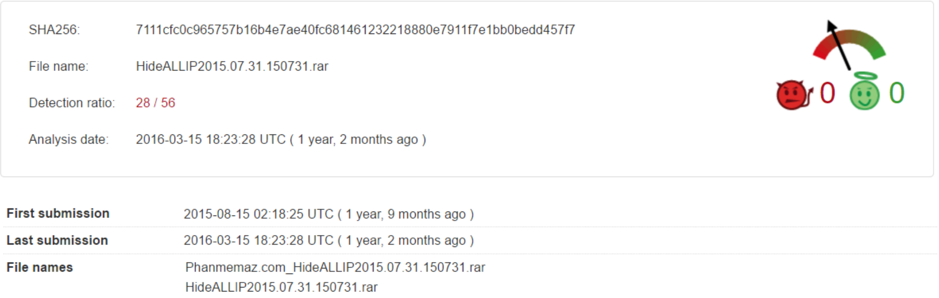
7 DNSbl's blacklisted this IP — 1 year, 8 months ago

zen.spamhaus.org
all.s5h.net
dnsbl.httpbl.org
dnsbl.ahbl.org
cbl.abuseat.org
tor.ahbl.org
xbl.spamhaus.org

Figure 16: IP addresses of potential malicious proxies

6.5.5 Malware

Malicious software cover an important part of the contents of this forum and hacking in general. Usually, hackers use these communication channels to share links to the malware which are uploaded in external sources. However, as we are going to see in the next section, the malware code in form of different scripts can also be embedded in the post. Trojans, keyloggers, RATs, etc. are the most common malware types present in the forum. We were interested and looked specifically for posts related to ransomware as they are very popular especially recently, however the data are from 2015 and therefore there are only few such posts in this forum. In order to illustrate the relevance of malware that can be found in this forum we consider one such example, an archive called **HideALLIP2015.07.31.150731.rar**. It was found by following a topic with keywords:**{com www https hide virustotal analysis file download en http}**. Its report from virustotal[8] is shown in figure 17.



SHA256:	7111cfc0c965757b16b4e7ae40fc681461232218880e7911f7e1bb0bedd457f7
File name:	HideALLIP2015.07.31.150731.rar
Detection ratio:	28 / 56
Analysis date:	2016-03-15 18:23:28 UTC (1 year, 2 months ago)

First submission	2015-08-15 02:18:25 UTC (1 year, 9 months ago)
Last submission	2016-03-15 18:23:28 UTC (1 year, 2 months ago)
File names	Phanmemaz.com_HideALLIP2015.07.31.150731.rar HideALLIP2015.07.31.150731.rar

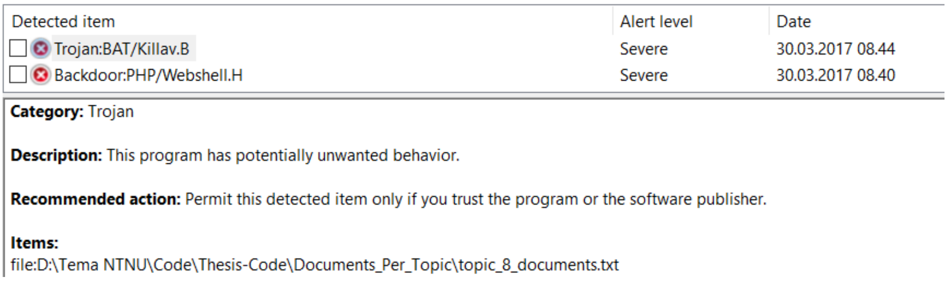
Figure 17: Virustotal report for a malicious archive

As of today, half of the anti-virus engines contacted by virustotal are able to detects its malicious behavior. What is important to note however, is the date in archive name and the first entry date in virustotal. As it can be seen from the figure, there is a difference of two weeks between the date from the name (31.07.2015) which we believe is the date when this malware was created, and the entry date in virustotal (15.08.2015). Should our assumption be correct, then this malware could have gone undetected from many anti-virus software for at least two weeks. Additionally, the date of this post in Nulled forum is just couple of hours after the submission date, which indicates that the malware present in the forum are recent and relevant.

[8] goo.gl/7CEX3Q

6.5.6 Attempts to infect other forum members

Since the forum is used to exchange cracked software, attempts to infect other members have been discovered. In addition to the posts from members claiming that certain software is malicious, during the analysis the antivirus defender in our system also detected three malware: two Trojans and a backdoor (figures 18 and 19).



Detected item	Alert level	Date
□ ⊗ Trojan:BAT/Killav.B	Severe	30.03.2017 08.44
□ ⊗ Backdoor:PHP/Webshell.H	Severe	30.03.2017 08.40

Category: Trojan

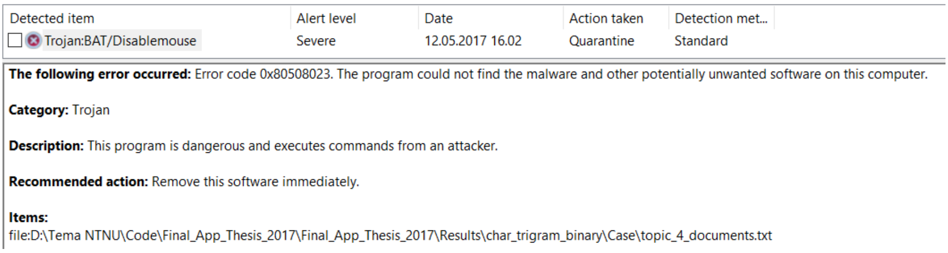
Description: This program has potentially unwanted behavior.

Recommended action: Permit this detected item only if you trust the program or the software publisher.

Items:
file:D:\Tema NTNU\Code\Thesis-Code\Documents_Per_Topic\topic_8_documents.txt

Figure 18: The detected malware (a Trojan and a backdoor) from Windows defender

It is important to emphasize that they were detected only after we stored documents according to their topic. Otherwise, they were silent. A more detailed analysis of the found malware is beyond the scope of this thesis, but from the description given from the Windows defender they should be taken seriously, given the alert is marked as *severe*.



Detected item	Alert level	Date	Action taken	Detection met...
□ ⊗ Trojan:BAT/Disablemouse	Severe	12.05.2017 16.02	Quarantine	Standard

The following error occurred: Error code 0x80508023. The program could not find the malware and other potentially unwanted software on this computer.

Category: Trojan

Description: This program is dangerous and executes commands from an attacker.

Recommended action: Remove this software immediately.

Items:
file:D:\Tema NTNU\Code\Final_App_Thesis_2017\Final_App_Thesis_2017\Results\char_trigram_binary\Case\topic_4_documents.txt

Figure 19: The detected malware (a Trojan) from Windows defender

6.6 A summary of the findings

We conclude this chapter with a summary of the findings of our experiments.

1. More traditional classifiers such as SVM with n-gram features yield at least as good performance as the deep learning classifiers used in this thesis. The differences in the classification accuracy between the two approaches are small (almost negligible), but deep learning is much more computationally demanding.

2. Topic modeling is an effective approach to explore the contents of hacker forums. The quality of the topics from these forums is improved when the irrelevant posts are filtered out prior to applying algorithms such as LDA. The main topics pertaining to cyber security in the analyzed forum are: leaked credentials, SQL injection, spamming, and malware (Trojans, RATs, keyloggers, crypters, etc).

3. A considerable number of the posts in these forums are not related to security. We support our assumption that these forums are not used only to exchange hacking related contents. Other topics that cover an important part of these platforms include music, sport, movies, drugs, pornography, etc. The ratio between security related and non-security related posts depend on the forum itself; the exact values are hardly able to be generalized. Both deep learning and more traditional classifier show a remarkable performance on filtering out these posts.

4. Highly qualitative intelligence can be extracted from the platforms used by the hacker communities. Some of the findings in the experiments run in Nulled.IO forum include 0days, IP addresses of malicious proxies, vulnerable SQL injection sites, leaked credentials, etc.

5. The forum contains posts intended to infect its members. This is supported by the malware detected by the anti-virus while running the experiments. We believe that this not properly enforcement of the forum policies might be one the reasons why the content of the forum has been leaked.

7 Discussion

Motivated by the sophistication of hacker capabilities the cyber security community has recently focused on more proactive approaches such as cyber threat intelligence to ensure the security and privacy of organizations' assets. Even though sharing within the community remains the preferred method of exchanging threat information [1], the results of this thesis have shown that hacker forums represent a valuable source of threat intelligence. The problem of these sources is, however, the discovery of the relevant intelligence, given the enormous number of available posts, which are not necessarily related to security. In order to tackle this Big Data issue, we have proposed the combination of different machine learning methods, and posed two research questions, a discussion of which is given the following.

Research Question 1:

How can different Machine Learning methods be used to classify the contents of hacker forums?

In the first phase of the proposed methodology, we classify the contents of the posts from hacker forums by regarding each post as a document, and applying machine learning algorithms for document classification. The classification model trained in a sample dataset that is manually labeled is then used to classify the remaining unlabeled samples. After the focus deep learning has gained recently, we compare its performance to more traditional algorithms. Based on the success it has shown in similar problems, we build the hypothesis that the deep ConvNN used in this thesis will outperform other conventional classifiers. The results of the experiments show that the traditional SVM classifier with character n-gram features perform at least as good as the ConvNN. Therefore, our hypothesis is not supported. We believe that the reason for this is the number of samples in the dataset; this was also stated by Zhang et al. [68] who proposed a different deep ConvNN and stated that its performance starts to get better than traditional approaches only when the dataset contains millions of samples.

It is important to emphasize that both traditional (SVM) and deep learning classifiers (ConvNN) yield high performance on the given datasets. The classification accuracy (on both datasets) is approximately 98%, which is considered a good performance in machine learning. What is maybe surprising is the high performance shown even when random vectors are used to represents individual words and their concatenation is used as the input to the ConvNN. A possible interpretation is that the neural network is able to successfully adjust the weights during training even when the inputs vectors are generated at random. Additionally, these random vectors are reused for each occurrence of a given word in the dataset, and their value even though random is limited to a value that was experimentally suggested by Kim [69].

The results are consistent with the work reported by Ebrahimi et al. [78] when it comes to vectors trained internally outperforming other pre-trained vectors. The difference is not significant however, and pre-trained vectors from word2vec or GloVe can be used should there be a lack of large number of samples required to build qualitative vectors.

To the best of our knowledge, there is no other work which studies the performance of deep convolutional neural networks on the data from hacker forums. On the other hand, Nunes et al. [5] used traditional SVM classifier with n-grams as features in a similar dataset. But, since the dataset is not publicly available a comparison of the results is impossible. However, since the difference on the reported performance is significant we have identified some possible reasons; first, we do not preserve the title (thread topic) features. The forum is organized in topics where each topic can have hundred or thousands of posts. For each of these posts the title of the post will be the same, even if the post is not relevant to security. Consequently, given the short length of the posts, the features from the title can have an important role in classification performance. Secondly, our binary dataset consists of significantly more training samples (16,000) than the dataset in [5]. Finally, we explain the high classification performance according to the "Garbage-In Garbage-Out" theorem, which makes a crucial connection between the quality of the data with the performance of the algorithm. We believe that the difference in posts length between classes in binary classification, and the presence of specific keywords in multi-class classification are some of the reasons of this high accuracy. In general, posts related to security have in average more words than posts not related to security. This is at least true for the analyzed forum, but an assessment of the generalization of this assumption is beyond the scope of this thesis. During the labeling of the posts we have also noticed that for some classes the use of some words is unavoidable. For example, it is very likely to use words such (D)DOS, IP, flood, network, ACK, target, etc. when posting about denial of service attacks, while rare are cases where these words are used when for example posting about SQL injection.

Research Question 2:

What are the main topics related to cyber security on such forums, and how does filtering through classification affect the discovered topics?

This research question consists of two parts; the first part aims to discover the main topics of the hacker posts, while the second studies the effect of pre-filtering in the quality of the discovered topics. The main topics related to security in the Nulled.IO forum are consistent with the topics reported on the related work [4, 49]. In general, leaked credentials, web and network attacks (SQL injection, DDOS), spamming, and different malware types (Trojans, keyloggers, crypters, RATs, etc) are the dominant topics on the analyzed forum. However, we believe that the distribution of these topics is different in each forum. Even though members are free to discuss different topics, there is a tendency of the forums to be more specialized in certain topics. For example, from the discovered topics from binary dataset and the number of post after multi-class classification, it seems that Nulled.IO is more focused in leaked credentials.

On the other hand, the effect of filtering irrelevant posts is measured by two factors: the topic coherence and the efficiency of the algorithms. The latter can be evaluated based on the total number of the remaining posts and the time required to run the topic modeling. The results of the experiment has shown that classification significantly reduces the number of posts that are regarded as relevant to security, and therefore the time to run the topic modeling algorithms. Similarly, the filtering out of irrelevant posts also increases the quality of the topics, measured in terms of topic coherence, making it more easier for an analyst to understand its subject and differentiate it from other topics.

7.1 The potential for practical application

The knowledge extracted from the experiments suggests that this research on CTI has commercial potential. First of all, the presence of zero-day exploits is a clear evidence that the indicators of compromise that can be found in hacker forums are of high *relevance*. Even though the assessment of the reliability of the discovered zero-days is beyond the scope of this thesis, and also difficult since the data is old, their thorough analysis can help organizations update their security controls in a timely manner, and therefore increase the intrusion detection and prevention rate. Secondly, the extracted intelligence is also *actionable*. For example, all the incoming traffic from the extracted malicious IP addresses can be denied by simply adding them to the blacklist in the firewall. The same approach can be followed for other indicators such as domain names, hash values of malicious files etc. Additionally, the intelligence from hacker forums can be used to identify the assets that require immediate patching. For example, should a corporation's website be in the list of sites vulnerable to SQL injection, then the necessary measures should be taken immediately to remedy this vulnerability. The same holds should credentials belonging to the users of a company are found in these forums. Furthermore, should these forums be analyzed in real-time then the extracted intelligence will also be *timely*. We have shown an example of a malware which was posted in the Nulled.IO forum only hours after its first submission to Virustotal. Finally, should the analysis be performed in a larger scale, then the intelligence will also be collected from *different independent sources*.

All of these are properties of good intelligence as explained in chapter 3. Therefore, we believe that this thesis can serve as foundation for a more general framework or portal for real-time analysis of the data from hacker forums and other platforms in Dark Web. An interactive method which allows better visualization and exploration(e.g. browsing) of the data would enable the analyst or investigators to have a better understanding of the real threats on these sources.

7.2 Limitations

Due to the computational requirements of the deep learning methods used in this thesis, we limited the size of the posts to 250 words. This is mainly because running ConvNN in CPU takes relatively a long time, while the GPU hardware (memory) we had at our disposal limited the length of the posts. Even though most of the posts in hacker forums are relatively short, there may always be posts which are significantly longer than others. This is a limitation of the ConvNN architecture used in this thesis which represents a single word with a vector of dimensionality D (e.g. D=300), and increasing posts length increases the computational complexity.

Another challenge of the proposed approach is the need to construct a training dataset, which is achieved through manual labeling of the posts. Manual labeling might be resource demanding especially in the case of the multi-class dataset. Using semi-supervised learning as in the case of Nunes et al. [5] reduces the number of the posts to be labeled, but does not eliminate the need for labeling. Additionally, the quality of the labeling was validated only by a limited number of experts/researchers on the field. Even though these posts have been chosen randomly, the study would benefit should all the labeled posts are validated from other peers with a related background.

Finally, the effect of the posts in a different language was not studied. Even though the main language in the forum is English, the forum contains posts in other languages as well.

8 Conclusion and Future work

The focus of this research has been to show the cyber threat intelligence potential in monitoring hacker forums, when these forums are used to share/trade hacking services with varying targets and skill levels. Even though the presence of such sources is known, little has been done to leverage their contents to enhance cyber security controls. Our research has shown that relevant, timely, and actionable cyber threat intelligence can be extracted from these forums. This conclusion is supported by our findings that included the discovery of zero-day exploits, malicious IP addresses, leaked credentials, sites vulnerable to SQL injection, etc. The value of the acquired CTI shows that this research also has a commercial development potential.

The enormous number of available posts in these forums, along with the fact that a significant proportion of posts are not necessarily relevant to security, represents a challenging analysis problem. Our solution to this problem is a method that combines highly advanced supervised and unsupervised machine learning algorithms including SVM, ConvNN, and LDA. The proposed method use supervised learning to filter out the posts irrelevant to security, and then unsupervised learning to reveal the main topics of discussion in the posts pertaining to cyber security. In our comparison with contemporary deep learning algorithms, we found that the more traditional supervised classifiers (such as SVM with character n-grams as features) remain good candidates for classification of forum posts. Given the computational complexity of deep learning architectures, our results when using traditional classifiers are especially beneficial for the purposes of practical, real-time CTI applications.

Furthermore, the supervised learning classifiers showed outstanding classification performance, by achieving an approximate accuracy of 98%. These classifiers were trained by using datasets that were constructed via the manual labeling of forum posts. Additionally, the results of our experiments have confirmed that topic modeling is an effective approach for exploring and organizing the massive contents of these forums. The topics discovered by our application of LDA has indicated the presence of the following discussion topics in the Nulled.IO forum: leaked credentials, SQL injection, spamming, and malware (Trojans, RATs, keyloggers, crypters, etc.).

8.1 Future Work

The research done in this thesis has many potential extensions. First, the performance of other deep learning algorithms should be compared with the ones used in this thesis. Specifically, while there is a large variety of deep learning algorithms to choose from, the limited scope of Master's research creates significant time constraints and so we chose to focus our efforts on Paragraph Vectors and Convolutional Neural Networks due to their architectural simplicity and the high performance shown on similar problems. So, in addition to testing other models of feed-forward neural networks, the utility of other architectures (such as recurrent or recursive deep neural networks) should be explored and studied.

Similarly, we only considered LDA as the topic extraction method. Other topic discovery methods (including variants on LDA) should be applied; for example, the utilization of some type of dynamic topic modeling to study how (and why) topics evolve over time.

In addition, further experiments should be performed using other data sources. In this thesis, we only considered posts written in English. We are confident that relevant cyber-threat intelligence can also be extracted from hacker forums that use other languages, such as Russian and Chinese. Furthermore, a comparison of the main topics extracted from forums in English and from forums in Russian may yield insights into the differences between two "schools" of hacking. While our analysis was performed using data from a single representative forum, the cited literature shows that other forums and platforms (such as underground marketplaces and IRC chats) can also be valuable sources of intelligence. So the scalability of our methods should be tested by extending them to other forums and platforms. Finally, a comparison of the topics extracted from these different sources will serve to support (or refute) our assumption that these sources are specific to certain hacker assets.

Bibliography

[1] Shackleford, D. The SANS state of cyber threat intelligence survey: CTI important and maturing. Technical report, SANS Institute, 2016. URL: https://www.sans.org/reading-room/whitepapers/bestprac/state-cyber-threat-intelligence-survey-cti-important-maturing-37177.

[2] Samtani, S., Chinn, R., & Chen, H. May 2015. Exploring hacker assets in underground forums. In *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 31–36. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2015.7165935.

[3] Benjamin, V. & Chen, H. May 2015. Developing understanding of hacker language through the use of lexical semantics. In *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 79–84. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2015.7165943.

[4] Samtani, S., Chinn, K., Larson, C., & Chen, H. Sept 2016. AZSecure hacker assets portal: Cyber threat intelligence and malware analysis. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 19–24. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745437.

[5] Nunes, E., Diab, A., Gunn, A., Marin, E., Mishra, V., Paliath, V., Robertson, J., Shakarian, J., Thart, A., & Shakarian, P. Sept 2016. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 7–12. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745435.

[6] Kononenko, I. & Kukar, M. 2007. *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Elsevier Science.

[7] Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill.

[8] Jain, A., Duin, P., & Mao, J. 2000. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 4–37. doi:10.1109/34.824819.

[9] Guyon, I., Nikravesh, M., Gunn, S., & Zadeh, L. A., eds. 2006. *Feature Extraction*. Springer Berlin Heidelberg. doi:10.1007/978-3-540-35488-8.

[10] Nguyen, H. T., Franke, K., & Petrovic, S. Nov 2011. A new ensemble-feature-selection framework for intrusion detection. In *2011 11th International Conference on Intelligent Systems Design and Applications*. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/ISDA.2011.6121657.

[11] Nguyen, H. T. *Reliable Machine Learning Algorithms for Intrusion Detection Systems*. PhD dissertation, Gjovik University College, 2012.

[12] Kaelbling, L. P., Littman, M. L., & Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4, 237–285. doi:doi:10.1613/jair.301.

[13] Cortes, C. & Vapnik, V. 1995. Support-vector networks. *Machine Learning*, 20(3), 273–297. doi:10.1023/a:1022627411411.

[14] Burges, C. J. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 121–167. doi:10.1023/a:1009715923555.

[15] Basheer, I. & Hajmeer, M. Dec 2000. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1), 3–31. doi:10.1016/s0167-7012(00)00201-3.

[16] McCulloch, W. S. & Pitts, W. Dec 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133. doi:10.1007/bf02478259.

[17] Jain, A. K., Mao, J., & Mohiuddin, K. 1996. Artificial neural networks: A tutorial. *IEEE Computer*, 29, 31–44.

[18] Jain, A. K., Murty, M. N., & Flynn, P. J. Sept 1999. Data clustering: a review. *ACM Computing Surveys*, 31(3), 264–323. doi:10.1145/331499.331504.

[19] Hastie, T., Tibshirani, R., & Friedman, J. 2009. Unsupervised learning. In *The Elements of Statistical Learning*, 485–585. Springer New York. doi:10.1007/978-0-387-84858-7_14.

[20] Griffiths, T. L. & Steyvers, M. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1), 5228–5235.

[21] Blei, D. M., Ng, A. Y., & Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993–1022.

[22] Steyvers, M. & Griffiths, T. 2007. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7), 424–440.

[23] Blei, D. M. Apr 2012. Probabilistic topic models. *Communications of the ACM*, 55(4), 77. doi:10.1145/2133806.2133826.

[24] Hoffman, M. D., Blei, D. M., & Bach, F. 2010. Online learning for latent dirichlet allocation. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, NIPS'10, 856–864, USA. Curran Associates Inc. URL: http://dl.acm.org/citation.cfm?id=2997189.2997285.

[25] Chang, J., Gerrish, S., Wang, C., Boyd-graber, J. L., & Blei, D. M. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems 22*, 288–296. Curran Associates, Inc. URL: http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models.pdf.

[26] Turian, J., Ratinov, L., & Bengio, Y. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=1858681.1858721.

[27] Bruni, E., Tran, N. K., & Baroni, M. Jan 2014. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1), 1–47. URL: http://dl.acm.org/citation.cfm?id=2655713.2655714.

[28] Lenci, A. 2008. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1), 1–31.

[29] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), 391.

[30] Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. Mar 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3, 1137–1155. URL: http://dl.acm.org/citation.cfm?id=944919.944966.

[31] Mikolov, T., Chen, K., Corrado, G., & Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781. URL: http://arxiv.org/abs/1301.3781.

[32] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. Q., eds, 3111–3119. Curran Associates, Inc.

[33] Rong, X. 2014. word2vec parameter learning explained. *CoRR*, abs/1411.2738. URL: http://arxiv.org/abs/1411.2738.

[34] Goldberg, Y. & Levy, O. 2014. word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *CoRR*, abs/1402.3722. URL: http://arxiv.org/abs/1402.3722.

[35] Pennington, J., Socher, R., & Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, 1532–1543.

[36] LeCun, Y., Bengio, Y., & Hinton, G. May 2015. Deep learning. *Nature*, 521(7553), 436–444. doi:10.1038/nature14539.

[37] Bengio, Y., Courville, A., & Vincent, P. Aug 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828. doi:10.1109/tpami.2013.50.

[38] Goodfellow, I., Bengio, Y., & Courville, A. 2016. *Deep Learning*. MIT Press, http://www.deeplearningbook.org.

[39] Schmidhuber, J. Jan 2015. Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. doi:10.1016/j.neunet.2014.09.003.

[40] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. doi:10.1109/5.726791.

[41] Gartner. Definition: Threat intelligence (online). 2013. URL: https://www.gartner.com/doc/2487216/definition-threat-intelligence (Visited 15.02.2017).

[42] ThreatConnect. Threat intelligence platforms everything you've ever wanted to know but didn't know to ask. Technical report, ThreatConnect, 2015.

[43] Friedman, J. & Bouchard, M. Definitive guide to cyber threat intelligencce. using knowlege about adversaries to win the war against targeted attacks. Technical report, iSIGHT Partners, 2015. URL: https://cryptome.org/2015/09/cti-guide.pdf.

[44] DoD, U. Joint intelligence (online). 2013. URL: http://www.dtic.mil/doctrine/new_pubs/jp2_0.pdf (Visited 02.01.2017).

[45] Chismon, D. & Ruks, M. Threat intelligence:Collecting, Analysing, Evaluating. Technical report, MWR InfoSecurity, 2015. URL: https://www.mwrinfosecurity.com/assets/Whitepapers/Threat-Intelligence-Whitepaper.pdf.

[46] Holt, T. J., Strumsky, D., Smirnova, O., & Kilger, M. 2012. Examining the social networks of malware writers and hackers. *International Journal of Cyber Criminology*, 6(1), 891–903.

[47] Motoyama, M., McCoy, D., Levchenko, K., Savage, S., & Voelker, G. M. 2011. An analysis of underground forums. In *Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference, IMC '11, Berlin, Germany, November 2-, 2011*, 71–80. doi:10.1145/2068816.2068824.

[48] Allodi, L., Corradin, M., & Massacci, F. Jan 2016. Then and now: On the maturity of the cybercrime markets the lesson that black-hat marketeers learned. *IEEE Transactions on Emerging Topics in Computing*, 4(1), 35–46. doi:10.1109/tetc.2015.2397395.

[49] Samtani, S. & Chen, H. Sept 2016. Using social network analysis to identify key hackers for keylogging tools in hacker forums. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 319–321. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745500.

[50] Benjamin, V. & Chen, H. Jun 2012. Securing cyberspace: Identifying key actors in hacker communities. In *2012 IEEE International Conference on Intelligence and Security Informatics*. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2012.6283296.

[51] Abbasi, A., Li, W., Benjamin, V., Hu, S., & Chen, H. Sep 2014. Descriptive analytics: Examining expert hackers in web forums. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, 56–63. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/jisic.2014.18.

[52] Li, W. & Chen, H. Sept 2014. Identifying top sellers in underground economy using deep learning-based sentiment analysis. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, 64,67. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/jisic.2014.19.

[53] Huang, S.-Y. & Chen, H. Sept 2016. Exploring the online underground marketplaces through topic-based social network and clustering. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 145–150. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745458.

[54] Fang, Z., Zhao, X., Wei, Q., Chen, G., Zhang, Y., Xing, C., Li, W., & Chen, H. Sept 2016. Exploring key hackers and cybersecurity threats in chinese hacker communities. In *2016 IEEE Conference on Intelligence and Security Informatics*

(ISI), 13–18. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745436.

[55] Grisham, J., Barreras, C., Afarin, C., Patton, M., & Chen, H. Sept 2016. Identifying top listers in alphabay using latent dirichlet allocation. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 219. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745477.

[56] Zhao, K., Zhang, Y., Xing, C., Li, W., & Chen, H. Sept 2016. Chinese underground market jargon analysis based on unsupervised learning. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 97–102. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745450.

[57] Benjamin, V. & Chen, H. Sept 2016. Identifying language groups within multilingual cybercriminal forums. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 205–208. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745471.

[58] Marin, E., Diab, A., & Shakarian, P. Sept 2016. Product offerings in malicious hacker markets. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 187–189. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2016.7745465.

[59] Benjamin, V., Li, W., Holt, T., & Chen, H. May 2015. Exploring threats and vulnerabilities in hacker web: Forums, IRC and carding shops. In *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 85–90. Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/isi.2015.7165944.

[60] Macdonald, M., Frank, R., Mei, J., & Monk, B. Aug 2015. Identifying digital threats in a hacker web forum. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15*, 926–933. Association for Computing Machinery (ACM). doi:10.1145/2808797.2808878.

[61] Johnson, R. & Zhang, T. 2014. Effective use of word order for text categorization with convolutional neural networks. *CoRR*, abs/1412.1058. URL: http://arxiv.org/abs/1412.1058.

[62] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759. URL: http://arxiv.org/abs/1607.01759.

[63] Le, Q. V. & Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 1188–1196. JMLR.org. URL: http://jmlr.org/proceedings/papers/v32/le14.html.

[64] Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, 1097–1105, USA. Curran Associates Inc. URL: http://dl.acm.org/citation.cfm?id=2999134.2999257.

[65] Lai, S., Xu, L., Liu, K., & Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, 2267–2273. AAAI Press.

[66] Zhang, X. & LeCun, Y. 2015. Text understanding from scratch. *CoRR*, abs/1502.01710. URL: http://arxiv.org/abs/1502.01710.

[67] Conneau, A., Schwenk, H., Barrault, L., & LeCun, Y. 2016. Very deep convolutional networks for natural language processing. *CoRR*, abs/1606.01781. URL: http://arxiv.org/abs/1606.01781.

[68] Zhang, X., Zhao, J. J., & LeCun, Y. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626. URL: http://arxiv.org/abs/1509.01626.

[69] Kim, Y. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882. URL: http://arxiv.org/abs/1408.5882.

[70] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. 2014. A convolutional neural network for modelling sentences. *CoRR*, abs/1404.2188. URL: http://arxiv.org/abs/1404.2188.

[71] Zhang, Y. & Wallace, B. C. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *CoRR*, abs/1510.03820. URL: http://arxiv.org/abs/1510.03820.

[72] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[73] www.riskbasedsecurity.com. Nulled.io: Should've expected the unexpected! (online). 2016. URL: https://www.riskbasedsecurity.com/2016/05/nulled-io-shouldve-expected-the-unexpected/ (Visited 10.04.2017).

[74] Paganini, P. The popular crime forum nulled.io pwned by hackers (online). 2016. URL: http://securityaffairs.co/wordpress/47388/data-breach/data-breach-nulled-io.html (Visited 10.04.2017).

[75] Pang, B. & Lee, L. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, 115–124, Stroudsburg, PA, USA. Association for Computational Linguistics. URL: https://doi.org/10.3115/1219840.1219855, doi:10.3115/1219840.1219855.

[76] Pang, B. & Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics. URL: http://dx.doi.org/10.3115/1218955.1218990, doi:10.3115/1218955.1218990.

[77] Ganesan, K., Zhai, C., & Han, J. 2010. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, 340–348, Stroudsburg, PA, USA. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=1873781.1873820.

[78] Ebrahimi, M., Suen, C. Y., & Ormandjieva, O. Sept 2016. Detecting predatory conversations in social media by deep convolutional neural networks. *Digital Investigation*, 18, 33–49. doi:10.1016/j.diin.2016.07.001.

A Appendix

As was explained earlier in this thesis, the original code for the deep learning methods was obtained from the original author and is freely available on the internet. That includes the method for preprocessing the posts; the only method we believe is worth sharing is the method we used to strip the HTML tags, necessary to extract the contents of the posts.

Listing A.1: SQL method to strip HTML tags

```
DELIMITER $$
CREATE DEFINER=‘root‘@‘localhost‘ FUNCTION ‘strip_tags‘
($str text)

RETURNS longtext CHARSET utf8

BEGIN

DECLARE $start, $end INT DEFAULT 1;

LOOP
        SET $start = LOCATE("<", $str, $start);
        IF (!$start) THEN RETURN $str; END IF;
        SET $end = LOCATE(">", $str, $start);
        IF (!$end) THEN SET $end = $start; END IF;
        SET $str =INSERT($str, $start,$end – $start + 1,"");

END LOOP;

END$$
DELIMITER ;
```

A.1 Goverment and Edcuational domains present in the leaked credentials

In table 35 we present a more comprehensive list of the domains belonging to educational(.edu) and government(.gov) institutions which were discovered in the leaked credentials.

Table 35: The email domains of accounts found in the forum (Educational only)

Domains
uwec.edu, hocking.edu, mail.slc.edu, wisc.edu, bsu.edu, nshs.edu, daralhekma.edu, ccse.kfupm.edu, gwmail.gwu.edu, osu.edu, lynn.edu, loop.colum.edu, csu.fullerton.edu,mscd.edu, connect.wcsu.edu, hawaii.edu, yhc.edu, uow.edu, elmhurst.edu, ucdavis.edu, gatech.edu, utsa.edu, cornell.edu, american.edu, stetson.edu, mnstate.edu, ucsc.edu, wmich.edu, isu.edu, alum.mit.edu, bc.edu, franklincollege.edu, mail.umkc.edu, uark.edu, ttu.edu, uni.edu, mayo.edu, fq.edu, aesop.rutgers.edu, upr.edu, monash.edu, esf.edu, uncg.edu, uky.edu, uws.edu, swinburne.edu, msu.edu, uq.edu, thapar.edu, biochem.umass.edu, qau.edu, uga.edu, utk.edu, hsc.edu, clarku.edu, sydney.edu, ucr.edu, umn.edu, gate.sinica.edu, berkeley.edu, tmu.edu, cau.edu, utar.edu, duke.edu, syr.edu, purdue.edu, njnu.edu, ksu.edu, hcmiu.edu, thu.edu, deakin.edu, camden.rutgers.edu, cqu.edu, uwsuper.edu, aya.yale.edu, lsu.edu, student.mccneb.edu, spsu.edu, fiu.edu, colorado.edu, umflint.edu, cc.hwh.edu, wpi.edu, brookwood.edu, mail.amc.edu, uscga.edu, students.wwu.edu, cmich.edu, umd.edu, bears.unco.edu, lab.icc.edu, mail.gvsu.edu, ufl.edu, emich.edu, mst.edu, upstate.edu, ntu.edu, williams.edu, lclark.edu, humboldt.edu, tcu.edu, mail.roanoke.edu, aucegypt.edu, mail.usf.edu, pace.edu, buffalo.edu, albright.edu, psu.edu, mtu.edu, stanford.edu, musc.edu, kean.edu, unomaha.edu, connect.qut.edu, mit.edu, columbia.edu, uci.edu, temple.edu, pop.belmont.edu, umiami.edu, bulldogs.barton.edu, virginia.edu, csulb.edu, uwlax.edu, juniata.edu, aggiemail.usu.edu, student.gvsu.edu, onu.edu, odu.edu, knights.ucf.edu, niu.edu, pegs.vic.edu, radford.act.edu, moody.edu, vt.edu, mail.weber.edu, uvm.edu, mccn.edu, anadolu.edu, uwm.edu, smu.edu, email.unc.edu, jmu.edu, umich.edu, exeter.edu, nmu.edu, Mercyhurst.edu, chartercollege.edu, mednet.ucla.edu, uchicago.edu, ohsu.edu, iastate.edu, shsu.edu, email.vccs.edu, kent.edu, clemson.edu, sendit.nodak.edu, mail.buffalostate.edu, wiu.edu, email.arizona.edu, uwstout.edu, uakron.edu, tmcc.edu, wustl.edu, hamline.edu, my.normandale.edu, asu.edu, indiana.edu, tufts.edu, uvas.edu, vet.upm.edu, sgu.edu, princeton.edu, usyd.edu, colostate.edu, vet.upenn.edu, wvdl.wisc.edu, chem.ucla.edu, stolaf.edu, students.ecu.edu, jhu.edu, mavs.uta.edu, uab.edu, olemiss.edu, email.sc.edu, mcdaniel.edu, northgeorgia.edu, emory.edu, tcnj.edu, usnh.edu, unm.edu, kctcs.edu, ohio.edu, dhips.ttct.edu, liberty.edu, fsu.edu, upmc.edu, uabc.edu, iupui.edu, umkc.edu, regis.edu, sage.edu, auburn.edu, georgetown.edu, mail.missouri.edu, students.deltacollege.edu, mail.chapman.edu, dartmouth.edu, jtsa.edu, nwu.edu, noctrl.edu, yale.edu, plymouth.edu, huskymail.uconn.edu, glennie.qld.edu, quinnipiac.edu, buckeyemail.osu.edu, ualr.edu, my.macu.edu, post.harvard.edu, memphis.edu, unc.edu, daltonstate.edu, gps.caltech.edu, willamette.edu, cal.berkeley.edu, ifms.edu, cs.unc.edu, rohan.sdsu.edu, spu.edu, uiowa.edu, eastms.edu, calvarycc.qld.edu, doane.edu, jpc.vic.edu, rpi.edu, mc.duke.edu, London.edu, mymsmc.la.edu, cardinalmail.cua.edu, wudosis.wustl.edu, rutgers.edu, student.uwa.edu, cnu.edu, my.fsu.edu, unh.newhaven.edu, ulm.edu, ashland.edu, student.fdu.edu, pointloma.edu, brenau.edu, ist.ucf.edu, u.washington.edu, hsph.harvard.edu, my.lonestar.edu, ncnu.edu, frc.edu, cuw.edu, ivytech.edu, sandiego.edu, csbsju.edu, ramapo.edu, uiuc.edu, jjay.cuny.edu, eden.rutgers.edu, med.unc.edu, deu.edu, med.umich.edu, students.kennesaw.edu, agruni.edu, dukes.jmu.edu, oregonstate.edu, calpoly.edu, iuk.edu, spartan.northampton.edu, mail.utexas.edu, mail.bw.edu, student.bridgew.edu, park.edu, email.msmary.edu, gustavus.edu, mail.harvard.edu, bridgew.edu, coastal.edu, latech.edu, v txstate.edu, cougars.ccis.edu, nsu.edu, unsa.edu, my.uu.edu, uoregon.edu, unisa.edu, alumni.princeton.edu, alumni.brown.edu, norwich.edu, alumni.pitt.edu, potsdam.edu, miracosta.edu, mail.med.upenn.edu, gac.edu, citadel.edu, oakland.edu, det.nsw.edu, oglethorpe.edu

Similarly, in table 36 we present the government domains that were present in the leaked credentials.

Table 36: The email domains of accounts found in the forum (Government only)

Domains
sefaz.ba.gov, eletrosul.gov, caixa.gov, joufmail.gov, aljoufedu.gov, moe.gov, rcjubail.gov, swcc.gov, sagia.gov, courts.phila.gov, tmag.tas.gov, frim.gov, mardi.gov, mpob.gov, bop.gov, barnet.gov, lbl.gov, maine.gov, enigma.rs.poznan.uw.gov, moag.gov, daff.gov, peo.gov, calpers.ca.gov, usdoj.gov, loc.gov, opm.gov, aqsiq.gov, trt10.gov, efeis1.bomba.gov, putra6.spa.gov, putra2.spa.gov, cdc.gov, hants.gov, la.gov, ssa.gov, nwpg.gov, rochdale.gov, rushcliffe.gov, epa.gov, sayistay.gov, adfa.arkansas.gov, schools.nyc.gov, bradford.gov, llnl.gov, avonfire.gov, southglos.gov, ukho.gov, patchwaytowncouncil.gov, mail.ncpb.gov, mail.nih.gov, bls.gov, dhs.gov, campinas.sp.gov, pmmg.mg.gov, uberlandia.mg.gov, crt.la.gov, parks.ny.gov, camden.gov, newham.gov, fssa.IN.gov, tce.sp.gov, nasa.gov, camarailhacomprida.sp.gov, rpa.gsi.gov, treasury.ap.gov, giris.turkiye.gov, sonuc.osym.gov, ais.osym.gov, esgm.sgk.gov, staffordshire.gov

A.2 Topics of posts from Nulled.IO forum

In this section we list the common topics discovered by using LDA in the posts from Nulled.IO forum after multi-class classification.

Table 37: Topics from posts classified as Leaked Credentials

Topic #1	**Topic #2**	**Topic #3**	**Topic #4**	**Topic #5**	**Topic #6**	**Topic #7**	**Topic #8**	**Topic #9**	**Topic #10**
premium	bol	account	80	hide	com	hide	amp	hide	hide
netflix	use	accounts	8080	enjoy	gmail	http	gt	123456a	https
origin	cracked	email	3128	upvote	hotmail	com	http	123456789a	file
spotify	script	password	8123	http	yahoo	www	com	euw	com
pass	download	change	117	com	hide	https	lt	accounts	zippyshare
combo	crack	cracked	120	kappa	net	pastebin	login	30	html
email	scripts	just	195	gyazo	aol	download	user	level	fun
accounts	account	crack	190	forget	uk	virustotal	members	eune	mega
games	click	hide	177	rep	live	enjoy	password	skins	leecher
2015	using	thanks	202	don	fr	mediafire	username	na	github

Table 38: Topics from posts classified as (D)DOS attack

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
ddos	skype	hide	booter	bol	xd	ddos	www	boost	account
friends	ipstresser	http	kappa	scripts	dos	like	com	free	boot
game	resolver	com	thanks	master	ddosed	booter	https	stresser	bot
thx	booster	download	nice	lua	work	good	nulled	booters	just
league	api	www	ddos	script	friend	know	analysis	hope	pm
man	resolve	https	ddoser	cracked	mate	just	io	gold	paypal
ip	application	link	drop	raw	pass	ip	virustotal	quezstresser	10
drophack	steam	php	buy	githubusercontent	does	free	topic	lets	ip
fix	ip	script	time	boots	ddos	people	en	works	banned
server	wanna	file	max	paid	auth	attack	http	booter	skype

Table 39: Topics from posts classified as Keyloggers

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
step	working	bol	account	thanks	legends	rebot	keylogger	com	added
wow	version	use	accounts	wub	bot	steam	keyloggers	hide	new
install	crack	download	unverified	newest	dominate	cdpatcher	logger	www	keyboard
click	spoiler	just	euw	kept	login	proxies	best	http	clashbot
phone	combo	hide	30	versions	io	auth	thanks	gt	fixed
exe	script	account	verified	date	download	regards	nz	https	th
enter	cracked	login	email	thread	nulled	premium	mega	virustotal	gt
net	work	file	acc	let	auth	gt	hide	file	troops
open	anymore	key	skins	process	vendor	patcher	ty	analysis	bot
program	scripts	cracked	level	bugs	http	key	nice	download	spoiler

Table 40: Topics from posts classified as Crypters

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
crypter	hide	virustotal	source	hope	key	hide	thank	service	clean
thanks	file	analysis	decrypted	fuck	decryption	http	crypter	color	cryptex
good	use	file	gt	needed	decrypt	com	sharing	selling	js
nice	crypter	thanks	topic	apk	ty	www	application	native	works
need	know	crack	nulled	crypter	mega	download	ip	going	ok
work	just	com	crypter	op	link	link	vpn	master	best
fud	like	www	add	release	thx	https	thanks	pm	legit
man	want	https	io	person	edit	net	facebook	bin	got
crypt	code	looking	www	thanks	nz	php	test	admin	botnet
crypters	http	crypter	https	jpg	need	html	good	net	35

Table 41: Topics from posts classified as Trojans

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
detected	fixed	trojan	bol	th	trojan	report	com	color	wordpress
trojan	clashbot	win32	just	github	wp	zoom	www	coder	58
says	added	gen	hide	snipe	virus	hack	https	updated	theme
thx	spoiler	generic	use	troops	nice	file	hide	lua	plugin
kaspersky	boost	fuck	download	tree	avast	91	virustotal	hide	direct
virus	custom	91	work	gamebot	ty	exe	analysis	size	item
antivirus	attack	hope	version	modified	program	size	file	scripts	links
hey	trophy	ratio	bot	attack	pro	amp	download	nitroflare	sales
detect	release	detection	file	master	thanks	original	en	github	virusscan
avg	barracks	virustotal	gt	forums	download	amazon	http	com	ratio

Table 42: Topics from posts classified as SQL Injection

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
sqli	traffic	mysql	rghost	kali	sqli	2015	gt	sql	hide
combo	wordpress	js	pl	linux	sql	11	lt	injection	http
pass	area	bot	net	ref	use	14	91	proxy	com
thanks	plugins	protocol	injectable	ssl	need	12	item	spoiler	pastebin
hq	plugin	node_modules	botnet	hacking	just	script	row	file	www
combolist	servers	error	http	pre	thanks	10	lowest_price	https	php
user	support	lib	interested	hi	script	13	var	use	file
good	file	connection	udemy	red	know	15	function	sqli	html
dumper	source	database	password	session	combos	zip	amp	tool	download
ty	demo	users	hash	secure	mysql	91	php	malwr	58

Table 43: Topics from posts classified as RATs

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
legends	http	rat	rats	bol	clashbot	xerath	just	hide	script
bot	www	thanks	account	run	added	script	account	com	lua
dominate	nulled	rate	accounts	folder	gt	bol	like	http	sac
login	topic	man	paypal	open	administrator	scripts	know	download	combo
available	https	nice	pm	exe	notes	cracked	want	www	master
download	com	hope	10	try	positive	banned	don	file	wow
vip	php	ty	30	error	needs	kappa	time	https	auto
http	io	good	selling	problem	privileges	use	people	rar	best
free	net	bro	btc	administrator	false	lol	use	link	raw
io	steam	thank	skins	leaguesharp	antivirus	rate	need	use	enemy

Table 44: Topics from posts classified as security-irrelevant

Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6	Topic #7	Topic #8	Topic #9	Topic #10
ty	thank	hope	nice	rep	thx	lets	lol	thanks	good
try	works	lt	check	help	bro	im	best	man	work
let	kappa	love	dude	link	thanks	god	account	test	working
share	great	xd	like	hide	mate	amazing	use	wow	sharing
thanks	really	com	thanks	bol	need	sure	time	ll	thanks
gonna	testing	free	looking	just	lot	don	accounts	look	awesome
wub	cool	upvote	leak	nulled	want	leecher	just	post	job
friend	ok	https	update	download	looks	crack	game	m8	edit
think	ill	yes	buddy	10	oh	guy	pm	try	does
interesting	vayne	watch	banned	http	good	trying	new	going	dont

A.3 Using word2vec to find word similarities

In this section, we present the word similarities results by applying word2vec in the data from Nulled.IO forum. In table 45 we depict the similar words to Netflix and Spotify. As it can be seen from the table, there is considerable similarity between the returned words as they often appear in the leaked credentials together.

Words similar to Netflix	Words similar to Spotify
spotify	netflix
hulu	hulu
crunchyroll	directv
starbucks	tidal
psn	crunchyroll
tidal	deezer
minecraft	minecraft
deezer	playstation
directv	premium
origin	starbucks

Table 45: Top similar words to Netflix and Spotify

Similarly, the top 10 similar words to Darkcomet (a type of RAT) and Trojan are shown in table 46. These similarities are useful to an investigator to for example understand the types of trending RATs in the forum. This can be inferred from the words such as njrat, babylon, nanocore, netwire, jrat, etc. that represent different types of RATs.

Words similar to Darkcomet	Words similar to Trojan
njrat	backdoor
babylon	virus
nanocore	nod32
rat	malware
keylogger	av
netwire	msil
jrat	win32
cybergat	generic
crypter	kaspersky
blackshades	variant

Table 46: Top similar words to Darkcomet and Trojan