Detecting authorship deception: a supervised machine learning approach using author writeprints

Lisa Pearl and Mark Steyvers
University of California, Irvine

Abstract
We describe a new supervised machine learning approach for detecting authorship deception, a specific type of authorship attribution task particularly relevant for cybercrime forensic investigations, and demonstrate its validity on two case studies drawn from realistic online data sets. The core of our approach involves identifying uncharacteristic behavior for an author, based on a writeprint extracted from unstructured text samples of the author’s writing. The writeprints used here involve stylometric features and content features derived from topic models, an unsupervised approach for identifying relevant keywords that relate to the content areas of a document. One innovation of our approach is to transform the writeprint feature values into a representation that individually balances characteristic and uncharacteristic traits of an author, and we subsequently apply a Sparse Multinomial Logistic Regression classifier to this novel representation. Our method yields high accuracy for authorship deception detection on the two case studies, confirming its utility.

1 Introduction
Detecting authorship deception can be thought of as a specific type of authorship attribution task (Baayen et al., 1996; Diederich et al., 2003; Holmes and Forsyth, 1995; Tweedie et al., 1996 among others), where the basic decision is whether to attribute a given text sample to a given author. This can be particularly relevant for cybercrime forensic investigations where the authorship of a document is in question, and there is little additional evidence to go on beyond the message itself. This process is also sometimes called authorship identification (de Vel et al., 2001; Iqbal et al., 2008, 2010; Li et al., 2006; Zheng et al., 2003) or authorship verification (Koppel et al., 2009), particularly if there is a suspected author who may be trying to anonymize his/her message or actively imitate another author’s writing in order to conceal his/her true identity.

In this article, we describe a new supervised machine learning approach for detecting authorship deception in unstructured text, and show its utility on two case studies drawn from realistic online data sets. The core of our approach involves identifying uncharacteristic behavior for an author, given the document in question, and we do this by extracting a writeprint (Abbasi and Chen, 2008; Iqbal et al., 2008, 2010; Li et al., 2006) for the author from known samples of the author’s writing. We follow a computational stylistics approach (Stamatatos et al., 2000) and draw from both stylometric and content features to define an author’s writeprint.

Stylometric features traditionally involve the internal statistics of a document, and are thought to reflect an unconscious ‘aspect’ (Holmes, 1998) of an author’s style, appearing in the form of distinctive, quantifiable features that are salient, structural, and frequent (Bailey, 1979). Given this, an implicit
assumption is that these features cannot be consciously manipulated (though see Brennan and Greenstadt (2009) for some evidence suggesting this is not true). Some stylometric features investigated previously include lexical features such as function word frequency and vocabulary richness, syntactic features such as passive structures and grammatical category sequences, and structural features such as the format of a signature or paragraph indentation style (Abbasi and Chen, 2008; Baayen et al., 2002; Brennan and Greenstadt, 2009; Burrows, 1987, 1989; de Vel, 2000; Gamon, 2004; Holmes and Forsyth, 1995; Iqbal et al., 2008, 2010; Juola, 2003, 2009; Li et al., 2006; Mosteller and Wallace, 1964; Morton, 1978; Stamatatos et al., 2000; Tweedie et al., 1996).

Content features are based on the semantic content of a message, and are generally less used in authorship attribution tasks since they are seen as more variable, depending on the content the author wishes to express, and so under more conscious control of the author. This would then make them easier to manipulate. Previous studies in authorship attribution have employed content features based on high-frequency domain-specific keywords chosen a priori (Iqbal et al., 2010) and semantic features derived from highly detailed syntactic annotation (Gamon, 2004), with the idea that these semantic features may be less subject to conscious manipulation.

We recognize that feature selection for writeprint is a serious issue (Liu and Motoda, 1998; Iqbal et al., 2008, 2010) and endeavor to incorporate stylometric and content features that are relatively simple to calculate from unstructured text. Because of this, the stylometric features we have chosen are a subset of those explored previously and the content features are derived from topic models (Griffiths and Steyvers, 2004; Rosen-Zvi et al., 2004; Steyvers et al., 2004), an unsupervised approach for identifying relevant keywords that relate to the content areas of a document.

Our machine learning approach differs from previous statistical and machine learning techniques used in authorship attribution studies (e.g. simple frequency comparisons (Mosteller and Wallace, 1964), cross entropy (Burrows, 1987), principal components analysis (Burrows, 1989, 2003; Baayen et al., 2002), genetic algorithms (Holmes and Forsyth, 1995; Li et al., 2006), neural networks (Tweedie et al., 1996), support vector machines (Diederich et al., 2003; Gamon, 2004), and linear discriminant analysis (Baayen et al., 2002), among many others—see work by Juola (2003, 2006, 2009) and Koppel et al. (2009) for a comparison of different methods). In particular, we represent the features used as input to a machine learning method not with the raw values calculated from the unstructured texts, but rather as a set of feature values that individually balance characteristic and uncharacteristic traits of an author. This has the advantage of distilling the unique components of a particular author’s writeprint, such that standout traits are made more salient. We then apply a Sparse Multinomial Logistic Regression (SMLR) classifier (Krishnapuram et al., 2005) to this feature representation, as the SMLR classifier also has the property of identifying a small number of informative writeprint features to base its decision on. This process thus allows the classifier to determine if a document is written by the author in question.

As a demonstration of our approach, we present the results from two practical cases of authorship deception detection. In the first case study, we verify the author of a blog entry from a set of approximately 2,200 blog authors with ten to twenty posts each, derived from the Spinn3r Personal Story Subset (Gordon, 2008). This corresponds to the detection of authors who are attempting to post anonymously or attempting to post under someone else’s name, and our approach achieves 89% accuracy using a combination of stylometric and content features. In the second case study, we detect imita-
tion attacks (from the Attack Corpus of Brennan and Greenstadt (2009)) on a particular author, where the imitators consciously attempted to alter their normal writing style to match the author’s. This corresponds to the detection of online document forgery, and our approach achieves 100% accuracy using stylometric features alone.

In the remainder of this article, we will describe the features of our writeprint implementation, and how they are derived from unstructured text. We will then discuss the machine learning approach...
in more detail, and verify its effectiveness on the two realistic case studies mentioned. For each case study, we will describe the data set, how training and test sets were created, and detailed results of the machine learning approach. We will conclude with general discussion of the results, implications for writeprint characterization, and areas of future research.

2 Writeprint Characterization

Our writeprint characterization can include both stylometric and content features. We have nine principle stylometric feature types, corresponding to eighty-one individual features, as shown in Table 1: character distribution, punctuation mark distribution, fine-grained grammatical category distribution, coarse-grained grammatical category distribution, first person pronoun frequency, lexical diversity, average sentence length, average word length, and total words. We note that these are a subset of available stylometric features used in previous studies, and correspond to fairly shallow linguistic information that is easy to extract from unstructured text. In particular, all these features can be extracted directly using a text manipulation programming language such as PERL and a part-of-speech tagger such as the freely available Stanford Log-linear Part-of-Speech Tagger


Our content features consist of topics, which are probability distributions over keywords that relate to a cohesive concept (see Fig. 1 for some sample topics). These topics, and the keywords that comprise them, are identified in an unsupervised fashion using topic models (Griffiths and Steyvers, 2004) from a collection of documents. Without any additional information beyond the documents themselves, topic models can use the words contained in the documents to identify both the topics

<table>
<thead>
<tr>
<th>Feature type</th>
<th>Description</th>
<th># Implementation</th>
<th>Example calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characters</td>
<td>Letters a, b, c, ... , z, all digits, all punctuation marks</td>
<td>28</td>
<td>(# digits)/(total # character tokens)</td>
</tr>
<tr>
<td>Punctuation marks</td>
<td>?!, ;, ;</td>
<td>6</td>
<td>(#)/(total # punctuation tokens)</td>
</tr>
<tr>
<td>Fine-grained grammatical categories</td>
<td>Part-of-speech tags</td>
<td>37</td>
<td>(# Vb)/(total # word tokens)</td>
</tr>
<tr>
<td>Coarse-grained grammatical categories</td>
<td>Nouns, adjectives, verbs, adverbs, function words(^a)</td>
<td>5</td>
<td>(# nouns)/(total # word tokens)</td>
</tr>
<tr>
<td>1st person pronouns</td>
<td>I, me, my, mine, we, us, our, ours</td>
<td>1</td>
<td>(# 1st person pronouns)/(total # word tokens)</td>
</tr>
<tr>
<td>Lexical diversity(^b)</td>
<td>Word type to word token ratio</td>
<td>1</td>
<td>Same as implementation</td>
</tr>
<tr>
<td>Average sentence length</td>
<td>Average sentence length, based on word tokens in sentence</td>
<td>1</td>
<td>Same as implementation</td>
</tr>
<tr>
<td>Average word length</td>
<td>Average number of characters in a word document, based on alphabetic word tokens</td>
<td>1</td>
<td>Same as implementation</td>
</tr>
<tr>
<td>Total words</td>
<td>Total words</td>
<td>1</td>
<td>Same as implementation</td>
</tr>
</tbody>
</table>

Note that for all proportion calculations (the first five feature types), a smoothing constant (1) was added to the raw counts.

\(^a\)Nouns consist of tags NN, NNS, NNP, NNPS, PRP, and WP. Adjectives consist of tags JJ, JJR, JJS, PDT, PRPS, and WP$. Verbs consist of tags MD, VB, VBD, VBG, VBN, VB, and VBZ. Adverbs consist of tags RB, RBR, RBS, and WRB. Function words consist of tags CC, DT, EX, IN, TO, and WDT.

\(^b\)Values range between 0 and 1, with values near 1 indicating more diverse usage (each word type is used only once or twice).
expressed in a given document and which topic each word, sentence, or subsection of the document most likely belongs to. We report results from content features based on fifty extracted topics.\textsuperscript{2,3} We can also calculate for each author the topics that author is most likely to write about. \textbf{Figure 2} shows the distribution over topics for some sample authors. The most likely topics in this distribution give a high-level summary of the typical content that is associated with the author, such as \textit{gambling games} and \textit{electronic communication} for the author poker\_star. We will use the probability distribution over topics for a given author as the set of content features, with the probability of a given topic for that author being the value for that topic’s content feature for the author.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Probability of keyword, given topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>game</td>
<td>0.03981</td>
</tr>
<tr>
<td>team</td>
<td>0.01637</td>
</tr>
<tr>
<td>games</td>
<td>0.01577</td>
</tr>
<tr>
<td>play</td>
<td>0.01560</td>
</tr>
<tr>
<td>win</td>
<td>0.01014</td>
</tr>
<tr>
<td>poker</td>
<td>0.00961</td>
</tr>
<tr>
<td>casino</td>
<td>0.00942</td>
</tr>
</tbody>
</table>

\textbf{Concept} ≈ gambling games  
\textbf{Topic probability in collection} = 0.01541

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Probability of keyword, given topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>food</td>
<td>0.01274</td>
</tr>
<tr>
<td>eat</td>
<td>0.00830</td>
</tr>
<tr>
<td>chicken</td>
<td>0.00751</td>
</tr>
<tr>
<td>cream</td>
<td>0.00724</td>
</tr>
<tr>
<td>cheese</td>
<td>0.00654</td>
</tr>
<tr>
<td>cake</td>
<td>0.00651</td>
</tr>
<tr>
<td>chocolate</td>
<td>0.00644</td>
</tr>
</tbody>
</table>

\textbf{Concept} ≈ food  
\textbf{Topic probability in collection} = 0.02728

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Probability of keyword, given topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>run</td>
<td>0.01827</td>
</tr>
<tr>
<td>time</td>
<td>0.01704</td>
</tr>
<tr>
<td>running</td>
<td>0.01144</td>
</tr>
<tr>
<td>bike</td>
<td>0.00784</td>
</tr>
<tr>
<td>race</td>
<td>0.00782</td>
</tr>
<tr>
<td>started</td>
<td>0.00780</td>
</tr>
<tr>
<td>miles</td>
<td>0.00741</td>
</tr>
</tbody>
</table>

\textbf{Concept} ≈ racing  
\textbf{Topic probability in collection} = 0.02070

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Probability of keyword, given topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.01700</td>
</tr>
<tr>
<td>store</td>
<td>0.01508</td>
</tr>
<tr>
<td>card</td>
<td>0.01499</td>
</tr>
<tr>
<td>item</td>
<td>0.01362</td>
</tr>
<tr>
<td>items</td>
<td>0.01344</td>
</tr>
<tr>
<td>price</td>
<td>0.01323</td>
</tr>
<tr>
<td>money</td>
<td>0.01117</td>
</tr>
</tbody>
</table>

\textbf{Concept} ≈ commerce  
\textbf{Topic probability in collection} = 0.01101

Fig. 1 Sample topics automatically extracted from a collection of blog entries using a topic model. The top seven keywords most associated with each topic are listed from highest to lowest probability. Given these keywords, an interpretation of the concept the topic represents is provided.

3 Application of Machine Learning Techniques

The basic representation of the problem the authorship deception classifier is designed to solve involves a comparison between a document with unknown authorship (the target document) to a document or set of documents from a known author (the reference document(s)). The classifier must decide if the target document is by the same author as the reference document(s). In order to develop the classifier, the following sets of documents are created for each author:

(i) \( A_1 = \) single randomly chosen target document from the author
The set $A_2$ serves as the reference set for our classifier. These are the known documents that belong to the author. The documents $A_1$ and $X_1$ then serve as target documents that either belong to the original author ($A_1$) or a different author ($X_1$). The training cases for the classifier are not based on single documents but on sets of documents, such that the classifier can learn from the relationships between these sets of documents. In one set, we pair $A_1$ and $A_2$, and based on the input features for this set of documents, the classifier should learn to say that the target document is the same author as the original author. In another set, we pair $X_1$ and $A_2$, and the classifier should learn to say that the target document belongs to a different author from the original author. We also create test cases for the classifier in order to test the generalization performance. For test cases, we again create pairs ($A_1$, $A_2$) and ($X_1$, $A_2$) but now select authors for the reference documents that do not appear in the training set. By doing this, we can test the classifier’s ability to detect authorship deception for new authors.

Instead of applying the classifier to the raw feature values for $A_1$, $A_2$, and $X_1$, we take the additional step of transforming the raw feature values into a more informative representation. Specifically, the classifier examines the target document’s value for a given feature, and compares the likelihood of

**Fig. 2** Sample topic distributions over authors, derived from a collection of blog entries. Lighter shading indicates higher probability, whereas darker shading indicates lower probability. Note that the blog author’s user handle sometimes corresponds directly to the topic, such as *poker_star* and *gamblingnet* for the gambling games topic, *home_cookin*, *Culinarily Obsessed*, and *Ruby Red Vegan* for the food topic, and *Japan’s Garage Sale*, *i_love_to_shop*, and *Shopaholic Heaven!* for the commerce topic. Also note that interpretable correlations among topics emerge. For example, authors who write about commerce also tend to write about fashion, and authors who write about gambling games also tend to write about electronic communication.
that value being produced from the general population of authors against the likelihood of that value being produced by the author in the reference documents. This comparison allows the classifier to determine if the target document’s feature value is unusual for the reference documents’ author.

More specifically, suppose we have feature \( f \) for the target document (either A1 or X1). We first log transform all values for \( f \) from the entire document collection (including all authors), which creates a distribution of values that are roughly normally distributed, provided the sample size is large enough (see Fig. 3 for an example). We then estimate the best-fitting normal distribution for this observed distribution, and call this the \( H_0 \) distribution. This represents the distribution of feature values we expect from the general population of authors.

We then apply the same process to the author reference documents alone (A2). In particular, we log transform all values for \( f \) from the author reference documents, and then estimate the best-fitting normal distribution for this observed distribution. We call this the \( H_1 \) distribution, and this represents the feature distribution for the reference author. The potential difference between the \( H_0 \) distribution and \( H_1 \) distribution is illustrated by the example in Fig. 3. For that feature, the author’s distribution is significantly different from the distribution in the general population, and so this feature could be helpful in identifying that author.

For each feature value \( f_v \) in the target document (either A1 or X1), we then calculate the log odds ratio, which compares the probability of \( f_v \) coming from the general population’s distribution \( H_0 \) to the probability of \( f_v \) coming from the reference author’s distribution \( H_1 \):

\[
\text{Log odds ratio comparison: } \log\left(\frac{p(f_v|H_0)}{p(f_v|H_1)}\right) \quad (1)
\]

A negative value indicates the probability of \( f_v \) coming from the author’s distribution \( H_1 \) is larger than the probability of \( f_v \) coming from the general population’s distribution \( H_0 \), which suggests this feature value is typical for that author and so likely to be from that author. For example, feature value \( f_v = y \) in Fig. 3 illustrates this outcome. Conversely, a positive value indicates the probability of \( f_v \) coming from the author’s distribution \( H_1 \) is smaller than the probability of \( f_v \) coming from the general population’s distribution \( H_0 \), which suggests this feature value is atypical for that author and so likely to be from a different author. In Fig. 3, feature value \( f_v = x \) illustrates this outcome. It is important to note that the distributions \( H_0 \) and \( H_1 \) are calculated separately for each feature in the writeprint. Therefore, the model analyzes the diagnosticity of each feature separately. Whether the particular raw feature values are associated with large or small values is irrelevant—what matters is how much evidence each feature value gives about the target author as compared to any randomly chosen author from the population of all authors.

We then apply a SMLR classifier (Krishnapuram et al., 2005) to the log odds ratios for each feature, training it on the data points from the training set and evaluating its performance on the data points from the test set. This kind of regression analysis allows us to identify classifier features that are particularly useful for detecting authorship deception in the training set. In particular, not all features may be useful and this analysis allows us to downweight the features that are less discriminative.

During training, the classifier is provided with the correct classification for each data point, either same (1) or different (0) (e.g. A1 = 1, X1 = 0).
During testing, the classifier predicts the probability that the target document is from the same author as the reference document (e.g. target = example of A1 document with some probability). The classifier can rank test data points by their probability score, which is the probability the classifier believes the target document is not from the author in question (e.g. target documents with low same probability are likely to be examples of X1 documents, and so from a different author).

4 Two Practical Demonstrations

We now demonstrate our authorship deception detection classifier on two real world data sets.

4.1 Identifying blog authors

In the first case study, we examine the authorship of blog entries, based on a subset of the Spinn3r Personal Story Subset (Gordon, 2008) from the Spinn3r Blog Dataset (Burton et al., 2009), consisting of approximately 28,500 blog posts from 2,194 unique authors and containing approximately 5.3 million words. This subset included authors who had between ten and twenty blog posts, which ensured a reasonable amount of data from each author. The average length of a blog post from this subset sample was 404 words (excluding punctuation). Authorship deception in this case can be thought of as an author attempting to post anonymously and thus conceal his/her identity, or attempting to post as a different author (perhaps by using that author’s username and account). Here, the classifier attempts to decide if a given blog entry belongs to the author in question.

Training and test data sets were constructed, with 75% of the total data used for training and 25% used for testing. This led to 1,646 training authors with 16,460 training cases, and 548 test authors with 5,480 test cases. Note that there was no overlap in authors between training and test, so anything the classifier learned from the training set would be about characterizing author writeprints in general rather than characterizing specific author writeprints.

To construct the writeprint, the classifier used the eighty-one stylometric and fifty content features described in Section 2: character distribution, punctuation mark distribution, fine-grained grammatical category distribution, coarse-grained grammatical category distribution, first person pronoun frequency, lexical diversity, average sentence length, average word length, total words, and fifty topics extracted from the entire blog corpus subset.

As mentioned previously, the SMLR classifier is able to weight features, based on the training set, such that some features are deemed more diagnostic while many are deemed less diagnostic. The thirty most diagnostic writeprint features are shown in Table 2, including both stylometric and content features. Most of these are stylometric, ranging in granularity from individual characters up through document-level characteristics like lexical diversity, though many are at the individual character (e.g. !, r, and :) and fine-grained grammatical category (e.g. proper nouns, possessive pronouns, and non-3rd person singular present tense verbs) level. One content feature does get ranked as highly diagnostic, which is the feature based on a topic that likely corresponds to an informal writing style, given the keywords associated with it (e.g. oh, lol, yeah).

We subsequently applied the SMLR classifier, with its internal ranking of diagnostic features, to the test cases. In order to evaluate the classifier’s performance, we calculated true positive rate (TPR) versus false positive rate (FPR), which are standard metrics in signal detection theory. The TPR describes how often the classifier says the target document is from a different author when it really is from a different author (i.e. detecting deception when it is present), whereas the FPR describes how often the classifier says the target document is from a different author when it is really from the same author as the referent documents (i.e. detecting deception when there is none). This can be represented as shown in (2). The goal of the classifier would be to maximize the TPR while minimizing the FPR.

(2) TPR versus FPR calculations

\[
\text{TP (true positive case)} = \text{classified as different when really different}
\]
FP (false positive case) = classified as different when really same
TN (true negative case) = classified as same when really same
FN (false negative case) = classified as same when really different

TPR versus FPR
TPR = TP/(TP + FN)
FPR = FP/(FP + TN)

Given how these metrics are defined, there are tradeoffs between TPR and FPR. For example, it is easy to get a 100% TPR by classifying every document as from a different author—however, this means that the FPR will also be quite high, which is bad. The receiver operating characteristic (ROC) curve in Fig. 4 shows the tradeoff between TPR and FPR for the classifier on the test data from the blog data set. The area under the ROC curve (abbreviated as AUC in Fig. 4) represents the average probability of the classifier making the correct classification across all true positive/false positive thresholds.

Table 3 shows more detailed results for the classifier on this data set, including how well the
classifier does using writeprints constructed from only stylometric and only content features. In addition, Table 3 provides the results of several existing machine learning methods on this same data set as a baseline for performance, but using only the raw values for the writeprint features, rather than the transformed feature values. These additional methods were run using the freely available Waffles toolkit.\(^4\) In particular, for each comparison machine learning method, the training set consisted of pairs of documents, one target document (A1) and one referent document (X1 or A2), along with a classification of ‘same’ or ‘different’ for each document pair. The test set was similarly constructed.

From these results, we can see quite good performance from our classifier. Using writeprints made of both stylometric and content features that were transformed, our classifier will identify a deceptive author (i.e. a different author from the reference author) 89% of the time on average, with a TPR of 81% and a FPR of 19%. This is a notable improvement over the best-performing comparison machine learning method (the KNN algorithm with 50 neighbors) operating over raw feature values, which had a TPR of 69% and a FPR of 28%. Interestingly, it appears that the contributions of stylometric and content features in the writeprint are not additive. Stylometric features on their own achieve correct classification 88% of the time while content features on their own achieve correct classification 83% of the time. This suggests that stylometric features on their own can be very distinctive of an author, irrespective of the content of an author’s message. However, we also find that sophisticated content features can be fairly distinctive on their own, even if the best performance is found by combining stylometric and content features. We note that the high performance of a classifier using stylometric features only in the writeprint is useful since the topic models that are used to generate the content features require a fairly large number of writing samples (on the order of thousands). This is relevant for situations where we do not have a large quantity of data to work with, as in the next case study.

### 4.2 Identifying imitation attacks

The second case study we examined used data gathered from writers who specifically tried to conduct imitation attacks on an existing author (Brennan and Greenstadt, 2009). Authorship deception in this case can be thought of as an author attempting
to impersonate another author and write a message as that author, concealing his/her own true identity. We used a subset of the Attack Corpus (Brennan and Greenstadt, 2009) that contained a writing sample from one author with a fairly distinctive style (writer Cormac McCarthy) and twelve imitation attacks by writers who saw this sample and attempted to mimic the original author’s style. The attacks ranged in length from 478 to 521 words, with an average length of 497 words. The McCarthy sample they used as a basis for their imitation was 2,541 words long. We obtained 24 additional samples from Cormac McCarthy to supplement the set of author reference documents, for a total of 80,262 words (with an average of 3,210 words per sample).

Because this data set was too small to extract topics as content features, we used a writeprint that consisted only of the 81 stylometric features used in the first case study. Training and test sets were constructed, with 70% of the writing samples used for training and 30% of the writing samples used for testing. There was no overlap in the imitation samples between training and test. This led to a training set consisting of 80 different author cases (X1 versus A2) and 50 same author cases (A1 versus A2), and a test set consisting of 40 different author cases (X1 versus A2) and 70 same author cases (A1 versus A2).

The results of our classifier were excellent, with the classifier achieving perfect detection of imitation attacks, as shown in Table 4: 100% TPR and 0% FPR. This is a substantial improvement over the results found with state-of-the-art methods tested by Brennan and Greenstadt (2009), such as the Signature stylometric system, neural networks, and a synonym-based classifier, which had average accuracy scores below 5%, and a highest accuracy score around 10%. In addition, Table 4 provides the results of several existing machine learning methods on this same data set as a baseline for performance, but using only the raw values for the writeprint features, rather than the transformed feature values. These additional methods were run using the freely available Waffles toolkit. In particular, for each comparison machine learning method, the training set consisted of pairs of documents, one target document (A1) and one referent document (X1 or A2), along with a classification of ‘same’ or ‘different’ for each document pair. The test set was similarly constructed.

Surprisingly, we found that all of the machine learning methods we applied to this corpus using the 81 stylometric features gave this same excellent performance—whether the feature values were transformed (as in the case of the SMLR classifier) or not (as in the case of the rest of the machine learning methods). This may be due to the diagnostic nature of some of the stylometric features—it turned out that many of the stylometric features were able to individually distinguish true author samples from imitations, such as lexical diversity, part-of-speech usage, and average sentence length. Figure 5 shows the transformed feature values for lexical diversity, average sentence length, proportion of adjectives, and proportions of verbs when comparing writing samples from the original author, McCarthy, with imitation writing samples and additional normal writing samples available for imitators from the Attack Corpus (there were 63,000 words in the normal writing samples, with an average length of 500 words per sample). Notably, despite the imitators’ conscious attempts to mimic McCarthy’s style, their lexical diversity and adjective

<table>
<thead>
<tr>
<th>Classifier performance</th>
<th>TPR</th>
<th>FPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMLR writeprint features (transformed values)</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>81 stylometric features</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The best balance of TPR and FPR is provided for the SMLR classifier. In addition, the results of several other machine learning methods that use the same stylometric features (but raw feature values rather than transformed values) as well as the same training and test sets are provided for comparison.
usage was much higher and therefore more similar to their normal writing style. Similarly, Fig. 5 shows that even though verb usage and average sentence length did change in the imitation condition (and so were more subject to conscious manipulation), the imitators did not change their style sufficiently to successfully mimic McCarthy's style.

Both these case studies on realistic authorship data demonstrate the effectiveness of our approach for detecting authorship deception. In each case, our classifier has yielded excellent results, using writeprints based either on both stylometric and content features, or stylometric features alone. Notably, while the particular representation of features that we propose is not always necessary to achieve good performance when paired with state-of-the-art machine learning methods (as in the case of the imitation attacks), it can significantly improve performance in some more challenging cases (as in the blog entries). In particular, we feel the blog data set was likely more challenging due to the diversity of authors used as target authors—compared with the target author in the imitation attack data set (Cormac McCarthy), who had a markedly distinctive writing style, the target authors for the blog data set were less likely to all have such distinctive styles.

Fig. 5 An illustration of several stylometric feature distributions for McCarthy's writing samples, the imitation attack samples, and the imitators' normal writing samples. The line in the center of each box indicates the median value, while the boxes indicate the 25th and 75th percentiles, and the dashed lines indicate the full range of values (excluding outliers).
This can make them more difficult to identify as distinct from each other, and as such, may represent a more realistic authorship deception scenario.

5 Discussion and Conclusion

In this article, we have described a new supervised machine learning approach to detecting authorship deception, involving a novel representation of writeprint feature values that is paired with a classifier which bases its decisions on highly informative features that it identifies from the training set. Importantly, this approach can be combined with a writeprint characterized by any kind and number of features, as demonstrated in the two case studies, where the writeprints were defined differently in each. Moreover, the feature representation may be combined with a number of different machine learning methods to achieve very good performance.

The characterization of an author's writeprint is one of the keys to successful authorship deception detection. Here, we have identified a more sophisticated kind of content feature for writeprints, based on topic models, that may be successfully integrated into a writeprint characterization. Interestingly, we found that these topic-based features are highly successful at characterizing an author on their own, though they also improve the classifier’s performance when integrated into a writeprint containing stylometric features.

Incorporating more sophisticated features that combine stylometry and content is an important area for writeprint research. For example, some features worth considering, particularly in the realm of online communication where messages may be created in a more fluid manner similar to conversational patterns, are distinctive capitalization patterns and emoticon patterns (e.g. I’m versus I’m; :) versus :-D) and distinctive synonym usage (e.g. Daddy versus Dad; heya versus hi; fabulous versus great; heck versus hell). Some of these features may be more or less difficult to consciously manipulate, which allows us to gauge their utility in author writeprints.

To conclude, we believe that this study highlights how stylometrics, computational linguistics, and machine learning can be combined to yield informative writeprints for authorship deception detection, and that this provides a useful basis for future cybercrime forensic investigations.

References


detect authorship deception


Notes
1 The list of tags from the Stanford Part-of-Speech tagger used is as follows, with an example of each tag in parentheses: CC = coordinating conjunction (and), CD = cardinal number (one penguin), DT = determiner (the), EOS = end of sentence marker (there’s a penguin here!), EX = existential there (there’s a penguin here), FW = foreign word (hola), IN = preposition or subordinating conjunction (after), JJ = adjective (good), JJR = comparative adjective (better), JJ$S$ = superlative adjective (best), LS = list item marker (one, two, three, . . .), MD = modal (could), NN = singular or mass noun (penguin, ice), NNS = plural noun (penguins), NNP = proper noun (Jack), NNPS = plural proper noun (There are two Jacks?), PDT = predeterminer (all the penguins), POS = possessive ending (penguin’s), PRP = personal pronoun (me), PRP$S$ = possessive pronoun (my), RB = adverb (easily), RBR = comparative adverb (later), RBS = superlative adverb (most easily), RP = particle (look it up), SYM = symbol (this = that), TO = infinitival to (I want to go), UH = interjection (oh), VB = base form of verb (we should go), VBD = past tense verb (we went), VBG = gerund or present participle (we are going), VBN = past participle (we should have gone), VBP = non-3rd person singular present tense verb (you go), VBZ = 3rd singular present tense verb (he goes), WDT = wh-determiner (which one), WP = wh-pronoun (who), WP$S$ = possessive wh-pronoun (whose), WRB = wh-adverb (how).
2 The topic model is implemented using a hierarchical Bayesian model, and Gibbs sampling is used to converge on the topics (see Griffiths and Steyvers (2004) for details). We use model parameters $\alpha = 0.1$ and $\beta = 0.001$, with 250 iterations, four chains, and a single sample saved at the end of each chain.
3 Note that we also looked at using 100 and 200 extracted topics, but found that results with fifty were better, perhaps due to the quantity of data available. In addition, the topics were more easily interpretable when we used fifty topics, as opposed to a higher number of topics.
4 http://waffles.sourceforge.net/