THINK INDIA JOURNAL

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

A Machine Learning Approach To Mobile Agent Platform Protection: Towards Eliminating The Curse of Dimensionality

Harinderjit Kaur[1]
Department of Computer Science and Engineering
Lovely Professional University, Phagwara, India

Abstract

In the past many decades, the escalating threat of malicious mobile agents has been calling for the automated techniques of malicious mobile agent detection. The machine learning (ML) algorithms have been ascertained superior in this context rather than signature-based and behavior based approaches, specifically in high-dimensional feature space. With regard to this, two prime contributions are made in this paper: Firstly, detection of the unidentified malevolent mobile agents depends upon n-gram structures with managed ML methodology, that has not been implemented till now in the area of the mobile agent systems safety by other investigators. Secondly, to encourage the utility of "feature selection methods" for the purpose of classification. To perform the experiment, the n-grams ranging from 3 to 9 are fetched from a dataset consisting forty malevolent as well as non- malevolent mobile agents. Since the number of extracted distinctive "n-gram" structures is very large, selection method such as Chi Square Statistic ($\chi 2$) has been used to reduce the feature space. Finally, the classification is performed using different classifiers such as "Naïve Bayesian" (NB), "Instance Based Learner" (IBK), "Sequential Minimal Optimization" (SMO) and "J48 Decision Tree". The extensive experiments have been performed with different profile lengths at the best parameter settings using a resampling method known as a Cross Validation. The job done in this research is adequate for the work the unidentified malevolent mobile agent discovered in a Mobile Agent Domain and is of huge concern to the investigators who are specifically concerned with the domain.

Keywords: Malicious Mobile Agents, N-gram Feature Extraction, Nested Cross Validation, Feature Selection, Classification

1. Introduction

A Mobile Agent (MA) is a collection of executable programs which performs different tasks on the basis of its user and transfers from a implementation platform to other in a varied network [1]. The mobile agents have gained acceptance in the recent times since they provides numerous profits to the dispersed computing along with the decline of network load, elimination of network latency, performing dynamically, asynchronously and independently [2]. However, while working in a grid, they carry the fright of "Trojan horses" alongwith, worms and added intrusive resources or units [3]. This is due to the spasms which can occur if the mobile agents traverse in the transmission network and there could be few attackers suspecting the network to get few of the evidence approved by the agents or material stored in the agent platform or mutating that data for their own benefit [4]. In past many decades, various researchers have contributed to avert malevolent mobile agents triggering any detriment effect to Mobile Agent Platform (MAP).

Machine Learning algorithms rely upon the choice of features or dimensions representing the salient structure of considered dataset [86]. However, it is also acknowledged that in machine learning applications, the curse of dimensionality [87], or the large number of features/dimensions (much of them does not participate in the accurateness and may even reduce features) in many realms demonstrates a gigantic issue [7]. Apparently, reducing the high-dimensional space or lessening the number of n-gram features is essential in malicious detection problem, but it should be executed while sustaining a higher rate of accurateness.

Since, applying effective and efficient feature selection methods can enhance the performance of n-gram analysis in terms of accuracy and time to train the classifier, in the present work, feature selection method "Chi Square Statistic Method", is applied, in order to choose a subset of features which are the finest for perceptive among dual kinds of agent grouping (malevolent and non-malevolent). The different set operations of features chosen from these 3 methods are also used. The selected features are then given into 4 popularly employed grouping algorithms: Naive Bayesian algorithm, IBK method, SMO technique, J48 Decision Tree method, maintained by WEKA (Waikato Environment for Knowledge Analysis) tool [56]. The general experiments are stimulated on a pool of eighty records, in which 50 percent of the whole records are malevolent. The simulated outcomes are analysed depending on general routine outcome measures like "Sensitivity Rate", "Specificity Rate", "Positive Predictive Value", "Negative Predictive Value", "F-score", "Receiver Operating Characteristics – Area Under Curve", "Miss Rate", "Fall out" and "Accuracy Rate", during implementation of the five-fold cross verification method.

In the following sections, the implemented framework is assessed for the automated detection of unknown malevolent mobile agents for a specific dataset (described in Section 2.1) while considering several approaches and situations of the framework, by answering the following 6 questions:

Q1. Which feature selection method is better: Chi Square Statistic, Gain Ratio, Information Gain, Union and Intersection of three?

Q2. Which n-gram is the best: three-gram, four-gram, five-gram, six-gram, seven-gram, eight-gram, and nine-gram?

Q3. Which profile length is the best: 40, 60, 80 and 100?

Q4. Which classifier is the best: Instance Based Learner, Sequential Minimization Optimization, Naïve Bayesian, J48 Decision Tree?

The rest of the work is structured as follows: Sect. 2 sheds light over material and methods for proposed approach. Sect. 3 presents the results and discussions. Finally, conclusion is stated in Sect. 3.

2. Material and Methods

2.1. Dataset used

No typical data set is obtainable for the detection of malevolent mobile agents. That is why, the standard dataset of malevolent files termed as CSDMC2010[1] "API sequence corpus", consisting "Windows API/System-Call trace files, has been selected for the task of grouping. The dataset consists of 378 files including 315 malware samples as well as 62 benign traces (taken as non-malevolent in the work). Only fourty malevolent as well as non-malevolent files are collected for the training dataset of current work after random sampling (equal count for both is taken to elude the Class-imbalance issue). This typical dataset is desirable for the planned method as agent byte code can be observed as a series of agent API function calls. This assumption is made on the basis of the preceding studies of mining API call sequences from byte codes [82].

2.2. Performance Evaluation Measures

Identification of appropriate performance metrices is essential to assess the grouping outcome of discovering malevolent mobile agents effectively. The confusion matrix mentions the proper and improper grouping results found by the classifier when it is compared with the actual classification performance. The measures except Accuracy Rate and Misclassification Rate are deliberated to find out whether the current framework proves worthy for the grouping of either malevolent mobile agents or non-malevolent mobile agents or both.

True Positives (TP): Count of malevolent agents categorized as malicious.

True Negatives (TN): Count of non-malevolent agents categorized as non-malicious.
False Positives (FP): Count of non-malicious agents classified as malicious.
False Negatives (FN): Count of malevolent agents classified as non-malicious.

Performance Metric	Formula	Expected Rate
Sensitivity	$TP/(TP+FN)$	Maximum
Specificity	$TN/(TN+FP)$	Maximum
(PPV)	$TP/(TP+FP)$	Maximum
(NPV)	$TN/(TN+FN)$	Maximum
Miss Rate	$FN/(TP+FN)$	Minimum
Fall out	$FP/(FP+TN)$	Minimum
ROC-AUC	NA	Range of 0.9 and 1
Accuracy	$(TP+TN)/(TP+TN+FP+FN)$	Maximum
F-measure	$2.Precision.Recall/(Precision+Recall)$	Maximum

2.3 Methodology

The framework implemented in this work is displayed in Fig. 2. It comprises of three sequential steps such as extraction of mobile agent n-gram feature, feature selection, and finally, grouping. These points are elaborated in successive sub-parts.

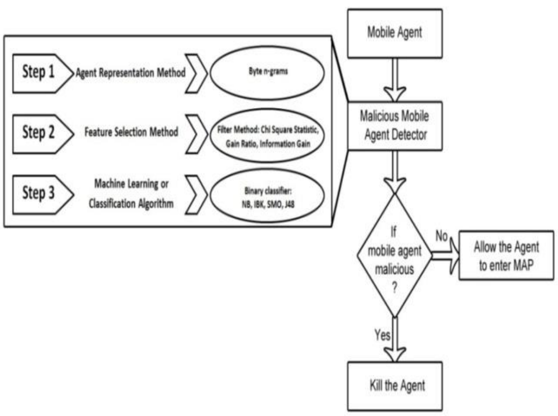


Fig. 1 Framework for Malicious Mobile Agent Detection

2.3.1 Data Preparation - Representation of Mobile Agent by Byte n-grams

Features from the malevolent and non- malevolent files are extracted using a standard n-gram analysis. This standard technique is solely machine-learning based technique which exploits "Natural Language Processing" (NLP) also [80]. Sliding-window fashion is used to extract n-grams, wherein a space of static length (n) slides a byte at one time. Generally, n-grams consists of all the substrings of a bigger string with size "n" [66, 35]. Presently, byte n-grams are considered as "API" call based structures. Importance of n-gram based methods has been released by many researchers in recent years in malware detection, as this technique of extracting features is basic and easy to implement. To bound the experiments for current study, the changing n-grams are implemented with the value of "n" between 3 to 9 only.

2.3.2 Feature Selection

Considering n-gram as a factor, the total count of likely mined features which form malevolent as well as non-malevolent records is very high and, exhausting all of them is likely to result in very high dimensionality respectively which burdens the classification process. In regard to this, the applicability of widely used various filter feature selection methods such as, Chi Square Statistic (χ^2) [74] is also explored in this paper, for improving n-gram based classification. Filter methods are used rather than wrapper methods because they are computationally less expensive and act independently of the grouping algorithm [97], thus allow us to equate the performances of the various classification algorithms. Filter methods tend to obtain a reduced set of features and so a threshold (profile size) is required to choose a subset. To achieve this purpose, for each feature selection technique, only four different thresholds or profile size (L): 40, 60, 80 and 100, are taken for performing different experiments, to bound the count of tests, which implies the highest L discerning n-grams are merely taken for designing training datasets.

2.3.3 Classification

"The Binary Classification" is taken into consideration because the unidentified mobile agent could be categorized as either malevolent or non-malevolent. The standard popularly used grouping algorithms like Instance based Learner[57], Naïve Bayesian [57], "Sequential Optimization" [81-87], and "J48 Decision Tree" [57], are employed.

Results and Discussion

On the basis of n-gram features, the grouping of mobile agent into 2 classes has been implemented on eighty agent files of dataset of Application Programming Interface calls sequence. To improve the presentation of each grouping algorithm, a widespread setting of parameters is done like "value of k", "distance measure", or "nearest neighbor search algorithm" in "IBK", "pruning", or "confidence factor" in "J48 decision tree", "complexity parameter", or "kernel" in SMO. Unbiased evaluation results are obtained by performing nested five-fold cross validation scheme [85]. In nested 5-fold cross validation scheme, the information is arbitrarily distributed into 5 disjoint folds. Alteration of classifier factors is done by four folds and afterwards the modified classifier is authorized on left out fold. The same process iterates for 5 times, every time with a varied left-out folds. Moreover, the standard parameters like Sensitivity, Specificity, F-measure and Accurateness, assess routine outcomes and the effects of all repetitions are averaged to obtain the concluding result. It is verified that the presentation of current job greatly depends upon feature selection technique and the selection of classifier.

3. Outcome of different classifiers by using χ^2 Feature Selection Method

The performance of different classifiers (NB, IBK, J48 and SMO) using χ^2 is analyzed at best parameter settings, which are repetitively tuned using WEKA tool, as shown in Tables 5, 6, and 7. The results provide more positive evidence for IBK classifier.

Fig. 2 Graph demonstrating Sensitivity Rate of using Chi Square Statistic Feature Selection Method for different Profile Lengths (40, 60, 80 and 100)

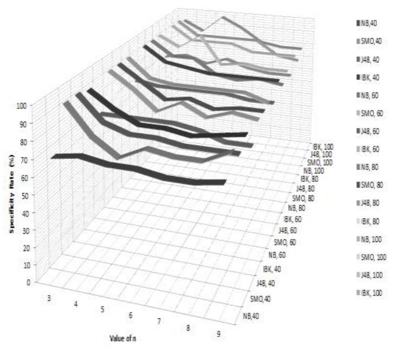
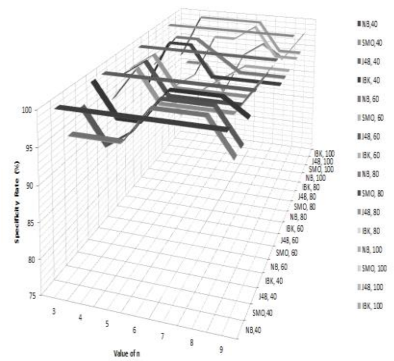


Fig. 3 Graph demonstrating Specificity Rate using Chi Square Statistic Feature Selection Method for different Profile Lengths (40, 60, 80 and 100)



The higher accuracy rates along with the miss rates of each classifier are summarized as follows:

Using Chi Square Statistic feature selection method

Classifier	n	Accuracy Rate (%)	Miss Rate (%)	Profile Length (L)
NB	4,6	88.75	22.50	100
SMO	3	95.00	5.00	40, 60
J48	3	96.25	5.00	40, 60
IBK	3	96.25	5.00	60

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

L	Classifier	n	Accuracy Rate (%)	Miss Rate (%)	Fall out (%)	PPV (%)	NPV (%)	F-measure (%)	ROC area	n	Accuracy Rate (%)	Miss Rate (%)	Fall out (%)	PPV (%)	NPV (%)	F-measure (%)	ROC area	n	Accuracy Rate (%)	Miss Rate (%)	Fall out (%)	PPV (%)	NPV (%)	F-measure (%)	ROC area
40	**Bayesian**	3	85.00	30.00	0.00	100.00	76.92	82.35	0.93	4	86.25	27.50	0.00	100.00	78.43	84.06	0.87	5	85.00	30.00	0.00	100.00	76.92	82.35	0.85
	SMO		95.00	5.00	5.00	95.00	95.00	95.00	0.95		86.25	22.50	5.00	93.94	80.85	84.93	0.86		81.25	32.50	5.00	93.10	74.51	78.26	0.81
	J48		96.25	5.00	2.50	97.44	95.12	96.20	0.94		86.25	20.00	7.50	91.43	82.22	85.33	0.87		85.00	25.00	5.00	93.75	79.17	83.33	0.83
	IBK		96.25	7.50	0.00	100.00	93.02	96.10	0.97		88.75	17.50	5.00	94.29	84.44	88.00	0.87		85.00	25.00	5.00	93.75	79.17	83.33	0.85
60	**Bayesian**		86.25	27.50	0.00	100.00	78.43	84.06	0.96		86.25	27.50	0.00	100.00	78.43	84.06	0.91		86.25	27.50	0.00	100.00	78.43	84.06	0.92
	SMO		95.00	5.00	5.00	95.00	95.00	95.00	0.95		92.50	15.00	0.00	100.00	86.96	91.89	0.93		83.75	27.50	5.00	93.55	77.55	81.69	0.84
	J48		96.25	5.00	2.50	97.44	95.12	96.20	0.94		92.50	15.00	0.00	100.00	86.96	91.89	0.89		85.00	25.00	5.00	93.75	79.17	83.33	0.83
	IBK		96.25	5.00	2.50	97.44	95.12	96.20	0.98		91.25	17.50	0.00	100.00	85.11	90.41	0.95		87.50	20.00	5.00	94.12	82.61	86.49	0.90
80	**Bayesian**		87.50	25.00	0.00	100.00	80.00	85.71	0.97		86.25	27.50	0.00	100.00	78.43	84.06	0.92		86.25	27.50	0.00	100.00	78.43	84.06	0.93
	SMO		93.75	5.00	7.50	92.68	94.87	93.83	0.94		92.50	15.00	0.00	100.00	86.96	91.89	0.93		91.25	17.50	0.00	100.00	85.11	90.41	0.91
	J48		95.00	5.00	5.00	95.00	95.00	95.00	0.93		92.50	15.00	0.00	100.00	86.96	91.89	0.89		92.50	15.00	0.00	100.00	86.96	91.89	0.90
	IBK		92.50	5.00	10.00	90.48	94.74	92.68	0.96		86.25	12.50	15.00	85.37	87.18	86.42	0.96		93.75	2.50	10.00	90.70	97.30	93.98	0.97
100	**Bayesian**		87.50	25.00	0.00	100.00	80.00	85.71	0.93		88.75	22.50	0.00	100.00	81.63	87.32	0.94		86.25	27.50	0.00	100.00	78.43	84.06	0.94
	SMO		91.25	5.00	12.50	88.37	94.59	91.57	0.91		87.50	15.00	10.00	89.47	85.71	87.18	0.88		91.25	15.00	2.50	97.14	86.67	90.67	0.91
	J48		93.75	5.00	7.50	92.68	94.87	93.83	0.92		92.50	15.00	0.00	100.00	86.96	91.89	0.96		92.50	15.00	0.00	100.00	86.96	91.89	0.90
	IBK		91.25	7.50	10.00	90.24	92.31	91.36	0.95		87.50	15.00	10.00	89.47	85.71	87.18	0.94		91.25	2.50	15.00	86.67	97.14	91.76	0.98
40	**Bayesian**	6	85.00	30.00	0.00	100.00	76.92	82.35	0.86	7	83.75	32.50	0.00	100.00	75.47	80.60	0.85	8	83.75	32.50	0.00	100.00	75.47	80.60	0.86
	SMO		87.50	25.00	0.00	100.00	80.00	85.71	0.88		86.25	27.50	0.00	100.00	78.43	84.06	0.86		86.25	27.50	0.00	100.00	78.43	84.06	0.86
	J48		87.50	25.00	0.00	100.00	80.00	85.71	0.85		86.25	27.50	0.00	100.00	78.43	84.06	0.82		86.25	27.50	0.00	100.00	78.43	84.06	0.84
	IBK		87.50	25.00	0.00	100.00	80.00	85.71	0.85		86.25	27.50	0.00	100.00	78.43	84.06	0.83		87.50	25.00	0.00	100.00	80.00	85.71	0.85
60	**Bayesian**		86.25	27.50	0.00	100.00	78.43	84.06	0.92		85.00	30.00	0.00	100.00	76.92	82.35	0.88		82.50	35.00	0.00	100.00	74.07	78.79	0.87
	SMO		87.50	20.00	5.00	94.12	82.61	86.49	0.88		83.75	27.50	5.00	93.55	77.55	81.69	0.84		86.25	22.50	5.00	93.94	80.85	84.93	0.86
	J48		86.25	22.50	5.00	93.94	80.85	84.93	0.85		83.75	27.50	5.00	93.55	77.55	81.69	0.85		85.00	25.00	5.00	93.75	79.17	83.33	0.85
	IBK		87.50	20.00	5.00	94.12	82.61	86.49	0.90		87.50	20.00	5.00	94.12	82.61	86.49	0.86		87.50	20.00	5.00	94.12	82.61	86.49	0.86
80	**Bayesian**		86.25	27.50	0.00	100.00	78.43	84.06	0.91		86.25	27.50	0.00	100.00	78.43	84.06	0.92		85.00	30.00	0.00	100.00	76.92	82.35	0.91
	SMO		87.50	20.00	5.00	94.12	82.61	86.49	0.88		87.50	20.00	5.00	94.12	82.61	86.49	0.88		87.50	20.00	5.00	94.12	82.61	86.49	0.88
	J48		86.25	25.00	2.50	96.77	79.59	84.51	0.84		83.75	27.50	5.00	93.55	77.55	81.69	0.85		85.00	25.00	5.00	93.75	79.17	83.33	0.85
	IBK		85.00	22.50	7.50	91.18	80.43	83.78	0.88		87.50	20.00	5.00	94.12	82.61	86.49	0.91		86.25	22.50	5.00	93.94	80.85	84.93	0.90
100	**Bayesian**		88.75	22.50	0.00	100.00	81.63	87.32	0.95		86.25	27.50	0.00	100.00	78.43	84.06	0.92		85.00	30.00	0.00	100.00	76.92	82.35	0.91
	SMO		91.25	15.00	2.50	97.14	86.67	90.67	0.91		90.00	20.00	0.00	100.00	83.33	88.89	0.90		87.50	20.00	5.00	94.12	82.61	86.49	0.88
	J48		92.50	15.00	0.00	100.00	86.96	91.89	0.89		92.50	15.00	0.00	100.00	86.96	91.89	0.92		85.00	25.00	5.00	93.75	79.17	83.33	0.85
	IBK		91.25	10.00	7.50	92.31	90.24	91.14	0.96		87.50	20.00	5.00	94.12	82.61	86.49	0.92		85.00	22.50	7.50	91.18	80.43	83.78	0.89
40	**Bayesian**	9	85.00	30.00	0.00	100.00	76.92	82.35	0.91																
	SMO		87.50	20.00	5.00	94.12	82.61	86.49	0.88																
	J48		83.75	27.50	5.00	93.55	77.55	81.69	0.86																
	IBK		87.50	22.50	2.50	96.88	81.25	86.11	0.91																
60	**Bayesian**		82.50	35.00	0.00	100.00	74.07	78.79	0.85																

Table Summary:

NB classifier gives maximum Accurateness Rate of 88.75% and minimum Miss Rate of 22.50 % for Profile Length equals to 100, with 3-grams and 6-grams. Thus, Bayesian classifier works well with large number of features selected using χ^2 selection model. Furthermore, the maximum F-

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

	SMO	87.50	25.00	0.00	100.00	80.00	85.71	0.88
	J48	87.50	25.00	0.00	100.00	80.00	85.71	0.84
	IBK	87.50	25.00	0.00	100.00	80.00	85.71	0.84
80	**Bayesian**	85.00	30.00	0.00	100.00	76.92	82.35	0.91
	SMO	87.50	20.00	5.00	94.12	82.61	86.49	0.88
	J48	83.75	27.50	5.00	93.55	77.55	81.69	0.86
	IBK	86.25	22.50	5.00	93.94	80.85	84.93	0.91
100	**Bayesian**	85.00	30.00	0.00	100.00	76.92	82.35	0.91
	SMO	87.50	20.00	5.00	94.12	82.61	86.49	0.88
	J48	83.75	27.50	5.00	93.55	77.55	81.69	0.86
	IBK	87.50	22.50	2.50	96.88	81.25	86.11	0.91

measure has been obtained for three-grams i.e. 96.20% with classifiers J48 and IBK for profile lengths 40 and 60. Pondering over, J48 gives accuracy rate of 96.25 % and miss rate of 5% for 3-grams (L= 40 and 60). Likewise, IBK provides accurateness rate of 96.25% and miss rate of five% for three-grams with profile length 40 and 60 respectively. SMO gives accuracy rate of 95.00% and 5.00% miss rate for 3-grams and L=40. IBK with χ^2 gives minimum miss rate of 2.50% when compared to other classifiers. Thus, IBK is the best in classifying correctly malicious agents as malicious for 5-grams and L=80 and 100.

Probing further, it has been observed that the Miss rate highly increases with increase in value of n for n-grams. For NB, the value of Fall-out is 0% always, irrespective of the values of n and L. Therefore, NB in conjunction with χ^2 never wrongly classifies the non-malicious as malicious agent. In other words, Naïve Bayesian always correctly classifies the non-malicious agents. The value of PPV for NB classifier is 100.00% for all values of n and L. It means the agents categorized as malevolent using NB classifier are actually malevolent. The (n, L) pairs of (4,60), (6, 40), (7, 40) and (8,40) with different classifiers give 100.00% PPV. The value of NPV is more than 95.00% for 3-grams only which means that more than 95.00% agents categorized as non-malicious are truthfully non-malicious.

4. Conclusions

This paper examines the suitability of Kernel-based Extreme Learning Machine algorithm for the work of classifying the incoming mobile agents which could be malevolent or non-malevolent in a Mobile Agent Environment while breaking the curse of dimensionality of a particular dataset. Specifically, the classification process make use of n-grams as the features. Various feature selection methods such as Chi Square Statistic ($\chi 2$) is employed to monitor the significance of feature selection in improving the classification accuracy.

In the extensive experiment, the authors have investigated the use of feature choice methods during grouping process. Both J48 and IBK give accuracy rate of 97.50% (more than that without feature selection methods) and low miss rate of 5.00% for 3-grams and 4-grams with profile length equals to 40 and 100 respectively using ($\chi 2$) . More accuracy rate obtained with the reduced feature space (i.e. IBK with 40 3-gram features and J48 with 74 4-gram features) encourages the use of feature selection methods. Hence, IBK is considered to be the finest classifier and the optimal outcomes uplift the usage of current study for Mobile Agent Platform protection. It has also been observed that Naïve Bayesian classifier gives Specificity rate of 100.00% for all n-grams at all profile lengths, deducing that Naïve Bayesian classifier is the best in correctly classifying non-malevolent agents as non-malevolent.

References

[1] Aneiba, A. & Rees, S. J. (2004). Mobile Agents Technology and Mobility. In Proceedings of the 5th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, pp. 14-20.

[2] Lange, D.B. & Oshima, M. (1999). Seven good reasons for Mobile Agents. Communications of the ACM, 42(3), 88-89. doi: 10.1145/295685.298136.

[3] Thomsen, L. L. & Thomsen, B. (1997). Mobile Agents – The new paradigm in computing. ICL Systems Journal, 12, 14-40.

[4] Oppliger, R. (1999). Security issues related to mobile code and agent-based systems. Computer Communications, 22 (12), 1165-1170. doi: 10.1016/S0140-3664(99)00083-3.

[5] Venugopal, D. & Hu, Guoning. (2008). Efficient signature based malware detection on mobile devices. Mobile Information Systems, 4 (1), 33-49. doi: 10.1155/2008/712353.

[6] Ma, W., Duan, P., Liu, S. Gu, G. & Liu, J. (2012). Shadow attacks: automatically evading system-call-behavior based malware detection. Journal in Computer Virology, 8 (1), 1-13. doi: 10.1007/s11416-011-0157-5.

[7] Moskovitch, R., Feher, C. Tzachar, N., Berger, E., Gitelman, M., Shlomi, D. & Elovici, Y. (2008). Unknown Malcode Detection Using OPCODE Representation. In D. Ortiz-Arroyo, H. L. Larsen, D. D. Zeng, D. Hicks, & G. Wagner (Eds.), Intelligence and Security Informatics, Series Volume 5376, (pp. 204-215). Springer Berlin Heidelberg. doi: 10.1007/978-3-540-89900-6_21.

[8] Jain, S. & Meena, Y. K. Byte Level n-Gram Analysis for Malware Detection. In K. R. Venugopal & L. M. Patnaik (Eds.), Computer Networks and Intelligent Computing, Series Volume 157, (pp. 51-59). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-22786-8_6.

[9] Wahbe, R., Lucco, S., Anderson, T. E. & Graham, S. L. (1994). Efficient Software-Based Fault Isolation. In Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles (SOSP '93), pp. 203-216. doi: 10.1145/168619.168635.

[10] van't Noordende, G. Brazier F. M. & Tannenbaum, A. S. (2002). A security framework for a mobile agent system. In Proceedings of the SEMAS-2002, pp. 43-50.

[11] Marikkannu, P. & Jovin, A. (2011). A Secure Mobile Agent System against Tailgating Attacks. Journal of Computer Science, 7(4), 488-492.

[12] Alfalayeh, M. & Brankovic, L. (2005). An overview of security issues and techniques in mobile agents. In Proceedings of the 8th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, pp. 59-78. doi: 10.1007/0-387-24486-7_5.

[13] Lee, P. & Necula, G. (1997). Research on proof-carrying code for mobile-code security. In DARPA workshop on foundations for secure mobile code, pp. 26-28.

[14] Ordille, J. J. (1996). When agents roam, who can you trust? In Proceedings of the IEEE First Annual conference on Emerging Technologies and Applications in Communications, pp. 188-191. doi: 10.1109/ETACOM.1996.502505.

[15] Cao, C. & Lu, J. (2006). Path-history-based access control for mobile agents. International Journal of Parallel, Emergent and distributed Systems, 21 (3), 215-225. doi: 10.1080/17445760500356205.

[16] Venkatesan, S. & Chellappan, S. (2008). Protection of mobile agent platform through Attack Identification Scanner (AIS) by Malicious Identification Police (MIP). In Proceedings of the IEEE First International Conference on Emerging Trends in Engineering and Technology (ICETET '08), pp. 1228-1231. doi: 10.1109/ICETET.2008.89.

[17] Venkatesan, S., Chellappan, C., Vengattaraman, T.,

Dhavachelvan, P. & Vaish, A. (2010). Advanced mobile agent security models for code integrity and malicious availability check. Journal of Network and Computer applications, 33 (6), 661-671. doi: 10.1016/j.jnca.2010.03.010.

[18] Venkatesan, S., Baskaran, R. Chellappan, C. Vaish, A. & Dhavachelvan, P. (2013). Artificial immune system based mobile agent platform protection. Computer Standards & Interfaces, 35 (4), 365-373. doi: 10.1016/j.csi.2012.10.003

[19] Kešelj, V., Peng, F., Cercone, N. & Thomas, C. (2003). N-GRAM-BASED AUTHOR PROFILES FOR AUTHORSHIP ATTRIBUTION. In Proceedings of the Conference Pacific Association for Computational Linguistics (PACLING '03), pp. 255-264.

[20] Sidorov, G., Velasquez, F. Stamatatos, E. Gelbukh, A. & Chanona-Hernandez, L. (2013). Syntactic Dependency-Based N-grams as Classification Features. In I. Batyrshin & M. G. Mendoza (Eds.), Advances in Computational Intelligence, Series Volume 7630, (pp. 1-11). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-37798-3_1.

[21] Riseman, E. M. (1974). A Contextual Postprocessing System for Error Correction Using Binary n-Grams. IEEE Transactions on Computers, C-23 (5), 480-493. doi: 0.1109/T-C.1974.223971.

[22] Kukich, K. (1992). Techniques for automatically correcting words in text. ACM Computing Surveys (CSUR). 24(4), 377-439. doi: 10.1145/146370.146380.

[23] McNamee, P. & Mayfield, J. (2004). Character N-Gram Tokenization for European Language Text Retrieval. Information Retrieval, 7(1), 73-97. doi: 10.1023/B:INRT.0000009441.78971.be.

[24] Lee, J. H. & Ahn, J. S. Using n-grams for Korean text retrieval. (1996) In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 216-224. doi: 10.1145/243199.243269.

[25] Wang, X., McCallum, A. & Wei, X. (2007). Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval. In Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, pp. 697-702. doi: 10.1109/ICDM.2007.86.

[26] Järvelin, A., Järvelin, A. & Järvelin, K. (2007). s-grams: Defining generalized n-grams for information retrieval. International Journal of Information Processing and Management, 43(4), 1005-1019. doi: 10.1016/j.ipm.2006.09.016.

[27] Zissman, M. A. & Singer, E. (1994). Automatic language identification of telephone speech messages using phoneme recognition and N-gram modeling. In Proceedings of the 1994 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-94), pp. 1/305-1/308. doi: 10.1109/ICASSP.1994.389377.

[28] Ahmed, B., Cha, S. H. & Tappert, C. (2004). Language Identification from Text Using N-gram Based Cumulative Frequency Addition. In Proceedings of Student/ Faculty Research Day, pp. 12/1-12/8.

[29] Schulze, B. M. (2000). Automatic Language Identification Using Both N-gram and Word Information. United States Patent No. 6,167,369.

[30] Gyawali, B., Ramirez, G. & Solorio, T. (2013). Native Language Identification: A Simple N-gram Based Approach. In Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 224-231.

[31] Tromp, E. & Pechenizkiy, M. (2011). Graph-Based N-gram Language Identification on Short Texts. In Proceedings of the 20th Machine Learning Conference of Belgium and the Netherlands, pp. 27-34.

[32] Jiang, G., Chen, H., Ungureanu, C. & Yoshihira, K. (2007). Multiresolution Abnormal Trace Detection Using Varied-Length n-Grams and Automata. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 37 (1), 86-97. doi: 10.1109/TSMCC.2006.871569.

[33] Abou-Assaleh, T., Cercone, N., Kešelj, V. & Sweidan, R. (2004). Detection of new malicious code using n-grams signatures. In Second annual conference on Privacy, Security and Trust (PST), pp. 193-196.

[34] Kanaris, I., Kanaris, K., Houvardas, I. and Stamatatos, E. (2007). WORDS VS. CHARACTER N-GRAMS FOR ANTI-SPAM FILTERING. International Journal on Artificial Intelligence Tools, 16(6), 1047 - 1067. doi: 10.1142/S0218213007003692.

[35] Schultz, M.G., Eskin, E., Zadok, F. & Stolfo, S. J. (2001). Data mining methods for detection of new malicious executables. In Proceedings of the 2001 IEEE Symposium on Security and Privacy, pp. 38-49. doi: 10.1109/SECPRI.2001.924286.

[36] Downie, J.S. (1999). Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams as text. Doctoral dissertation, The University of Western Ontario (London).

[37] Doraisamy, S. & Rüger, S. (2003). Robust Polyphonic Music Retrieval with N-grams. Journal of Intelligent Information Systems, 21(1), 53-70. doi: 10.1023/A:1023553801115.

[38] Harding, S.M., Croft, W. B. & Weir, C. (1997). Probabilistic Retrieval of OCR Degraded Text Using N-grams. In Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, pp. 345-359.

[39] Stolcke, A., Coccaro, N., Bates, R., Taylor, P., Ess-Dykema, C.V., Ries, K. Shriberg, E., Jurafsky, D., Martin, R. & Meteer, M. (2000). Dialogue Act Modeling for Automatic

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

Tagging and Recognition of Conversational Speech. Computational Linguistics, 26(3), 339-373. doi: 10.1162/089120100561737.

[40] Hirsimaki, T., Pylkkonen, J. & Kurimo, M. (2009). Importance of High-Order N-gram Models in Morph-Based Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing, 17(4), 724-732. doi: 10.1109/TASL.2008.2012323.

[41] Siu, M. & Ostendorf, M. (2000). Variable n-grams and extensions for conversational speech language modeling. IEEE Transactions on Speech and Audio Processing, 8(1), 63-75. doi: 10.1109/89.817454.

[42] Mustafa, S.H. & Al-Radaideh, Q.A. (2004). Using N-grams for Arabic text searching. In Journal of the Association for Information Science and Technology, 55(11), 1002-1007. doi: 10.1002/asi.20051.

[43] Robertson, A.M. & Willett, P. (1998). Applications of N-grams in Textual Information Systems. Journal of Documentation, 54(1), pp. 48-67. doi: 10.1108/EUM0000000007161.

[44] Rangarajan, V & Ravichandran, N. (1998). SYSTEM AND METHOD FOR PORTABLE DOCUMENT INDEXING USING N-GRAM WORD DECOMPOSITION. United States Patent No. 5,706,365.

[45] Zamora, E.M., Pollock, J.J., & Zamora A. (1981). The use of trigram analysis for spelling error detection. Information Processing & Management, 17(6), 305-316. doi: 10.1016/0306-4573(81)90044-3.

[46] Bassil, Y. (2012). Parallel Spell-Checking Algorithm Based on Yahoo! N-Grams Dataset. International Journal of Research and Reviews in Computer Science, 3(1), 2079-2557.

[47] Ahmed, F., De Luca, E.W., & Nürnberger, A. (2009). Revise N-gram based automatic spelling correction tool to improve retrieval effectiveness. Polibits, 40, 39-48.

[48] Ullmann, J. R. (1977). A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words. The Computer Journal, 20(2), 141-147. doi: 10.1093/comjnl/20.2.141.

[49] Islam, A. & Inkpen, D. (2009). Real-word spelling correction using Google web 1tn-gram data set. In Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09), pp. 1689-1692. doi: 10.1145/1645953.1646205.

[50] Flor. M. (2012). Four Types of context for automatic spelling correction. TAL, 53(3), 61-99.

[51] Samanta, P. & Chaudhuri, B.B. (2013). A Simple Real-Word Error Detection and Correction Using Local Word Bigram and Trigram. In Proceedings of ROCLING.

[52] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reuteman, P. & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter, 11(1), 10-18. doi: 10.1145/1656274.1656278.

[53] Han, J. (2000). Data mining: concepts and Techniques. doi: 10.1109/CMPSAC.2004.1342667.

[54] Dalkilic, G. & Yalcin, C. (2009). Turkish spelling error detection and correction by using word n-grams. In Proceedings of the Fifth International Conference on Soft Computing, computing with Words and Perceptions in System Analysis, Decision and Control, pp. 1-4. doi: 10.1109/ICSCCW.2009.5379481.

[55] Peng, F. & Schuurmans, D. (2003). Combining Naive Bayes and n-Gram Language Models for Text Classification. In Proceedings of the 25th European conference on IR Research, pp. 335-350.

[56] Cavnar, W. B. & Trenkle, J. M. (1994). N-Gram-Based Text Categorization. In Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94), pp. 161-175.

[57] Fürnkranz, J. (1998). A Study Using N-gram Features for Text Categorization. Technical Report oefai-tr-9830, Austrian Research Institute for Artificial Intelligence.

[58] Khreisat, L. (2006). Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study. In Proceedings of the 2006 International Conference on Data Mining, pp. 78-82.

[59] Damashek, M. (1995). Gauging Similarity with n-Grams: Language-Independent Categorization of Text. Science, 267 (5199), 843-848.

[60] Rahmoun, A. & Elberrichi, Z. (2007). Experimenting N-Grams in Text Categorization. The International Arab Journal of Information Technology, 4(4), 377-385.

[61] Silic, A., Chauchat J., Bašić, B.D. & Morin, A. (2007). N-Grams and Morphological Normalization in Text Classification: A Comparison on a Croatian-English Parallel Corpus. In J. Neves, M.F. Santos & J. M. Machado (Eds.), Progress in Artificial Intelligence, Series Volume 4874, (pp. 671-682). Springer Berlin Heidelberg. doi: 10.1007/978-3-540-77002-2_56.

[62] Wei, Z., Miao, Duoqian, Chauchat, J., Zhao, R. & Li, W. (2009). N-grams Based Feature Selection and Text Representation for Chinese Text Classification. International Journal of Computational Intelligence Systems, 2(4), 365-374. doi: 10.1080/18756891.2009.9727668.

[63] Ifrim, G., Bakir, G., & Weikum, G. (2008). Fast Logistic Regression for Text Categorization with variable-length n-grams. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 354-362. doi: 10.1145/1401890.1401936.

[64] Graovac, J. (2013). Text categorization using n-gram based language independent technique. In 35th Anniversary

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

of computational Linguistics in Serbia, Book of Abstracts, to appear in the Proceedings of the Conference.

[65] Graovac, J. (2012). Serbian text categorization using byte level n-grams. BCI (Local), 93-96.

[66] Khreisat, L. (2009). A machine learning approach for Arabic Text Classification using N-gram frequency statistics. Journal of Informetrics, 3(1), 72-77. doi: 0.1016/j.joi.2008.11.005.

[67] Oroumchian, F., Darrudi, E. Taghiyareh, F. & Angoshtari, N. (2004). Experiments with Persian Text Compression for Web. In Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, pp. 478-479. doi: 10.1145/1013367.1013534.

[68] Witten, I.H. & Bell, T.C. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. IEEE Transactions on Information Theory, 37(4), 1085-1094. doi: 10.1109/18.87000.

[69] Hull, J.J. & Srihari, S.N. (1982). Experiments in Text Recognition with Binary n-Gram and Viterbi Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-4(5), 520-530. doi: 10.1109/TPAMI.1982.4767297.

[70] Liu, H. & Setiono R. Chi2: Feature Selection and Discretization of Numeric Attributes. In Proceedings of the 2012 IEEE 24th International conference on tools with Artificial Intelligence, doi: 10.1109/TAI.1995.479783.

[71] Bickel, S., Haider, P. & Scheffer, T. (2005). Predicting Sentences Using N-gram Language Models. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 193-200. doi: 10.3115/1220575.1220600.

[72] Ziólko, B., Gałka, J. & Ziólko, M. (2009). Phoneme Ngrams Based on a Polish Newspaper Corpus. WORLDCOMP, 59(803).

[73] Tauritz, D.R. & Sprinkhuizen-Kuyper. (2000). Adaptive Information Filtering: Evolutionary Computation and N-gram Representation. In Proceedings of the Twelfth Belgium-Netherlands Artificial Intelligence Conference, pp. 157-164.

[74] Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research, 3, 1289-1305.

[75] Guyon, I. & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3, 1157-1182.

[76] Stańczyk, U. (2015). Feature Evaluation by Filter, Wrapper, and Embedded Approaches. In U. Stańczyk, & L. C. Jain (Eds.), Feature Selection for Data and Pattern Recognition, Series Volume 584, (pp. 29-44). Springer Berlin Heidelberg. doi: 10.1007/978-3-662-45620-0_3.

[77] Sokolova, M. & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing and Management, 45(4), 427-437. doi: 10.1016/j.ipm.2009.03.002.

[78] Burbidge, R. & Buxton, B. (2001). An Introduction to Support Vector Machines for Data Mining. Keynote Speakers, young OR12, 3-15.

[79] Jakkula, V. (2006). Tutorial on Support Vector Machine (SVM). School of EECS, Washington State University.

[80] Hsu, C.W, Chang, C.C. & Lin, C.J. (2003). A Practical Guide to Support Vector Classification. Technical Report, Department of computer Science, National Taiwan University.

[81] Witten, I. H., Frank, E., Trigg, L. Hall, M., Holmes, G. & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java Implementations. In Proceedings of ANNES '99 International Workshop on emerging Engineering and Connectionist-based Information Systems, pp. 192-196.

[82] Suen, C. Y. (1979). n-Gram Statistics for Natural Language Understanding and Text Processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-1 (2), 164-172. doi: 10.1109/TPAMI.1979.4766902.

[83] Nguyen, T, T., Pham, H. V., Vu, P. M., &Nguyen, T. T. (2016). Learning API Usages from Bytecode: A Statistical Approach. In Proceedings of the 38th International Conference on Software Engineering (ICSE '16), ACM, pp. 416-427. doi:10.1145/2884781.2884873.

[84] Qiao, Y., Yang, Y., He, Jie, Tang, C. & Liu, Zhixue. (2012). CBM: Free, Automatic Malware Analysis Framework Using API Call Sequences. In F. Sun, T. Li, & H. Li. (Eds.), Proceedings of the Seventh International Conference on Intelligent Systems and Knowledge Engineering, Series Volume 214, (pp. 225-236). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-37832-4_21.

[85] Varma, S. & Simon, R. (2006) Bias in error estimation when using cross validation for model selection. BMC Informatics, 7:91. doi: 10.1186/1471-2105-7-91.

[86] Zhou, L., Pan, S., Wang, J. & Vasilakos, A.V. (2017). Machine Learning on Big Data: Opportunities and Challenges. Neurocomputing. (Accepted)

[87] Keogh E. & Mueen A. (2011) Curse of Dimensionality. In C, Sammut, & G. I. Webb (Eds.), Encyclopedia of Machine Learning, Springer US, pp. 257-258.