Classification in a Skewed Online Trade Fraud Complaint Corpus

William Kos, Marijn Schraagen, Matthieu Brinkhuis, and Floris Bex

Department of Information and Computing Sciences, Utrecht University

{w.h.kos, m.p.schraagen[1], m.j.s.brinkhuis, f.j.bex}@uu.nl

Abstract. This paper explores how machine learning techniques can be used to support handling of skewed online trade fraud complaints, by predicting whether a complaint will be withdrawn or not. To optimize the performance of each classifier, the influence of resampling, word weighting, and word normalization on the classification performance is assessed. It is found that machine learning can indeed be used for this purpose, by improving the baseline performance in comparison to the skewness ratio up to 13 pp using Logistic Regression. Furthermore, the results show that data alteration techniques can improve classifier performance on a skewed dataset up to 13.5 pp.

Keywords: Classification, Law Enforcement, Skewed Data.

1 Introduction

The Dutch National Police maintains an online interface that allows civilians to report their complaints regarding trade fraud over an online medium (e.g. eBay). Since an increasing amount of complaints are being filed [9], it is desirable to make an automatic distinction between complaints worth investigating and those not worth investigating. One valuable distinction which can be made early in the process is that between a complaint which will be withdrawn by either the complainant or the police and a complaint that will not be withdrawn. The current research examines whether either one of nine machine learning classifiers trained on free text complaint data can be used for this purpose. Complicating this task is the class distribution in the data, where a majority of 83.3% is labelled as "not withdrawn". To prevent this skewness from affecting classifier performance, data alteration techniques are applied, of which the influence on the classification performance is assessed. Specifically, resampling using multiple training sample distributions [12, 16], word normalization using either stemming [17] or lemmatization [4], and word weighting using either binary [19] or TF-IDF weighting [24] were applied to examine their influence on a classifier trained using a textual skewed dataset.

[1] Corresponding author

2 Related Work

Data mining techniques, including classification, are increasingly being applied to the field of crime analysis. Chen et al. explored data mining techniques used for crime analysis [7], and Sharma and Panigrahi have categorized over 40 approaches using machine learning techniques for fraud detection [21].

In [1], associative classification has been used, which is a technique where both association rules and classification are combined, to accurately discriminate phishing websites from legitimate websites. Making use of Bayesian networks, [2] have been able to predict the characteristics of a homicide offender based on crime scene variables (e.g. police report or autopsy report) more accurate than a team of police experts. These characteristics could be used by police officers to identify a possible suspect. In [9], naive Bayes, Bayesian network, decision tree, and association rule techniques were compared for their speed and accuracy in classifying crimes and accidents in Denver City, and it was found that association rules result in the highest accuracy. In their collaboration with the West Midlands Police, [15] used a Bayesian network to predict whether a certain property in the UK's Midlands will be re-victimized or not and within what timespan. Next to this, a neural network was used to classify possible offenders for their likelihood of conducting unsolved crimes.

Some of the research performed in this field focuses mainly on how to deal with a skewed dataset (i.e. one class is overly present). In their research, [5] combined a rule-based association system with a neural network to detect credit card fraud. Using their combined classifier, they are able to achieve an accuracy of 99.955%. In [16], research has been performed on which classification method is best to be used for fraud detection with a skewed dataset. It was proposed to use a stacking-bagging method, in which a naive Bayes, neural network and decision tree classifier are combined.

Previous work on the classification of online police complaints has to the knowledge of the authors not yet been presented.

3 Experimental Setup

3.1 Dataset

The online trade fraud complaints dataset used in this research has been provided by the Dutch National Police. This dataset consists of 51.386 entries, manually labelled by police employees on whether a complaint has been withdrawn and if so, for what reason. In total, 8.609 (16.7%) entries have been labelled *withdrawn*, resulting in a skewed dataset. The dataset contains a total of 60 attributes, including the binary class labels, and contains a free-text field in which the complainant's story that led to the complaint is included. Note that only the textual complaints description is used in the current research (see Section 5 for further discussion). An anonymized and translated example of both complaint types is included in Table 1.

Table 1: Data example

Withdrawn	John Doe advertised a rental home. In hindsight it all appears to be fake.
Not withdrawn	I have bought a bottle of Dom Perignon and a bottle of Crystal 1999 from John Doe via Marktplaats and transferred 100 euro to NL01ABCD0123456789. Up to now, I have not received anything and John Doe does not respond to my e-mails.

3.2 Classification technique selection

As no previous work on classifying online trade fraud complaints has yet been presented, it was unknown which classification techniques would result in the best performance. Therefore, the set of techniques used in this research was based upon a combination of previous work on classification for textual, skewed, and criminal data. An overview of the techniques used in this research is described by Sebastiani [20], who compared the classification results of ensembling based (e.g. AdaBoost), SVM, logistic regression, association rule (e.g. RIPPER), KNN, decision tree, neural network, and probabilistic classifiers used in individual papers on the highly skewed Reuters dataset, which contains a collection of news documents.

3.3 Research framework

In order to ensure the analyses in this research were performed under equal conditions, a research framework was created. Analysis is conducted using Weka [22] combined with R [18] using the package RWeka [10]. Linguistic analysis is performed by Frog [20], incorporated in the framework using the package Frogr[2]. The research framework consists of two sections: the preprocessing section and the training/testing section.

Preprocessing

Since this research is based upon the prediction of the *withdrawn* label using a complaint's free-text field, these variables were first selected from the dataset, after which a corpus was created where each document represented a single complaint. The words in each document were transformed to their base form using either a Dutch adapted stemmer [17] or lemmatizer [4]. In a subsequent cleaning phase, all punctuation was removed and Dutch stop words were removed according to a predefined list[3]. Next, the corpus was split using stratified 10-fold cross validation [14], so that the ratio of minority to majority classes in each fold was maintained and each complaint was used once for testing and nine times for training.

[2] W. Van Atteveldt. *Frogr: R client for the frog tagger and parser for Dutch*. R package version 0.100, 2014.

[3] http://snowball.tartarus.org/algorithms/dutch/stop.txt

Training and testing

Each of the 10 folds resulting from the preprocessing section was used for training and testing a classifier following the same procedure. First, 90% of the fold assigned for training the classifier was transformed into a term-document matrix containing 1:3-grams. All terms present in the term-document matrix were evaluated for their presence in the documents and the term-document matrix was reduced to the 100 most occurring n-grams, which are used as features for machine learning. Here, it has been opted to use the 100 most occurring terms over a standard feature selection algorithm to reduce the dimensionality of the problem and thereby clearly discriminate the influence of the mentioned data alteration techniques on the classification performance, which is the focus of this research. In preliminary experiments it was found that the classification performance improved when increasing the amount of features used for training up to 100, after which it stabilized. Furthermore, it was found that using 100 features greatly improves the framework runtime, while barely influencing the classification performance in comparison to using more features.

Next, the term-document matrix was resampled using either random undersampling or SMOTE [6] following the resampling techniques described by Japkowicz and Stephen [12] to reduce dataset skewness from influencing a classifier's performance. With random undersampling, the minority class entries were kept constant and the majority class entries were randomly downscaled according to the training sample distribution. For SMOTE, the minority class entries were scaled up to match the amount of majority class entries and the majority class entries were randomly up- or downscaled according to the training sample distribution. The classifier was then build on the resampled term-document matrix using Weka's default classifier implementations [22]. To avoid interference with the aspects under investigation in this research, we have opted to use the basic parameter settings provided by Weka.

After the classifier had been built it was tested using the assigned 10% of the fold. If a term used for training the classifier did not occur in the term-document matrix of the test set, a value of 0 was assigned to it for each document indicating its absence.

Finally, the evaluation results of each individual fold were combined and averaged resulting in an overall evaluation for the classifier under the predefined settings.

3.4 Evaluation

The performance on classifying online trade fraud complaints was evaluated using the macro-averaged F_1-measure, following the approach of Yang and Liu [23]. When using the accuracy as a metric, the overall performance would be highly influenced by the data skewness. The macro-averaged F_1-measure, however, equally weighs both recall and precision of the minority and majority class, thereby clearly showing the overall performance evenly influenced by both classes. The computation of the F_1-measure defines the *not withdrawn* and *withdrawn* classes as positive and negative, respectively:

	Predicted class		
Actual class		not withdrawn	withdrawn
	not withdrawn (nwd)	TP	FN
	withdrawn (wd)	FP	TN

The F1-measure is defined in terms of precision and recall, as follows:

$$F_1 = \frac{F_{wd} + F_{nwd}}{2} = \frac{\frac{2 \cdot P_{wd} \cdot R_{wd}}{P_{wd} + R_{wd}} + \frac{2 \cdot P_{nwd} \cdot R_{nwd}}{P_{nwd} + R_{nwd}}}{2} =$$

$$\frac{\frac{2 \cdot \frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}}{\frac{TP}{TP + FP} + \frac{TP}{TP + FN}} + \frac{2 \cdot \frac{TN}{TN + FN} \cdot \frac{TN}{TN + FP}}{\frac{TN}{TN + FN} + \frac{TN}{TN + FP}}}{2}$$

To determine which classifier holds the most additional value, classification results should also be compared to a baseline [23]. As this research focused on the influence of data alteration techniques to the performance of a classifier, it was opted to compare all results to their unimproved counterpart. Regarding word normalization, however, it was determined to omit test cases where no word normalization was applied for runtime optimization. Based on three randomly generated feature lists using both stemming and lemmatization, it was decided to use stemming as the baseline for word normalization, as lemmatization generally diverges further from the non-normalized word forms. Overall, when evaluating the influence of data alteration techniques on classifier performance, it was thus compared to the results obtained using a training set which was not sampled, unweighted, and normalized using stemming.

4 Results and Discussion

4.1 Evaluating the overall results

During this research a total of 9 classifiers have been evaluated under different resampling, word normalization and word weighting conditions. An overview of the individual results has been combined in Table 2, which shows for each classifier the baseline compared to the optimal training setup, shown as resampling type, percentage of minority cases, weighting type.

When evaluating the observations of each individual classification algorithm, it becomes apparent that the difference in optimized performance between the best classifier (i.e. Logistic regression) and the worst classifier (i.e. K-nearest neighbor) is only minor with 4.2 percentage points (pp). Five classifiers hold the best optimized classification performance within a range of 1 pp, namely logistic regression, multinomial naive Bayes, support vector machine, multivariate naive Bayes, and association rule classifi-

Table 2: Overview individual classifier results (F1 score)

U/O: Undersampling/Oversampling; N/B/T: No/Binary/TF-IDF weighting

Classifier	**Baseline**	**Optimal**	**Setup**	**Difference**
Multinomial naive Bayes	**0.572**	0.590	U30B	0.018
Logistic regression	0.466	**0.594**	U40N	0.128
Decision tree	0.527	0.560	U30T	0.033
Multivariate naive Bayes	0.560	0.586	N16B	0.026
Association rule	0.456	0.585	U40N	0.129
Neural network	0.454	0.564	U30B	0.110
K-nearest-neighbor	0.503	0.552	U40B	0.049
Support vector machine	0.454	0.589	O40T	**0.135**
AdaBoost	0.454	0.556	U40B	0.102

ers. Even though the differences between the individual baseline and optimal classification results vary in size, this minor difference in optimal performance suggests that a classifier can only be improved up to a certain extent. This would thus imply that, after optimizing, the selection of an appropriate classification algorithm should depend less upon performance, but more on other metrics (e.g. runtime). In our results, the macro-averaged F-measures do not exceed 0.594, which results from a low performance on the minority class. Underlying this relatively low overall performance could be two reasons:

- The features used in the free-text field for both withdrawn and not withdrawn complaints show a high resemblance. When, e.g., warning the police for a possible fraudster, which is not a valid complaint, words like *marktplaats* (online trading website), *oplichter* (fraudster), and *product* (idem) are often used. Such explanative features are also used in a complaint which has not been withdrawn. Comparing a subset of individual complaints revealed that it is possible that the free-text fields do not contain enough information to be distinct, thereby reducing the overall performance of a classifier trained on this dataset using features resulting from a bag-of-words approach.
- Since in this research it has been opted to use the 100 most occurring terms, which are used in both withdrawn and not withdrawn complaints, the overall performance of the classifiers could be reduced compared to when the more distinctive features would be selected using a feature selection algorithm.

4.2 Evaluating data alteration results

Resampling

With respect to resampling, each individual classification technique follows a similar trend as illustrated for an undersampled multinomial naive Bayes classifier in Figure 1. When increasing the training sample distribution, the precision of the minority class slightly decreases, while the recall of the minority class strongly increases. The majority class follows an opposite pattern where the precision of the majority class slightly increases, while the recall of the majority class strongly decreases. Even though

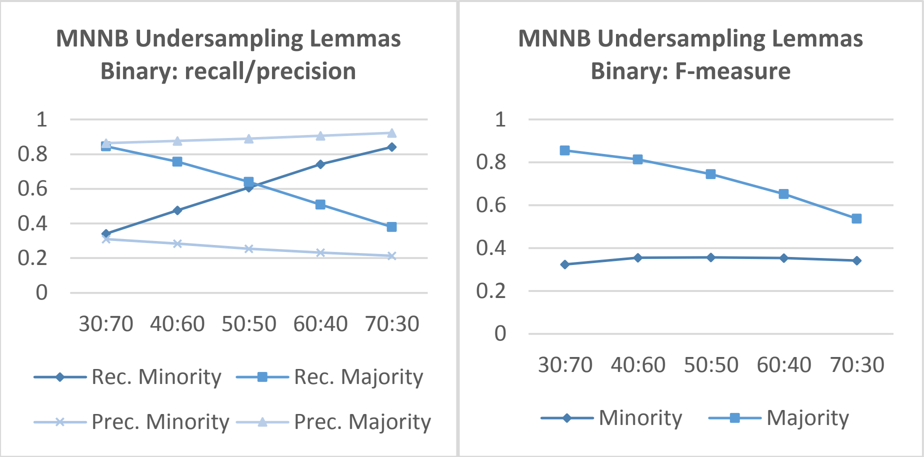

Figure 1: Recall, precision (left) and F-measure (right) using multinomial naive Bayes. X-axis: training sample distribution. Y-axis: recall/precision/F-measure.

resampling is intended to increase the minority performance without detriment of the majority performance, the observed pattern can be explained. During the training phase a classifier will, at first, become less inclined to automatically label a test case as the majority class as the skewness is lifted, after which it will be inclined towards the minority class as it becomes overly present in the training set. The rate at which the recall and precision change differs per classification technique, however, the precision of the minority class does, independent of the training sample distribution, remain centered around 30% for each classification technique. In a similar way, the precision of the majority class remains centered around 90%. These patterns do, however, imply that, in the current setup, the F-measure of the minority class can only be improved up to a certain extent, due to its harmonic nature as illustrated in 1. For the majority class, the F-measure only decreases due to the decline in recall, thus nullifying the influence of the high precision. Combining both patterns results in the conclusion that an optimal resampling percentage should thus be at a level where the initial increase in minority performance neutralizes the decrease in majority performance. This initial increase is the largest when there are less minority cases, which implies that the resampling percentage should be in favor of the majority class, which is confirmed by the optimal training sample distributions in Table 2.

Word weighting

Table 2 shows that for most classifiers, the right choice of weighting technique can outperform an unweighted baseline. Which technique to use depends upon the classification technique, however, binary weighting often outperforms TF-IDF weighting. This finding could imply that merely the presence of a term is enough for a classifier to be based upon, which is consistent with earlier findings with respect to SVM and naive Bayes classifiers [13, 19].

Word normalization

For word normalization Table 3 lists the difference in performance using either stemming or lemmatization for all classifiers under the optimal setup. The table shows that all classifiers perform better using lemmatization, however, the differences are only minor (0.5 to 1.8pp). Overall, with a few exceptions, lemmatization has little but consistently improved recall and precision of both the minority and majority class.

Table 3: Overview of word normalization influence

Classifier	Stemming	Lemmatization	Difference
Multinomial naive Bayes	0.584	0.590	0.006
Logistic regression	0.583	0.594	0.011
Decision tree	0.549	0.560	0.011
Multivariate naive Bayes	0.581	0.586	0.005
Association rule	0.574	0.585	0.011
Neural network	0.551	0.564	0.013
K-nearest-neighbor	0.543	0.552	0.009
Support vector machine	0.578	0.589	0.011
AdaBoost	0.538	0.556	0.018

4.3 Evaluating additional results

In section 4.1, two possible reasons have been mentioned with respect to the relatively low classification performance on the minority class. To initially examine these possible causes, follow-up experiments have been performed in which the influence of each cause is independently verified. Furthermore, an experiment has been executed regarding the probabilistic classifiers as the findings in this research did not match the prescribed literature.

The performance of probabilistic classifiers

As outlined in Section 4.1, probabilistic classifiers (i.e. multinomial naive Bayes and multivariate naive Bayes) have a good performance on the current data. In contrast, Yang and Liu [23] found logistic regression, SVM, and KNN to significantly outperform neural networks and naive Bayes with respect to the macro-averaged F-measure on the Reuters-21578 corpus. Even though it is based upon the micro-averaged F-measure, the result comparison of Sebastiani [20] also showed the lower performance of probabilistic classifiers on the skewed Reuters corpus in comparison to the other classifiers in his research. Here it should be noted that the Reuters corpus is a multi-labeled dataset instead of binary, and that the classifiers in the overview of Sebastiani [20] have not been improved using resampling. However, when comparing the unsampled results, the probabilistic classifiers still outperform all other classifiers in this research including logistic regression and association rules. Looking at the probabilistic classifiers individually, it was observed that the difference in performance between an unsampled and a resampled classifier was negligible. Combining this with the skewed characteristic of the dataset led to the assumption that on a highly skewed dataset, determining the

Table 4: Results per feature set using multinomial naive Bayes

#Features in test set	Set size	Recall minority	Recall majority	Precision minority	Precision majority	F-measure minority	F-measure majority
0-9	1529.2	0.068	0.965	0.313	0.825	0.110	0.889
10-19	1921.8	0.273	0.884	0.331	0.856	0.298	0.870
20-29	1016.4	0.343	0.844	0.298	0.871	0.318	0.857
30-39	459.9	0.443	0.827	0.309	0.894	0.361	0.859
40+	211.3	0.482	0.819	0.281	0.920	0.348	0.866

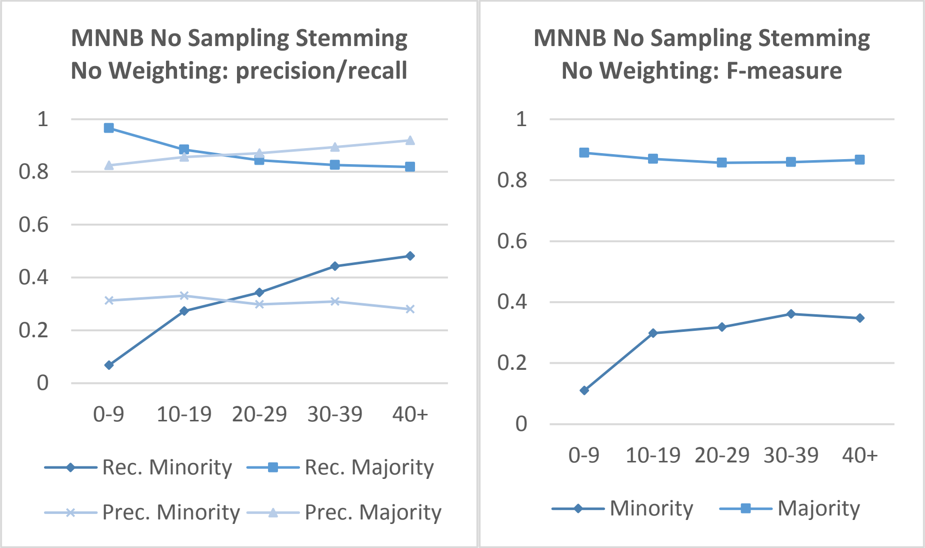

Figure 2: Recall, precision (left) and F-measure (right) using multinomial naive Bayes. X-axis: #features in test set, Y-axis: recall/precision/F-measure.

posterior probability in a naive Bayes classifier is mainly influenced by the class likelihood and less by the class priori. To test this hypothesis, an experiment has been set up in which the class predictions were evaluated with respect to the total amount of features in the test set which have been used for building the classifier. For this experiment, the multinomial naive Bayes classifier has been trained according to the baseline and framework as described in section 3.4. The results of this experiment are contained in Table 4 and depicted in Figure 2.

Evaluating the above results reveals a clear relationship between the predictive accuracy of a classifier and the amount of features used in the test set. Using less features causes a multinomial naive Bayes classifier to predict more test cases as the majority class compared to using more features. This observation supports the assumption that the influence of the priori decreases as more features are present in the test set. Furthermore, this experiment shows that the predictive power of a probabilistic classifier on a skewed dataset depends upon the amount of features used in the test set. Since in this research the 100 most common features have been used, the amount of features used in the test set are likely to be correlated with the length of the test set, thereby resulting in the assumption that the predictive power of a classifier on textual data depends on the length of the document.

Table 5: Results using multinomial naive Bayes with bi-normal separation

Selection method	**Recall minority**	**Recall majority**	**Precision minority**	**Precision majority**	**F-measure minority**	**F-measure majority**
BNS	0.033	0.989	0.384	0.836	0.060	0.906
Occurrence	0.241	0.892	0.314	0.854	0.272	0.872

The influence of features

In section 4.2, it was mentioned that independent of the training sample distribution, the precision of the minority class remains fixed around 30%. This implies that, even though the 100 most common features contain more information than simply classifying all test cases as either the minority or majority class, the information richness is restricted. To examine whether feature selection could be used to overcome this limitation in information richness, an experiment following the same training procedure as above has been set up in which bi-normal separation [8] is used to select the most informative features. The results of this experiment are compared to the classifier trained using the 100 most occurring features in Table 5.

Given that the selected features are highly distinctive, the low minority recall using bi-normal separation suggests a reduced influence of the priori, which, as was discussed earlier, implies that few features are used for testing when training the classifier with 100 selected features. Combining this with the above example leads to the conclusion that the distinctive features selected using bi-normal separation do not cover a wide spectrum of the test cases. Since the most distinctive features occur in only a limited amount of complaints, the assumption is supported that the features in the dataset do not contain enough information to be used for making a distinction on whether a complaint will be withdrawn or not.

Comparing the results of the two feature selection methods, it can be observed that for 100 features using bi-normal separation as the feature selection metric only reduces the performance of multinomial naive Bayes classifier (-8.9pp). Combining this with above conclusion that the distinctive features in this dataset cover only a small spectrum of the test cases, it can be concluded that when training a multinomial naive Bayes classifier on a dataset with little distinctive features the best feature selection metric is to use the most common features. Further research is required to conclude whether this finding also applies to other classifiers.

5 Conclusions and Future Research

In this research, it has been examined whether a complaint’s free-text field can be used to predict whether a complaint will be withdrawn or not. Literature studies on the use of machine learning techniques for characteristics of the dataset (i.e. criminal, textual, and skewed) revealed the combination of machine learning with data alteration techniques to result in the best classification performance, which was confirmed during the data analysis. The data analysis furthermore revealed that the best optimized machine learning technique to be used for making the prediction is Logistic regression. Using a Logistic regression classifier, a recall of 33.5% and a precision of 29.7% can be attained for the minority class, while for the majority class a recall of 84.0% and a precision of 86.3% can be attained, which exceeds the unsampled ratio of minority to majority complaints (16.7%/83.3%).

Regarding probabilistic classifiers, it was found that using more features in the test set can improve the predictive power. Furthermore, it was found that when training a multinomial naive Bayes classifier on a dataset with little distinctive features the best feature selection metric is to use the most common features.

Even though it can be concluded that a complaint's free-text field can indeed be used to some extent to predict whether a complaint will be withdrawn or not, the performance was restricted. Since a complaint does not solely exist of a free-text field, but contains 59 other attributes, such as location fields, payment method, social media usage. Exploratory analysis showed the potential of this approach. Textual meta-attributes such as word or document length could be included as features as well. In future research a selection of these attributes may lead to a more accurate classifier. In addition, the decision to reject a complaint may be influenced by developments during the investigation following the initial filing of the complaint, which is outside of the scope of the current dataset.

Acknowledgements

This research has been funded by the Dutch National Police in the project 'Intelligence Amplification for Cybercrime' (IAC) [3].

References

1. Abdelhamid, N., Ayesh, A, Thabtah, F.: Phishing detection based Associative Classification data mining. *Expert Systems with Applications,* 41(13): 5948–5959, 2014.
2. Baumgartner, K., Ferrari, S., Palermo, G.: Constructing Bayesian networks for criminal profiling from limited data. *Knowledge-Based Systems,* 21(7):563–572, 2008.
3. Bex, F.J., Testerink, B. and Peters, J. (2016) A.I. for Online Criminal Complaints: From Natural Dialogues to Structured Scenarios. *ECAI 2016 workshop on Artificial Intelligence for Justice*.
4. van den Bosch, A, Busser, G.J., Daelemans, W, Canisius, S: An efficient memory-based morphosyntactic tagger and parser for Dutch. In: van Eynde, F., Dirix, P., Schuurman, I, Vandeghinste, V. (eds.) *Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting,* pages 99–114, 2007.
5. Brause, R., Langsdorf, T., Hepp, M.: Neural Data Mining for Credit Card Fraud Detection. In: *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence,* pages 103–106, 1999.

6. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, 16:321-357, 2002.
7. Chen, H., Chung, W., Xu, J., Wang, G., Qin, Y., Chau, M.: Crime Data Mining: A General Framework and Some Examples. *Computer*, 37(4):50–56, 2004.
8. Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research,* 3:1289-1305, 2003.
9. Gupta, A., Syed, A., Mohammad, A., Halgaguge, M.: A Comparative Study of Classification Algorithms using Data Mining: Crime and Accidents in Denver City the USA. *International Journal of Advanced Computer Science and Applications*, 7(7):374–381, 2016.
10. Hornik, K., Buchta, C., Zeileis, A.: Open-Source Machine Learning: R Meets Weka. *Computational Statistics*, 24(2):225–232, 2009.
11. Inspectie Veiligheid en Justitie. Aanpak van internetoplichting door de politie: inspectieonderzoek naar een vorm van cybercrime, 2015.
12. Japkowicz, N., Stephen, S.: The Class Imbalance Problem: A Systematic Study. *Intelligent data analysis*, 6(5):429–449, 2002.
13. Jurafsky, D., Martin, J.: *Speech and Language Processing*. Prentice Hall, 2014.
14. Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Mellish, C (ed.) *Proceedings of the 14th International Joint Conference on Articial Intelligence*, pages 1137-1143, 1995.
15. Oatley, G., Ewart, B: Crimes analysis software: 'pins in maps', clustering and Bayes net prediction. *Expert Systems with Applications,* 25(4):569588, 2003.
16. Phua, C, Alahakoon, D, Lee, V: Minority Report in Fraud Detection: Classification of Skewed Data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59, 2004.
17. Porter, M.: An algorithm for suffix stripping. *Program*, 14(3):130-137, 1980.
18. R Core Team. R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria, 2015.
19. Schneider, K-M: On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification. In: Vicedo, J., Martínez-Barco, P, Muńoz, R., Saiz Noeda, M. (eds.) *Advances in Natural Language Processing. Lecture Notes in Computer Science, 3230*, pages 474-485, 2004.
20. Sebastiani, F.: Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.
21. Sharma, A., Panigrahi, P.: A Review of Financial Accounting Fraud Detection based on Data Mining Techniques. *International Journal of Computer Applications,* 39(1):37–47, 2012.
22. Witten, I, Frank, E.: *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2005.
23. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, 1999.
24. Zhang, W., Yoshida, T., Tang, X.: A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758-2765, 2011.