Stability of Filter- and Wrapper-Based Feature Subset Selection

Randall Wald, Taghi M. Khoshgoftaar, Amri Napolitano
Florida Atlantic University
Email: {rwald1, khoshgof}@fau.edu, amrifau@gmail.com

***Abstract*—High dimensionality (too many features) is found across many data science domains. Feature selection techniques address this problem by choosing a subset of features which are more relevant to the problem at hand. These techniques can simply rank the features, but this risks including multiple features which are individually useful but which contain redundant information; subset evaluation techniques, on the other hand, consider the usefulness of whole subsets, and therefore avoid selecting redundant features. Subset-based techniques can either be filters, which apply some statistical test to the subsets to measure their worth, or wrappers, which judge features based on how effective they are when building a model. One known problem with subset-based techniques is stability: because redundant features are not included, slight changes to the input data can have a significant effect on which features are chosen. In this study, we explore the stability of feature subset selection, including two filter-based techniques and five choices for both the wrapper learner and the wrapper performance metric. We also introduce a new stability metric, the modified Kuncheva's consistency index, which is able to compare two feature subsets of different size. We also consider both the stability of the feature selection technique and the average/standard deviation of feature subset size. Our results show that the Consistency feature subset evaluator has the greatest stability overall, but CFS (Correlation-Based Feature Selection) shows moderate stability with a much smaller standard deviation of feature subset size. All of the wrapper-based techniques are less stable than the filter-based techniques, although the Naïve Bayes learner using the AUC performance metric is the most stable wrapper-based approach.**

***Keywords*-Feature selection, feature subsets, stability, filters, wrappers**

I. Introduction

Many datasets have a very large number of features, and thus suffer from the problem of high dimensionality. Feature selection algorithms designed to address this problem generally fall into one of two categories: rankers or subset evaluators. Rankers consider each feature individually, while subset evaluators consider whole subsets at a time. Although rankers are more efficient, only subset evaluation can reduce feature redundancy, by finding subsets which have more unique features, even if those features may not perform as well individually. Subset evaluators can themselves be divided into filter- vs. wrapper-based subset evaluation. Filter-based techniques use statistical tests to decide on subset scoring, while wrappers use the feature(s) being tested to build a classification model (using a chosen learner) and then use the performance of this model (measured with a chosen performance metric) to grade the feature(s). Filters are generally faster (because they do not require building a full classification model), but wrappers are able to determine exactly which features will perform best for the actual problem of building a classification model.

While feature selection techniques are often evaluated in terms of how they can help improve classification performance, another important metric is stability, the ability of a technique to produce similar results even in the face of changes to the data. Stability is an especially challenging problem for subset evaluation because the effort to reduce redundancy can also directly reduce stability: if two features are both very useful but also very similar, slight changes to the data may change their order. One challenge for considering the stability of feature subset selection is that many stability metrics assume that the feature subsets being compared are identical in size. While this is usually true for feature rankers, as the top N features may be chosen from all lists, subset selection techniques usually include their own stopping criteria, and choose a certain number of features based on which performs best on the actual data. Thus, metrics must be used which can handle a varying number of features. In this paper, we propose the modified Kuncheva's consistency index, which extends the original to handle feature subsets of unequal size.

As no previous work has considered the stability of both filter-based subset evaluation and wrapper-based subset selection on the same dataset, we perform a case study with our proposed stability metric using data from the domain of social network profile mining. We test two filter-based techniques, along with five learners and five performance metrics for wrapper-based selection, and we use four levels of perturbation for stability analysis. We also report both the stability of each technique and the average and standard deviation of the number of features it selects, to demonstrate how the number of chosen features can affect stability. We find that the Consistency-based filter produces the smallest feature subsets but is also the most stable technique, while the CFS-based filter is the second-most stable with slightly larger and much more consistent (in terms of standard deviation) feature subset sizes, leading to our recommendation of CFS over Consistency. The Naïve Bayes learner using the AUC (Area Under the Receiver Operating Characteristic (ROC) Curve) performance metric produces the most stable wrapper, although this is still less stable than either filter.

1082-3409/13 $31.00 © 2013 IEEE
DOI 10.1109/ICTAI.2013.63

Also, the most stable wrapper learners are also those which select the most features, despite the Consistency-based filter (which is the most stable technique overall) producing very small feature subsets. Overall, the relationship between feature subset sizes and the stability of feature subset selection merits further research, as do the more general questions about how to choose a feature subset selection technique which will give reliable results.

The remainder of this paper is organized as follows: Section II presents related work on the topic of feature subset selection stability. Section III discusses our methods and techniques, including the newly-proposed modified Kuncheva's consistency index. In Section IV, we explain our case study. Section V contains our results and discussion of these results. Finally, Section VI holds our conclusions and suggestions for future research.

II. Related Work

Feature selection has been heavily studied for many years, although due to their increased complexity, filter-based subset evaluation and wrapper-based subset selection have been less well-studied [8]. Evaluating the stability of these techniques has received even less focus. He and Yu [5] reviewed causes of instability and stability metrics, including metrics which may be applied towards feature subset selection. Somol and Novovičová [11] also evaluated stability metrics, using both simulated and real data to observe how different metrics can give different results as well as how three wrapper-based techniques (using a Bayesian classifier, 3-Nearest Neighbor, and Support Vector Machines) compare to one another. Yu et al. [19] proposed both a new filter-based feature subset evaluation technique and a new stability metric for evaluating such techniques, both based on the idea of feature groups (selecting a collection of feature subsets, where the features within each subset are expected to be redundant with one another, rather than simply selecting individual features). Lustgarten et al. [9] proposed a new stability metric for feature subset evaluation (based on Kuncheva's consistency index, but extended somewhat differently than the modified Kuncheva's consistency index proposed in the present work), and compare this metric with the Jaccard index using three wrapper-based subset selection techniques (Logistic Regression, Naïve Bayes, and SVM). Dunne et al. [2] considered wrappers using a 3-nearest neighbor learner and three choices of search technique, evaluating stability by resampling the original dataset and finding the Hamming distance between the various feature subset masks. Haury et al. [4] evaluated a number of different feature selection techniques (primarily rankers, but including one wrapper-based subset evaluation technique using least squares regression) and consider stability in terms of how many features are in common between two subsets generated from independent subsamples of the original data. Overall, no work has considered both filter-based subset evaluation and wrapper-based subset selection at the same time, or has examined a wide range of both learners and performance metrics within the context of wrapper-based feature selection.

The large number of users on Twitter makes it a major target for agencies seeking to create buzz for a product or service, and thus automatically-generated advertising messages have become a significant problem for Twitter users [16]. This has led to research focused on detecting spam on Twitter, using techniques including traditional classifiers [10], considering the network relationships between the sender and receiver [12], and evaluation of the URLs being promoted by the spammers [15].

In 2011, the Web Ecology Project began their Socialbot Challenge [6] to promote the study of Twitter social bots. Three teams created bot clusters in order to elicit real users to follow and reply to their bots, accruing points based on how many users interacted with the "lead" bot on each team. Over the course of the two-week challenge, the top team was able to accumulate approximately 8 followers per day and 14 replies per day, with the latter metric far outweighing the other two competitors.

III. Methods

In this work, we introduce the modified Kuncheva's consistency index, and we also employ a number of different filter- and wrapper-based feature selection approaches, along with our stability testing framework. The details of our feature selection algorithms are discussed in Section III-A, while details pertaining to stability measurement are found in Section III-B.

A. Feature Selection

Two classes of feature selection are explored in this work: filter-based subset evaluation and wrapper-based subset evaluation. Both of these operate on the principle of evaluating various subsets, deciding how good they are according to some metric and then using this information to find an ideal subset. The distinction is that the filter-based methods do not employ a learner to perform this evaluation, while the wrapper-based methods do. In either case, a key aspect of feature selection is the search strategy which determines how the subsets are created in the first place. For all experiments in this work (both filter-based and wrapper-based), we use Forward Greedy Stepwise as our subset search algorithm. This begins with the empty set as the "working" set of features and progressively attempts to add exactly one feature at a time into the working set. That is, the set including both the current working set and exactly one feature not in that set is considered, and in fact all features not yet in the working set are tested this way. The feature which most improves performance when temporarily added to the working set becomes a permanent member, and the algorithm iterates until no feature actually improves performance.

1) Filter-Based Feature Subset Evaluation: Two forms of filter-based subset evaluation are used to find the quality of the various feature subsets, Correlation-Based Feature Selection (CFS) [3] and Consistency [1]. The first of these uses the Pearson correlation coefficient in order to balance the need to have the features correlate with the class and the need to have the features not correlate with one another. The second metric, Consistency, seeks to find the largest possible feature subset which is "consistent" (defined as "does not result in two or more instances sharing all the same values for the given features but differing in their class labels"). Due to space limitations, we refer interested readers to the cited works for further detail on these algorithms.

2) Wrapper-Based Feature Subset Evaluation: Conceptually, wrapper-based feature subset evaluation is very simple: the chosen feature subset is used as the basis for a classification model, and the performance of this model is then used as the score for that feature subset. This does leave two key questions unanswered, however: which learner will be used for building the model, and which performance metric will be used to evaluate the model? In this paper, five diverse learners are used: 5-Nearest Neighbor (5-NN), Logistic Regression (LR), Multi-Layer Perceptron (MLP), Naïve Bayes (NB), and Support Vector Machines (SVM). Because these are each well-understood techniques, we provide only a brief discussion of how they predict class labels; an interested reader may consult Witten and Frank [18] for further information. All models were built using the WEKA Machine Learning Toolkit [18], using the default parameter values unless otherwise noted.

5-Nearest Neighbor classifies instances by finding the five closest instances to the test instance and comparing the total weight of the instances from each class (using 1/Distance as the weighting factor). Logistic Regression builds a simple logistic model using all of the features in order to predict the class variable. Multilayer Perceptron builds an artificial neural network with three nodes in its single hidden layer, and 10% of the data held aside in order to validate when to stop the backpropagation procedure. Naïve Bayes uses Bayes' Theorem to determine the posterior probability of membership in a given class based on the values of the various features, assuming that all of the features are independent of one another. Finally, Support Vector Machines finds a maximal-margin hyperplane which cuts through the space of instances (such that instances on one side are in one class and those on the other side are in the other class), choosing the plane which preserves the greatest distance between each of the classes. In this paper, for SVM the complexity constant "c" was set to 5.0 and the "buildLogisticModels" parameter was set to "true."

In addition to the choice of learner, the choice of performance metric can have a significant impact on the wrapper selection process. This is especially true with imbalanced data where one class has more instances than the others. In this work we consider four performance metrics which take class imbalance into account: Area under the Receiver Operating Characteristic (ROC) Curve (AUC), Area Under the Precision-Recall Curve (PRC), Best Arithmetic Mean (BAM), and Best Geometric Mean (BGM). The first two of these plot two metrics against one another as the decision threshold changes (AUC uses True Positive Rate and False Positive Rate; PRC uses Precision and Recall), and then use the area under the resulting curve as a measure of the trade-off between the metrics. The two mean-based metrics consider the mean of True Positive Rate and True Negative Rate; they differ in whether the arithmetic or geometric mean is used. Finally, for comparison, we also consider using overall accuracy as a metric.

Finally, when building the models we use 5-fold cross-validation to avoid the problems of training and testing on the same instances. Each time the wrapper technique begins to build a model using a specified dataset and feature subset, the dataset is broken into five folds, and the model is built using the first four folds (and is then tested on the final hold-out fold). This process is repeated a total of five times, such that each fold is the hold-out fold once. Cross-validation itself is only performed once; more repetitions of cross-validation (or more folds within cross-validation) are not used because wrappers are already quite time-consuming.

B. Stability Measurement

In order to measure stability, we need both a system for producing the diversity which will create the various feature subsets for comparison, as well as a measure to actually compare these subsets. For this experiment, the former is accomplished through perturbation. A fraction of the instances (either 95%, 90%, 80%, or 67%) from the original dataset are randomly chosen (without replacement, and with preserving the original class ratio), and this procedure is repeated 30 times to create 30 subsamples of the original data per perturbation level. Feature selection is applied independently on each of these 30 subsamples, creating 30 feature subsets which are then compared pairwise, resulting in $(30 \cdot 29)/2 = 435$ pairings. The stability measure is used to find a score for each pairing, and the average value is taken as the stability measure for the full collection.

The stability measure we use in this paper is a modified version of Kuncheva's consistency index [7], hence our referring to it as the "modified Kuncheva's consistency index." The original motivating idea behind the Kuncheva's consistency index was to find how close the observed level of overlap between the feature subsets is to the maximum, taking into account the amount of overlap which would be expected by chance:

$$\frac{\text{Observed overlap} - \text{Expected overlap}}{\text{Maximum overlap} - \text{Expected overlap}} \tag{1}$$

In Kuncheva's original formulation, it was assumed that the two feature subsets A and B being compared have the same number of features, k, and there are exactly r features in common; also, there are a total of n features in the original dataset. If the two subsets were chosen randomly, we would expect any one feature should have a $\frac{k}{n}$ chance of being in either subset, and thus of the k features in the first subset, $\frac{k^2}{n}$ will also be in the second subset (and thus, the expected overlap is $\frac{k^2}{n}$). Inserted into Equation 1, this gives us:

$$I_C(A,B) = \frac{r - \frac{k^2}{n}}{k - \frac{k^2}{n}} = \frac{rn - k^2}{k(n-k)} \tag{2}$$

where $I_C(A,B)$ is defined as the consistency index between feature subsets A and B. This metric takes a maximum value of 1 when the overlap between the subsets is at a maximum, a value of 0 when their overlap is equal to what would be expected by random chance, and a minimum value of -1 when the lack of overlap would actually be statistically unlikely due to random chance (e.g., if the two subsets each comprise half of the original feature collection, but nonetheless have no overlap whatsoever).

Working with the same principles as Equation 1, we can extend this for the case where A and B have two different feature subset sizes, k_1 and k_2. First, we assume without loss of generality that $k_1 > k_2$. We find that by the same argument as above, the expected overlap between the two feature subsets is $\frac{k_1 k_2}{n}$, which produces an equation much like Equation 2:

$$MI_C(A,B) = \frac{r - \frac{k_1 k_2}{n}}{k_2 - \frac{k_1 k_2}{n}} = \frac{rn - k_1 k_2}{k_2(n-k_1)} \text{ for } k_2 < k_1 \tag{3}$$

The primary distinction between Equation 2 and Equation 3 is that the squared term on top becomes the product of the two different sizes, and the two k's on the bottom refer to two different sizes. Recall that $k_2 < k_1$, which is why the denominator was initially $k_2 - \frac{k_1 k_2}{n}$: k_2, the smaller of the two feature subset sizes, is the largest possible overlap, and the metric will thus take on a maximum value when the actual overlap reaches this value. It should be noted that the modified Kuncheva's consistency index does not take into account how different k_1 and k_2 are from one another: in fact, if one set is small and the other is large, but the small set is completely contained within the larger one, the modified Kuncheva's consistency index will be 1, its maximum value. Thus care must be taken to consider how variability in feature subset size may be an additional source of instability.

IV. Case Study

The case study for this experiment uses data prepared by the Online Privacy Foundation[1], an organization dedicated to understanding how users interact with social networks and the privacy implications of their actions. In a previous experiment, called the Twitter Big Five Experiment [13], [17] (relevant because it uses almost the same independent features as the present work, although the class value differs), a number of users were solicited to take an online personality survey, rating these users according to the Big Five personality index (Agreeableness, Conscientiousness, Extroversion, Openness, Neuroticism) and the Dark Triad of negative personality traits (Narcissism, Machiavellianism, and Psychopathy).

In addition, three classes of features were extracted from each individual's profile: demographic features, linguistic features, and count-based features. The 22 demographic features covered numeric information which could be extracted directly from each user's profile, such as number of friends, length of self-description, and Klout.com score (a broad measure of engagement on social networking sites). The linguistic features were created by using the Linguistic Inquiry and Word Count package [14] to place the words in each individual's tweets into 80 categories centered around different linguistic contexts. In addition to the full collection of tweets, tweets from different groups (such as tweets originating from the user, retweets, etc.) were also considered separately, resulting in 480 lingustic features. Finally, the number of days wherein the user posted more than a certain number of a certain type of tweet (e.g., more than 40 original tweets, or more than 10 retweets) were considered, along with the maximum and average number of tweets/day for each category; a total of 74 count-based features were created with different numbers and categories.

Following the Twitter Big Five Experiment, a second experiment was performed on the same users, to study their response to social bots. The original pool of subjects was reduced to 610 users, who were then divided into two groups to be studied separately. However, the procedure for both was identical (the only difference being the name of the bot account used, and one group being tested approximately one month prior to the other, to avoid the problem of many users discovering the bot and realizing its nature), and as such these groups are pooled for the present work. For both groups, a Twitter bot was created which performed two tasks: it would post tweets meant to be representative of what normal Twitter users would post, and it would ask specific questions of the users in the group being tested, using Twitter's @ syntax to send these messages at the target users. A user was considered to have interacted with the bot if they replied to this message or if they subsequently chose to follow the bot. Only users who were part of the original

[1] https://www.onlineprivacyfoundation.org/

Feature Selection Approach		Perturbation Level			
		95%	90%	80%	67%
CFS Filter		*0.73410*	*0.61726*	*0.43861*	*0.30226*
Consistency Filter		**0.99230**	**0.84321**	**0.67150**	**0.35114**
Wrapper with AUC Metric	5-NN	0.09796	0.17413	0.08336	0.10300
	LR	0.21118	0.17296	0.14310	0.06720
	MLP	0.10539	0.13465	0.09980	*0.05639*
	NB	**0.33778**	**0.30657**	**0.27941**	**0.19428**
	SVM	*0.05972*	*0.05783*	*0.06184*	0.05822
Wrapper with PRC Metric	5-NN	0.14341	0.11433	0.14231	0.08014
	LR	0.21551	0.19698	0.14497	0.10459
	MLP	*0.11001*	*0.08236*	*0.06552*	*0.06474*
	NB	**0.29348**	**0.20562**	**0.17574**	**0.12346**
	SVM	0.15153	0.10019	0.12511	0.10394
Wrapper with BAM Metric	5-NN	0.09974	0.11187	0.07492	*0.03985*
	LR	**0.17832**	0.13726	0.11457	0.06592
	MLP	0.13511	0.10014	0.08962	0.05694
	NB	0.16880	**0.14405**	**0.13975**	**0.12743**
	SVM	*0.06955*	*0.09244*	*0.06328*	0.04273
Wrapper with BGM Metric	5-NN	0.07361	0.10257	0.07834	0.05206
	LR	**0.17063**	0.14982	0.08831	0.07103
	MLP	0.13846	0.12965	0.11022	0.07189
	NB	0.14215	**0.18531**	**0.14646**	**0.12538**
	SVM	*0.05619*	*0.07694*	*0.04702*	*0.03605*
Wrapper with Overall Accuracy Metric	5-NN	0.21100	*0.05623*	*0.04082*	*0.02554*
	LR	**0.29193**	**0.28310**	**0.18295**	0.09608
	MLP	*0.03959*	0.06112	0.05963	0.04761
	NB	0.24773	0.22940	0.12528	0.08479
	SVM	0.22508	0.16298	0.16687	**0.09618**

Table I
STABILITY OF FEATURE SUBSET EVALUATION

Twitter Big Five Experiment were considered, even if other users followed (or sent messages to) the bot of their own accord. Thus, for each user, the 576 features were paired with one binary class variable: whether or not that user interacted with the bot.

V. RESULTS

In Table I, we see the average stability results for each subset-based feature selection method, either filter-based or wrapper-based. For each of the 30 runs, the instances were sampled from the original dataset (with the column header specifying the percentage, from 95% to 67%), and the specified feature selection technique was applied to each. In the case of the filter-based subset evaluators, the technique listed in the first column shows which filter was used, while for the wrapper-based subset selection techniques, the first column lists the performance metric used within the wrapper while the second column presents the learner. Once these 30 feature subsets were created, they were compared pairwise (with 435 pairings) using the modified Kuncheva's similarity metric described in Section III. The results were averaged across all pairings, except for the Consistency Index with 67% perturbation, where 110 of the pairings did not return any values (due to one of their subsets being empty); in that case, the remaining 325 pairings were averaged. For each block (either the filter-based block at the top or each choice of wrapper performance metric), the largest (e.g., most stable) value in a given column is presented in **bold**, while the least stable value is in *italics*.

We also present the average and standard deviation of the number of features chosen across the 30 runs of perturbation in Table II (with the standard deviation values being the ones in parenthesis). As with the previous tables, the blocks have their largest and smallest values per column typeset in **bold** and *italics*, respectively; values for average and standard deviation are considered separately. While these high and low values may not have the same meaning as "best" and "worst," they still serve to highlight notable patterns in terms of feature subset size.

As we can see from these results, the Consistency Subset Evaluation method is significantly more stable than the other techniques, even for the 67% overlap level (where it suffers a significant drop, possibly due to the pairs which cannot be included); in second place is the CFS Subset Evaluation method, and the wrapper based approaches are the least stable. One major source of this distinction is the difference in how many features are selected by these three techniques: due to the nature of the modified Kuncheva's similarity metric, if one subset has a very small number of features while the other has more features (but includes all features from the smaller subset), it will give a perfect stability score. In fact, we see that Consistency Subset Evaluation has standard deviation values always exceeding the average number of features chosen, so it is not uncommon for two paired subsets to have sizes which vary by more than 100%. Thus, the greater stability of the Consistency Subset Evaluation may not be indicative of its increased utility. CFS, on the other hand, has lower standard deviation values than Consistency while producing significantly larger subsets; although it is less stable than Consistency, it is more stable than any choice of wrapper, and produces relatively manageable feature subset sizes (in the 10–15 range). Thus, despite the increased stability of Consistency, we feel that CFS gives the best balance of stability and consistency of feature subset sizes.

When we look specifically at the wrapper results, we find additional patterns. First of all, for all performance metrics other than Overall Accuracy, the greatest stability for permutation levels other than 95% is found with the Naïve Bayes learner. Logistic Regression is almost always in second place for stability, and in fact is best for the 95% permutation level when Naïve Bayes isn't. Logistic Regression is also best when using the Overall Accuracy metric (except for the 67% permutation level, where SVM is best). It is notable that both Naïve Bayes and Logistic Regression tend to produce the largest feature subset sizes, especially with the AUC and PRC metrics; this result is at odds with the observation among the filters that the technique producing the fewest features is most stable. However, the relatively high standard deviation values for NB and LR when using the AUC and PRC performance metrics suggest that a similar effect may be in play, where feature subsets that significantly differ in size are giving a deceptively high stability reading. Also, this result may arise from larger feature subsets being more likely to include the same high-performing features, which may suggest the

Feature Selection Approach		Perturbation Level			
		95%	90%	80%	67%
CFS Filter		**13.76667** (*2.40235*)	**12.93333** (*2.49044*)	**12.43333** (*3.53976*)	**10.26667** (*4.08473*)
Consistency Filter		*2.50000* (**3.32960**)	*3.00000* (**4.51052**)	*3.53333* (**4.86885**)	*5.40000* (**6.88126**)
Wrapper with AUC Metric	5-NN	6.36667 (3.60539)	6.53333 (3.40115)	*5.96667* (*2.04237*)	7.50000 (4.15020)
	LR	**56.30000** (**12.51248**)	**56.70000** (**10.22556**)	**48.20000** (**11.06532**)	**42.73333** (**12.37610**)
	MLP	9.93333 (*2.77841*)	9.13333 (*3.33976*)	9.30000 (3.10894)	9.90000 (*3.65164*)
	NB	32.50000 (8.66523)	29.96667 (8.33143)	36.46667 (9.52577)	32.53333 (10.31481)
	SVM	*5.26667* (3.05053)	*4.76667* (3.35984)	6.70000 (5.28596)	*7.20000* (5.68361)
Wrapper with PRC Metric	5-NN	*8.46667* (4.41575)	*9.30000* (5.06611)	*8.26667* (*4.32262*)	*7.76667* (*4.26439*)
	LR	**32.46667** (**10.60167**)	**32.06667** (9.57343)	**32.30000** (**10.05211**)	**25.26667** (8.11101)
	MLP	9.43333 (*3.08146*)	9.40000 (*2.32824*)	9.46667 (4.51613)	10.03333 (4.32701)
	NB	24.26667 (7.40891)	23.26667 (5.30409)	22.96667 (7.19906)	24.10000 (8.52724)
	SVM	9.60000 (8.67656)	13.46667 (**12.08514**)	9.66667 (9.35998)	10.90000 (**8.65567**)
Wrapper with BAM Metric	5-NN	5.00000 (2.47748)	*4.96667* (2.65854)	*4.16667* (*1.36668*)	*4.76667* (2.80004)
	LR	**11.10000** (**3.34612**)	**9.83333** (2.81723)	8.96667 (2.79758)	**10.00000** (**3.41397**)
	MLP	6.63333 (2.65854)	6.86667 (2.73840)	6.53333 (2.27025)	6.70000 (*2.40903*)
	NB	10.30000 (2.64119)	9.63333 (**4.42160**)	**10.66667** (**4.09654**)	9.50000 (3.33994)
	SVM	*4.06667* (*1.36289*)	5.26667 (*2.14851*)	5.86667 (3.59821)	5.60000 (3.15791)
Wrapper with BGM Metric	5-NN	4.86667 (2.33021)	4.90000 (2.36862)	*4.13333* (*1.45586*)	*5.40000* (*2.48582*)
	LR	**11.46667** (3.02556)	**11.30000** (2.57508)	**10.60000** (3.00115)	9.53333 (2.97962)
	MLP	6.06667 (2.39156)	6.30000 (2.43749)	5.96667 (2.45628)	6.86667 (2.68756)
	NB	10.53333 (**3.75760**)	9.13333 (**4.04060**)	9.13333 (**3.22419**)	**9.80000** (**4.34225**)
	SVM	*4.43333* (*1.97717*)	*4.40000* (*1.42877*)	4.43333 (1.85106)	5.70000 (2.96124)
Wrapper with Overall Accuracy Metric	5-NN	*3.56667* (*1.97717*)	3.53333 (1.99540)	*2.70000* (*1.46570*)	*3.40000* (*1.30252*)
	LR	6.00000 (2.30442)	**6.46667** (**2.47377**)	6.63333 (2.55266)	5.70000 (1.96784)
	MLP	3.80000 (**2.75931**)	*3.16667* (2.33538)	3.93333 (2.30342)	4.20000 (**2.39828**)
	NB	**6.23333** (2.09570)	5.80000 (2.38385)	**6.70000** (**3.31298**)	**5.96667** (2.14127)
	SVM	3.80000 (2.18774)	3.40000 (*1.97571*)	3.83333 (2.15092)	3.53333 (1.69651)

Table II
AVERAGE AND STANDARD DEVIATION OF FEATURE SUBSET SIZE FOR FEATURE SUBSET EVALUATION

existence of a "pessimal" feature subset size in terms of stability: a size where techniques cannot reliably select the top 1–3 features, but also don't select enough for chance to include most of the high-performing features.

As for poor stability, MLP and SVM tend to show the worst performance with the non-Overall Accuracy metrics: with PRC and one permutation level of AUC, MLP is worst, and for three permutation levels of AUC and BAM as well as all levels for BGM, SVM is worst. 5-NN has the worst stability in the one remaining combination, but generally falls between the bottom two and top two in terms of stability as long as Overall Accuracy is not the performance metric (if it is, 5-NN is usually worst). Thus, overall we would recommend Naïve Bayes as the most stable learner for use within a wrapper-based feature selection technique, unless the Overall Accuracy metric will be used (which is generally a bad idea when working with imbalanced data such as in our case study). Also, although LR is generally second-place in terms of stability, its unusually large feature subset sizes and correspondingly large standard deviation values are sufficiently suspicious to not merit recommendation.

The performance metrics themselves show notable differences, especially in terms of feature subset sizes. We find that AUC and PRC lead to significantly larger feature subsets, especially with the Naïve Bayes and Logisitic Regression learners, while Overall Accuracy produces extremely small feature subsets. This connection seems to also affect subset stability: the top two learners tend to give their best results for the AUC and PRC metrics, although the results are less clear-cut for the other learners. Overall Accuracy is a special case: because the dataset is known to be imbalanced, this metric has the potential to disregard the most important class in the dataset. Perhaps because of this change, it also shows abnormal behavior: its feature subsets are smallest among the various metrics, and unlike with the other metrics Naïve Bayes is often not the first or second-place choice in terms of stability. It also exhibits the pattern of most learners showing decent stability at the 95% perturbation level and then suddenly dropping by a large margin at one of the other perturbation levels; only MLP is immune to this precipitous drop, by virtue of starting out low. Generally, we see that Overall Accuracy has unusual results which make it unreliable, while AUC and PRC give the largest and most stable feature subsets for the top two learners; with these two facts combined, we would recommend Naïve Bayes with AUC as the most stable wrapper-based approach, or perhaps Naïve Bayes with PRC to slightly temper the large feature subset sizes while not sacrificing greatly in terms of stability.

VI. CONCLUSION

In this paper, we propose a new feature selection stability metric, the modified Kuncheva's consistency index, to measure the stability of feature subset selection techniques and then apply this to find the stability of both filter- and wrapper-based feature selection. Our primary findings are that the Consistency-based filter subset evaluation technique is the most stable overall, with the CFS-based filter in second place and all wrappers being less stable than all filters. Consistency was also noted for having the smallest feature subset sizes, although this suggests that the increased stability for this technique might result from extremely small feature subsets (with just one or two features) which happen to be wholely contained within the subsets chosen from other perturbations of the data. CFS, on the other hand, produces relatively stable feature subsets with larger and more consistent feature subset sizes.

As for the wrapper-based techniques, we find that Naïve Bayes and Logistic Regression produce the largest and most stable feature subsets, especially when using the AUC and PRC performance metrics. Conversely, SVM and MLP produce the least stable feature subsets, at least for four of the performance metrics; with Overall Accuracy, however, 5-NN becomes the worst, despite usually being the middle learner in terms of stability. This is consistent with other unusual results when using Overall Accuracy, which suggests that this metric should be avoided when working with imbalanced data. Overall, we find that CFS is the best all-around subset selection technique due to its stability and low feature subset size standard deviation, although Consistency is the most stable; Naïve Bayes with AUC is the most stable wrapper, but Naïve Bayes with PRC can help reduce the feature subset sizes somewhat without hurting stability too much.

Looking towards the future, we feel these techniques could be compared using a wider range of stability measures, in order to evaluate how the choice of measure can impact the conclusions. In addition, these results could be extended by considering a wider range of learners and metrics within the wrapper and by considering datasets with different properties to ensure the generality of these results.

References

[1] M. Dash, H. Liu, and H. Motoda, "Consistency based feature selection," in *Knowledge Discovery and Data Mining. Current Issues and New Applications*, ser. Lecture Notes in Computer Science, T. Terano, H. Liu, and A. Chen, Eds. Springer Berlin Heidelberg, 2000, vol. 1805, pp. 98–109.

[2] K. Dunne, P. Cunningham, and F. Azuaje, "Solutions to instability problems with sequential wrapper-based approaches to feature selection," *Journal of Machine Learning Research*, 2002.

[3] M. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, Hamilton, New Zealand, April 1997.

[4] A.-C. Haury, P. Gestraud, and J.-P. Vert, "The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures," *PLoS ONE*, vol. 6, no. 12, p. e28210, 12 2011.

[5] Z. He and W. Yu, "Stable feature selection for biomarker discovery," *Computational Biology and Chemistry*, vol. 34, no. 4, pp. 215–225, 2010.

[6] T. Hwang. (2011) Help robots take over the internet: The socialbots 2011 competition : Web ecology project.

[7] L. I. Kuncheva, "A stability index for feature selection," in *Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications*. Anaheim, CA, USA: ACTA Press, 2007, pp. 390–395.

[8] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, April 2005.

[9] J. L. Lustgarten, V. Gopalakrishnan, and S. Visweswaran, "Measuring stability of feature selection in biomedical datasets," in *AMIA 2009 Annual Symposium Proceedings*, 2009, pp. 406–410.

[10] M. McCord and M. Chuah, "Spam detection on Twitter using traditional classifiers," in *Autonomic and Trusted Computing*, ser. Lecture Notes in Computer Science, J. M. A. Calero, L. T. Yang, F. G. Mármol, L. J. García Villalba, A. X. Li, and Y. Wang, Eds. Springer Berlin Heidelberg, 2011, vol. 6906, pp. 175–186.

[11] P. Somol and J. Novovičová, "Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 1921–1939, 2010.

[12] J. Song, S. Lee, and J. Kim, "Spam filtering in Twitter using sender-receiver relationship," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science, R. Sommer, D. Balzarotti, and G. Maier, Eds. Springer Berlin Heidelberg, 2011, vol. 6961, pp. 301–317.

[13] C. Sumner, A. Byers, R. Boochever, and G. Park, "Predicting dark triad personality traits from Twitter usage and a linguistic analysis of tweets," in *11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, 2012, pp. 386–393.

[14] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, 2010.

[15] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2011, pp. 447–462.

[16] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect: an analysis of twitter spam," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 243–258.

[17] R. Wald, T. M. Khoshgoftaar, A. Napolitano, and C. Sumner, "Using Twitter content to predict psychopathy," in *11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, 2012, pp. 394–401.

[18] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical machine learning tools and techniques*, 3rd ed. Burlington, MA: Morgan Kaufmann, January 2011.

[19] L. Yu, C. Ding, and S. Loscalzo, "Stable feature selection via dense feature groups," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 803–811.