Journal of the Association for Information Systems

Volume 11 | Issue 9 Article 1

9-28-2010

A Hybrid Attribute Selection Approach for Text Classification

Chen-Huei Chou
College of Charleston, chouc@cofc.edu

Atish P. Sinha
University of Wisconsin - Milwaukee, sinha@uwm.edu

Huimin Zhao
University of Wisconsin - Milwaukee, hzhao@uwm.edu

Recommended Citation
Chou, Chen-Huei; Sinha, Atish P.; and Zhao, Huimin (2010) "A Hybrid Attribute Selection Approach for Text Classification," *Journal of the Association for Information Systems*, 11(9), .
DOI: 10.17705/1jais.00236
Available at: https://aisel.aisnet.org/jais/vol11/iss9/1

Journal of the Association for Information Systems
JAIS

Research Article

A Hybrid Attribute Selection Approach for Text Classification

Chen-Huei Chou
College of Charleston
chouc@cofc.edu

Atish P. Sinha
University of Wisconsin-Milwaukee
sinha@uwm.edu

Huimin Zhao
University of Wisconsin-Milwaukee
hzhao@uwm.edu

Abstract

The application of text mining in organizations is growing. Text classification, an important type of text mining problem, is characterized by a large attribute space and entails an efficient and effective attribute selection procedure. There are two general attribute selection approaches: the filter approach and the wrapper approach. While the wrapper approach is potentially more effective in finding the best attribute subset, it is cost-prohibitive in most text classification applications. In this paper, we propose a hybrid attribute selection approach that is both efficient and effective for text classification problems. We apply the proposed approach to detect and prevent Internet abuse in the workplace, which is becoming a major problem in modern organizations. The empirical evaluations we conducted using a variety of classification algorithms, indexing schemes, and attribute selection methods demonstrate the utility of the proposed approach. We found that combining the filter and wrapper approaches not only boosts the accuracies of text classifiers but also brings down the computational costs significantly.

Keywords: *text mining, text classification, data mining, attribute selection, Internet abuse detection*

Volume 11, Issue 9, pp. *491-518*, September 2010

* Jeffrey Parsons was the accepting senior editor. Tao Chen, Gautam Pant, and Yilu Zhou were the reviewers. This article was submitted on Mach 25, 2009 and went through two revisions.

A Hybrid Attribute Selection Approach for Text Classification

1. Introduction

As organizations are being flooded with massive volumes of textual data—such as written documents, web pages, and emails—several of them have started to apply text mining techniques to sift through the unstructured or semi-structured data and discover useful patterns and models (Fan et al. 2006). Text mining and data mining utilize similar machine learning techniques, but work with different types of data (unstructured/semi-structured vs. structured). *Text classification* is an important type of text mining problem, where the *class* (a categorical dependent variable) of a document is predicted based on several attributes (independent variables) describing the document. Examples of text classification include junk e-mail filtering (Sakkis et al. 2003; Schneider 2003), web page classification (Chen and Hsieh 2006; Kwon and Lee 2003), anticipatory event detection (He et al. 2007), and online deception detection (Zhou et al. 2004). Internet abuse detection is another domain where text classification techniques can be applied. Various machine learning techniques can be used to automatically learn classification models (called *classifiers*) based on training examples with known cases of *abuse* and *non-abuse*. The learned classifiers can then be applied to predict the classes of new documents.

Support vector machine (SVM) has been found to be one of the most accurate text classifiers across the board for a large number of existing document collections (Chakrabarti 2003). But, as we argue in this paper, accuracy is only one of the performance measures for text classifiers, and there are other measures—such as attribute selection time, classifier training time, and classifier testing time—that are equally, if not more, important. The question that arises, then, is if it is possible to boost the performance of other classifiers to bring them closer to SVM accuracy levels. To address that question, we propose a hybrid attribute selection approach that combines the filter and wrapper approaches.

Attribute selection (also called feature selection)—i.e., selecting a subset of the attributes (features) that are most relevant to a classification problem—is a common preprocessing step. There are two general attribute selection approaches: the *filter* approach and the *wrapper* approach (Dash and Liu 1997; Hall and Holmes 2003; Witten and Frank 2005). In the filter approach, the attributes are evaluated by some relevance measure and filtered without invoking a learning algorithm. In the wrapper approach, the learning algorithm used to build the classifier is wrapped into the attribute selection procedure, so that multiple classifiers can be generated based on different subsets of attributes, and the subset that results in the best performance can be selected. The filter approach is independent of any learning algorithm, while the wrapper approach fundamentally relies on a learning algorithm. The wrapper approach could potentially find a better subset of attributes for a particular learning algorithm than the filter approach does. However, the wrapper approach is much more computationally expensive and may become infeasible when the number of original attributes is very large.

For text classification problems, the attributes are usually weights (defined by an indexing method) of terms that appear in the documents. They are typically characterized by very large numbers (thousands or more) of attributes, thereby rendering attribute selection almost inevitable. Furthermore, as the sheer size of the space of attribute subsets makes the wrapper approach cost-prohibitive, text classification applications are often forced to settle for the filter approach (Sebastiani 2002).

An alternative to selecting an attribute subset (feature selection) is to extract a subset of transformed attributes (feature extraction), which can approximate the original attributes (Sebastiani 2002). Some examples are independent component analysis, principal component analysis, and factor analysis. A feature extraction method developed specifically for text data is latent semantic analysis (Deerwester et al. 1999). It extracts a small number of linearly transformed features through singular value decomposition of the original term-document matrix. Another method for text data is term clustering (Baker and McCallum 1998; Lewis 1992), which tries to group words with a high degree of pairwise

semantic relatedness, so that the groups (or their centroids, or a representative of them), instead of the individual terms, may be used as dimensions of the vector space.

In this paper, we propose a hybrid attribute selection approach that is both efficient and effective for text classification problems. It first applies the filter approach to reduce the full attribute set to a much smaller subset and then applies the wrapper approach to further tune the attribute subset. We apply the proposed hybrid approach to address the organizational problem of Internet abuse and demonstrate empirically the utility of the approach.

The rest of the paper is organized as follows. We first review the text classification and attribute selection literature. We then propose and describe the hybrid attribute selection approach. Next, we describe how we empirically evaluate the proposed approach in the domain of workplace Internet abuse and discuss the findings. Finally, we conclude the paper and outline potential future research directions.

2. Background

The objective of classification is to predict the class (a categorical dependent variable) of a case based on several attributes (independent variables) describing the case. Some examples of classification problems include profiling web usage in the workplace (Anandarajan 2002), generating document taxonomies (Spangler et al. 2003), assessing the risks of prostate cancer patients (Churilov et al. 2005), forecasting financial performance (Walczak 2001), and credit evaluation (Sinha and May 2005). In text classification, a case is usually a text document, such as a written article, an email message, or a web page. Attribute selection is a common preprocessing step in text classification applications. We now review the literature on text classification and attribute selection.

2.1. Text Classification

Text classification is the activity of classifying a text document into one of several pre-defined categories (Sebastiani 2002). Each document to be classified is represented by a vector of attribute values, $\mathbf{x} = \langle x_1, x_2, \ldots, x_m \rangle$, and the class value, y. The attributes are usually term weights (defined by an indexing method). The class is a categorical variable that takes its value from a set of categories $\{c_1, c_2, \ldots, c_n\}$. A given machine learning algorithm is used to learn a prediction model, $\hat{y} = f(\mathbf{x})$, called a *classifier*, from a set of pre-classified training text documents. The trained classifier can be applied to classify other documents in the future. Several machine learning methods, including naïve Bayes, multinomial naïve Bayes, decision tree, neural network, and SVM, have been applied in text classification problems (Sebastiani 2002). A description of these methods can be found in Witten and Frank (2005).

Figure 1 shows the general procedure for text classification. After a set of pre-labeled documents is prepared, several preprocessing steps are applied before machine learning algorithms can be used to learn classifiers. There are two critical preprocessing steps: *indexing* and *dimensionality reduction*.

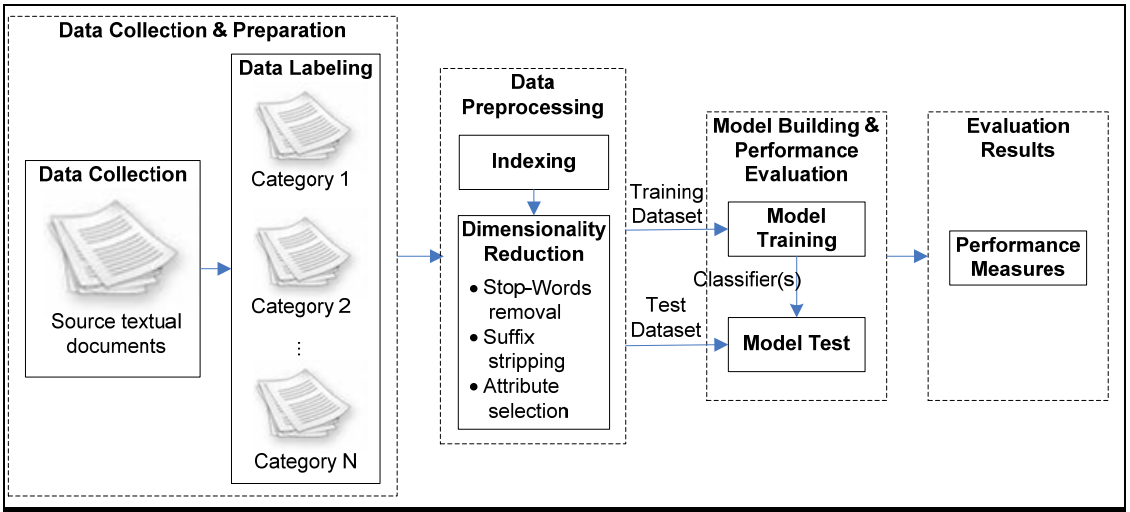

Figure 1. Text Classification Procedure

The indexing step maps a document into a compact representation of its content, consisting of a set of attributes, which correspond with some chosen meaningful units of text. Previous studies have found that words (or *word stems*) work well as representation units, and more sophisticated representations do not significantly improve classification performance (Sebastiani 2002). Using word stems as representation units, a document is described by a vector of term weights $< w_1, w_2, \ldots, w_{|T|} >$, where T is a set of selected word stems (terms) and w_i is the weight of the i-th word stem.

There are various schemes for weighting terms. TF (term frequency) and TFIDF (term frequency/inverse document frequency) are two of the most commonly used weighting schemes in text classification applications. The TF of term t with regard to document d is the number of times t appears in d. TFIDF adjusts TF by inverse document frequency (IDF), where the document frequency (DF) of term t with regard to a document set D is the number of documents in D that contain t. While there are many variants, the standard TFIDF is defined as

$$\text{TFIDF}(t, d, D) = \text{TF}(t, d) \log \frac{|D|}{\text{DF}(t, D)}.$$

Text classification problems typically involve large numbers (normally exceeding thousands) of attributes. Large dimensionality of the attribute space incurs high computational costs and long training times. More importantly, it leads to *overfitting* for many classification methods. Overfitting happens when a classifier fits the training dataset well but does not perform well on cases outside the training dataset. Dimensionality reduction can help reduce computational cost and overfitting (Sebastiani 2002). There are three basic ways to achieve dimensionality reduction: stop-word removal, suffix stripping, and attribute selection. Stop-words are frequent words, such as conjunctions, prepositions, and articles, which are not very useful in discriminating different classes of documents. Words with a common stem usually have similar meanings and can be merged into a single term through a suffix stripping process. For example, “invent,” “invented,” “inventing,” “inventive,” “invention,” and “inventions” can be combined into the same term “invent” by removing the suffixes.

2.2. Attribute Selection

Apart from the simple steps of stop-word removal and suffix stripping, attribute selection is a critical step that can often substantially reduce the number of attributes. The major difference between the two general attribute selection approaches is that while the filter approach is independent of any learning algorithm, the wrapper approach fundamentally depends on the learning algorithm.

In the filter approach, the attributes are evaluated based on some relevance measure, independent of any learning algorithm. In this paper, we use the term "relevance" informally to refer to the degree to which an attribute is relevant to the prediction of the class. For formal definitions of "relevance," please see Avrim and Pat (1997). The relevance measure is designed to measure the dependency between the class and an attribute. Attributes that are deemed most relevant to predicting the class are selected, and the remaining ones are filtered out. As the attributes need to be evaluated only once, the filter approach is computationally efficient. However, since the learning algorithm that is eventually employed for building the classifier is not involved in the process, the selected attributes are not specifically tuned to the learning algorithm used. Furthermore, since the attributes are usually individually evaluated, the selected attributes, when taken together, may not form the best possible subset.

Some examples of relevance measures that have been shown to be effective in text classification applications include *information gain*, *gain ratio*, and *Chi-square* (χ^2) (Sebastiani 2002). Chi-square is a standard statistic that measures the lack of independence between two variables (the class y and an attribute x_i in the current context) (Liu and Setiono 1995). Information gain and gain ratio are used in the C4.5 decision tree learning algorithm as criteria in selecting decision attributes at intermediate nodes (Quinlan 1993).

Information gain measures the amount of uncertainty associated with the class y that can be reduced—or stated differently, the amount of information about the class y that can be gained—given the knowledge of the value of an attribute x_i. The amount of uncertainty associated with the class y is measured by its entropy, defined as

$$Entropy(y) = -\sum_{l=1}^{n} \Pr(y = c_l) \log \Pr(y = c_l).$$

Given a representational sample, $\Pr(y=c_l)$ can be estimated by the proportion of instances in the sample that falls into class c_l. The information gain of an attribute x_i is defined as

$$IG(x_i) = Entropy(y) - Entropy(y \mid x_i),$$

where $Entropy(y \mid x_i)$ is the conditional entropy of y given x_i.

Gain ratio is defined as

$$GR(x_i) = IG(x_i) / Entropy(x_i).$$

$GR(x_i)$ is in the range [0, 1]. It equals zero if and only if y and x_i are independent, and reaches one if and only if there is a one-to-one mapping between y and x_i. Gain ratio takes into account the efficiency of an attribute in reducing the uncertainty on the class. If two attributes lead to the same amount of uncertainty reduction on the class, the one with lower uncertainty itself is favored.

We also considered mRMR (minimum redundancy and maximum relevance) (Ding and Peng 2005) as a relevance measure. mRMR selects a subset of attributes that are mutually unrelated to each other, but are related to the class. It attempts to minimize the average mutual dependency among the selected attributes, $\frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j)$, where S is the subset of attributes the process is seeking, $|S|$ is the number of attributes in the subset S, and $I(x_i, x_j)$ is the mutual information between two attributes x_i and x_j. At the same time, it tries to maximize the average dependency between the selected attributes and the class, $\frac{1}{|S|} \sum_{x_i \in S} I(y, x_i)$.

There are two major components in the wrapper approach: the performance evaluation method and the search method. *Cross-validation* has been shown to be an effective performance evaluation

method (Kohavi and John 1996; Witten and Frank 2005). This method splits the full dataset into *k* approximately equal-sized subsets (called folds), trains a classifier based on *k*-1 subsets and evaluates it based on the remaining subset, repeats the classifier training and evaluation *k* times, and takes the average performance as an estimate.

A search method may search the attribute space greedily in one of two directions: forward or backward. *Forward search* starts without any attribute and adds attributes one at a time until a termination condition is met (e.g., no new attribute leads to further performance improvement when added). Backward attribute elimination starts with a set of attributes and keeps eliminating attributes one at a time until some condition is met. These two search strategies can be combined into a more sophisticated method. For example, the best-first search method keeps an ordered list of attribute subsets evaluated until that point, and can backtrack to a previous subset when the current subset cannot be further improved.

Table 1 summarizes recent studies comparing attribute selection methods. The wrapper approach has been shown to be prohibitively expensive to large datasets with a large number of attributes (Hall and Holmes 2003). When the filter approach is adopted, information gain, gain ratio, and Chi-square have been shown to provide relatively good performance, in comparison to other relevance measures, although the wrapper approach might have given even better performance, time allowing. For example, Yang and Pedersen (1997) reported that information gain and Chi-square were more effective than a few other measures. Debole and Sebastiani (2003) reported that gain ratio and Chi-square outperformed information gain. Forman (2003) found that information gain outperformed 10 other attribute selection methods in most experiments.

Table 1: Prior Studies on Attribute Selection Methods

Reference	Dataset	Initial Number of Attributes	Attribute Selection Method	Number of Attributes Selected/ Tested	Learning Method	Outcome
Debole and Sebastiani (2003)	Reuters-21578		χ^2, IG, GR		SVM	GR and χ^2 outperformed IG
Forman (2003)	229 text classification tasks gathered from Reuters, TREC, and OHSUMED		ACC, ACC2, χ^2, DF, F1, IG, ODDN, OR, POW, PR, RAND	10 to 2,000 initially; 500 and below	SVM, NB, C4.5, LR	IG outperformed other methods in most situations
Hall and Holmes (2003)	18 datasets from the UCI repository	3 datasets with over 100 attributes (227, 293, and 1557 respectively)	CFS, IG, WRP, RLF, CNS, PC	Ranging from 1% to 95% of attributes	C4.5, NB	Wrapper was not applicable on the dataset with 1557 attributes due to time limitation. Wrapper and CFS outperformed other methods on the dataset with 293 attributes by NB.
Lewis and Ringuette (1994)	Reuters-21450 and FBIS	22,791 in Reuters and 8,876 in FBIS	IG	Peak when 10 in Reuters and 15 in FBIS by	PropBayes, DT-min10	Both PropBayes and DT-min10 provided reasonable

				PropBayes; Peak when 90 in Reuters and around 4 to10 in FBIS by DT-min10		performance
Liu (2004)	KDD dataset with 2543 instances	139,351	IG, MI, χ^2, OR, GSS	200, 1,000, 5,000, 10,000, 50,000, and full	NB, SVM	IG and χ^2 were most effective for NB. No benefit for SVM was found.
McCallum and Nigam (1998)	Yahoo Science pages, Industry Sector, Newsgroups, WebKB, Reuters-21578	44,383, 29,964, 42,191, 23,830, and 19,371, respectively	IG	20 to 20,000	MNB, NB	MNB outperformed NB in large attribute set
Mladenic (1998)	HomeNet project two users' data	Around 4,000	IG, OR, WF, RAND	2% to 5% of attributes	NB	OR outperformed other methods
Riboni (2002)	8,000 Web documents	131,730	IG, WF, DF	500 and 1,000	NB, NN	IG>WF>DF
Rogati and Yang (2002)	Reuters-21578 and RCV1		DF, IG, χ^2	3% of full set	NB, kNN, Rocchio, SVM	χ^2 outperformed other methods
Joachims (1998)	Reuters-21578 and OHSUMED	9,947 in Reuters and 15,561 in OHSUMED	IG	1,000 for kNN, Racchio, and C4.5; full for NB	NB, kNN, SVM, C4.5, Racchio	SVM outperformed other classifiers
Liu (2002)	Acute lymphoblastic leukemia (ALL), Ovarian cancer	12,558 in ALL and 15,154 in Ovarian cancer	χ^2, CFS, MIT correlation, Entropy, t-statistic	17 for CFS and 20 for others	kNN, C4.5, NB, SVM, PCL	Entropy was the best, followed by χ^2, on the ALL dataset. CFS outperformed others on the Ovarian cancer dataset
Sebastiani (2002)						Summary of previous studies as {OR, NGL, GSS} > { χ^2, IG} > MI
Yang and Pedersen (1997)	Reuters-22173 and OHSUMED	16,039 in Reuters and 72,072 in OHSUMED	DF, IG, MI, χ^2, TS	321 by IG on Reuters	kNN, LLSF	IG and χ^2 were most effective. Performance improved after attribute selection

Note 1: Abbreviations of attribute selection methods: ACC—Accuracy; ACC2—Accuracy balanced; BNS—Bi-Normal Separation; CFS—Correlation-based Feature Selection; χ^2—chi-square; CNS-Consistency-based; DF—document frequency; F_1—F1 Measure; GSS—GSS coefficient (simplified chi-square); IG—information gain; MI—mutual information; NGL—NGL coefficient; ODDN—odds ratio

numerator; OR—odds ratios; PC—Principal Components; POW—Power; PR—Probability Ratio; RAND—Random; RLF—Relief; TS—term strength; WF—word frequency; WRP—Wrapper.
Note 2: Abbreviations of classification methods: C4.5—decision tree; DT-min10—decision tree; kNN—k-Nearest Neighbors; LLSF—Linear Least Squares Fit; LR—Logistic Regression; NN—Neural Network; NB—Naïve Bayes; MNB—Multinomial Naïve Bayes; PCL—Prediction by Collective Likelihood; PropBayes—Bayesian classifier; SVM—Support Vector Machine.
Note 3: ">" means "performed better than".

3. Proposed Hybrid Attribute Selection Approach

Traditionally, either the filter approach or the wrapper approach has been applied to select a good subset of attributes from the full attribute set. Both approaches have their advantages and disadvantages. While the filter approach is computationally efficient and is cost-feasible in most classification applications, it encounters problems in determining the size of the final attribute subset and tends to deliver lower performance than the wrapper approach does (Inza et al. 2004; Kohavi and John 1996). As the filter approach usually only evaluates the attributes individually and not collectively, the best subset of attributes cannot be determined without performance measures on different subsets. There are only some rough heuristics in the literature that help select satisfactory attribute subsets. For example, Fuhr and Buckley (1991) suggested that 50 to 100 training examples per attribute are needed. In addition, as different learning algorithms may favor different attribute subsets, the universal subset produced by the filter approach may not be optimal for a particular learning algorithm. On the other hand, the wrapper approach can determine the size of the final attribute subset by itself and provide better results than the filter approach, in general. However, it is very time-consuming and is cost-prohibitive in realistic text classification applications, where the number of attributes is typically very large (Sebastiani 2002).

To address the problems inherent in the two approaches, we propose a hybrid approach that combines the two. Figure 2 outlines the proposed approach. The initial input is the full set of attributes. We first apply the filter approach, which evaluates the attributes based on a relevance measure and then selects a proper subset of the top-ranked attributes. We then apply the wrapper approach to these attributes by wrapping the target learning algorithm into the process to search for the best attribute subset through repeated classifier training and evaluation on different attribute subsets.

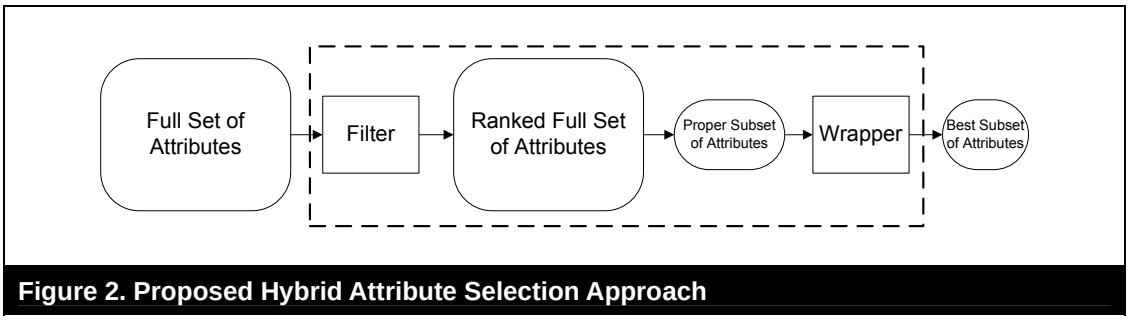

Figure 2. Proposed Hybrid Attribute Selection Approach

The primary objective of our study is to combine the filter and wrapper approaches so that the resultant approach performs better than either of the individual approaches. In developing the hybrid approach, we exploit the strengths of the two basic attribute selection approaches and, at the same time, address their shortcomings. Table 2 provides a comparison of the three approaches.

Using the filter approach for pre-selection makes the wrapper approach cost-feasible. And, using the wrapper approach, the size of the final subset is determined automatically rather than based on a rough heuristic, and the subset is tuned specifically to the target learning algorithm. We expect this hybrid approach to be sufficiently efficient to be applicable in realistic text classification applications and, at the same time, to be more effective than the filter approach alone.

The criteria for comparison of the attribute selection approaches include computational complexity, computing time, feasibility, automatic selection of final subset of attributes, and classification performance (see Table 2). These criteria provide the rationale for the choices we will make in developing the hybrid approach, which should: i) be cost-efficient; ii) automatically select the best final subset of attributes; iii) be feasible for text classification; and iv) boost classification performance using the target classifier for selecting the final subset of attributes.

Table 2: Comparison of Attribute Selection Approaches

Approach	Advantages	Disadvantages
Filter	• Relatively low computational complexity • Feasible for text classification	• Final best subset of attributes can not be automatically determined • Rough heuristics needed for determining the size of the selected attribute subset • Selected subset is not tuned to the particular classification method used
Wrapper	• Final best subset of attributes can be automatically determined • Better performance on classification accuracy due to the use of target classifier on the selection process	• Time consuming • High computational complexity • Not feasible for text classification
Proposed (Hybrid)	• Moderate computational complexity • Feasible for text classification • Final best subset of attributes can be automatically determined • Expected better performance on classification accuracy due to the use of target classifier on the selection process	• Extra acceptable time spent on wrapping process compared to filter method

Determining the number of attributes to keep after the first step of the hybrid approach involves a tradeoff between effectiveness and efficiency. Keeping more attributes increases the computing time during the second step, but at the same time, increases the chance to find the overall "optimal" attribute subset. The extra time investment in attribute selection at this early stage may very well pay off subsequently in repeated applications of the resulting classifier. It may be beneficial to keep as many attributes after the first step as feasible in the second step.

The hybrid approach does not require substantial implementation. Relevance measures and search algorithms available in some existing tools (e.g., the options of the filter and wrapper of Weka (Witten and Frank 2005)) can be easily combined and configured following the hybrid approach. The wrapper step uses the same classification algorithm that is later used in training the final classifier.

4. Empirical Evaluation

We have empirically evaluated our proposed approach in workplace Internet abuse detection and prevention. In this section, we report on our evaluation and discuss our findings.

4.1. Domain Selection

As Internet access is becoming widespread in the workplace, it is also causing severe problems. One of them is Internet abuse, which refers to employees' use of the Internet for non-work-related purposes. Such abuse includes undesirable activities such as investing, shopping, online chatting, gaming, illegal downloading, viewing pornography, engaging in cybersex, and committing online crimes (Greenfield and Davis 2005). Internet abuse results in a waste of employees' time (Malachowski 2005), loss of workers' productivity (Sharma and Gupta 2003), network congestion, and other problems. The Internet abuse problem has been attracting a lot of attention in the research community lately. It has been studied under several names, including *Internet abuse in the workplace* (Galletta and Polak 2003; Griffiths 2003; Mahatanankoon 2006; Sharma and Gupta 2003; Siau et al. 2002; Woon and Pee 2004; Young and Case 2004), *Internet misuse in the workplace* (Wyatt and Phillips 2005), *problematic Internet use in the workplace* (Davis et al. 2002), *personal web usage in the workplace* (Anandarajan and Simmers 2004; Mahatanankoon et al. 2004), *non-work related computing* (Lee et al. 2005; Wong et al. 2005), and *cyberloafing* (Lim 2002).

Many organizations have started to address the Internet abuse problem by adopting Internet usage policies (Siau et al. 2002), conducting management training (Young and Case 2004), and monitoring and blocking employees' abuse usage using software packages (Urbaczewski and Jessup 2002). Internet usage policies and management training provide general guidelines for employees' Internet usage. They can be used for Internet abuse prevention, but have little control on employees' actual Internet usage. Using software packages to automatically monitor, filter, and block employees' Internet abuse behavior is another popular approach for Internet abuse detection and prevention. Current filtering software packages mainly rely on white lists or black lists to control Internet access. The lists need to be frequently updated, as new websites and web pages are constantly emerging. Some software vendors, such as CYBERSitter (http://www.cybersitter.com/), provide periodic updates of filtering lists for various categories. However, the lists may still lack sufficient coverage of newly developed websites and web pages. A few products also perform content-based filtering. They usually rely on pre-defined key words and phrases, profile analysis based on characteristics such as the ratio of images to text, and links to known undesirable sites (Greenfield et al. 2001). Such simple matching often results in incorrectly blocking web pages with acceptable content. Past studies evaluating commercial filtering software packages have found that they did not yield satisfactory performance, with an overall accuracy around 80 percent (Greenfield et al. 2001; Hunter 2000).

Other than using black lists, white lists, keyword blocking, and rating systems, machine learning classification methods have also been applied to web filtering, especially for screening pornographic web pages (Du et al. 2003; Hammami et al. 2006; Hu et al. 2007; Lee et al. 2002; Polpinij et al. 2006). Lee et al. (2002) used 61 indicated terms as feature vectors to represent textual contents of pornographic pages and applied neural networks to classify pages. Du et al. (2003) proposed a system to filter undesired web pages using cosine similarity coefficient. They applied vectors of words to represent page content and used only pornographic pages to train the system. Undesired pages were determined based on the cosine similarity coefficient. Similarly, Polpinij et al. (2006) built content-based text classifiers for filtering pornographic web pages, represented by vectors of words. The results showed that Support Vector Machine performed better than naïve Bayes. Hu et al. (2007) utilized C4.5 decision tree to first classify pages into three categories—continuous text, discrete text, and image pages—for further classification by respective text and contour-based image classifiers. The results were then fused to make the final decision of filtering. The study showed that the fusion approach outperformed the individual classifiers. Likewise, Hammami et al. (2006) proposed WebGuide to combine analysis of textual content, structural content such as hyperlinks, and skin color-related visual content on filtering pornographic pages. When these three techniques were combined, the proposed system reached a 97.4 percent accuracy rate in classifying a balanced collection of 400 sites.

Internet abuse detection can be formulated as a binary classification problem, where Web pages are classified into two categories, *abuse* or *non-abuse*, depending on the job profile of the employees' duties. In this study, we apply text classification techniques for detecting and preventing Internet

abuse. Figure 3 outlines our evaluation procedure.

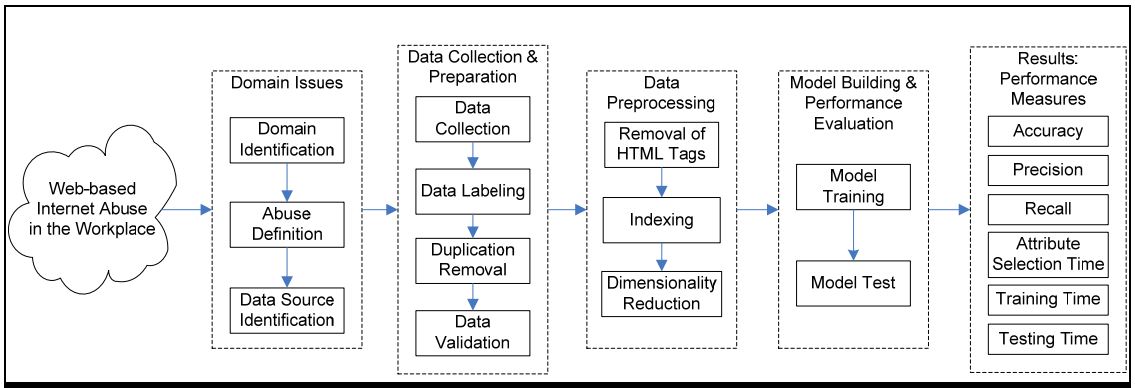

Figure 3. Internet Abuse Detection Using Text Classification

Internet abuse detection is a domain-specific task. Although there are some general abuse types (e.g., pornography, profanity, hate-crimes, etc.) common to many domains (job duties), other abuse types are only specific to particular domains. For instance, reading financial news can be considered to be abuse for programmers in the IT industry working on non-financial projects, but it is not considered to be abuse for financial analysts. Thus, each domain has its own profile of abuse types.

To simplify domain complexity, we set our experiment in a particular domain, the workplace of programmers in the IT industry. To define the abuse types in the selected domain, we consulted with four domain experts, including three experienced programmers and a professor teaching programming languages. Based on their feedback, we classified as abuse cases web pages containing personal investment, business news, general news, entertainment news, and sports news, which are not related to programmers' work. On the other hand, web pages containing technology news and online resources related to programming (e.g., PHP, ASP.NET, Visual C++, Visual Basic, Visual C#, Visual Studio, and Web Service) are crucial to programmers; thus, we classified these as non-abuse cases. Daily news websites and auction web pages were identified as the source of abuse cases, and official programming resource web pages were identified as the source of non-abuse cases.

Note that the Internet abuse detection and prevention we consider in this study is at the web page level, rather than at the website level. If a web page is classified as "abuse" for an employee, his or her access to the page is blocked. It is possible that an employee is allowed to access some of the pages, but not others, at a website. For instance, for an IT security employee, viewing general news at some website may be deemed "abuse" and hence should be blocked, but viewing a story (page) at the same site dealing with a major vulnerability in a new version of a software package may be considered appropriate and should be allowed. As new pages are added to many sites on a continual basis, the detection and prevention of potential access to abuse pages must be performed "on the fly" in real time, demanding efficient page classification to avoid unacceptable delays in user experience while browsing legitimate pages. The text mining approach we incorporate *detects* the likely "abuse" pages and *prevents* access to those pages by representing the problem as a binary classification problem.

4.2. Data Collection and Preprocessing

We collected a balanced sample of 10,000 web pages during a half-month period, with 5,000 abuse cases and 5,000 non-abuse cases, for the domain of programmers in the IT industry. We collected abuse pages from 16 popular news websites, including CNN (http://www.cnn.com), New York Times (http://www.nytimes.com), Reuters (http://www.reuters.com), and Washington Post

(http://www.washingtonpost.com), and the auction site eBay (http://www.ebay.com/). We collected non-abuse cases from web pages of official resources on programming languages, such as MSDN (http://msdn.microsoft.com/) and PHP (http://www.php.net). We used pairwise byte-by-byte comparison and filename comparison for all pages in the dataset to ensure that each page was unique.

The web pages were preprocessed into a representation suitable for classification. We discarded all HTML tags, since they are mainly used for the layout of the pages and do not carry much additional meaning to the content of the pages. We then parsed the web pages and extracted the words, removing stop words. We applied the Porter stemming algorithm (Porter 1980) for suffix stripping. We then calculated the TF and TFIDF indices of the resulting word stems as the raw attributes.

While we used the collected and classified dataset for our empirical evaluation purposes, we are not recommending the particular classification scheme we used as a practical guideline for all programmers. An organization may have a different definition of Internet abuse for each employee group (or even each individual employee) at a particular time and have to classify training examples according to its own policies. For example, a programmer working at a financial company may need to browse financial news, personal investment, and other related web pages to gain more domain knowledge. On the other hand, for another programmer working with COBOL program development, surfing the web to search for C++ information may actually be considered an abuse case. Even non-work-related Internet usage is not necessarily considered abusive. Some non-work-related Internet usage behaviors may be constructive (Anandarajan et al. 2004), or may satisfy prolific employees (Stanton 2002). Certain categories of non-work-related Internet activities may also enhance the employees' well-being (Anandarajan et al. 2004; Mahatanankoon and Igbaria 2004). An organization may also have its own policy on a level of control for each employee group. For example, an organization may simply focus on web pages that are considered abusive and offensive by social norms (e.g., pornography, profanity, hate-crimes, etc.), if defining other abuse types is controversial and overwhelming. An organization's policies may also change over time, which means that training examples need to be reclassified and web page classifiers retrained.

4.3. Performance Measure and Evaluation Method

We used classification accuracy (the percentage of web pages correctly classified by a classifier) as the main performance measure. In addition to accuracy, we employed precision, recall, and F-measure—used in the information retrieval field—as performance measures; these additional measures are defined as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

We also measured the computing time, including attribute selection time (the time spent on attribute selection), training time (the time spent on classifier training), and testing time (the time spent on classifier testing). We used 10-fold cross-validation to estimate the performance of each classifier. All accuracy measures reported below are cross-validated estimates. In addition, we conducted 10-fold cross-validation 20 times for each configuration to obtain a more reliable estimate. Each time, a different set of 10-folds was randomly generated, leading to a different estimation. We then averaged the results from the 20 runs.

4.4. Experimental Design

We examined four factors: attribute selection approach, classification algorithm, attribute relevance measure, and term weighting scheme. The attribute selection approaches evaluated include the filter approach (Filter), the hybrid approach with greedy forward search in the wrapper step (HybridGreedy), and the hybrid approach with best-first search in the wrapper step (HybridBest). The attribute relevance measures used in the filter step of attribute selection include information gain (IG), gain ratio (GR), Chi-square (χ^2), and minimum redundancy and maximum relevance (mRMR).[1] The classification algorithms evaluated include naïve Bayes (NB), multinominal naïve Bayes (MNB), backpropagation neural network (NN), the C4.5 decision tree learner, and support vector machine (SVM). Finally, the term weighting schemes include TF and TFIDF.

4.5. Implementation and Experimental Environment

We developed a document indexing program based on the WekaIndex tool (http://www.ainetsolutions.com/eng/soluciones/aplicaciones/ir.html) and used the Weka data mining software (Witten and Frank 2005) for the experiment. The decision tree method was J4.8, Weka's implementation of C4.5. We kept Weka's default parameters for most algorithms. For NN, we used one hidden layer with two nodes and 50 training epochs. We used a small number of hidden nodes to prevent severe *overfitting,* as the ratio between the number of training examples and the number of input nodes was low. For the wrapper step in the hybrid attribute selection approach, we used five-fold cross-validation for evaluating classifiers built for different attribute subsets. Note that this five-fold cross-validation for attribute selection is different from the 10-fold cross-validation for the final performance estimation.

We ran the experiment in a uniform environment. We used a personal computer with Intel Pentium 4 3.0GHz CPU, HyperThreading technology, and 1 GB RAM, running Microsoft Windows XP Professional edition with service pack 2. Therefore, all time measures (attribute selection time, training time, and testing time) for different configurations are comparable.

5. Experimental Results

5.1. Pilot study

We conducted a pilot study to determine the number of top-ranked attributes for the filter approach, as attribute relevance measures only rank the attributes and do not select the best subset of attributes. We obtained 42,144 unique attributes from the dataset after removing the HTML tags, stop-words, and suffixes. In order to search for the potentially best subset of attributes, we examined the classification performance by selecting the top 20 to 1,000 attributes. Starting with 20, we increased the number of selected top-ranked attributes in increments of 20, up to 200. Above 200, we increased the number of selected top-ranked attributes in increments of 100, up to 1,000.

The pilot study indicated that 200 attributes (with 50 training examples per attribute) appeared to be appropriate. This is somewhat consistent with Fuhr and Buckley (1991), who found that the performance peaked when 200 attributes were used in all classifiers except neural network. They also suggested that 50 to 100 training examples per attribute are needed. Therefore, we used the top 200 attributes selected by the filter methods in subsequent experiments.

5.2. Classification Performance

In this section, we report the experimental results of all configurations on the datasets with attributes selected by different approaches, measuring the results by accuracy, precision, recall, and F-measure. Table 3 lists the average accuracy from 20 runs of 10-fold cross-validation for the same configuration. For the attribute selection approach, "Full" refers to the use of the full set of attributes

[1] We also tried the latent semantic analysis algorithm for dimensionality reduction. But even after increasing the heap size to 1GB to run the algorithm, memory overflow occurred.

without applying any selection method. We report the results using precision, recall, and F-measure in Tables 4, 5, and 6, respectively.

We were not able to obtain the average accuracy of NN with the full set of attributes due to memory overflow problems in Weka. We examined the following eight configurations (two term weighting schemes and four attribute relevance measures) for each classification algorithm: (TF, IG), (TF, GR), (TF, χ^2), (TF, mRMR), (TFIDF, IG), (TFIDF, GR), (TFIDF, χ^2), and (TFIDF, mRMR).

For every classification algorithm and every term weighting scheme, mRMR produced the least accurate results among the four attribute relevance measures (see Table 3). Because mRMR was dominated uniformly by the other three attribute relevance measures, we do not discuss the results using this measure any further in the paper. Our discussion will be limited to the six remaining configurations: (TF, IG), (TF, GR), (TF, χ^2), (TFIDF, IG), (TFIDF, GR), and (TFIDF, χ^2).

Table 3 Average Accuracy for All Configurations (%)

Classification Algorithm	Term Weighting Scheme	Attribute Relevance Measure	Full Set	Attribute Selection Approach		
				Filter	HybridBest	HybridGreedy
NB	TF	IG GR χ^2 mRMR	86.91 [0.0151]	87.22 [0.0186] 90.23 [0.0159] 87.34 [0.0202] 73.11 [0.0177]	95.29 [0.0194] 93.89 [0.0245] 96.97 [0.0127] 90.92 [0.0249]	95.29 [0.0202] 93.89 [0.0245] 96.97 [0.0127] 90.46 [0.1880]
	TFIDF	IG GR χ^2 mRMR	86.81 [0.0141]	87.22 [0.0173] 90.23 [0.0162] 87.35 [0.0193] 72.62 [0.0177]	95.17 [0.0097] 93.99 [0.0354] 96.03 [0.0134] 91.35 [0.0825]	95.17 [0.0097] 93.99 [0.0354] 96.01 [0.0116] 90.36 [0.0818]
MNB	TF	IG GR χ^2 mRMR	90.11 [0.0032]	90.00 [0.0000] 91.45 [0.0033] 89.99 [0.0000] 86.58 [0.0058]	96.39 [0.0022] 96.15 [0.0030] 96.61 [0.0117] 88.44 [0.0093]	96.37 [0.0025] 96.15 [0.0030] 93.17 [0.0198] 88.23 [0.0091]
	TFIDF	IG GR χ^2 mRMR	91.81 [0.0106]	90.15 [0.0000] 92.32 [0.0033] 90.09 [0.0011] 86.71 [0.0068]	95.84 [0.0167] 96.27 [0.0024] 95.46 [0.0148] 89.50 [0.0074]	94.80 [0.0050] 96.00 [0.0043] 94.82 [0.0032] 89.30 [0.0073]
NN	TF	IG GR χ^2 mRMR	N/A (out of memory)	92.99 [0.2711] 88.84 [0.2466] 94.22 [0.2733] 73.90 [0.6819]	97.18 [0.1218] 93.82 [0.0664] 97.62 [0.0581] 84.16 [0.5598]	97.18 [0.1218] 93.57 [0.0777] 96.88 [0.0673] 84.16 [0.5598]
	TFIDF	IG GR χ^2 mRMR	N/A (out of memory)	92.87 [0.2319] 88.70 [0.2736] 94.47 [0.2722] 73.91 [0.6953]	97.36 [0.0940] 93.82 [0.0664] 96.75 [0.0587] 81.34 [0.6745]	97.36 [0.0940] 93.57 [0.0777] 96.68 [0.0597] 80.99 [0.4809]
SVM	TF	IG GR χ^2 mRMR	99.75 [0.0058]	98.39 [0.0122] 92.80 [0.0121] 98.53 [0.0080] 89.39 [0.0178]	97.26 [0.0074] 93.08 [0.0168] 97.97 [0.0089] 89.49 [0.0162]	97.26 [0.0074] 93.08 [0.0168] 97.97 [0.0066] 89.48 [0.0171]
	TFIDF	IG GR χ^2 mRMR	99.75 [0.0054]	98.39 [0.0118] 92.80 [0.0123] 98.53 [0.0080] 88.78 [0.0238]	98.52 [0.0081] 93.07 [0.0166] 98.00 [0.0073] 87.84 [0.0203]	98.51 [0.0069] 93.07 [0.0161] 97.99 [0.0084] 87.52 [0.0189]

J4.8	TF	IG GR χ^2 mRMR	99.72 [0.0077]	99.46 [0.0123] 98.06 [0.0199] 99.53 [0.0100] 96.37 [0.0328]	99.67 [0.0106] 98.42 [0.0089] 99.65 [0.0116] 96.52 [0.0172]	99.67 [0.0106] 98.42 [0.0089] 99.65 [0.0116] 96.52 [0.0172]
	TFIDF	IG GR χ^2 mRMR	99.72 [0.0077]	99.46 [0.0124] 98.06 [0.0199] 99.53 [0.0100] 96.88 [0.0224]	99.67 [0.0106] 98.42 [0.0089] 99.65 [0.0116] 96.99 [0.0172]	99.67 [0.0106] 98.42 [0.0089] 99.65 [0.0116] 96.99 [0.0172]

Note: The numbers in brackets are standard errors.

Table 4: Average Precision for All Configurations

Classification Algorithm	Term Weighting Scheme	Attribute Relevance Measure	Full Set	Attribute Selection Approach		
				Filter	HybridBest	HybridGreedy
NB	TF	IG GR χ^2	0.7949 [0.0002]	0.7998 [0.0002] 0.8417 [0.0002] 0.8019 [0.0003]	0.9346 [0.0004] 0.9051 [0.0002] 0.9663 [0.0002]	0.9353 [0.0004] 0.9051 [0.0002] 0.9663 [0.0002]
	TFIDF	IG GR χ^2	0.7937 [0.0002]	0.7997 [0.0002] 0.8417 [0.0002] 0.8021 [0.0003]	0.9256 [0.0002] 0.9164 [0.0004] 0.9439 [0.0002]	0.9256 [0.0002] 0.9164 [0.0004] 0.9440 [0.0002]
MNB	TF	IG GR χ^2	0.8350 [0.0000]	0.8335 [0.0000] 0.8561 [0.0001] 0.8334 [0.0000]	0.9423 [0.0000] 0.9460 [0.0000] 0.9517 [0.0001]	0.9421 [0.0000] 0.9460 [0.0000] 0.8889 [0.0000]
	TFIDF	IG GR χ^2	0.8597 [0.0002]	0.8357 [0.0000] 0.8740 [0.0000] 0.8348 [0.0000]	0.9329 [0.0002] 0.9456 [0.0000] 0.9269 [0.0002]	0.9295 [0.0000] 0.9432 [0.0000] 0.9303 [0.0000]
NN	TF	IG GR χ^2	N/A (out of memory)	0.9241 [0.0032] 0.8465 [0.0042] 0.9377 [0.0033]	0.9692 [0.0020] 0.9130 [0.0016] 0.9787 [0.0010]	0.9692 [0.0020] 0.9084 [0.0021] 0.9808 [0.0008]
	TFIDF	IG GR χ^2	N/A (out of memory)	0.9208 [0.0038] 0.8457 [0.0037] 0.9395 [0.0029]	0.9730 [0.0012] 0.9130 [0.0016] 0.9796 [0.0010]	0.9730 [0.0012] 0.9084 [0.0021] 0.9805 [0.0009]
SVM	TF	IG GR χ^2	0.9974 [0.0001]	0.9804 [0.0002] 0.8920 [0.0003] 0.9818 [0.0001]	0.9702 [0.0001] 0.8981 [0.0002] 0.9721 [0.0001]	0.9702 [0.0001] 0.8981 [0.0002] 0.9717 [0.0001]
	TFIDF	IG GR χ^2	0.9974 [0.0001]	0.9804 [0.0001] 0.8920 [0.0003] 0.9818 [0.0001]	0.9813 [0.0002] 0.8979 [0.0002] 0.9724 [0.0001]	0.9812 [0.0001] 0.8979 [0.0002] 0.9724 [0.0001]
J4.8	TF	IG GR χ^2	0.9976 [0.0001]	0.9955 [0.0002] 0.9764 [0.0003] 0.9961 [0.0001]	0.9975 [0.0001] 0.9854 [0.0002] 0.9971 [0.0001]	0.9975 [0.0001] 0.9854 [0.0018] 0.9971 [0.0001]
	TFIDF	IG GR χ^2	0.9976 [0.0001]	0.9955 [0.0001] 0.9764 [0.0003] 0.9961 [0.0001]	0.9975 [0.0001] 0.9854 [0.0002] 0.9971 [0.0001]	0.9975 [0.0001] 0.9854 [0.0002] 0.9971 [0.0001]

Note: The numbers in brackets are standard errors.

Table 5: Average Recall for All Configurations

Classification Algorithm	Term Weighting Scheme	Attribute Relevance Measure	Full Set	Attribute Selection Approach		
				Filter	HybridBest	HybridGreedy
NB	TF	IG GR X^2	0.9953 [0.0000]	0.9941 [0.0001] 0.9914 [0.0002] 0.9929 [0.0001]	0.9743 [0.0004] 0.9809 [0.0004] 0.9733 [0.0002]	0.9734 [0.0003] 0.9809 [0.0004] 0.9733 [0.0002]
	TFIDF	IG GR X^2	0.9954 [0.0000]	0.9940 [0.0001] 0.9914 [0.0002] 0.9927 [0.0001]	0.9826 [0.0001] 0.9683 [0.0004] 0.9788 [0.0003]	0.9826 [0.0001] 0.9683 [0.0004] 0.9784 [0.0001]
MNB	TF	IG GR X^2	1.0000 [0.0000]	1.0000 [0.0000] 0.9967 [0.0000] 1.0000 [0.0000]	0.9885 [0.0000] 0.9790 [0.0000] 0.9822 [0.0001]	0.9883 [0.0000] 0.9790 [0.0000] 0.9868 [0.0004]
	TFIDF	IG GR X^2	0.9996 [0.0000]	0.9998 [0.0000] 0.9894 [0.0000] 0.9998 [0.0000]	0.9879 [0.0002] 0.9819 [0.0000] 0.9872 [0.0001]	0.9696 [0.0001] 0.9790 [0.0001] 0.9691 [0.0001]
NN	TF	IG GR X^2	N/A (out of memory)	0.9428 [0.0053] 0.9605 [0.0085] 0.9531 [0.0059]	0.9754 [0.0015] 0.9711 [0.0027] 0.9740 [0.0014]	0.9754 [0.0015] 0.9718 [0.0032] 0.9565 [0.0013]
	TFIDF	IG GR X^2	N/A (out of memory)	0.9456 [0.0062] 0.9590 [0.0094] 0.9565 [0.0036]	0.9748 [0.0016] 0.9711 [0.0027] 0.9552 [0.0013]	0.9748 [0.0016] 0.9718 [0.0032] 0.9529 [0.0012]
SVM	TF	IG GR X^2	0.9976 [0.0001]	0.9876 [0.0001] 0.9743 [0.0002] 0.9889 [0.0001]	0.9752 [0.0001] 0.9723 [0.0003] 0.9878 [0.0001]	0.9752 [0.0001] 0.9723 [0.0003] 0.9882 [0.0001]
	TFIDF	IG GR X^2	0.9976 [0.0001]	0.9875 [0.0001] 0.9743 [0.0002] 0.9889 [0.0001]	0.9893 [0.0001] 0.9723 [0.0003] 0.9880 [0.0001]	0.9893 [0.0001] 0.9723 [0.0003] 0.9879 [0.0001]
J4.8	TF	IG GR X^2	0.9969 [0.0001]	0.9937 [0.0002] 0.9851 [0.0002] 0.9944 [0.0002]	0.9958 [0.0001] 0.9829 [0.0001] 0.9959 [0.0001]	0.9958 [0.0001] 0.9829 [0.0001] 0.9959 [0.0001]
	TFIDF	IG GR X^2	0.9969 [0.0001]	0.9937 [0.0002] 0.9851 [0.0002] 0.9944 [0.0002]	0.9958 [0.0001] 0.9829 [0.0001] 0.9959 [0.0001]	0.9956 [0.0001] 0.9829 [0.0001] 0.9959 [0.0001]

Note: The numbers in brackets are standard errors.

Table 6: Average F-Measure for All Configurations

Classification Algorithm	Term Weighting Scheme	Attribute Relevance Measure	Full Set	Attribute Selection Approach		
				Filter	HybridBest	HybridGreedy
NB	TF	IG	0.8838 [0.0001]	0.8863 [0.0001]	0.9539 [0.0002]	0.9539 [0.0002]
		GR		0.9103 [0.0001]	0.9414 [0.0002]	0.9414 [0.0002]
		χ^2		0.8871 [0.0002]	0.9698 [0.0001]	0.9698 [0.0001]
	TFIDF	IG	0.8831 [0.0001]	0.8862 [0.0001]	0.9532 [0.0001]	0.9532 [0.0001]
		GR		0.9103 [0.0001]	0.9416 [0.0003]	0.9416 [0.0003]
		χ^2		0.8871 [0.0002]	0.9698 [0.0001]	0.9608 [0.0001]
MNB	TF	IG	0.9101 [0.0000]	0.9090 [0.0000]	0.9648 [0.0000]	0.9646 [0.0000]
		GR		0.9210 [0.0000]	0.9622 [0.0000]	0.9622 [0.0000]
		χ^2		0.9091 [0.0000]	0.9667 [0.0001]	0.9353 [0.0002]
	TFIDF	IG	0.9243 [0.0001]	0.9104 [0.0000]	0.9596 [0.0002]	0.9491 [0.0001]
		GR		0.9281 [0.0000]	0.9634 [0.0000]	0.9608 [0.0000]
		χ^2		0.9098 [0.0000]	0.9353 [0.0002]	0.9492 [0.0000]
NN	TF	IG	N/A (out of memory)	0.9292 [0.0031]	0.9719 [0.0011]	0.9719 [0.0011]
		GR		0.8921 [0.0040]	0.9398 [0.0008]	0.9374 [0.0009]
		χ^2		0.9419 [0.0032]	0.9761 [0.0006]	0.9683 [0.0007]
	TFIDF	IG	N/A (out of memory)	0.9295 [0.0026]	0.9736 [0.0009]	0.9736 [0.0009]
		GR		0.8902 [0.0045]	0.9398 [0.0008]	0.9374 [0.0777]
		χ^2		0.9452 [0.0027]	0.9683 [0.0007]	0.9663 [0.0006]
SVM	TF	IG	0.9975 [0.0001]	0.9839 [0.0001]	0.9727 [0.0001]	0.9727 [0.0001]
		GR		0.9313 [0.0001]	0.9336 [0.0002]	0.9336 [0.0002]
		χ^2		0.9853 [0.0000]	0.9799 [0.0001]	0.9799 [0.0001]
	TFIDF	IG	0.9975 [0.0001]	0.9839 [0.0001]	0.9852 [0.0001]	0.9852 [0.0001]
		GR		0.9313 [0.0001]	0.9335 [0.0002]	0.9335 [0.0002]
		χ^2		0.9854 [0.0001]	0.9799 [0.0001]	0.9801 [0.0001]
J4.8	TF	IG	0.9972 [0.0001]	0.9946 [0.0001]	0.9966 [0.0001]	0.9966 [0.0001]
		GR		0.9807 [0.0002]	0.9841 [0.0001]	0.9841 [0.0001]
		χ^2		0.9953 [0.0001]	0.9965 [0.0001]	0.9965 [0.0001]
	TFIDF	IG	0.9972 [0.0001]	0.9946 [0.0001]	0.9966 [0.0001]	0.9966 [0.0001]
		GR		0.9807 [0.0002]	0.9841 [0.0001]	0.9841 [0.0001]
		χ^2		0.9953 [0.0001]	0.9965 [0.0001]	0.9965 [0.0001]

Note: The numbers in brackets are standard errors.

The hybrid approach—both HybridBest and HybridGreedy—outperformed the filter approach for every configuration. Among the six configurations, using the hybrid approach, best-first search provided much better performance than greedy forward search in two configurations. In configurations (TF, IG), (TF, GR), (TF, χ^2), and (TFIDF, GR), there was not much of a difference in accuracy between the two search methods.

As described before, we used a balanced dataset (5,000 abuse cases and 5,000 non-abuse cases) to train the classifiers. Using a balanced dataset is common practice for training data/text mining classifiers because by giving equal importance to each class, it does not bias the accuracy results in favor of any class. But we also experimented with datasets using different class distributions. Specifically, we used two other distributions: one with a 30-70 split and the other with a 70-30 split. We found that all the results obtained on the balanced sample still held on the two additional sample distributions. The relative ranks of the classifiers remained the same in the new samples.

As the results demonstrate, SVM and J4.8 were much more accurate than NB, MNB, and NN for the text classification problem (see Tables 3-6). Not only that, their accuracy was very high even when they were applied to the full set of attributes. We also found that the accuracy levels of J4.8 and SVM using the hybrid approach were more or less the same as those using the filter approach for each configuration. That may be because J4.8 and SVM exhibited high performance (over 99 percent accuracy and precision) without using the hybrid approach, and there was not much room for improvement. Also, J4.8 employs the *gain ratio* relevance measure in selecting the attributes, one at a time, for building a decision tree (Quinlan 1993). This process is similar to the filter method, which uses a relevance measure in the first step of the hybrid approach to select the attributes. Therefore, the additional gain ratio-based attribute selection process in the hybrid approach turned out to be redundant in terms of improving classification performance for J4.8.

Text classification typically involves a very high volume of attributes. Some machine learning classifiers such as Neural Network suffer from the overfitting problem when the number of attributes is large. Thus, attribute selection is necessary to reduce dimensionality and enhance performance. The proposed hybrid attribute selection approach addresses these issues; the experimental results show that our approach enhances the performance of most of the examined classifiers, but not SVM. This is not very surprising, given that it has been theoretically and empirically shown that SVM is robust to learning in high-dimensional spaces (Joachims 1998). Because it handles high-dimensional spaces effectively, there is not a great need for attribute selection.

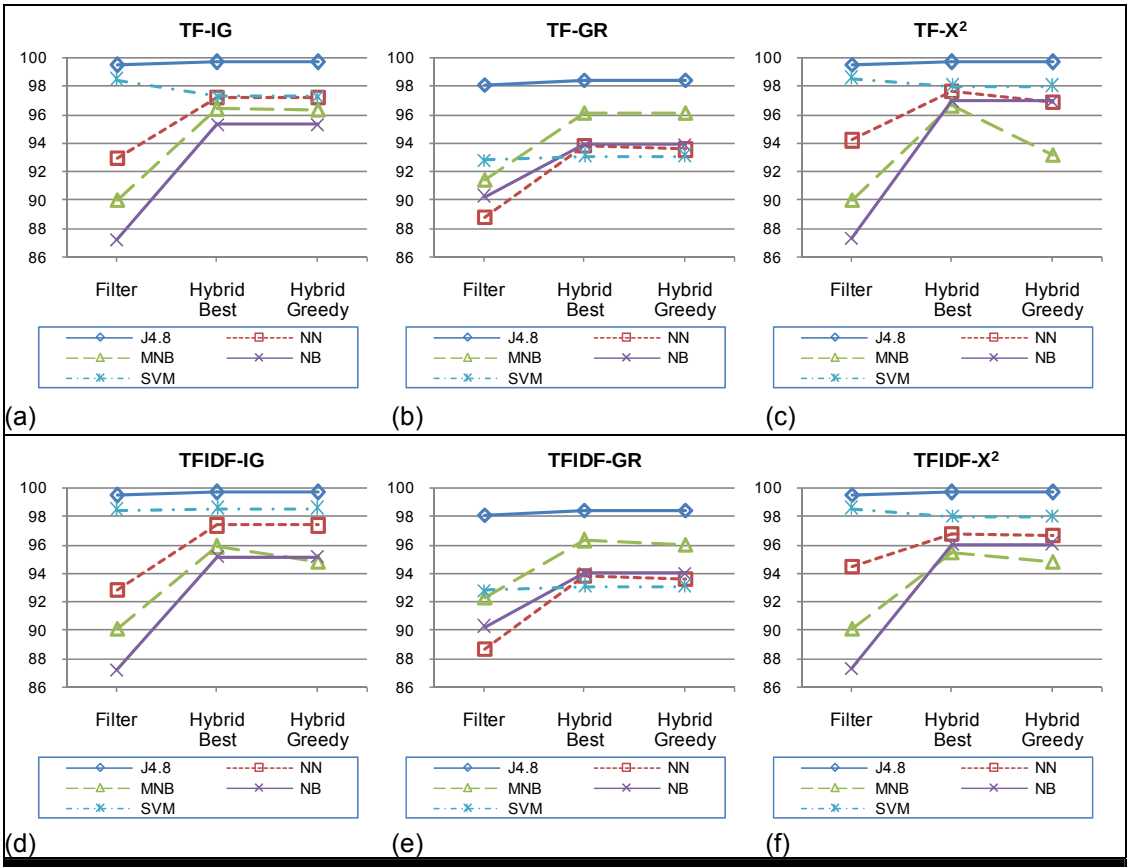

Figure 4. Accuracy Plots for Filter and Hybrid Approaches

Figure 4 shows the accuracy plots for all three levels of attribute selection—Filter, Hybrid Best, and Hybrid Greedy. The plots show that, for J4.8 and SVM, the performance remained relatively flat across the three attribute selection approaches, while there was a substantial performance

improvement of the hybrid approach over the filter approach for the other three classification algorithms.

J4.8 consistently outperformed the other classifiers in terms of accuracy on all possible configurations (see Figure 4). For the filter approach, SVM was always the second best, but for the hybrid approach, the results were mixed. In general, though, when the hybrid approach was adopted, SVM and NN exhibited similar accuracy levels on most configurations. While they were more accurate than MNB on some configurations—(e.g., TF-IG, TFIDF-IG), they were dominated by MNB on others (e.g., TF-GR, TFIDF-GR). On the TF-χ^2 and TFIDF-χ^2 configurations, SVM was the second most accurate classifier after J4.8. The results on precision, recall, and F-measure (see Tables 4, 5, and 6) were more or less consistent with the accuracy results. The relative ranks of the classifiers remained similar for precision and F-measure, but MNB and NB performed the best with respect to recall on filter set.

5.3. Computing Time

The time complexity of the filter approach is O(*m*), where *m* is the number of original attributes, multiplied by the complexity of the method for measuring the relevance of an attribute. The time complexity of the second step in the hybrid approach equals the complexity of the search method multiplied by the complexity of the learning algorithm. The time complexity of the greedy search is O(*m*'^{2}), where *m*' is the number of remaining attributes after the first step, in the worst case. The best-first search can be exhaustive and have a time complexity that is exponential in *m*' in the worst case, although it usually terminates much earlier.

We analyze the time spent on attribute selection, training, and testing separately. Table 7 lists the time spent by the attribute selection approaches. The time spent by all filter methods was quite close, ranging from 61.88 to 67.60 minutes. After the filter method was applied, the wrapper method was used in the second step in the hybrid approach. The time spent on the second step is denoted as "2nd Step-Best" and "2nd Step-Greedy" for best-first search and greedy forward search, respectively. The time spent by the hybrid approach varies depending on the classifier used, because wrapper is a classifier-dependent method. The results also show that greedy forward search was more efficient than best-first search in all configurations.

Applying wrapper attribute selection directly on large attribute sets is often not feasible. In our pilot test, when the wrapper approach was applied on the full attribute set using MNB—the most efficient algorithm among the five algorithms used in this study—along with greedy forward search and TF weighting scheme, it took 18,211.10 minutes (12.65 days) to generate the best subset of attributes. Although the accuracy rate was very high (99.60 percent), applying the wrapper approach on the full attribute set for real-world applications such as Internet abuse detection is not viable. In contrast, when the wrapper method was used by MNB as the second step in the hybrid approach for the same configuration, the time spent on the second step varied between 16.63 minutes and 38.62 minutes, whereas the shortest attribute selection times for NN, SVM, and J4.8 across all configurations were around 24 hours, 10 hours, and 13 hours, respectively.

Table 7: Time Spent on Attribute Selection (in Minutes)

Classification Algorithm	Term Weighting Scheme	Attribute Relevance Measure	Attribute Selection Approach		
			Filter	2nd Step-Best	2nd Step-Greedy
NB	TF	IG	62.02	774.05	298.15
		GR	67.60	154.23	79.93
		χ^2	62.00	459.05	334.33
	TFIDF	IG	61.88	193.70	118.68
		GR	67.70	268.57	185.67
		χ^2	62.15	290.95	104.92
MNB	TF	IG	62.02	50.90	38.62
		GR	67.60	27.82	23.57
		χ^2	62.00	44.63	16.63
	TFIDF	IG	61.88	45.63	17.00
		GR	67.70	39.58	24.78
		χ^2	62.15	37.03	19.02
NN	TF	IG	62.02	3414.02	2283.73
		GR	67.60	4278.27	3182.93
		χ^2	62.00	6754.12	1451.05
	TFIDF	IG	61.88	3130.53	1988.45
		GR	67.70	4264.67	1736.65
		χ^2	62.15	4315.97	1943.60
SVM	TF	IG	62.02	1628.95	1456.48
		GR	67.60	854.50	702.45
		χ^2	62.00	3197.12	2162.12
	TFIDF	IG	61.88	4401.22	3398.83
		GR	67.70	918.07	613.77
		χ^2	62.15	3339.38	2299.62
J4.8	TF	IG	62.02	2522.90	1832.48
		GR	67.60	1064.78	792.48
		χ^2	62.00	3406.77	2596.23
	TFIDF	IG	61.88	2523.60	1833.77
		GR	67.70	1061.42	793.18
		χ^2	62.15	3412.6	2597.62

Tables 8 and 9 report the training time and testing time of four different settings—Full, Filter, HybridBest, and HybridGreedy. We saw earlier that, in terms of accuracy measures (e.g., accuracy, precision, etc.), J4.8 performed very well, even when the full set of attributes was used. However, as we can see from Table 8, training a decision tree on our web page classification task using the full set can take over 45 hours. This would clearly be unacceptable in a real-world setting.

The training time for SVM on the full set was close to an hour, much better than the time for decision tree. Applying attribute selection as a preprocessing step radically reduced the training time without substantially lowering the accuracy. When the hybrid approach was used, the total training time—for filter plus wrapper—reduced drastically to between six and nine minutes for decision tree, and between one and two minutes for SVM. More importantly, attribute selection reduced testing time for SVM significantly, from more than 30 seconds to around one-tenth of a second (see Table 9). For an organization that is serious about addressing the Internet abuse problem, waiting longer than is absolutely needed while employees browse across sites may not be a viable option.

Table 8: Average Time Spent on Training over 20 Runs (in Seconds)

Classification Algorithm	Term Weighting Scheme	Attribute Relevance Measure	Attribute Selection Approach			
			Full	Filter	HybridBest	HybridGreedy
NB	TF	IG	23899.18	88.09	5.27	3.73
		GR		65.44	1.38	1.38
		χ^2		89.57	4.45	4.46
	TFIDF	IG	24140.86	87.75	2.49	2.45
		GR		65.25	3.10	3.07
		χ^2		90.15	3.33	2.15
MNB	TF	IG	5.50	0.94	0.18	0.16
		GR		0.40	0.07	0.07
		χ^2		0.95	0.16	0.09
	TFIDF	IG	5.49	0.90	0.15	0.08
		GR		0.40	0.09	0.06
		χ^2		0.98	0.15	0.10
NN	TF	IG	N/A (out of memory)	732.52	41.85	41.36
		GR		679.88	47.66	39.74
		χ^2		732.80	51.04	34.68
	TFIDF	IG	N/A (out of memory)	725.29	38.57	40.54
		GR		679.36	47.75	39.53
		χ^2		735.99	38.73	38.84
SVM	TF	IG	3093.65	80.66	16.70	16.37
		GR		49.71	19.04	18.15
		χ^2		88.97	28.23	29.59
	TFIDF	IG	3130.76	80.93	33.99	31.50
		GR		49.77	17.81	16.50
		χ^2		89.64	30.05	28.48
J4.8	TF	IG	163259.7	528.36	26.74	25.98
		GR		402.99	20.06	18.86
		χ^2		538.30	31.33	30.77
	TFIDF	IG	163717.2	526.80	29.21	26.83
		GR		403.74	19.28	19.07
		χ^2		535.81	31.31	30.81

The hybrid approach does incur extra computing time over the filter approach alone. However, this time investment may well pay off in the subsequent constant application of the resulting classifier. The hybrid approach tends to lead to a more accurate and faster classifier. For example, in our experiment, the classification accuracy of the NB classifier increased from 87.22 percent (using the filter approach alone) to 95.29 percent (using the hybrid approach) for the TF-IG configuration (see Table 3). Using the filter approach alone, 62 minutes were spent on attribute selection and 88.09 seconds on classifier training. Using the hybrid approach, an additional 298 minutes (beyond the 62 minutes) were spent on attribute selection (see Table 7) and 3.73 seconds on classifier training (see Table 8). However, the extra investment in attribute selection would be required only when the classifier needs to be retrained, whereas the trained classifier would be repeatedly applied. The classifier resulting from the filter approach alone took 6.79 seconds to classify 10,000 web pages, whereas the classifier resulting from the hybrid approach took only 0.69 seconds (see Table 9).

Table 9: Average Time Spent on Testing over 20 Runs (in Seconds)

Classification Algorithm	Term Weighting Scheme	Attribute Relevance Measure	Attribute Selection Approach			
			Full	Filter	HybridBest	HybridGreedy
NB	TF	IG	1950.29	6.79	0.91	0.69
		GR		8.12	0.40	0.38
		χ^2		6.98	0.69	0.68
	TFIDF	IG	1945.46	7.08	0.49	0.53
		GR		7.96	0.79	0.80
		χ^2		7.10	0.60	0.45
MNB	TF	IG	0.4534	0.09	0.08	0.07
		GR		0.09	0.06	0.06
		χ^2		0.11	0.09	0.08
	TFIDF	IG	0.4482	0.12	0.05	0.07
		GR		0.09	0.07	0.09
		χ^2		0.10	0.07	0.06
NN	TF	IG	N/A (out of memory)	3.93	0.19	0.17
		GR		3.83	0.21	0.16
		χ^2		3.95	0.21	0.14
	TFIDF	IG	N/A (out of memory)	3.94	0.17	0.15
		GR		3.86	0.20	0.17
		χ^2		4.04	0.16	0.15
SVM	TF	IG	33.62	0.32	0.11	0.13
		GR		0.28	0.10	0.09
		χ^2		0.32	0.10	0.12
	TFIDF	IG	33.83	0.32	0.14	0.14
		GR		0.30	0.10	0.08
		χ^2		0.33	0.11	0.10
J4.8	TF	IG	0.14	0.08	0.06	0.06
		GR		0.07	0.05	0.05
		χ^2		0.08	0.06	0.05
	TFIDF	IG	0.16	0.08	0.07	0.05
		GR		0.08	0.06	0.05
		χ^2		0.07	0.07	0.06

The training time and testing time spent by a classifier are highly related to the number of attributes used for training and testing. Table 10 shows the final number of attributes used in each classifier. The full attribute set includes 42,144 attributes. In our experiments, we fixed the attribute set size to 200 for the filter attribute selection approach. We obtained three sets of 200 attributes ranked by three attribute relevance measures. The three attribute sets were different, but largely overlapping. The 200 top-ranked attributes—based on each of the three attribute relevance measures—were then used as inputs to the wrapper in the hybrid approach. The number of attributes finally selected by the wrapper was typically much smaller than 200. Greedy forward search (in the wrapper) resulted in the same, or a smaller, attribute set, compared to best-first search.

Table 10: Final Number of Attributes Used in Classifiers

Classification Algorithm	Term Weighting Scheme	Attribute Relevance Measure	Attribute Selection Approach			
			Full	Filter	HybridBest	HybridGreedy
NB	TF	IG	42,144	200	22	17
		GR			8	8
		χ^2			18	18
	TFIDF	IG	42,144	200	12	12
		GR			17	17
		χ^2			16	11
MNB	TF	IG	42,144	200	38	36
		GR			25	25
		χ^2			35	18
	TFIDF	IG	42,144	200	31	17
		GR			37	26
		χ^2			29	19
NN	TF	IG	42,144	200	11	11
		GR			14	10
		χ^2			15	8
	TFIDF	IG	42,144	200	10	10
		GR			14	10
		χ^2			10	10
SVM	TF	IG	42,144	200	34	34
		GR			20	20
		χ^2			33	29
	TFIDF	IG	42,144	200	52	49
		GR			19	18
		χ^2			36	31
J4.8	TF	IG	42,144	200	20	20
		GR			16	16
		χ^2			24	24
	TFIDF	IG	42,144	200	20	20
		GR			16	16
		χ^2			24	24

6. Conclusions and Future Directions

We proposed a text mining approach for web page classification, applying it to the organizational problem of Internet abuse detection and prevention in the workplace. Specifically, we proposed a hybrid attribute selection approach for text classification and chose the domain of programmers' workplace as the test bed for our experiments. The experimental results show that text mining is an attractive technique to use for Internet abuse detection. In our pilot study, we found that the wrapper attribute selection approach, using the most efficient classification method (MNB), took more than 12 days to generate the best subset of attributes. This indicates that the wrapper approach is not suitable for text classification. The experimental results show that the hybrid approach is more efficient than the wrapper approach and is more effective than the filter attribute selection approach.

The main objective of our study was to combine the filter and wrapper approaches. To that end, we proposed a hybrid approach that exploits the relative strengths of these two approaches. The findings of the study are summarized in Table 11. Building classifiers on the full set of attributes, however accurate, is an impractical proposition because of the computational costs involved. As we saw, training a J4.8 decision tree using the full set can take close to two days. Even for SVM, the training time using the full set took close to an hour. Applying the hybrid approach substantially reduced the

total training time to between six and nine minutes for decision tree, and between one and two minutes for SVM. More importantly, the testing time for SVM fell sharply from more than 30 seconds to around one-tenth of a second. This is a great benefit for real-time detection and prevention of Internet abuse without causing unacceptable delays in user experience while browsing legitimate content.

As has been discussed in detail above, J4.8 and to a lesser extent, SVM, produced more accurate results overall. However, as emphasized in this paper, accuracy measures only provide one perspective of performance. For organizational problems such as Internet abuse, where computational efficiency is an important concern, attribute selection time, training time, and, more importantly, testing time could have a major bearing on the type of classifier that the organization would be willing to adopt. The results of the study strongly suggest that using the hybrid approach on MNB—a relatively lower performer accuracy-wise than J4.8 and SVM when using only the Filter approach—not only brought it closer to the other two by boosting its accuracy level, but also enabled it to outperform J4.8 and SVM with respect to attribute selection time and training time.

In the second step of the hybrid approach, when the wrapper was applied for attribute selection, the time taken by MNB was the lowest (~17 minutes), followed by NB (~1hour 20 mins), SVM (~10 hours), J4.8 (~13 hours), and NN (~24 hours).[2] Using the hybrid approach, MNB, along with NB, took significantly less time to train than J4.8, SVM, and NN. As far as the time taken to classify a web page as abuse or non-abuse is concerned, MNB still performed the best, along with J4.8. Therefore, insofar as computing times are concerned, MNB dominated all other classifiers.

The above findings have important implications. One of the important factors to consider is the frequency of retraining in response to changes in abuse policies or employee profiles. Note that retraining would also be needed when new websites enter the fray, allowing employees to access new content. Retraining would consume attribute selection times by both the filter and the wrapper, as well as the time needed to train the classifier on the attribute subset generated by the wrapper. As is apparent from the results above, if retraining happens frequently enough, SVM, J4.8, and NN would not remain competitive.

Lower testing times imply that the classifier is quicker at detecting abuse pages at run time. If these times are relatively short, it implies that the organization could dynamically decide on whether the page that an employee attempts to visit falls under “abuse” or not. The advantage of dynamically classifying pages is that abuse detection and prevention could be made “on the fly” as employees visit those pages. From this perspective, MNB and SVM are the two top performers. If page classification is done *a priori* in a static fashion, the advantage is that the classifier would not incur any costs at the time of a visit, but the shortcoming is that it would not be able to respond adequately to changes in page content.

Another important finding of this study is that use of the hybrid approach (HybridBest or HybridGreedy), as opposed to the filter approach, always resulted in much more accurate classifiers for NB, MNB, and NN. For J4.8 and SVM, the differences in accuracy between the two approaches were negligible.

In a text classification task such as Internet abuse detection, there are tens of thousands of attributes, and applying text mining techniques to the full attribute set is not practically feasible. The filter approach, on the other hand, provides a viable option. However, the accuracy levels of NB, MNB, and NN classifiers built using the filter approach still cannot compete with those of J4.8 and SVM classifiers. Using the hybrid approach addresses the shortcomings of the filter approach by boosting the accuracies of NB, MNB, and NN classifiers close to the levels attained by J4.8 and SVM, and by lowering the training and testing times appreciably. The hybrid approach proposed in this paper makes a valuable contribution to the field by providing a more effective, efficient, and viable method for real-world text classification tasks than either the filter approach or the wrapper approach.

[2] The times reported are the shortest attribute selection times across all six configurations.

The findings of this research would benefit both practitioners and researchers. From a practitioner's perspective, our content-based Internet abuse detection can supplement current state-of-the-art Internet filtering software products using lists. It can be implemented as a real-time Internet abuse detector. From a research standpoint, our study provides a new perspective on text classification and opens up new avenues for research.

There are several potential future research directions. While our evaluation was limited to one dataset, one may replicate our evaluation with more datasets to test the generalizability of our findings. The evaluation could also be extended to other domains and other text classification techniques. Although we limited the empirical experiments to one specific application, the proposed hybrid attribute selection approach is general and can be applied in any situation where selection of the best representative subset of attributes is necessary, especially on high dimensional datasets. One possible direction of future research is to empirically examine the proposed hybrid approach in other applications. Also, an examination of configurations suggested by this study can be conducted. Another direction is to apply the text mining approach for Internet abuse detection in the real-world workplace.

Table 11: Summary of Research Findings

Comparison	Research Findings
Full vs. Filter	• Text classification using full attribute set is time consuming (extremely long training and testing times). • Text classification using full attribute set may not be feasible for some classifiers (e.g., out of memory problem for NN).
Filter vs. Hybrid	• Based on computation complexity, both filter and hybrid approaches are suitable for text classification. • Applying the hybrid approach boosted the accuracies of NB, MNB, and NN much closer to the levels of J4.8 and SVM. • Text classifiers built using the hybrid approach are much more cost-efficient—with respect to training and testing—than the corresponding classifiers built using the filter approach. • Hybrid approach performed better on accuracy measures than filter approach for NB, MNB, NN, and J4.8 using all three relevance measures.
Full vs. Hybrid	• Text classification using full attribute set is time consuming (extremely long training and testing times). • Text classification using full attribute set may not be feasible for some classifiers (e.g., out of memory problem for NN). • Text classifiers built using the hybrid approach are much more cost-efficient than the corresponding classifiers built using the full set. • NB and MNB classifiers built using the hybrid selected subset exhibited much higher accuracy rates than the corresponding classifiers built using the full set.

Acknowledgments

The authors would like to thank the Senior Editor and the three referees for their detailed and constructive feedback during the review process. Their input and feedback helped us to improve the quality of the paper significantly.

References

Anandarajan, M. "Profiling Web Usage in the Workplace: A Behavior-Based Artificial Intelligence Approach," *Journal of Management Information Systems* (19:1) 2002, pp 243 - 254.

Anandarajan, M., Devine, P., and Simmers, C. "A Multidimensional Scaling Approach to Personal Web Usage in the Workplace, in Personal Web Usage in the Workplace: A Guide to Effective Human Resources Management.," M. Anandarajan, C. Simmers and P. Hershey (eds.), Idea Group Inc., 2004, pp. 61-78.

Anandarajan, M., and Simmers, C.A. *Constructive and dysfunctional personal web usage in the workplace: mapping employee attitudes* Idea Group Inc, 2004.

Avrim, L.B., and Pat, L. "Selection of Relevant Features and Examples in Machine Learning.," *Artificial Intelligence* (97:1-2) 1997, pp 245-271.

Baker, L.D., and McCallum, A.K. "Distributional Clustering of Words for Text Classification," *Proceedings of 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 1998, pp. 96-103.

Chakrabarti, S. *Mining the Web: Discovering Knowledge from Hypertext Data* Morgan Kaufmann, San Francisco, CA, 2003.

Chen, R.-C., and Hsieh, C.-H. "Web page classification based on a support vector machine using a weighted vote schema," *Expert Systems with Applications* (31:2) 2006, pp 427-435.

Churilov, L., Bagirov, A., Schwartz, D., Smith, K., and Dally, M. "Data Mining with Combined Use of Optimization Techniques and Self-Organizing Maps for Improving Risk Grouping Rules: Application to Prostate Cancer Patients," *Journal of Management Information Systems* (21:4) 2005.

Dash, M., and Liu, H. "Feature Selection for Classification," *Intelligent Data Analysis* (1:3) 1997, pp 131-156.

Davis, R.A., Flett, G.L., and Besser, A. "Validation of a new scale for measuring problematic Internet use: implications for pre-employment screening," *CyberPsychology & Behavior* (5:4) 2002, pp 331-345.

Debole, F., and Sebastiani, F. "Supervised term weighting for automated text categorization," *Proceedings of the 2003 ACM symposium on Applied computing*, Melbourne, Florida 2003.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R. "Indexing by Latent Semantic Analysis," *Journal of the American Society of Information Science and Technology* (41:6) 1999, pp 391-407.

Ding, C., and Peng, H. "Minimum redundancy feature selection from microarray gene expression data," *Journal of Bioinformatics and Computational Biology* (3:2) 2005, pp 185-205.

Du, R., Safavi-Naini, R., and Susilo, W. "Web Filtering Using Text Classification," *Proceedings of The 11th IEEE International Conference on Networks*, 2003, pp. 325-330.

Fan, W., Wallace, L., Rich, S., and Zhang, Z. "Tapping The Power of Text Mining," *Communications of the ACM* (49:9) 2006, pp 77-82.

Forman, G. "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research* (3:7) 2003, pp 1289-1305.

Fuhr, N., and Buckley, C. "A Probabilistic learning approach for document indexing," *ACM Transactions on Information Systems* (9:3) 1991, pp 223-248.

Galletta, D.F., and Polak, P. "An empirical investigation of antecedents of Internet abuse in the workplace," *Proceedings of the Second Annual Workshop on HCI Research in MIS*, Seattle, WA, 2003, pp. 47-51.

Greenfield, D.N., and Davis, R.A. "Lost in cyberspace: the web @ work," *CyberPsychology & Behavior* (5:4) 2005, pp 347-353.

Greenfield, P., Rickwood, P., and Tran, H.C. *Effectiveness of Internet filtering software products* CSIRO Mathematical and Information Sciences, Sydney, Australia, 2001.

Griffiths, M. "Internet abuse in the workplace: issues and concerns for employers and employment counselors," *Journal of Employment Counseling* (40:2) 2003, pp 87-96.

Hall, M.A., and Holmes, G. "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining," *IEEE Transactions on Knowledge and Data Engineering* (15:6) 2003, pp 1437-1447.

Hammami, M., Chahir, Y., and Chen, L. "WebGuard: A Web Filtering Engine Combining Textual, Structural, and Visual Content-Based Analysis," *IEEE Transactions on Knowledge and Data Engineering* (18:2) 2006, pp 272- 284.

He, Q., Chang, K., and Lim, E.-P. "Anticipatory event detection via classification," *Information Systems and E-Business Management* (5:3) 2007, pp 275-294.

Hu, W., Wu, O., Chen, Z., Fu, Z., and Maybank, S. "Recognition of Pornographic Web Pages by Classifying Texts and Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (29:6) 2007, pp 1019-1034.

Hunter, C.D. "Social impacts: Internet filter effectiveness—testing over-and underinclusive blocking decisions of four popular web filters.," *Social Science Computer Review* (18:2) 2000, pp 214-222.

Inza, I., Larrañaga, P., Blanco, R., and Cerrolaza, A. "Filter Versus Wrapper Gene Selection Approaches in DNA Microarray Domains," *Artificial Intelligence in Medicine* (31:2) 2004, pp 91-103.

Joachims, T. "Text categorization with support vector machine: learning with many relevant features," *Proceedings of the 10th European Conference on Machine Learning*, 1998, pp. 137-142.

Kohavi, R., and John, G.H. "Wrappers for Feature Subset Selection," *Artificial Intelligence* (97:1-2) 1996, pp 273-324.

Kwon, O.-W., and Lee, J.-H. "Text categorization based on k-nearest neighbor approach for Web site classification," *Information Processing & Management* (39:1) 2003, pp 25-44.

Lee, O., Lim, K.H., and Wong, W.M. "Why employees do non-work-related computing: an exploratory investigation through multiple theoretical perspectives," *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005, pp. 185c-185c.

Lee, P.Y., Hui, S.C., and Fong, A.C.M. "Neural Networks for Web Content Filtering," *IEEE Intelligent Systems* (17:5) 2002, pp 48-57.

Lewis, D.D. "An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task," *Proceedings of 15st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Copenhagen, Denmark, 1992, pp. 37-50.

Lewis, D.D., and Ringuette, M. "A Comparison of Two Learning Algorithms for Text Categorization," *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US, 1994.

Lim, K.G. "The IT way of loafing on the job: cyberloafing, neutralizing and organizational justice," *Journal of Organizational Behavior* (23:5) 2002, pp 675-694.

Liu, H., Li, J., and Wong, L. "A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns," *Genome Informatics* (13) 2002, pp 51-60.

Liu, H., and Setiono, R. "Chi2: Feature Selection and Discretization of Numeric Attributes," *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence*, 1995.

Liu, Y. "A comparative study on feature selection methods for drug discovery," *Journal of Chemical Information and Computer Sciences* (44:5) 2004, pp 1823-1828.

Mahatanankoon, P. "Predicting Cyber-Production deviance in the workplace," *International Journal of Internet and Enterprise Management* (4:4) 2006, pp 314-330.

Mahatanankoon, P., Anandarajan, M., and Igbaria, M. "Development of a measure of personal web usage in the workplace," *CyberPsychology & Behavior* (7:1) 2004, pp 93-104.

Mahatanankoon, P., and Igbaria, M. "Impact of Personal Internet Usage on Employee's Well-Being, in Personal Web Usage in the Workplace: A Guide to Effective Human Resources Management," M. Anandarajan, C. Simmers and P. Hershey (eds.), Idea Group Inc., 2004, pp. 246-263.

Malachowski, D. "Wasted time at work costing companies billions. http://salary.com," 2005.

McCallum, A., and Nigam, K. "A Comparison of Event Models for Naive Bayes Text Classification," *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, 1998, pp. 41-48.

Mladenic, D. "Feature Subset Selection in Text-Learning," *Proceedings of European Conference on Machine Learning*, 1998, pp. 95-100.

Polpinij, J., Chotthanom, A., Sibunruang, C., Chamchong, R., and Puangpronpitag, S. "Content-Based Text Classifiers for Pornographic Web Filtering," *Proceedings of IEEE International*

Conference on Systems, Man and Cybernetics,, Taipei, Taiwan, 2006.
Porter, M.F. "An algorithm for suffix stripping " *Program* (14:3) 1980, pp 130-137.
Quinlan, J.R. *C4.5: programs for machine learning* Morgan Kaufmann Publishers, 1993.
Riboni, D. "Feature Selection for Web Page Classification," *Proceedings of EURASIA-ICT 2002 Proceedings of the Workshop*, 2002, pp. 473-477.
Rogati, M., and Yang, Y. "High-Performing Feature Selection for Text Classification," *Proceedings of CIKM'02, ACM*, 2002.
Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C., and Stamatopoulos, P. "A memory-based approach to anti-spam filtering for mailing lists," *Information Retrieval* (6:1) 2003, pp 49–73.
Schneider, K. "A comparison of event models for naive Bayes anti-spam e-mail filtering," *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, 2003.
Sebastiani, F. "Machine learning in automated text categorization," *ACM Computing Surveys* (34:1) 2002, pp 1-47.
Sharma, S., and Gupta, J. "Improving workers' productivity and reducing Internet abuse," *Journal of Computer Information Systems* (44:2) 2003, pp 74-78.
Siau, K., Nah, F., and Teng, L. "Acceptable Internet use policy," *Communications of the ACM* (45:1) 2002, pp 75-79.
Sinha, A.P., and May, J.H. "Evaluating and Tuning Predictive Data Mining Models Using Receiver Operating Characteristic Curves," *Journal of Management Information Systems* (21:3) 2005, pp 249 - 280.
Spangler, S., Kreulen, J.T., and Lessler, J. "Generating and Browsing Multiple Taxonomies Over a Document Collection," *Journal of Management Information Systems* (19:4) 2003, pp 191 - 212.
Stanton, J.M. "Company Profile of the Frequent Internet User," *Communications of the ACM* (45:1) 2002, pp 55-59.
Urbaczewski, A., and Jessup, L.M. "Does electronic monitoring of employee Internet usage work? ," *Communications of the ACM* (45:1) 2002, pp 80-83.
Walczak, S. "An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks," *Journal of Management Information Systems* (17:4) 2001, pp 203 - 222.
Witten, I.H., and Frank, E. *Data mining: practical machine learning tools and techniques, second edition* Morgan Kaufmann Publishers, 2005.
Wong, W.M., Lee, O., and Lim, K.H. "Managing non-work related computing within an organization: the effects of two disciplinary approaches on employees' commitment to change," *Proceedings of the Ninth Pacific Asia Conference on Information Systems*, Bangkok, Thailand, 2005, pp. 441-454.
Woon, I., and Pee, L.G. "Behavioral factors affecting Internet abuse in the workplace – an empirical investigation," *Proceedings of the Third Annual Workshop on HCI Research in MIS*, Washington, DC, 2004, pp. 80-84.
Wyatt, K., and Phillips, J.G. "Internet use and misuse in the workplace," *Proceedings of the 19th conference of the computer-human interaction special interest group of Australia*, 2005.
Yang, Y., and Pedersen, J.O. "A comparative study on feature selection in text categorization," *Proceedings of ICML-97, 14th International Conference on Machine Learning*, 1997, pp. 412-420.
Young, K.S., and Case, C.J. "Internet abuse in the workplace: new trends in risk management," *CyberPsychology & Behavior* (7:1) 2004, pp 105-111.
Zhou, L., Burgoon, J.K., Twitchell, D., and Qin, T. "A comparison of classification methods for predicting deception in computer-mediated communication," *Journal of Management Information Systems* (20:4) 2004, pp 139-166.

About the Authors

Chen-Huei Chou is an Assistant Professor of Management Information Systems and Decision Sciences in the School of Business at the College of Charleston. He received his Ph.D. in MIS from University of Wisconsin-Milwaukee. His areas of interests include Web design issues in disaster management, ontology development, and data mining. His research has been published in MIS journals and major conference proceedings, including *Decision Support Systems, IEEE Transactions on Systems, Man, and Cybernetics, Journal of Information Systems and e-Business Management, Americas Conference on Information Systems, International Conference on Design Science Research in Information Systems and Technology, and Workshop on e-Business.*

Atish P. Sinha is a Professor of MIS at the Sheldon B. Lubar School of Business, University of Wisconsin-Milwaukee. He earned his Ph.D. in business, with a concentration in Artificial Intelligence, from the University of Pittsburgh. His current research interests are in the areas of business intelligence, data mining, text mining, data warehousing, web analytics, and service-oriented computing. His research has been published in several journals, including *Communications of the ACM, Decision Support Systems, IEEE Transactions on Engineering Management, IEEE Transactions On Software Engineering, IEEE Transactions On Systems, Man, And Cybernetics, Information Systems Research, International Journal of Human-Computer Studies,* and *Journal of Management Information Systems*. Professor Sinha is a member of ACM, AIS, and INFORMS. He served as the co-chair of the *16th Workshop on Information Technologies and Systems* (WITS) in 2006.

Huimin Zhao is an Associate Professor of MIS at the Sheldon B. Lubar School of Business, University of Wisconsin-Milwaukee. He earned his Ph.D. in MIS from The University of Arizona. His current research interests are in the areas of data mining, recommendation systems, and data integration. His research has been published in several journals, including *Communications of the ACM*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Systems, Man, and Cybernetics*, *Information Systems*, *Data and Knowledge Engineering*, *Journal of Management Information Systems*, and *Decision Support Systems*. He serves on the editorial review board of the Journal of Database Management and as the treasurer of the INFORMS College on Artificial Intelligence. He served as a co-chair of the *19th Workshop on Information Technologies and Systems* in 2009 and a co-chair of the *5th INFORMS Workshop on Data Mining and Health Informatics* in 2010.