2015 IEEE 16th International Conference on Information Reuse and Integration

Using Random Undersampling to Alleviate Class Imbalance on Tweet Sentiment Data

Joseph Prusa, Taghi M. Khoshgoftaar, David J. Dittman, Amri Napolitano
Florida Atlantic University
Email: {jprusa, khoshgof, ddittman, amrifau} @fau.edu

***Abstract*—**
Sentiment classification of tweets is used for a variety of social sensing tasks and provides a means of discerning public opinion on a wide range of topics. A potential concern when performing sentiment classification is that the training data may contain class imbalance, which can negatively affect classification performance. A classifier trained on imbalanced data may be biased in favor of the majority class. One possibile method of addressing this is to use data sampling to achieve a more balanced class distribution. In this work, we seek to observe how data sampling (using random undersampling with either a 50:50 or 35:65 positive:negative post-sampling class distribution ratio) affects the classification performance on tweet sentiment data. Our experimental results show that Random Undersampling significantly improves classification performance in comparison to not using any data sampling. Furthermore, there is no significant difference between selecting a 50:50 or 35:65 post-sampling class distribution ratio.

***Keywords*-sentiment analysis; tweet mining; classification; data sampling;**

I. Introduction

Opinion mining, performed by mining social media sites and classifying user posts or microblogs, is a powerful tool that can be used to survey a large number of people to collect information that can be used for a wide range of applications, such as prediction of election results [18], product sales [12], or movie box office performance [13]. By collecting large quantities of posts relating to a specific topic and performing sentiment analysis, a statement can be made about the general population's view towards that topic.

Sentiment analysis refers to a number of methods used to determine the emotional polarity of text. In particular, mining Twitter and conducting sentiment analysis of tweets (user posts on twitter containing 140 or fewer characters) is appealing as Twitter has a large user base of over 270 million active users and over 500 million tweets are submitted daily [1].

As the data is collected from real world sources the dataset may exhibit class imbalance (have more instances in one class than the other class or classes) [10]. The presence of a large majority class may skew classifier performance to favor classification of unknown instances as belonging to the majority class. Despite tweet data frequently being imbalanced, such as the SemEval dataset [3], addressing class imbalance in tweet sentiment datasets has not been thoroughly studied.

One technique used to reduce the impact of class imbalance is to perform data sampling to create a training set with a more balanced class distribution. Data sampling creates a subset of the original data and selects instances in a manner that reduces class imbalance. The reduction in class imbalance reduces bias towards classification of new instances as the majority class. While many data sampling techniques exist, Random UnderSampling (RUS) is of particular interest as the datasets are high dimensional and may contain thousands of instances, so reducing the number of instances is beneficial in terms of computational costs. Random undersampling is often used with a 50:50 post-sampling class ratio (both classes have an equal number of instances), however less aggressive resampling ratios may be beneficial as they delete less majority instances. We test both 50:50 and 35:65 positive:negative post-sampling ratios. The impact of class imbalance is tested on eight different machine learning algorithms: C4.5, C4.5D, Naïve Bayes, Multilayer Perceptron, 5-Nearest Neighbors, Support Vector Machine, Radial Basis Function Network, and Logistic Regression.

Each learner was trained and tested using either no data sampling, or random undersampling to a 50:50 or 35:65 post sampling class distribution, on six datasets with different levels of class imbalance (1:99, 5:95, 10:90, 15:85, 20:80, and 25:75 possitive:negative class ratios). Each dataset consists of 3000 instances selected from the sentiment140 corpus specified to have one of the six previously mentioned class ratios. Our results show that, for the majority of learners, using RUS improves classifier performance and that sampling to a 35:65 post-sampling class distribution has higher performance on highly imbalanced data (data with less than 10% minority instances), while sampling to a 50:50 post-sampling class distribution achieves higher performance on less imbalanced data. Statistical analysis (ANalysis Of VAriance (ANOVA) and Tukey's Honest Significant Difference (HSD)) tests are used to confirm the significance of these results, validating that the increase in classifier performance observed when implementing RUS is significant; however,

978-1-4673-6656-4/15 $31.00 © 2015 IEEE
DOI 10.1109/IRI.2015.39

the choice of post-sampling class distribution was found to not be significant. We conclude that RUS offers significant improvement to tweet sentiment classification performance when training from class imbalanced data.

The remaining sections are organized as follows. Section II describes previous research on tweet sentiment and how class imbalance has been addressed by current research on tweet sentiment classification. Section III explains the data sampling technique used in our experiment and the class ratios selected. Section IV is broken into subsections explaining the creation of the datasets, the different learners used, and how each model is trained and evaluated. Results and analysis of our experiments are presented in Section V. Finally, the Section VI presents our conclusions based on our results, and discusses potential for future work.

II. Related Works

Sentiment classification is concerned with extracting useful information (a text passage's sentiment) from patterns and trends in text. Methodologies proposed for performing sentiment classification of tweets have primarily followed the work of Go et al. [8], who applied the methodology developed by Pang et al. [14] for classification of movie review sentiment to tweet sentiment. Pang et al. extracted unigram, bigram, and part-of-speech features and used three learners (Naïve Bayes, Support Vector Machine, and maximum entropy) in their experiment. Go et al. collected a large volume of sentiment labeled tweets (using emoticons to automatically label tweets). Using these tweets as a training set, Go et al. extracted unigrams, bigrams, and part-of-speech tags as features and trained sentiment classifiers using three learners mirroring the experiment conducted by Pang et al. Though Go et al. collected tweets in a manner leading to a balanced dataset, many tweet sentiment datasets are collected without control over the class distribution and many more recent datasets contain class imbalance. The presence of imbalance in tweet sentiment training data has been noted to have a negative effect on tweet sentiment classifier performance [9], [16].

By creating a new dataset with a more balanced class distribution, data sampling can reduce classifier bias towards to the majority class. Data sampling has been shown to improve classifier performance for learners trained on imbalanced data in many machine learning domains. Van Hulse et al. [17] tested seven data sampling techniques on 35 datasets with 11 different classifiers. They found that data sampling techniques could significantly improve classifier performance on imbalanced data. Data sampling has also been shown to be effective for sentiment classification of product reviews. Li et al. [11] were one of the first groups to directly address class imbalance in sentiment classification. They chose to use RUS based on the work of Japkowicz and Stephen [10]. In their experiment, they used random subspace generation in a semi-supervised environment to train their classifiers from four sentiment datasets created from documents about books, DVDs, electronics, and kitchen appliances. In preliminary testing, using only labeled instances, they found that using RUS offered superior performance compared to using the full labeled training data, and also found RUS outperformed random oversampling. In addition to improving classifier performance, RUS also reduces the computational time required to train a classifier as less instances are considered. This is highly desirable in the domain of tweet sentiment as datasets may be high dimensional. Yet, despite its promising results in other data mining domains and potential benefits, data sampling has received little attention for tweet sentiment classification.

Agrawal and An [3] trained a tweet sentiment classifier using SVM and the SemEval 2014 tweet sentiment dataset. This dataset contains more positive than negative instances, and in preliminary experiments they found that the recall of negative instances was lower than that of positive instances. In an effort to improve classifier performance, they applied Synthetic Minority Over-sampling TechniquE (SMOTE). While SMOTE increased the number of negative instances, they did not observe an improvement in the recall of negative instances corresponding to the improvement in class balance.

Li et al. [11] showed that RUS improved the performance of sentiment classifiers on four datasets that had between 12% and 22% minority class instances, and confirmed that RUS outperforms random oversampling; however, they were not studying tweet sentiment, which (due to the brevity of tweets) may be significantly different than text in other sentiment domains [8]. Additionally, they tested only a single post-sampling class ratio and provided no statistical tests of the significance of the measured improvement. From their work, it is unclear if RUS will significantly improve performance for tweet sentiment classification and if choice of post-sampling class distribution ratios will impact classifier performance for different levels of imbalance.

Agrawal and An [3] acknowledged class imbalance in tweet sentiment datasets, and tried to directly address it using data sampling. They selected an oversampling technique instead of an undersampling technique and found no notable improvements. They did not make further efforts to try additional techniques after SMOTE failed to improve their classifier's recall of negative instances, despite undersampling having been shown to outperform oversampling in related sentiment classification domains. Agrawal and An also conducted their tests using SVM, a learner that does not greatly benefit from data sampling.

In this work, we conduct experiments to measure the impact of RUS on classifier performance for classifiers trained on imbalanced twitter data. To the best of our knowledge, we are the first group to directly address class imbalanced tweet sentiment data using RUS. We compare two different levels of sampling and use six different datasets with different levels of class imbalance. The two levels of

Table I: Datasets

Name	#minority	%minority	#attribute
1:99	30	1%	2380
5:95	150	5%	2371
10:90	300	10%	2375
15:85	450	15%	2362
20:80	600	20%	2370
25:75	750	25%	2368

post-sampling class distribution ratio, 50:50 and 35:65, were selected as 50:50 is a common sampling ratio representing a perfectly balanced dataset and 35:65 has been found to yield better performance for learners in some domains (especially in cases of high class imbalance [17]). The performance of RUS on all datasets is evaluated using eight different machine learning algorithms. Additionally, we conduct statistical analysis of our results to verify their significance using ANOVA and Tukey's HSD test.

III. Random Undersampling

Random undersampling is a form of data sampling that randomly selects majority class instances and removes them from the dataset until the desired class distribution is achieved [4]. This means that for a dataset containing 100 positive and 400 negative instances, RUS must remove 300 negative instances in order to achieve a 50:50 post-sampling positive:negative class ratio. As tweet sentiment datasets are large, this loss of instances should not be problematic as sufficient instances will remain from which to train the classifier. Oversampling techniques are less desirable as they increase the size of already high dimensional datasets (tweet datasets have thousands of instances and thousands of features). This paper compares the impact of two different post-sampling class distribution ratios, 50:50 (denoted as RUS50) and 35:65 (denoted as RUS35). The first is selected as it creates a perfectly balanced resampled dataset from which to train learners. The second post-sampling ratio is selected as it has been shown in other domains that sampling to a perfectly balanced class distribution is not always optimal in cases of high class imbalance as it eliminates less majority instances [17]. This sampling technique was implemented in the Weka data-mining toolkit [20].

IV. Methodology

A. Dataset

The datasets for this experiment were constructed from the sentiment140 corpus [8], a publicly available collection of 1.6 million tweets with positive or negative sentiment labels. This dataset was selected due to being previously used to construct datasets by multiple research groups and has been used in our previous work [15]. Its large size facilitates construction of multiple datasets with different class distributions but the same total number of instances.

The corpus was constructed through automated collection and labeling of tweets by searching Twitter for Tweets with emoticons associated with either positive or negative sentiment. For our experiment, six datasets with different class distributions were constructed by sampling (without replacement) a specified number of positive and negative instances from the sentiment140 corpus, creating a dataset containing 3000 instances with the specified class ratio. Details for our constructed datasets are provided in Table I.

Unigrams (individual words within the text of the tweet) were extracted as features with the requirement that each unigram be at least two characters in length and appears in at least two tweets in the dataset, following the methodology proposed by Forman [6]. In recent years, unigrams have been used for sentiment classification in related domains, such as movie review sentiment [14], and studies on tweet sentiment classification. Prior to extracting features, text was filtered and cleaned by removing symbols, punctuation marks and URLs, making all letters uniformly lower case, and removing excess character repetitions. As the datasets contain different instances, their features are not identical. The number of features extracted for each dataset is displayed in Table I, and shows all six datasets have a similar number of features.

B. Learners

We created inductive models using eight different classification algorithms, briefly described with accompanying parameters. These classifiers were selected as they are commonly used in machine learning, and several are commonly used for sentiment classification. All learners were implemented using the Weka toolkit [20] with default values unless otherwise noted. All changes were found to improve classification results by preliminary research or previous work [17].

In this study, we train and evaluate the performance of K Nearest Neighbor (KNN), two variants of C4.5, Support Vector Machines (SVM), Multilayer Perceptron (MLP), Radial Basis Function Network (RBF), Logistic Regression (LR), and Naïve Bayes (NB).

KNN, denoted IBk in Weka, was constructed using "$k = 5$" and the "$distanceWeighting$" parameter set to "$Weight\ by\ 1/distance$" to use inverse distance weighting in determining the classification of instances. It is denoted as 5NN in this study. The variants of the C4.5 decision tree were constructed using J48 in Weka. C4.5D uses default parameters while C4.5N uses no pruning and Laplace Smoothing [19].

SVM, called SMO in Weka, had the complexity constant "c" changed from 1.0 to 5.0, and the "$buildLogisticModels$" parameter set to "$true$" to allow proper probability estimates to be obtained [20]. The SVM learner used a linear kernel.

Table II: Classification Results

Dataset	Sampling	Learner							
		C4.5N	NB	MLP	5NN	SVM	RBF	LR	C4.5D
1:99	None	0.49987	**0.57909**	0.49827	0.50065	0.49195	0.51096	**0.49149**	0.50013
	RUS35	**0.53112**	0.53480	**0.50901**	0.47253	**0.50852**	**0.51951**	0.47283	**0.53384**
	RUS50	0.50942	0.50446	0.48117	**0.51708**	0.50234	0.51629	0.47981	0.51820
5:95	None	0.54387	0.58657	0.53729	0.54737	**0.61706**	0.50579	0.57273	0.50449
	RUS35	**0.58652**	**0.62940**	**0.60126**	**0.57961**	0.61026	0.57978	0.59752	**0.57137**
	RUS50	0.57822	0.62724	0.54270	0.57238	0.59635	**0.58449**	0.58499	0.56491
10:90	None	0.57065	0.64153	**0.60915**	**0.62295**	**0.67913**	0.57695	0.60407	0.53129
	RUS35	0.60481	0.67986	0.58922	0.61377	0.66126	**0.61741**	0.62639	0.55944
	RUS50	**0.61586**	**0.69630**	0.56964	0.61186	0.64100	0.60507	**0.63744**	**0.58800**
15:85	None	0.58273	0.67723	**0.62227**	**0.65847**	**0.70903**	0.62565	0.56310	0.52342
	RUS35	0.61696	0.70754	0.59079	0.63835	0.69660	0.63933	0.65109	0.56861
	RUS50	**0.62177**	**0.72115**	0.61437	0.61002	0.67912	**0.64404**	**0.65245**	**0.60978**
20:80	None	0.58823	0.71100	**0.64772**	**0.66428**	**0.71254**	0.63209	0.57199	0.52496
	RUS35	0.61825	0.72785	0.59648	0.64750	0.70166	**0.64519**	0.64727	0.56832
	RUS50	**0.62117**	**0.73636**	0.63212	0.62888	0.69120	0.64168	**0.66339**	**0.59795**
25:75	None	0.58446	0.72613	**0.65955**	**0.65428**	**0.69683**	0.65034	0.56000	0.53693
	RUS35	0.63248	0.73410	0.61938	0.64154	0.69416	**0.65115**	0.62884	0.57592
	RUS50	**0.64257**	**0.73790**	0.62511	0.62289	0.68194	0.65079	**0.64195**	**0.62524**

MLP had two parameters changed from default values. The "$hiddenLayers$" parameter was changed to "3" to define a network with 1 hidden layer containing three nodes and the "$validationSetSize$" parameter was set to "10" was chosen so that 10% of the training data would be left aside to use as validation to determine when to stop the training process. A second type of Artificial Neural Network, Radial Basis Function Network (RBF), was constructed with the parameter "$numClusters$" set to "10."

C. Cross-Validation and Performance Metric

Cross-validation (CV) is a technique used to allow for training and testing of inductive models without resorting to using the same dataset. In this paper, we use five-fold cross-validation, which splits the data into five partitions. In each iteration of CV the training data consists of four folds while the remaining fold serves as a test dataset. Additionally, we perform four runs of the five-fold cross validation in an effort to reduce any bias resulting from how the data was split in the creation of the five partitions. Data sampling is performed on each training dataset generated using CV. The classification performance of each model is evaluated using the Area Under the Receiver Operating Characteristic Curve (AUC) [20]. AUC is selected as a performance metric because we are measuring classifier performance on imbalanced data and AUC is insensitive to changes in class skew, making it suitable for measurement of classifier performance on highly skewed data [5].

V. Results

Using the methodology and datasets described above we measured the effect using RUS has on the performance of tweet sentiment classifiers. We compare the performance of eight learners on six imbalanced datasets, testing RUS50 and RUS35 on every dataset, and compare the performance of classifiers trained using RUS against classifiers trained with no sampling technique. Table II presents the results of our experiments, divided by dataset. AUC score for each learner is presented with no data sampling (None), RUS35, and RUS50. In each column the model with the highest AUC is indicated in **boldface**.

The 10:90, 15:85, 20:80, and 25:75 imbalanced datasets exhibit similar patterns. for all of these datasets, RUS improves performance of five of the eight learners, while SVM, 5-NN, and MPL were not improved. Also, RBF network performed best with RUS35, while the remaining four learners performed better using RUS50. The best performing classifier on these datasets was NB with RUS50 and best performing classifier with no data sampling was SVM, which was only outpeformed by NB with RUS50. Averaged across all learners, both RUS35 and RUS50 perform better than using no undersampling. RUS50 achieves higher AUC scores than RUS35 for these datasets; however, this difference is small and narrows as the level of class imbalance rises.

Figure 1: AUC vs. Level of Class Imbalance Averaged Across All Learners

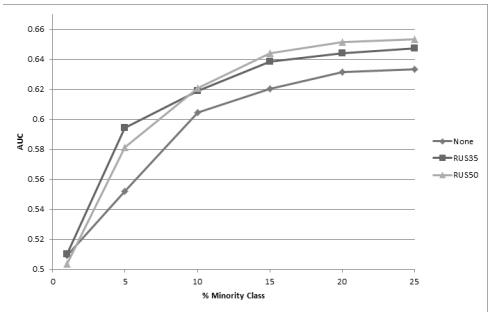


Table III: ANOVA Results: ANOVA

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Data Sampling	0.1903	2	0.09517	14.76	4.18027e-07
Learner	2.9944	7	0.42777	66.36	5.59068e-89
Error	4.69245	957	0.0049		
Total	4.8394	959			

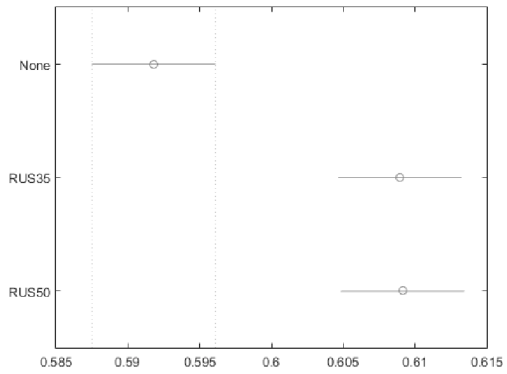
The 5:95 imbalanced dataset represents data with more severe class imbalance, containing only 5% (or 150) positive sentiment instances. Due to the high level of class imbalance, all learners have lower performance than is observed on less imbalanced datasets, since there are less available negative instances. Once again SVM achieves the highest AUC for learners when no sampling is used, and does not benefit from sampling. Unlike previous datasets, performance for all seven remaining learners is improved by both RUS50 and RUS35. RUS50 achieves higher performance for MLP, while all other learners perform better with RUS35. The higher level of class imbalance accounts for improvements observed for MLP and 5-NN when using RUS. Again NB is the best performing learner when using either RUS50 or RUS35, and achieves the highest overall AUC when using RUS35.

All learners performed poorly on the 1:99 imbalanced dataset and the results for this dataset are inconsistent with the trends and patterns observed for the other five. While RUS35 performs slightly better on average than using no data sampling, RUS50 performs slightly worse than using no data sampling. The inconsistency is likely due to an insufficient number of minority instances being present with which to train classifiers and very few instances remaining after resampling. Only 48 instances are used to train classifiers using RUS50, and 66 for those using RUS35. Due to this, no conclusions can be drawn from the results of this dataset.

Figure 1 plots the AUC (averaged across all learners) of each sampling technique, for each level of class imbalance. It can be observed that increasing the level of class imbalance has a greater impact on classifiers trained with no data sampling technique compared to those trained using RUS. For datasets with 15-25% minority class ratios, RUS50 performs better than RUS35, while for 5% minority class RUS35 performs better. RUS35 appears to overtake RUS50 once the minority class falls below 10%. For these five datasets there is little difference in performance between RUS35 and RUS50. As discussed above, the results for 1% minority class are not meaningful since there are insufficient minority instances with which to train the classifiers.

The statistical significance of the different sampling techniques was tested by performing two-way ANOVA [7] with a 5% confidence level using MATLAB. The results are presented in Table III and show that choice of RUS50, RUS35 or no sampling technique is a significant factor in determining classifier performance for both datasets. Tukey's HSD test [2] was also conducted to compare the performance of each technique averaged across all learners, the results of which are shown in Figure 2. In the figure, the average performance for the three sampling techniques (None, RUS50, and RUS35) are shown, along with their confidence interval. When comparing two techniques, if there is no overlap of the confidence intervals, then the averages are significantly different. It can be observed that that both RUS35 and RUS50 significantly improve classifier performance on class imbalanced data; however, while RUS50 has slightly higher performance than RUS35, this difference is not significant. Additional Tukey's HSD test (not presented due to space constraints) were conduced for each level of imbalance. No significant difference was found between RUS35 and RUS50 for any level of imbalance.

Figure 2: Tukey's HSD Results: Data Sampling



VI. Conclusion

In our study, we examined the impact six different levels of class imbalance had on the performance of tweet sentiment classifiers by comparing the performance of eight learners on six datasets constructed from the sentiment140 corpus, each containing 3000 instances. 1 %, 5% 10%, 15%, 20%, and 25% minority class imbalanced datasets were used to measure the impact of imbalance. For all learners, class imbalance decreased classifier performance, with greater imbalance leading to worse performance.

Random undersampling was implemented in an effort to improve performance on class imbalanced data. Two post-sampling class distribution ratios were tested, 35:65 and 50:50. We found that RUS improves classifier performance on all datasets, and has a greater impact on severely imbalanced data. RUS routinely improved performance for five of

the eight classifiers, and improved performance for seven of the eight learners on the 5% minority dataset. Unfortunately results for the 1% minority dataset are inconclusive, as this dataset contains an insufficient number of minority instances with which to train a classifier. For all imbalanced datasets (excluding the 1% minority dataset), RUS combined with NB learner yielded the best performance. The statistical significance of our results was tested and confirmed using ANOVA analysis and Tukey's HSD test. These tests showed that RUS significantly improves performance compared to using no data sampling, however there is no significant difference between RUS50 and RUS35 for our data at any level of class imbalance.

We conclude that RUS can be used to significantly improve classifier performance for most learners when conducting tweet sentiment classification. Based on our experimental results, we recommend using RUS50 for tweet sentiment datasets with low to moderate levels of class imbalance, while using RUS35 for more severe levels of imbalance. For both datasets, the best results were achieved using NB. Future work should investigate random undersampling on larger tweet sentiment datasets, in an effort to better investigate how resampling ratio effects performance for differents level of class imbalance.

ACKNOWLEDGEMENT

The authors gratefully acknowledge partial support by the National Science Foundation, under grant number CNS-1427536. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] (2015, Jun.) Twitter usage statistics. [Online]. Available: http://www.internetlivestats.com/twitter-statistics//

[2] H. Abdi and L. J. Williams, "Tukeys honestly significant difference (hsd) test," *Encyclopedia of Research Design. Thousand Oaks, CA: Sage*, pp. 1–5, 2010.

[3] A. Agrawal and A. An, "Kea: Sentiment analysis of phrases within short texts," *SemEval 2014*, p. 380, 2014.

[4] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[5] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[6] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *The Journal of machine learning research*, vol. 3, pp. 1289–1305, 2003.

[7] R. J. Freund and R. C. Littell, *SAS for linear models: a guide to the ANOVA and GLM procedures*. Sas Institute, 1981, vol. 1.

[8] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, pp. 1–12, 2009.

[9] A. Hassan, A. Abbasi, and D. Zeng, "Twitter sentiment analysis: A bootstrap ensemble framework," in *Social Computing (SocialCom), 2013 International Conference on*. IEEE, 2013, pp. 357–364.

[10] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, Jan 2002. [Online]. Available: http://iospress.metapress.com/content/MXUG8CJKJYLNK3N0

[11] S. Li, Z. Wang, G. Zhou, and S. Y. M. Lee, "Semi-supervised learning for imbalanced sentiment classification," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 3, 2011, p. 1826.

[12] Y. Liu, X. Huang, A. An, and X. Yu, "Arsa: a sentiment-aware model for predicting sales performance using blogs," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 607–614.

[13] C. Meador and J. Gluck, "Analyzing the relationship between tweets, box-office performance and stocks," *Methods*, 2009.

[14] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[15] J. D. Prusa, T. M. Khoshgoftaar, and D. J. Dittman, "Impact of feature selection techniques for tweet sentiment classification," in *Proceedings of the 28th International FLAIRS conference*, May 2015, pp. 299–304.

[16] N. F. Silva, E. R. Hruschka, and E. R. Hruschka Jr, "Biocom usp: Tweet sentiment analysis with adaptive boosting ensemble," *SemEval 2014*, p. 123, 2014.

[17] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 935–942. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273614

[18] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 2012, pp. 115–120.

[19] G. M. Weiss and F. J. Provost, "Learning when training data are costly: the effect of class distribution on tree induction," *J. Artif. Intell. Res.(JAIR)*, vol. 19, pp. 315–354, 2003.

[20] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Morgan Kaufmann, 2011.