

Instant Message Classification in Finnish Cyber Security Themed Free-Form Discussion

**Samir Puuska, Matti J. Kortelainen, Viljami Venekoski and
Jouko Vankka**

*Department of Military Technology
National Defence University
Helsinki, Finland*

ABSTRACT

Instant messaging enables rapid collaboration between professionals during cyber security incidents. However, monitoring discussion manually becomes challenging as the number of communication channels increases. Failure to identify relevant information from the free-form instant messages may lead to reduced situational awareness. In this paper, the problem was approached by developing a framework for classification of instant message topics of cyber security-themed discussion in Finnish. The program utilizes open source software components in morphological analysis, and subsequently converts the messages into Bag-of-Words representations before classifying them into predetermined incident categories. We compared Support vector machines (SVM), multinomial naïve Bayes (MNB), complement naïve Bayes classification methods (CNB) with five-fold cross-validation. A combination of SVM and CNB achieved classification accuracy of over 85%, while multiclass SVM achieved 87% accuracy. The implemented program recognizes cyber security related messages in IRC chat rooms and categorizes them accordingly.

Keywords: natural language processing, machine learning, language technology, text classification, classifiers, Finnish, instant messaging, cyber security.

1 INTRODUCTION

Instant messaging has become a common method of real-time collaboration between experts and professionals within and across organizational boundaries. Cyber security and incident response tasks require rapid communication between technical experts and management. External and internal attacks against information technology systems may have critical consequences, such as loss of confidential information, financial losses or damage to other organizational infrastructure and reputation (Jouini, Rabai, & Aissa, 2014).

On a national scale, numerous communication channels may be needed for the multitude of experts to discuss and analyze cyber security incidents. For both experts and operators, due to the cognitive demands of maintaining vigilance, constant manual real-time monitoring of several online chat rooms becomes time-consuming and challenging, and may lead to reduced situational awareness. It has been shown that emphasizing relevant content from chat messages with visual cues can assist operators in detecting relevant messages (Catanzaro, Risser, Gwynne, & Manes, 2006; Satterfield, Finomore, Castle, & Warm, 2011). Therefore, a program capable of automatic detection of chat message topics and subsequent message highlighting would make the monitoring process less exhausting. Analysis of instant message characteristics and content has been the subject of several studies (Dong, Hui, & He, 2006; Forsyth & Martell, 2007; Adams & Martell, 2008; Özyurt & Köse, 2010; Ramachandran, et al., 2011). Although suggested methods have yielded promising results, the majority of them have been tested mainly using English. Recovering the base form of words from written text is a more simple process in English than in morphologically complex languages such as Finnish where profound linguistic analysis is needed (Huvelin, et al., 2013).

The aim of this work is to develop a program for classification of cyber security related instant messages in Finnish. The task is approached by using the well-known Bag-of-Words vector space model, where the messages are regarded as combinations of words where inter-word grammatical dependencies are ignored (Adams & Martell, 2008; Ramachandran, et al., 2011).

2 MATERIAL AND METHODS

Data

Our data consisted of chat messages written during a five-day cyber security exercise session organized by the Finnish Defence Forces and other

authorities (Halminen, 2015; Jyväskylä Security Technology, 2015). The messages were written in eight separate chat rooms: Seven were used by individual teams participating in the exercise, and one was used by the moderators and admins. The team members used the chat to collaborate and discuss various cybersecurity incidents which occurred during the exercise. These incidents included Denial of Service (DoS) attacks, phishing attempts, unauthorized mapping scans, installation and execution of malicious software and even physical intrusions into IT systems.

A total of 3060 messages were written during the exercise. To obtain the ground truth about the topics, the messages were manually inspected for any content that could imply a possible cyber security incident or anomaly. When such a message was found, it was further assigned a class according to the Multinational Alliance for Collaborative Cyber Situational Awareness (MACCSA) incident categorization (Multinational Alliance for Collaborative Cyber Situational Awareness, 2013). Messages that implied no ongoing or possible cybersecurity anomaly were assigned the class tag ‘0’. The class distribution of messages is presented in Table I.

Table 1: Class Distribution of Messages.

Class	Class Name	Number of Messages (Percentage of All Messages)
0	NOT INCIDENT	2383 (77.9%)
1	Unauthorized Access	69 (2.3%)
2	Denial of Service	122 (4.0%)
3	Malicious Code	82 (2.7%)
4	Improper Usage	15 (0.5%)
5	Scans, Probes, Attempted Access	64 (2.1%)
6	Investigation	325 (10.6%)

Names of classes 1–6 from (Multinational Alliance for Collaborative Cyber Situational Awareness, 2013, pp. 68–75)

Preprocessing and Morphological Analysis

Message preprocessing included the removal of special characters, user names, numbers, and time tags. Standard punctuation characters were not removed at this stage. Each message was processed with an open source Finnish dependency parser (FDP; Haverinen, et al., 2013) which produced the base form for each word in the messages (e.g., the first infinitive short form for verbs, the nominative singular form for nouns).

The FDP uses finite state transducers, which deduce the base form and the morphological structure (Beesley & Karttunen, 2003) for each word. The default FDP transducers lacked numerous IT terms and proper nouns for organizations that were present in the chat messages, such as ‘*spammaaja*’

[a spammer] or ‘*konffata*’ [to configure]. We generated new transducers using open source OMorFi software (Pirinen, 2014) to replace the default transducers in FDP. Adding the missing words into OMorFi database and generating new transducers allows the FDP toolchain to perform morphological analysis also on new words. A total of 401 terms were added to the database and assigned a semantic tag ‘Cyber’ which appears in the morphological analysis result.

A single inflected word can have multiple morphological interpretations. The FDP uses a scoring system based on a precomputed statistical model to decide which morphological interpretation is most likely to be the correct one. For example, ‘*virpi*’ is a Finnish noun, and ‘*VIRVE*’ is the name of a communication network used by Finnish authorities. When either is inflected in the genitive case, their inflected form is identical (‘*virven*’), and here the FDP scoring system preferred the relatively rare noun ‘*virpi*’ to the common ‘*virve*’. We modified the default FDP scoring to prefer base forms which had the ‘Cyber’ tag to address this problem.

The analyzed sentences were further processed by including only the words tagged as nouns, verbs, proper nouns, and adjectives by a Part-of-Speech tagger in the FDP pipeline. Furthermore, a stopword list was used to remove the remaining common words. One-character words and remaining punctuation characters were also removed.

Building of Feature Vectors

A global dictionary was constructed from word forms appearing in the processed messages. Cyber security – related terms which were obviously synonyms, such as ‘*palvelunestohyökkäys*’ [Denial of Service attack] and its acronym, ‘*DoS*’, were associated with the same dictionary entry. In total, the global dictionary contained 2378 entries. Based on the dictionary, a feature vector representation for each message was built by assigning term frequency-inverse document frequency (TF-IDF) weights to the vector elements (Yang & Chute, 1994). The feature vectors were also normalized to unit length. For the rest of the paper, we use the term ‘message’ interchangeably with ‘feature vector’.

Classification

Support Vector Machines: Support Vector Machine (SVM) classifier separates two classes in the feature space with the widest possible margin (Cortes & Vapnik, 1995). In multi-class problems, the classification decision of a multi-class SVM (MSVM) is based on outputs of multiple binary classifiers. For example, in one-versus-all winner-takes-all approach,

one binary SVM classifier is constructed for each class to determine whether the message belongs to the class or its complement. The class whose classifier produces the highest output function value is chosen as the correct class (Duan & Keerthi, 2005).

Multinomial Naïve Bayes: In multinomial naive Bayes (MNB) classification, it is assumed that the elements of the feature vector have been created by sampling from a multinomial distribution. In MNB classification, the objective is to choose a class that maximizes the posterior probability of the class when the message is given (McCallum & Kamal, 1998). The complement naive Bayes (CNB) approach of Rennie et al. is a modification of MNB classifier. In CNB, the objective is to choose the class whose complement suits the given message the worst (Rennie, Shih, Teevan, & Karger, 2003).

Two-step Classification: Since the majority of our data consisted of class 0 chat messages (Table 1), a two-step classification strategy was justified. This strategy is a simple ensemble method (Dietterich, 2000): in the first step, the messages in classes 1–6 are pooled into a single class, denoted by $\Lambda = \{1, 2, 3, 4, 5, 6\}$, and a binary classifier is used to separate them from class 0 messages. In the second step, the separated class Λ messages are further classified in one of classes 1–6 with a multi-class classifier trained exclusively with messages in classes 1–6.

We hypothesized that compared to the conventional one-step classification strategy, the two-step strategy is less likely to misclassify a message in class 0 when the message's contents imply detection of cyber security incidents. Figure 1 illustrates the overall pipeline involving the two-step classification procedure.

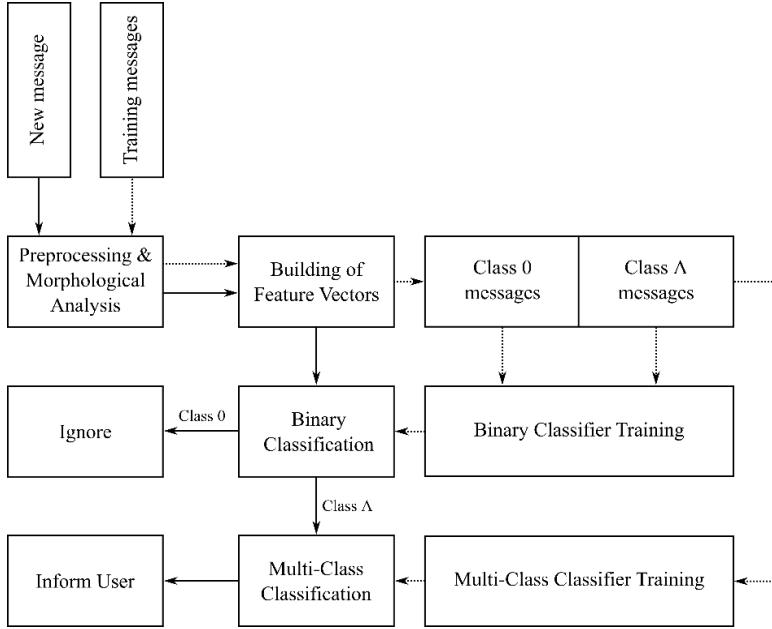


Figure 1: An overview of the two-step classification pipeline. In the first step, the binary classifier checks whether the new message contains cyber security incident related content (class A). Then, the multiclass classifier – trained exclusively with class A messages – assigns the new message a class according to the MACCSA categorization (Table 1).

Performance Evaluation

Evaluation Metrics: Based on the work of van Rijsbergen (1979), and Özgür, Özgür and Güngör (2005), the following metrics were calculated for evaluation purposes:

$$\text{Recall}_{\mu} = \frac{\sum_{c=0}^6 \text{TP}(c)}{\sum_{c=0}^6 \text{TP}(c) + \sum_{c=0}^6 \text{FN}(c)}. \quad (1)$$

$$\text{Recall}(\Lambda) = \frac{\text{TP}(\Lambda)}{\text{TP}(\Lambda) + \text{FN}(\Lambda)} \quad (2)$$

$$\text{FNR}(0) = \frac{\text{FN}(0)}{\text{TP}(0) + \text{FN}(0)} \quad (3)$$

In Eqs. (1) – (3), for class c , $\text{TP}(c)$ denotes true positives and $\text{FN}(c)$ false negatives. Recall_{μ} denotes the micro-averaged recall, $\text{Recall}(\Lambda)$ the recall of

class Λ messages and $\text{FNR}(0)$ the false negative rate for class 0. In addition, we defined an additional metric that we considered useful for our task: cyber incident classification accuracy (CICA):

$$\text{CICA} = \frac{\sum_{c=1}^6 \text{TP}(c)}{\sum_{c=1}^6 \text{TP}(c) + \sum_{c=1}^6 \text{FN}(c)}. \quad (4)$$

CICA resembles micro-averaged recall with the exception that it does not consider the classification accuracy of class 0 messages. CICA measures the fraction of messages in classes 1–6 that were assigned to the correct class. $\text{Recall}(\Lambda)$, on the other hand, measures the fraction of class Λ messages that were assigned to any of the classes 1–6, regardless of whether the class was actually correct. $\text{FNR}(0)$ serves as an estimate for false alarms, i.e., that chat user is notified of cyber security incident when there is none.

Classifiers: In addition to MSVM, MNB and CNB classifiers, six other methods based on two-step classification strategy were tested. Three methods used SVM in the first step binary classification and the other three used MNB. The second step multi-class classification was done with MSVM, MNB or CNB. We denote the two-step methods with syntax ‘binary classifier+multi-class classifier’, i.e., MNB+MSVM method uses MNB in the first step and MSVM in the second step.

Implementation of Tests: The performance evaluation of different classification methods was conducted with MATLAB R2013a (The MathWorks, Inc. 2013). For implementation of the linear kernel MSVM classifier, one-versus-all method with the winner-takes-all strategy was adopted (Duan & Keerthi, 2005). MATLAB’s built-in algorithm was used to train the binary classifiers for individual classes. Implementation of MNB and CNB classifiers followed the work of Rennie, Shih, Teevan and Karger (2003). Five-fold cross-validation was done for each method, computing the metrics (1) – (4) on each validation round (Refaeilzadeh, Tang, & Liu, 2009). Statistical tests comparing the methods were performed with SPSS Statistics 22 (IBM Corp, 2013). Normality of data was tested with Shapiro-Wilk test with 0.05 as the significance level, followed by pairwise Least Significant Difference t-tests with 0.05 as the significance level. Post hoc Bonferroni adjustment to the p values was done before the interpretation.

Software Implementation

Based on the performance evaluation results, an instant message classification program was devised. An IRC bot was implemented in Python using features from OMorFi, FDP, and the scikit-learn library (Pedragosa, et al., 2011). The bot was trained using all 3060 messages as training data. The

MSVM classification method was chosen due to satisfactory results (Table 2) and least complex implementation. In addition to text classification, the bot was programmed to recognize IP addresses and hashtags (Figure 1).

As a sanity check, the bot was tested by writing messages consisting of various Finnish texts from Project Gutenberg (Hart), as well as messages mimicking the cyber security incident reports written during the cyber security exercise (Halminen, 2015; Jyväskylä Security Technology, 2015).

3 RESULTS

Classification Performance

The results for multi-class classification are presented in Table 2. The results are expressed as averages over five cross-validation rounds.

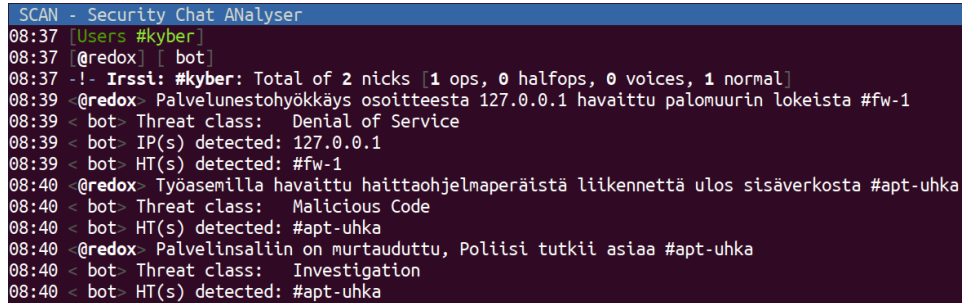
Table 2: Classification Results

Method	Recall _μ	Recall(Λ)	FNR(0)	CICA
CNB	0.767	0.833	0.223	0.733
MNB	0.735	0.762	0.220	0.657
MSVM	0.876	0.755	0.062	0.656
MNB+CNB	0.794	0.861	0.188	0.733
MNB+MNB	0.789	0.861	0.188	0.711
MNB+MSVM	0.779	0.861	0.188	0.665
SVM+CNB	0.860	0.758	0.081	0.651
SVM+MNB	0.858	0.758	0.081	0.640
SVM+MSVM	0.850	0.758	0.081	0.606

MSVM achieved the highest Recall_μ, but the differences were not statistically significant compared to the two-step methods using SVM in the first step ($p>0.99$). Two-step methods using MNB in the first step achieved the highest Recall(Λ), but the differences were not statistically significant compared to CNB ($p>0.99$). MSVM achieved the lowest FNR(0), but the differences were not statistically significant compared to two-step methods using SVM in the first step ($p>0.99$). CNB achieved the highest CICA value, but only the differences compared to the SVM+MSVM and SVM+MNB methods were statistically significant ($p<0.05$). Overall, no single method proved to be evidently superior in comparison to the rest.

Practical Implementation

The program was able to recognize novel cyber incident messages and assigned them to the correct class (Figure 1). In addition, the program correctly assigned the majority of the Project Gutenberg messages to class 0. The functionality was achieved at real-time speed.



```

SCAN - Security Chat ANalyser
08:37 [Users #kyber]
08:37 [redox] [ bot]
08:37 -!- Irssi: #kyber: Total of 2 nicks [1 ops, 0 halfops, 0 voices, 1 normal]
08:39 <@redox> Palvelunestohyökkäys osoitteesta 127.0.0.1 havaittu palomuurin lokeista #fw-1
08:39 < bot> Threat class: Denial of Service
08:39 < bot> IP(s) detected: 127.0.0.1
08:39 < bot> HT(s) detected: #fw-1
08:40 <@redox> Työasemilla havaittu haittaohjelmaperäistä liikennettä ulos sisäverkosta #apt-uhka
08:40 < bot> Threat class: Malicious Code
08:40 < bot> HT(s) detected: #apt-uhka
08:40 <@redox> Palvelinsaliin on murtauduttu, Poliisi tutkii asiaa #apt-uhka
08:40 < bot> Threat class: Investigation
08:40 < bot> HT(s) detected: #apt-uhka

```

Figure 2: A picture of the bot in operation.

The translations of the messages by the user 'redox' are as follows:

[Denial of service attack from address 127.0.0.1 detected from firewall log files #FW1]

[Malware-based traffic out of intranet detected on the workstations #apt-threat],

[The server hall has been breached, Police is investigating the matter #apt-threat].

4 DISCUSSION

In this work, a framework for automatic classification of free-form cyber security-themed instant messages was created for Finnish. The framework utilizes open source components for morphological analysis of Finnish which were modified to enable analysis of several IT-related Finnish terms. Conventional machine learning methods were tested to evaluate their message classification performance. Finally, based on the results, a MSVM message classifier bot was implemented using Python.

MSVM and the two-step classifiers using SVM in the first step achieved the highest Recall_μ scores in cyber security message classification, and can thus be considered the most accurate methods. Their average Recall(Δ) values of approximately 75% can be considered sufficient. In the study of Catanzaro et al. (2006), highlighting 75% of the messages conveying critical information resulted in a significant increase in critical event detection rate among the test participants. This implies that SVM-based classification tool could be utilized to assist in discussion monitoring. However, the false

alarm rate is still quite high (Table 2), which could cause some distrust towards the classification application.

Classifiers based on supervised learning suffer from limited and biased training datasets, such as ours (e.g. only 15 messages from class 4). Abundance of class 0 messages may guide the classifiers to prefer class 0 in the classification decision. In addition, the current framework ignores words not present in the training material. Utilizing neural network –based language models and resulting semantic representations of words could allow for estimating the senses of novel out-of-vocabulary words. Further, neural language models (e.g. (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013)) could enhance classification and allow extracting more sophisticated information from the cyber security messages. However, the use of such methods would require much more extensive collection of training material than the corpus utilized in our research. Nonetheless, we speculate that extending a domain-general neural language model with domain-specific cyber security corpus could make the neural models usable even with sparse target data. Therefore, a greater collection of cyber security terms and sample training messages would enhance the performance of the classifiers. A rudimentary spell-check would further benefit the language processing.

5 REFERENCES

- Adams, P. H., & Martell, C. H. (2008). Topic detection and extraction in chat. *2008 IEEE International Conference on Semantic Computing* (pp. 581-588). IEEE.
- Beesley, K. R., & Karttunen, L. (2003). *Finite State Morphology*. CSLI Publications.
- Catanzaro, J. M., Risser, M. R., Gwynne, J. W., & Manes, D. I. (2006). Military situation awareness: Facilitating critical event detection in chat. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 50, pp. 560-564. SAGE Publications.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems* (pp. 1-15). Berlin, Heidelberg: Springer. doi:10.1007/3-540-45014-9_1
- Dong, H., Hui, S. C., & He, Y. (2006). Structural analysis of chat messages for topic detection. *Online Information Review*, 30(5), 496-516.
- Duan, K.-B., & Keerthi, S. S. (2005). Which is the best multiclass SVM method? An empirical study. In N. C. Oza, R. Polikar, J. Kittler, & F. Roli (Eds.), *6th International Workshop on Multiple Classifier Systems* (pp. 278-285). Berlin, Heidelberg: Springer. doi:10.1007/11494683_28
- Erkan, H., Hassan, A., Diao, Q., & Radev, D. R. (2011). *Improved nearest neighbor methods for text classification*. Technical Report CSE-TR-576-11, University of Michigan. Department of Electrical Engineering and Computer Science.

- Forsyth, E. N., & Martell, C. H. (2007). Lexical and discourse analysis of online chat dialog. *International Conference on Semantic Computing* (pp. 19-26). IEEE.
- Halminen, L. (2015). *Kyberharjoitus alkaa Jyväskylässä [Cyber security exercise begins in Jyväskylä]*. Retrieved 11 27, 2015, from <http://www.hs.fi/kotimaa/a1431831015084>
- Hart, M. (n.d.). Project Gutenberg. *Project Gutenberg*. Retrieved December 16, 2015, from http://www.gutenberg.org/wiki/Main_Page
- Haverinen, K., Nyblom, J., Viljanen, T., Laippala, V., Kohonen, S., Missilä, A., . . . Ginter, F. (2013). Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, 48(3), 493-531.
- Huovelin, J., Gross, O., Solin, O., Linden, K., Maisala, S., Oittinen, T., . . . Silfverberg, M. (2013). Software Newsroom -- an approach to automation of news search and editing. *Journal of Print and Media Technology Research*, 2(3), 141-156.
- IBM Corp. (2013). IBM SPSS Statistics for Windows, Version 22.0. Armonk, NY: IBM Corp.
- Jouini, M., Rabai, L. B., & Aissa, A. B. (2014). Classification of security threats in information systems. *Procedia Computer Science*, 32, 489-496.
- Jyväskylä Security Technology. (2015). National Cyber Exercise 2015 focuses on security network environment. *National Cyber Exercise 2015 focuses on security network environment*. Retrieved June 27, 2016, from <http://jyvsectec.fi/en/national-cyber-exercise-2015-focuses-on-security-network-environment/>
- McCallum, A., & Kamal, N. (1998). A comparison of event models for naive bayes text classification. *AAAI-98 workshop on learning for text categorization*(752), 41 - 48.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111-3119.
- Multinational Alliance for Collaborative Cyber Situational Awareness. (2013). *CCSA Information Sharing Framework (ISF), version 2.4*. Retrieved June 27, 2016, from <https://www.terena.org/mail-archives/refeds/pdf/Jz1CRtYC4.pdf>
- Pedragosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning*(12), 2825-2830.
- Pirinen, T. (2014). *Weighted Finite-State Methods for Spell-Checking and Correction*. Helsinki: University of Helsinki.
- Ramachandran, S., Jensen, R., Bascara, O., Carpenter, T., Denning, T., & Sucillon, S. (2011). Untangling Topic Threads in Chat-Based Communication: A Case Study. *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence* (pp. 50-55). AAAI Publications.
- Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In L. Liu, & M. T. Özsu (Eds.), *Encyclopedia of Database Systems* (pp. 532-538). Springer. doi:10.1007/978-0-387-39940-9_565
- Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. *Proceedings of the Twentieth International Conference on Machine Learning*, 3, pp. 616-623. Washington DC.
- Satterfield, K., Finomore, V., Castle, C., & Warm, J. (2011). Evaluation Tools to Aid Command and Control Operators in Chat-Based Communication Monitoring. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 55, pp. 480-484. SAGE Publications.

- The MathWorks Inc. (2013). *MATLAB Statistics and Machine Learning Toolbox*. (MATLAB, Ed.) Natick, Massachusetts.
- van Rijsbergen, C. J. (1979). Evaluation. In *Information Retrieval* (2 ed., pp. 112-140). London: Butterworths.
- Yang, Y., & Chute, C. G. (1994). An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems (TOIS)*, 12(3), 252-277.
- Özgür, A., Özgür, L., & Güngör, T. (2005). Text categorization with class-based and corpus-based keyword selection. In P. Yolum, T. Güngör, F. Gürgen, & C. Özturan (Eds.), *International Symposium on Computer and Information Sciences* (pp. 606-615). Berlin, Heidelberg: Springer. doi:10.1007/11569596_63
- Özyurt, Ö., & Köse, C. (2010). Chat mining: Automatic determination of chat conversations topic in Turkish text based chat mediums. *Expert Systems with Applications*, 37(12), 8705-8710.

7 KEYWORDS

Natural language processing: A field between computer science and linguistics which is concerned mainly with developing and evaluation tools for processing and analyzing human created natural language data.

Machine learning: A subfield of computer science namely concerned with developing methods for learning patterns from large scale data.

Text classification: A common task in information sciences where algorithms are used to assign a collection of text documents to a number of different classes, and the algorithm's performance is evaluated against a pre-labelled golden standard.

Classifiers: A group of machine learning algorithms designed to assign an item to a class of similar items.

Instant messaging: A form of online communication where two or more people converse with one another in a real-time setting, often through relatively brief and informal text messages.

Finnish: An agglutinative Finno-Ugric language spoken mainly in Finland by approximately 5.4 million people. Generally considered a "difficult" language due to the language's complex inflectional system.

Language technology: A general term for technologies utilizing natural language processing and computational linguistics.

Cyber security: Protection of information systems from threats.

BIOGRAPHICAL NOTES:

The authors are researchers at National Defence University, Helsinki Finland, at the Department of Military Technology. This paper highlights some of the results from their research on language technologies for efficient analysis of the highly technical yet informal professional instant messaging conducted between different state authorities. The authors are interested in how the expertise and knowledge of such authorities could automatically be incorporated into the decision making process of critical infrastructure management.

Reference to this paper should be made as follows: Puuska, S., Kortelainen, M.J., Venekoski, V., & Vankka, J. (2016). Instant Message Classification in Finnish Cyber Security Themed Free-Form Discussion. *International Journal on Cyber Situational Awareness*, Vol. 1, No. 1, pp97-109