Predicting CyberSecurity Incidents Using Machine Learning Algorithms: A Case Study of Korean SMEs

Alaa Mohasseb[1], Benjamin Aziz[1], Jeyong Jung [2]and Julak Lee[3]
[1]*School of Computing, University of Portsmouth, United Kingdom*
[2]*Institute of Criminal Justice Studies, University of Portsmouth, United Kingdom*
[3]*Department of Security Management, Kyonggi University, Suwon, South Korea*
{*alaa.mohasseb, benjamin.aziz, jeyong.jung*}*@port.ac.uk, julaklee@kyonggi.ac.kr*

Keywords: Text Mining, Cybersecurity, Malware, Malicious Code, Machine Learning.

Abstract: The increasing amount and complexity of cyber security attacks in recent years have made text analysis and data-mining based techniques an important factor in detecting security threats. However, despite the popularity of text and other data mining techniques, the cyber security community has remained somehow reluctant in adopting an open approach to security-related data. In this paper, we analyze a dataset that has been collected from five Small and Medium companies in South Korea, this dataset represents cyber security incidents and response actions. We investigate how the data representing different incidents collected from multiple companies can help improve the classification accuracy and help the classifiers in distinguishing between different types of incidents. A model has been developed using text mining methods, such as n-gram, bag-of-words and machine learning algorithms for the classification of incidents and their response actions. Experimental results have demonstrated good performance of the classifiers for the prediction of different types of response and malware.

1 INTRODUCTION

The use of text analysis and data mining in detecting vulnerabilities and Cyber security threats is an activity that has been going on for a number of years now. The increasing amount and complexity of Cyber security attacks in recent years have brought data mining techniques into the attention of researchers and experts as an important technique in detecting such attacks through the analysis of data and the side-effects left by malware and spyware programs and the incidents of network and host intrusions.

Text analysis and mining is widely used in many Cyber security areas, such as malware detection and classification (Suh-Lee et al., 2016; Kakavand et al., 2015; Norouzi et al., 2016; Fan et al., 2015; Hellal and Romdhane, 2016; Lu et al., 2010; Fan et al., 2016; Rieck et al., 2011; Ding et al., 2013) and malicious code detection (Bahraminikoo et al., 2012; Schultz et al., 2001; Shabtai et al., 2012).

In addition, the popularity of social media has opened up the doors for text mining and analysis as important techniques for increasing the knowledge about users' context, e.g. their location and time, and combining that knowledge with other attributes related to important events, topics, emotions and interests (Inkpen, 2016). Other applications for such techniques have included predicting links (Bartal et al., 2009) and detecting leaks of confidential data (Ojoawo et al., 2014), for example, private health information that users may inadvertently share on social media (Ghazinour et al., 2013). Moreover, text clustering and analysis has also been used extensively in digital forensics, e.g. as in (Decherchi et al., 2009) where text clustering was applied to the Enron corpus (Klimt and Yang, 2004), or in (Xylogiannopoulos et al., 2017), where text mining algorithms were applied to unclean, noisy or scrambled datasets that can be obtained from electronic communications such as SMS communications, or in (Hicks et al., 2016) where text mining was used for performing Web text analysis and forensics.

The Cyber security community has remained somehow reluctant in adopting an open approach to security-related data despite all the popularity of text and other data mining techniques, due to many factors such as political factors, for example, the fear of many organizations to share their data since these data could reveal sensitive information. Others factors are more technical, such as the metrics that should be used to quantify the security data themselves (Hoo, 2000) and he consistency, quality and the lack of consensus on the nature of variables that should be monitored. Furthermore, There is also the factor that related to whether past data are relevant to future events (Parker, 1998). However, with the availability of large and open security datasets and data-sharing platforms backed by the reliability and reputation of well-established organisations in the area of Cyber security, e.g. VCDB (VERIZON,), CERT's Knowledge base at Carnegie Mellon University (CERT Coordination Center,), SecRepo (Mike Sconzo,), CAIDA (Center for Applied Internet Data Analysis,) and others, we are starting to witness an increasing trend in the usage of such datasets in gaining insight into how incidents occur.

In this paper, we analyse a dataset that has been collected from five Small and Medium Enterprises (SMEs), which represents textual data describing Cyber security incidents that occurred in those companies and the response actions that were applied. In addition, we investigate how the data representing different incidents collected from multiple companies can help improve the classification accuracy and help the classifiers in distinguishing between different types of incidents. This is achieved by focusing on two sets of questions: the first includes forward-looking predictive questions, such as the prediction of future responses from the type of malware or the name of the malicious code encountered in previous incidents, and the second includes backward-looking questions that reverse-engineer the type of the malware or the name of the malicious code from past responses to incidents. The main objective of our analysis is to demonstrate how a centralised hub may collect experience from multiple organisations in order to train a single classifier that can predict features of future Cyber security incidents.

The rest of the paper is structured as follows. Section 2, highlights other works in the literature related to the work presented in this paper. Section 3, outlines the research problem and our approach in solving it. The experimental setup and results in applying the classification problem to the Cyber security incidents dataset are presented in section 4. Finally, Section 5, concludes the paper and outlines some directions for future research.

2 Related Work

In this section we review related works for detecting and classifying malware and malicious code using text analysis and data mining methods. Data mining techniques have many applications related to malware detection. In (Suh-Lee et al., 2016) authors detect security threats using data mining, text classification, natural language processing, and machine learning by extracting relevant information from various unstructured log messages. Authors in (Kakavand et al., 2015) proposed an anomaly detector model called Text Mining-Based Anomaly Detection (TMAD) model to detect HTTP attacks in network traffic. The proposed model uses n-gram text categorization and (Term Frequency, Inverse Document Frequency) TF-IDF methods.

Different classification methods have been proposed by (Norouzi et al., 2016) in order to detect malware programs based on the feature and behavior of each malware program. In addition, authors in (Fan et al., 2015), utilised hooking techniques to trace the dynamic signatures that malware programs try to hide, for the classification process, machine learning algorithms were used such as Naïve Bayesian, J48 (Decision Tree), and Support Vector Machine.

In (Hellal and Romdhane, 2016) authors proposed an approach that combines static analysis and graph-mining techniques. In addition, a novel algorithm was proposed, called Minimal Contrast Frequent Subgraph Miner (MCFSM) algorithm, which is used for extracting minimal discriminative and widely employed malicious behavioral patterns. Furthermore, authors in (Lu et al., 2010) proposed a new ensemble learning model, SVM-AR. The proposed model combined features extracted from both content-based and behavior-based analysis to represent instances. While in (Rieck et al., 2011) a framework was proposed for the automatic analysis of malware behavior using machine learning. The framework allows for automatically identifying novel classes of malware with similar behavior (clustering) and assigning unknown malware to these discovered classes (classification).

In (Ding et al., 2013), an Application Programming Interface (API)-based association mining method was proposed for detecting malware. A classification method based on multiple association rules was adopted. Furthermore, data mining-based malicious code detectors have been proven to be successful in detecting clearly malicious code, e.g. like viruses and worms. In (Bahraminikoo et al., 2012) a method was proposed for spyware detection using data mining techniques. The framework focused on DM-based malicious code detectors using Breadth-First Search (BFS) approach, which are known to work well for detecting viruses and similar software.

In (Schultz et al., 2001) authors proposed a data-mining framework that detects new, previously unseen malicious executable accurately and automatically. The data-mining framework automatically found patterns in the data set and used these patterns to detect a set of new malicious binaries. A Machine learning algorithms were used such as RIPPER, Naïve Bayes and Multi-Naïve Bayes. The authors in (Fan et al., 2016) proposed a sequence mining algorithm to discover malicious sequential patterns, based on the instruction sequences extracted from the file sample set, and then a Nearest-Neighbor (ANN) classifier was constructed for malware detection based on the discovered patterns. The developed data mining framework composed of the proposed sequential pattern mining method and ANN classifier.

Furthermore, authors in (Wang et al., 2006) proposed an integrated architecture to defend against surveillance spyware and used features extracted from both static and dynamic analysis. These features were ranked according to their information gains. In addition, a machine learning algorithm was used. In (Abou-Assaleh et al., 2004), the authors presented a method based on byte n-gram analysis to detect malicious code using Common N-Gram analysis (CNG), which relies on profiles for class representation. The authors in (Shabtai et al., 2012) presented the inspected files using OpCode n-gram patterns, which are extracted from the files after disassembly for detecting unknown malicious code. The OpCode n-gram patterns are used as features for the classification process.

Other works used machine learning techniques for the detection classification of malicious code. In (Hou et al., 2010) authors proposed a malicious web page detection using of machine learning techniques by analyzing the characteristic of a malicious Web page. In addition, authors in (Zhang et al., 2007) proposed a method to automatically detecting malicious code using the n-gram analysis. The proposed method used selected features based on information gain. Finally, in (Elovici et al., 2007) authors proposed an approach for detecting malicious code using machine learning techniques. Three machine learning algorithms were used which are Decision trees, Neural Networks and Bayesian Networks in order to determine whether a suspicious executable file actually inhabits malicious code.

3 The Proposed Approach

The main scenario motivating our work is one in which a centralized hub, shown in Figure 1, collects data generated by multiple companies (organisations) and therefore maintains a centralized dataset representing the collective experience of those companies. The datasets collected from these companies are used to train one centralized classifier, which would then have better performance than any individual instance belonging to a single company.

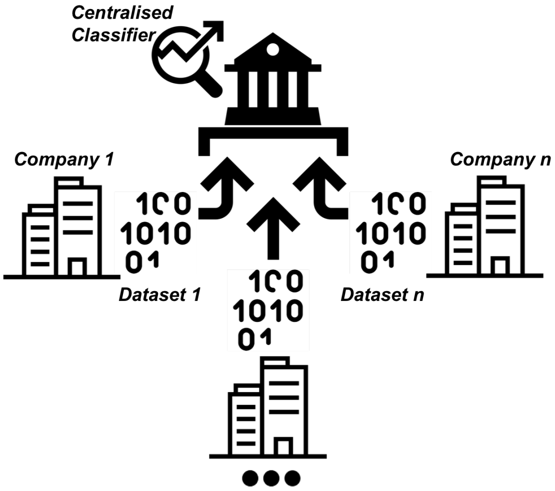

Figure 1: Experience collection from *n* number of companies

3.1 Description of the Dataset

The dataset represents Cyber security intrusion events in five Small and Medium Enterprises (SMEs) over a period of ten months, which were collected by the KAITS Industrial Technology Security Hub [1] in South Korea. The Hub is a public-private partnership supported by governmental agencies in order to support the sharing of knowledge, experience and expertise across SMEs.

The data for each SME are stored in a separate file. There are 4643 entries overall. Each entry, expressed as a row, has the following metadata:

- `Date and Time of Occurrence`: this is a value representing the date and time of the incident's occurrence.
- `End Device`: this is a value representing the name of the end device affected in the incident.
- `Malicious Code`: this is a value representing the name of the malicious code detected in the incident.
- `Response`: this is a value representing the response action that was applied to the malicious code.

[1] http://www.kaits.or.kr/index.do

- `Type of Malware`: this is a value representing the type of the malware (malicious code) detected in the incident.
- `Detail`: this is a free text value to describe any other detail about the incident.

An example entry from this dataset is shown below:

```
(14/02/2017 11:58, rc0208-pc,
Gen:Variant.Mikey.57034, deleted, virus,
C:\Users\RC0208\AppData\Local\Temp\is-ANFS3.
tmp\SetupG.exe)
```

3.2 Research Problems

Our research in this paper is aimed at investigating two kinds of problems, which we take a classification approach to solving them:

- The first problem is *forward-looking* to attempt to predict future aspects of Cyber security incidents. More specifically, how an organization can gain the ability to predict response actions to future Cyber security incidents involving malware. We consider two questions here: *a) how to predict a response action from the name of malicious code,* and *b) how to predict a response action from the type of malware involved in the incident.*
- The second problem is *backward-looking* to investigate, for example, as part of a digital forensics process, properties of current incidents. More specifically, how an organization can utilize its knowledge of the response actions in guiding digital forensics analysis to determine the type of malware or the name of the malicious code to investigate. We consider here two questions: *c) how to identify the type of the malware based on the name of malicious code,* and *d) how to identify the type of the malware based on the response action.*

3.3 Data Anaylsis and Classification Model

In this section, we describe the processes that have been taken for the analysis and classification of the dataset. A model has been developed using knime software [2], which makes use of the most used features in text mining such as n-gram, Bag-of-Words, Snowball Stemmer and stop words remover. This model consists of three main parts (1) Data analysis and Pre-processing, (2) Features Extraction and (3) Classification. The phases of the model are described below:

Phase 1: Data Analysis and Pre-processing
The main objective of pre-processing is to clean the data from noise in which this will help to improve the accuracy of the results by reducing the errors in the data. This is done by removing special characters and stop words such as ”a” and ”the”, punctuation marks such as question and exclamation marks, and numbers. In addition, all terms are converted to lowercase. The resulting terms are used to generate the n-gram features.

Phase 2: Features Extraction: Feature extraction help in the analysis and classification and also in improving accuracy. The most commonly used features in text mining are n-gram and bag-of-words. The model makes use of ”bigram” which is an n-gram for $n = 2$, every two adjacent words create a bigram e.g. ”malware detection”. In this phase, a bag-of-words is created containing all words (bigram). This bag-of-words is filtered based on the minimum frequency in which terms that occur in less than the minimum frequency are filtered out and not used as features using term frequency (TF) method.

Phase 3: Classification: The classification phase is executed using machine learning algorithms such as Naïve Bayes (NB) and Support Vector Machine (SVM). In this phase, the n-gram features predictive models are built, tested and compared. The dataset is split into training and test set. The training dataset is used for building the model, and the test dataset is used to evaluate the performance of the model.

4 Experimental Study and Results

The objective of the experimental study is to explore the ability of machine learning classifiers to distinguish between (1) the different types of response based on the given malicious code, (2) the different types of response based on the given malware, (3) the different types of malware according to the malicious code and (4) the different types of malware based on the different responses. Two machine learning algorithms were used for the classification process, which are Naïve Bayes (NB) and Support Vector Machine (SVM).

We have used the dataset that was selected from five different companies provided by the KAITS Industrial Technology Security Hub in South Korea. The distribution of the data is given in Table 1. As mentioned in Section 3, all the incidents of the five companies were collected by a centralised hub and the concatenated data were used for the experiment with the objective of evaluating the performance of the classifiers in distinguishing between different types

[2] `https://www.knime.com/`

of incidents and investigating how different data collected from multiple companies can help in improving the classification accuracy.

Table 1: Data distribution

Company Name	**Total Number of Incidents**
Company 1(DF)	932
Company 2(MT)	633
Company 3(SE)	923
Company 4(EP)	448
Company 5(MS)	1707
Total	4643

4.1 Performance Evaluation Metrics

To assess the performance of the machine learning classifiers performance indicators, such as accuracy, precision, recall, and F-measure (Chinchor, 1992), are calculated as shown in the following formulæ.

$$Accuracy = \frac{\#\ of\ correct\ predictions\ (TP{+}TN)}{\#\ of\ predictions\ (TP{+}TN{+}FP{+}FN)}$$

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- True Positive (TP): An instance that is positive and is classified correctly as positive.
- True Negative (TN): An instance that is negative and is classified correctly as negative.
- False Positive (FP): An instance that is negative but is classified wrongly as positive.
- False Negative (FN): An instance that is positive but is classified wrongly as negative.

4.2 Results

In this section, we present and analyse the results of the machine learning algorithms for the four different problems that were proposed.

4.2.1 Problem 1: Identifying the different types of response based on the given malicious code

Table 2 presents the classification performance details of the SVM and Naïve Bayes classifiers in identifying the different types of response based on the given malicious code. SVM has achieved an accuracy of 81% while NB achieved an accuracy of 73%. Comparing the performance of both classifiers, Both SVM and NB had a zero precision, recall and f-measure for response types ”Recovered” and ”Name Changed”. While both classifiers had a 100% precision, recall, and f-measure for response type ”Blocked”. For the rest of the types of response, NB could identify response type ”None” and achieved a recall of 50% while SVM had a zero precision, recall, and f-measure for this type. In addition, for the response type ”Segregated” and ”Deleted” SVM has the highest recall and f-measure while NB has the highest precision. Furthermore, for the response type ”Not defined” SVM has the highest precision while NB has the highest recall and f-measure.

Table 2: Performance of the classifiers in identifying the different types of response based on the malicious code.

	SVM			*NB*		
Accuracy:	81%			73%		
Class:	P	R	F	P	R	F
None	0.00	0.00	0.00	**0.11**	**0.50**	**0.18**
Recovered	0.00	0.00	0.00	0.00	0.00	0.00
Segregated	0.90	**0.72**	**0.80**	**0.91**	0.55	0.68
Deleted	0.61	**0.92**	**0.73**	**0.64**	0.80	0.71
Not defined	**1.00**	0.84	**0.91**	0.79	**0.89**	0.84
Blocked	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Name Changed	0.00	0.00	0.00	0.00	0.00	0.00

4.2.2 Problem 2: Identifying the different types of response based on the given malware

Table 3 presents the classification performance details of the SVM and Naïve Bayes classifiers in identifying the different types of response based on the given malware. SVM and NB achieved an accuracy of 73% and 72.8% respectively. Comparing the performance of both classifiers, both classifiers failed to identify the response type ”None, ”Recovered” and ”Name Changed”. In addition, both classifiers had similar precision, recall, and f-measure for response type ”Segregated and ”Blocked” and nearly similar precision, recall and f-measure for response types ”Not defined” and Deleted”.

4.2.3 Problem 3: Identifying the different types of malware according to the malicious code

Table 4 presents the classification performance details of the SVM and Naïve Bayes classifiers in identifying the different types of malware according to the malicious code. SVM achieved an accuracy of 77% while NB achieved an accuracy of 70%. Both

Table 3: Performance of the classifiers in identifying the different types of response based on malware.

	SVM			*NB*		
Accuracy:	73%			72.8%		
Class:	P	R	F	P	R	F
None	0.00	0.00	0.00	0.00	0.00	0.00
Recovered	0.00	0.00	0.00	0.00	0.00	0.00
Segregated	**0.83**	**0.13**	**0.23**	**0.83**	**0.13**	**0.23**
Deleted	**0.97**	**0.88**	**0.92**	**0.97**	0.87	**0.92**
Not defined	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.99
Blocked	**0.43**	**1.00**	**0.60**	**0.43**	**1.00**	**0.60**
Name Changed	0.00	0.00	0.00	0.00	0.00	0.00

classifiers could not identify malware type ”Web contents” and NB achieved very low precision, recall, and f-measure for malware type ”Downloaded file”, while SVM failed to identify this type. In addition, SVM has the highest recall of 100% for malware type ”Email attachment”, while NB has the highest precision and f-measure. Furthermore, SVM achieved the highest precision and f-measure for malware type ”Spyware”, while NB has the highest recall. For the malware type ”Virus”, SVM has the highest recall and f-measure while NB has the highest precision.

Table 4: Performance of the classifiers in identifying the different types of malware according to the malicious code

	SVM			*NB*		
Accuracy:	77%			70%		
Class:	P	R	F	P	R	F
Email attachment	0.53	**1.00**	0.69	**0.57**	0.89	**0.70**
Spyware	**1.00**	0.86	**0.92**	0.79	**0.89**	0.84
Virus	0.90	**0.75**	**0.82**	**0.98**	0.54	0.69
Downloaded file	0.00	0.00	0.00	**0.25**	**0.39**	**0.30**
Web Contents	0.00	0.00	0.00	0.00	0.00	0.00

4.2.4 Problem 4: Identifying the different types of malware based on the different responses

Table 5 presents the classification performance details of the SVM and Naïve Bayes classifiers in identifying the different types of malware based on the different responses. SVM and NB achieved similar accuracy of 92%. In addition, SVM and NB failed to identify malware types ”Downloaded file” and ”Web contents”, while both classifiers had similar precision, recall, and f-measure for malware types ”Email attachment, ”Spyware” and ”Virus”.

4.3 Discussion

In this research, many factors have affected the identification and classification process using machine learning. We will discuss the overall results below.

Table 5: Performance of the classifiers in identifying the different types of malware based on the different responses

	SVM			*NB*		
Accuracy:	92%			92%		
Class:	P	R	F	P	R	F
Email attachment	**0.99**	**0.88**	**0.93**	**0.99**	**0.88**	**0.93**
Spyware	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Virus	**0.93**	**0.89**	**0.91**	**0.93**	**0.89**	**0.91**
Downloaded file	0.00	0.00	0.00	0.00	0.00	0.00
Web Contents	0.00	0.00	0.00	0.00	0.00	0.00

The overall results for the identification of the different types of responses based on the given malicious code indicated that SVM was the best classifier, but NB performed better due to the fact that it could identify five different type of the responses while SVM only identify four types. While, the overall results for the identification of the different types of response based on the given malware showed that SVM and NB had nearly similar accuracy and precision, recall, and f-measure for most response types in which SVM and NB could identify four different types of the response. The poor performance from the classifiers was due to the fact that some types of malware were assigned by the companies to multiple response types (e.g segregated and name changed are assigned to malware type virus) and the high and low frequency of some types affected the classification this because of the imbalance of the categories.

Furthermore, the overall results for the identification of the different types of malware according to the malicious code indicated that SVM was the best classifier but NB performed better due to the fact that it could identify five different type of the malware while SVM only identify three types. While, the overall results or the identification of the different types of malware based on the different responses showed that SVM and NB performed well and had similar precision, recall, and f-measure for most response types in which SVM and NB could identify three different type of malware only.

Following from the discussion above, we observe the following about the overall results:

(1) The classification accuracy was affected by the imbalance of the (dataset) categories and the inconsistency of the categories that were used across the five companies (e.g type of responses and malware types) as shown in Tables 6 and 7. This problem could not be handled due to the fact that we are trying to solve real case problems and applying an algorithm to handle class imbalance will result in altering the given information.

(2) The classifiers performance was affected by the multi-labeling of some of the categories.

Table 6: Type of responses distribution for five companies shows the imbalance of the data which affect the classification performance of the classifiers

Type of Responses	Co1 (DF)	Co2 (MT)	Co3 (SE)	Co4 (EP)	Co5 (MS)	**Total**
Blocked	166	89	411	231	5	**902**
Deleted	39	98	172	91	1153	**1553**
Name Changed	2	2	10	4	2	**20**
None	65	3	43	9	4	**124**
Segregated	153	288	206	61	201	**909**
Not defined	0	153	81	28	326	**588**
Recovered	42	0	0	24	10	**76**

Table 7: Type of malware distribution for five companies shows the imbalance of the data which affect the classification performance of the classifiers

Type of Malware	Co1 (DF)	Co2 (MT)	Co3 (SE)	Co4 (EP)	Co5 (MS)	**Total**
Email attachment	372	0	117	99	1140	**1728**
Spyware	93	153	81	27	326	**680**
Virus	467	480	725	322	87	**2081**
Downloaded file	0	0	0	0	149	**149**
Web Contents	0	0	0	0	5	**5**

(3) Malware types could actually be used for the identification of malicious code even-though from a security point of view there is no explicit research showing that this is possible.

(4) Problem 2 was the most difficult problem to predict in which the classifiers' performance was the lowest in this case.

(5) SVM is more suitable for the detection of types of response using the malicious code, the detection of types of response using malware. In addition to the detection of malware based on the malicious code. While SVM and NB could be used for the detection of the types of malware using the different types of response as their performance results are similar.

5 Conclusion and Future Work

In this paper, a dataset collected from five SMEs in South Korea was analysed to demonstrate how a centralised hub may collect experience from multiple organisations in order to train a single classifier that can predict features of future Cyber security incidents. Moreover, a model has been developed using text mining methods. Using machine learning algorithms for the classification of these incidents and their response actions, experimental results showed good performance of the classifiers in predicting different types of response and malware.

As future work, we are planning to test other Cyber security datasets and evaluate the performance of different machine learning algorithms. In addition, we aim to investigate how handling class imbalance can help to improve the classification accuracy.

ACKNOWLEDGEMENTS

This work was supported by the University of Portsmouth Themes Research and Innovation Fund (TRIF) 2018 - grant number 41611.

REFERENCES

Abou-Assaleh, T., Cercone, N., Keselj, V., and Sweidan, R. (2004). N-gram-based detection of new malicious code. In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, volume 2, pages 41–42. IEEE.

Bahraminikoo, P., Yeganeh, M., and Babu, G. (2012). Utilization data mining to detect spyware. *IOSR Journal of Computer Engineering (IOSRJCE)*, 4(3):01–04.

Bartal, A., Sasson, E., and Ravid, G. (2009). Predicting links in social networks using text mining and sna. In *2009 International Conference on Advances in Social Network Analysis and Mining*, pages 131–136.

Center for Applied Internet Data Analysis. CAIDA Data. Last accessed: 05.01.2017.

CERT Coordination Center. CERT Knowledgebase. Last accessed: 14.01.2017.

Chinchor, N. (1992). Muc-4 evaluation metrics. In *Proceedings of the 4th Conference on Message Understanding*, MUC4 '92, pages 22–29, Stroudsburg, PA, USA. Association for Computational Linguistics.

Decherchi, S., Tacconi, S., Redi, J., Leoncini, A., Sangiacomo, F., and Zunino, R. (2009). Text clustering for digital forensics analysis. In Herrero, Á., Gastaldo, P., Zunino, R., and Corchado, E., editors, *Computational Intelligence in Security for Information Systems*, pages 29–36, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ding, Y., Yuan, X., Tang, K., Xiao, X., and Zhang, Y. (2013). A fast malware detection algorithm based on objective-oriented association mining. *computers & security*, 39:315–324.

Elovici, Y., Shabtai, A., Moskovitch, R., Tahan, G., and Glezer, C. (2007). Applying machine learning techniques for detection of malicious code in network traffic. In *Annual Conference on Artificial Intelligence*, pages 44–50. Springer.

Fan, C.-I., Hsiao, H.-W., Chou, C.-H., and Tseng, Y.-F. (2015). Malware detection systems based on api log data mining. In *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, volume 3, pages 255–260. IEEE.

Fan, Y., Ye, Y., and Chen, L. (2016). Malicious sequential pattern mining for automatic malware detection. *Expert Systems with Applications*, 52:16–25.

Ghazinour, K., Sokolova, M., and Matwin, S. (2013). Detecting health-related privacy leaks in social networks using text mining tools. In Zaïane, O. R. and Zilles, S., editors, *Advances in Artificial Intelligence*, pages 25–39, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hellal, A. and Romdhane, L. B. (2016). Minimal contrast frequent pattern mining for malware detection. *Computers & Security*, 62:19–32.

Hicks, C., Beebe, N., and Haliscak, B. (2016). Extending web mining to digital forensics text mining. In *AMCIS 2016: Surfing the IT Innovation Wave - 22nd Americas Conference on Information Systems*. Association for Information Systems.

Hoo, K. J. S. (2000). How Much is Enough? A Risk-Management Approach to Computer Security.

Hou, Y.-T., Chang, Y., Chen, T., Laih, C.-S., and Chen, C.-M. (2010). Malicious web content detection by machine learning. *Expert Systems with Applications*, 37(1):55–60.

Inkpen, D. (2016). *Text Mining in Social Media for Security Threats*, pages 491–517. Springer International Publishing, Cham.

Kakavand, M., Mustapha, N., Mustapha, A., and Abdullah, M. T. (2015). A text mining-based anomaly detection model in network security. *Global Journal of Computer Science and Technology*.

Klimt, B. and Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In Boulicaut, J.-F., Esposito, F., Giannotti, F., and Pedreschi, D., editors, *Machine Learning: ECML 2004*, pages 217–226, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lu, Y.-B., Din, S.-C., Zheng, C.-F., and Gao, B.-J. (2010). Using multi-feature and classifier ensembles to improve malware detection. *Journal of CCIT*, 39(2):57–72.

Mike Sconzo. SecRepo.com - Samples of Security Related Data. Last accessed: 05.01.2017.

Norouzi, M., Souri, A., and Samad Zamini, M. (2016). A data mining classification approach for behavioral malware detection. *Journal of Computer Networks and Communications*, 2016:1.

Ojoawo, A. O., Fagbolu, O. O., Olaniyan, A. S., and Sonubi, T. A. (2014). Data leak protection using text mining and social network analysis. *International Journal of Engineering Research and Development*, 10(12):14 – 22.

Parker, D. B. (1998). *Fighting Computer Crime: A New Framework for Protecting Information*. John Wiley & Sons, Inc., New York, NY, USA.

Rieck, K., Trinius, P., Willems, C., and Holz, T. (2011). Automatic analysis of malware behavior using machine learning. *Journal of Computer Security*, 19(4):639–668.

Schultz, M. G., Eskin, E., Zadok, F., and Stolfo, S. J. (2001). Data mining methods for detection of new malicious executables. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*, pages 38–49. IEEE.

Shabtai, A., Moskovitch, R., Feher, C., Dolev, S., and Elovici, Y. (2012). Detecting unknown malicious code by applying classification techniques on opcode patterns. *Security Informatics*, 1(1):1.

Suh-Lee, C., Jo, J.-Y., and Kim, Y. (2016). Text mining for security threat detection discovering hidden information in unstructured log messages. In *Communications and Network Security (CNS), 2016 IEEE Conference on*, pages 252–260. IEEE.

VERIZON. VERIS Community Database. Last accessed: 21.11.2016.

Wang, T.-Y., Horng, S.-J., Su, M.-Y., Wu, C.-H., Wang, P.-C., and Su, W.-Z. (2006). A surveillance spyware detection system based on data mining methods. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 3236–3241. IEEE.

Xylogiannopoulos, K., Karampelas, P., and Alhajj, R. (2017). Text mining in unclean, noisy or scrambled datasets for digital forensics analytics. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 76–83.

Zhang, B., Yin, J., Hao, J., Zhang, D., and Wang, S. (2007). Malicious codes detection based on ensemble learning. In *International Conference on Autonomic and Trusted Computing*, pages 468–477. Springer.