Classify, but Verify: Breaking the Closed-World Assumption in Stylometric Authorship Attribution

Ariel Stolerman, Rebekah Overdorf, Sadia Afroz and **Rachel Greenstadt**
The Privacy, Security and Automation Lab
Drexel University
Philadelphia, PA
stolerman,rjo43,sa499,greenie@cs.drexel.edu

Abstract

Forensic stylometry is a form of authorship attribution that relies on the linguistic information found in a document. While there has been significant work in stylometry, most research focuses on the closed-world problem where the document's author is in a known suspect set. For open-world problems where the author may not be in the suspect set, traditional methods used in classification are ineffective. We propose the *Classify-Verify* method, that augments classification with a binary verification step, evaluated on stylometric datasets, but can be generalized to any domain. We suggest augmentations to an existing distance-based authorship verification method, by adding per-feature standard deviations and per-author threshold normalization. The *Classify-Verify* method significantly outperforms traditional classifiers in open-world settings ($p\text{-}val < 0.01$) and attains F1-score of 0.87, comparable to traditional classifiers performance in closed-world settings. Moreover, *Classify-Verify* successfully detects adversarial documents where authors deliberately change their style, where closed-world classifiers fail.

Keywords. Forensic Stylometry, Authorship Attribution, Authorship Verification, Abstaining Classification, Machine Learning.

1 Introduction

The web is full of anonymous communication, with high value for digital forensics. For this purpose, forensic stylometry is used to analyze anonymous communication in order to "de-anonymize" it. Classic stlyometric analysis requires an exact set of suspects in order to perform reliable authorship attribution, settings that are often not met in real world problems. In this paper we break the closed-world assumption and explore a novel method for forensic stylometry that copes with the possibility that the true author is not in the set of suspects.

Stylometry is a form of authorship recognition that relies on the linguistic information found in a document. While stylometry existed before computers and artificial intelligence, the field is currently dominated by AI techniques such as neural networks and statistical pattern recognition. State-of-the-art stylometry approaches can identify individuals in sets of 50 authors with over 90% accuracy (Abbasi and Chen, 2008), and even scaled to over 100,000 authors (Narayanan et al., 2012). Stylometry is currently used in intelligence analysis and forensics, with increasing interest for digital communication analysis (Wayman et al., 2009). The 2009 Technology Assessment for the State of the Art Biometrics Excellence Roadmap (SABER) commissioned by the FBI stated that, "As non-handwritten communications become more prevalent, such as blogging, text messaging and emails, there is a growing need to identify writers not by their written script, but by analysis of the typed content (Wayman et al., 2009)." With the continuous increase in rigorousness, accuracy and scale of stylometric techniques, law practitioners turn to stylometry for forensic evidence (Chaski, 2013; Juola, 2013), albeit considered controversial at times and nontrivial to be admitted in court (Clark, 2011).

The effectiveness of stylometry has considerable implications for anonymous and pseudonymous speech. Recent work has exposed limits on stylometry through active circumvention (Brennan et al., 2012; McDonald et al., 2012). Stylometry has thus far focused on limited, closed-world models. In the classic stylometry problem, there are relatively few authors (usually fewer than 20, nearly always fewer than 100), the set of possible authors is known, every author has a large training set and all the text is from the same genre. However, problems faced in the real world often do not conform to these restrictions.

Controversial, pseudonymous documents that are published on the Internet often have an unbounded suspect list. Even if the list is known with certainty, training data may not exist for all suspects. Nonetheless, classical stylometry requires a fixed list and training data for each suspect, and an author is always selected from this list. This is problematic both for forensics analysts, as they have no way of knowing when widening their suspect pool is required, and for Internet activists as well, who may appear in these suspect lists and be falsely accused of writing certain documents.

We explore a mixed closed-world and open-world authorship attribution problem where we have a known set of suspect authors, but with some probability (known or unknown) that the author we seek is not in that set. The key contributions of this paper are:

1) We present the *Classify-Verify* method. This novel method augments authorship classification with a verification step, and obtains similar accuracy on open-world problems as traditional classifiers in closed-world problems. Even in the closed-world case, *Classify-Verify* can improve results by replacing wrongly identified authors with "unknown." Our method can be tuned to different levels of rigidity, to achieve desired false positive and false negative error rates. However, it can be automatically tuned, whether or not we know the expected proportion of documents by authors in the suspect list versus those that are absent.

2) *Classify-Verify* performs better in adversarial settings than traditional classification. Previous work has shown that traditional classification performs near random chance when faced with writers who change their style. *Classify-Verify* filters out most of the attacks in the Extended-Brennan-Greenstadt Adversarial corpus (Brennan et al., 2012), an improvement over previous work which requires training on adversarial data for attack detection (Afroz et al., 2012).

In addition we present the Sigma Verification method. This method is based on Noecker and Ryan's (Noecker and Ryan, 2012) *distractorless* verification method, which measures distance between an author and a document. Sigma Verification incorporates pairwise distances within the author's documents and the standard deviations of the author's features, and although not proven to statistically out-

perform the distractorless method always, it is yet shown as a better alternative suitable for datasets with certain characteristics.

In Sec. 2 we present a formal definition of the closed-world, open-world, verification, and classify-verify problems. Sec. 3 contextualizes our contribution in terms of related work. Sec. 4 describes the datasets we worked with. In Sec. 5 and Sec. 6 we discuss the closed-world classification and binary verification methodologies used later by our *Classify-Verify* method, presented in Sec. 7. Sec. 8 presents the evaluation of the *Classify-Verify* method on standard and adversarial datasets. We conclude with a discussion of interesting results (Sec. 9), how *Classify-Verify* can be applied in other security and privacy domains and directions for future work (Sec. 10).

2 Problem Statement

2.1 Definitions

In order to understand the problem we explore in this paper, first we describe the pure closed-world and open-world stylometry problems.

The closed-world stylometry problem, namely *authorship attribution*, is: given a document D of unknown authorship and documents by a set of known authors $\mathcal{A} = \{A_1, ..., A_n\}$, determine the author $A_i \in \mathcal{A}$ of D. This problem assumes D's author is in $\mathcal{A}$. The open-world stylometry problem is: given a document D, identify who its author is. *Authorship verification* is a slightly relaxed version (yet very hard): given a document D and an author A, determine whether D is written by A.

Finally, the problem we explore is a mixture of the two above: given a document D of unknown authorship and documents by a set of known authors $\mathcal{A}$, determine the author $A_i \in \mathcal{A}$ of D, *or* that D's author is not in $\mathcal{A}$. This problem is similar to the attribution problem, with the addition of the class "unknown." An extended definition also includes $p = Pr[A_D \in \mathcal{A}]$, the probability that D's author is in the set of candidates.

For the rest of the paper, we look at test documents in two settings: when the authors of these documents are in the set of suspects, denoted *in-set*, and when these documents are by an author outside the suspect set, denoted *not-in-set*.

2.2 Hypothetical Scenario

Consider Bob's workplace which he shares with $n-1$ other employees, under the management of Alice. Bob gets up to get a cup of coffee, and incautiously forgets to lock his computer. When he returns to his desk he discovers that a vicious (and sufficiently long) email has been sent in his name to Alice! He quickly goes to Alice in order to explain, and Alice decides to check the authorship of the email to assert Bob's innocence (or refute it). Luckily Alice has access to the company's email database, so she can model the writing style of her n employees. Unluckily, the security guard at the door tends to doze off every once in a while, resulting with unauthorized people wandering off in the company's halls, such that the portion of authorized people in the office at any given time is p.

A closed-world system would only be able to consider the n employees and identify one of them as the culprit. This would be problematic if the email was written by one of the unauthorized entrants. A *Classify-Verify* approach would be able to consider this possibility.

2.3 Problems with Closed-World Models

Applying closed-world stylometry in open-world settings suffers from a fundamental flaw: a closed-world classifier will *always* output some author in the suspect set. If it outputs an author, it merely means the document in question is written in a style more similar to that author's style than the others', and the probability estimates of the classifier reflect only who is the least-worst choice. Meanwhile, the absence of the document's author from the set of suspects remains unknown. If we relax the precision of our results to k-accuracy (Narayanan et al., 2012), i.e. target to narrow down our set of suspects to k rather than just one, the problem will not be solved – all k options will still be wrong.

This problem becomes prominent especially in online domains, where the number of potential suspects can be virtually unbounded. Failing to address the limitations of closed-world models may result in falsely attributed authors with consequences for both the forensic analyst and the innocent Internet user.

3 Related Work

3.1 Open-World Classification

Open-world classification deals with scenarios in which the set of classes is not known in advance. Approaches include unsupervised, semi-supervised and abstaining classification. Unsupervised stylometry clusters instances based on their feature vector dis-

tances (Abbasi and Chen, 2008; Koppel et al., 2011a). Semi-supervised methods are proposed to identify clusters (Sorio et al., 2010) which are later used in supervised classification. Abstaining classifiers refrain from classification to improve the classifier's reliability in certain situations, for example, to minimize misclassification rate by rejecting the results when the classifier's confidence is low (Pietraszek, 2005; Chow, 1970; Herbei and Wegkamp, 2006). The *Classify-Verify* method is an abstaining classifier that rejects/accepts an underlying classifier's output using a verification step based on the distances between the test author and the predicted author. The primary novelty of this work is that, unlike other stylometric techniques, *Classify-Verify* considers the open-world situation where the author may not be in the suspect set.

Another way is to make a model of the closed-world and reject everything that does not fit it. Although approaches like this are criticized for network intrusion detection (Sommer and Paxson, 2010), in biometric authentication systems distance-based methods for anomaly detection work well (Araujo et al., 2005; Killourhy, 2012; Lee and Cho, 2007).

3.2 Authorship Classification

In authorship classification, one of the authors in a fixed suspect set is attributed to the test document. Current stylometry methods achieve over 80% accuracy with 100 authors (Abbasi and Chen, 2008), over 30% accuracy with 10,000 authors (Koppel et al., 2011b), and over 20% precision with 100,000 authors (Narayanan et al., 2012). None consider the case where the true author is missing. Though stylometric techniques for classification work well, they can be easily circumvented by imitating another person or deliberate obfuscation (Brennan and Greenstadt, 2009).

3.3 Authorship Verification

In authorship verification we aim to determine whether a document D is written by an author A or not. This problem is harder than the closed-world stylometry discussed above. Authorship verification is essentially a one-class classification problem, on which a reasonable amount of research is done, mostly with support vector machines (Tax, 2001; Schölkopf et al., 2001; Manevitz and Yousef, 2007). However, little research is done in the domain of stylometry.

Most previous work addresses verification for plagiarism detection (van Halteren, 2004; Clough, 2000; Meyer zu Eissen et al., 2007). The *unmasking* algorithm (Koppel et al., 2007) is an example of a general approach to verification, that relies on measuring "depth-of-difference" between document and author models. It reaches 99% accuracy with similar false positive and false negative rates, however it is limited to tasks with large training data (Sanderson and Guenter, 2006).

Noecker and Ryan (Noecker and Ryan, 2012) propose the *distractorless* verification method, that avoids using negative samples to model the class of *not the author*. They use simplified feature sets constructed only of character or word n-grams, normalized dot-product (cosine distance) and an acceptance threshold. They evaluate their approach on two corpora (Juola, 2004; Potthast et al., 2011), and report up to 88% and 92% accuracy. Their work provides a verification framework robust across different types of writings (language, genre or length independent). However, their results also suffer from low F-score measurements (up to 47% and 51%), which suggest a skew in the test data (testing more non-matching document-author pairs than matching ones). A closer look at this method along with error rates is given in Sec. 6.2.

The main novelty of this work is in utilizing verification for elevating closed-world attribution to open-world, which, to the best of our knowledge, has not been done before.

4 Corpora

We experiment with two corpora: the Extended-Brennan-Greenstadt (EBG) Adversarial corpus (Brennan et al., 2012) and the ICWSM 2009 Spinn3r Blog dataset (blog corpus) (Burton et al., 2009).

The EBG corpus contains writings of 45 different authors, with at least 6,500 words per author. It also contains adversarial documents, where the authors change their writings style either by hiding it (obfuscation attack), or imitating another author (imitation attack). Most of the evaluations in this paper are done using the EBG corpus.

The Spinn3r blog corpus, provided by Spinn3r.com, is a set of 44 million blog posts made between August 1st and October 1st, 2008. The posts include the text as syndicated, as well as metadata such as the blog's homepage, timestamps, etc. This dataset has been previously used in internet scale authorship attribution (Narayanan et al., 2012). We use a subcorpus of 50 blogs with at least 7500 words as our blog cor-

pus. We use the blog corpus as control, evaluated under the same settings as the EBG corpus, in order to avoid overfitting configurations on the latter, and generalize our conclusions.

5 Closed-World Setup

Throughout the paper we utilize a closed-world classifier, both for baseline results to evaluate different methods, and as the underlying classifier for the *Classify-Verify* method presented in section 7. We use Weka's (Hall et al., 2009) linear kernel sequential minimal optimization support vector machine (Platt, 1998) (SMO SVM) with complexity parameter $C = 1$. SVMs are chosen due to their proven effectiveness for stylometry (Juola, 2008), specifically for the EBG corpus (Brennan et al., 2012).

In addition to the classifier selection, another important part of a stylometric analysis algorithm is the feature set used to quantify the documents, prior to learning and classification. The EBG corpus was originally quantified using the *Writeprints* feature set (Brennan et al., 2012), based on the Writeprints algorithm (Abbasi and Chen, 2008), which is proven accurate on high number of authors (over 90% accuracy for 50 authors). Writeprints uses a complex feature set that quantifies different linguistic levels of the text, including lexical, syntactic, and content related features (see original paper for details); however, for simplicity we choose to use a feature set that consists only of one type of feature. We evaluated the EBG corpus using 10-fold cross validation [1] with the k most common word n-grams or character n-grams, with k from 50 to 1000 with steps of 50, and n from 1 to 5 with steps of 1. The most-common feature selection heuristic is commonly used in stylometry (Abbasi and Chen, 2008; Koppel and Schler, 2004; Noecker and Ryan, 2012) to improve performance and avoid over-fitting, as are the chosen ranges of k and n. F1-score results for character n-grams are illustrated in Fig. 1.

Of word and character n-grams, the latter outperform the first, with the best F1-score results attained with character bigrams at ≈ 0.93 (for $k = 400$ and above), compared to the best score of 0.879 for words, attained using $n = 1$ and $k = 1000$. Both feature sets outperform the original EBG evaluation

[1] In k-fold cross validation the dataset is randomly divided into k equally-sized subsets (folds), where each subset is tested against models trained on the other subsets, and the final result is the weighted average of all subset results.

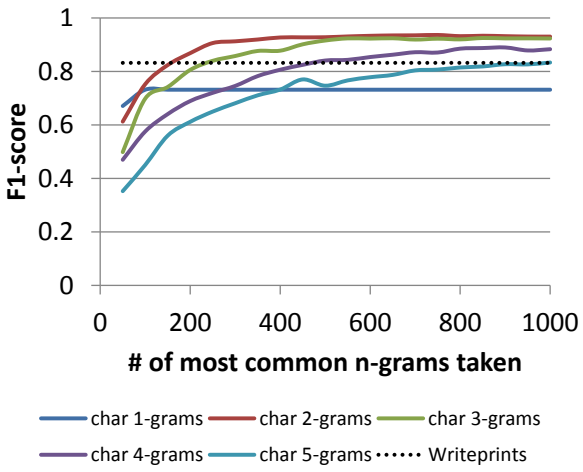

Figure 1: F1-scores for evaluation of the EBG corpus using different character n-grams with varying limits of the feature set size.

with *Writeprints* at F1-score of 0.832 (Brennan et al., 2012). Finally, we choose the 500 most common character bigrams as our feature set (at F1-score of 0.928), denoted $\langle 500, 2\rangle$-*chars*, used throughout all of our experiments. It is chosen for its simplicity, performance and effectiveness.

For control, we evaluated the effectiveness of using $\langle 500, 2\rangle$-*chars* compared to using the *Writeprints* feature set on the blog corpus, via 10-fold cross validation with an SMO SVM classifier. Although both results were lower than those obtained for the EBG corpus, $\langle 500, 2\rangle$-*chars* outperformed *Writeprints* with F1-score of 0.64 versus 0.509, respectively.

Feature extraction for all experiments in the paper is done using the JStylo [2] and JGAAP [3] authorship attribution frameworks API.

6 Verification

In authorship verification we aim to determine whether a document D is written by an author A or not, a problem for which two naïve approaches suggest themselves. The first and most intuitive is to reduce the problem to closed-world settings by creating a model for *not-A* (simply from documents not written by A) and train a binary classifier. This method suffers from a fundamental flaw, as if D is attributed to A it merely means D's style is less distant from A than it is from *not-A* [however the opposite direc-

[2] `https://github.com/psal`
[3] `http://evllabs.com/jgaap/w`

tion, when D is attributed to *not-A*, is useful (Koppel et al., 2007)]. Another approach is to train a binary model of D versus A, and test the model on itself using cross-validation; if D is written by A, we expect the accuracy to be close to random, due to the indistinguishability of the models. However this method does not work well, and requires D to contain a substantial amount of text for cross-validation, an uncommon privilege in real-world scenarios.

In the next sections we discuss and evaluate several verification methods. The first family of methods is *classifier-induced* verifiers, which require an underlying (closed-world) classifier and utilize its class probabilities output for verification.

The second family of methods is *standalone* verifiers, which rely on a model built using author training data, independent of other authors or classifiers. We evaluate two verification methods: the first is the *distractorless* verification method, denoted V, developed by Noecker and Ryan (Noecker and Ryan, 2012). It is presented as a baseline as it is a straight forward verification method, proven robust across different domains, and does not use a distractor set (model of "*not-A*"). Then we present the novel *Sigma Verification* method, which applies adjustments to V for increased accuracy: adding per-feature standard deviations normalization (denoted V_σ) and adding per-author threshold normalization (denoted V^a; the method with both adjustments combined is denoted V^a_σ). Finally, we evaluate and compare V with its new variants.

6.1 Classifier-Induced Verification

One promising aspect of the closed-world model that can be used in open-world scenarios is the confidence in the solution given by distance-based classifiers. A higher confidence in an author may, naturally, indicate that the author is in the suspect set while a lower confidence may indicate that he is not and that this problem is, in fact, an open-world situation.

Following classification, verification can be formulated simply by setting an acceptance threshold t, measure the confidence of the classifier in its classification, and accept the classification if and only if it is above t.

Next we discuss several verification schemes, based on classification probabilities outputted by closed-world classifiers. For each test document D with suspect authors $\mathcal{A} = \{A_1, ..., A_n\}$, a classifier produces a list of probabilities P_{A_i} which is, according to the classifier, the probability D is written by A_i ($\sum_{i=1}^{n} P_{A_i} = 1$). We denote the probabilities $P_1, ..., P_n$ as the reverse order statistic of P_{A_i}, i.e. P_1 is the highest probability given to some author (the chosen author), P_2 the second highest and so on.

These methods are obviously limited to classify-verify scenarios, as verification is dependent on classification results (thus they are not evaluated in this section, but rather in Sec. 8 as part of the *Classify-Verify* evaluation). For this purpose, and in order to extract the probability measurements required by the following methods, we use SMO SVM classifiers with the $\langle 500, 2\rangle$-*chars* feature set for all of our experiments in Sec. 8. We fit logistic regression models to the SVM outputs for proper probability estimates. The classifier-induced verification methods we evaluate are the following:

1) $\mathbf{P_1}$: The first measurement we look at is simply the classifier's probability output for the chosen author, namely P_1. The hypothesis behind this measurement is that as the likelihood the top author is the true author increases, relative to all others, so does its corresponding probability.

2) $\mathbf{P_1}$-$\mathbf{P_2}$-Diff: Another measurement we consider is the difference between the classifier's probability outputs of the chosen and second-to-chosen authors, i.e. $P_1 - P_2$, denoted as the *P_1-P_2-Diff* method.

3) Gap-Conf: The last classifier-induced method we consider is the gap confidence (Paskov, 2010). Here we do not train one SVM classifier; instead, for all n authors, we train corresponding n one-versus-all SVMs. For a given document D, each classifier i in turn produces 2 probabilities: the probability that D is written by A_i and the probability it is not (i.e. the probability it is written by an author other than A_i). For each i, denote the probability that D *is* written by A_i as $p^i(\text{Yes}|D)$; then the gap confidence is the difference between the highest and second-highest $p^i(\text{Yes}|D)$, which we denote briefly as *Gap-Conf*. The hypothesis is similar to *P_1-P_2-Diff*: the probability of the true author should be much higher than that of the second-best choice.

6.2 Standalone Verification

6.2.1 V: Distractorless Verification

In this section we describe the Distractorless verification method, denoted V, proposed by Noecker and Ryan (Noecker and Ryan, 2012). As discussed in Sec. 3, V uses straight-forward distance combined with a threshold: set an acceptance threshold t, model document D and author A as feature vectors, measure the distance between them, and determine

D is written by A if it is below t.

The algorithm begins with preprocessing of the documents (D and A's), in which whitespaces and character case are standardized. The authors use character n-grams and word n-grams as feature sets, extracted using a sliding window technique, due to their known high performance in stylometry, simplicity, fast calculation and robustness against errors that are found in more complex feature extractors, like part-of-speech taggers. Let n denote the size of the chosen feature set.

Next, a model $M = \langle m_1, m_2, ..., m_n \rangle$ is built from the centroid of the feature vectors of A's documents. For each i, m_i is the average *relative* frequency of feature i across A's documents, where relative frequency is used to eliminate document length variation effect. In addition, a feature vector $F = \langle f_1, f_2, ..., f_n \rangle$ is extracted from D, where f_i corresponds to feature i's relative frequency in D.

Finally, a distance function δ and a threshold t are set, such that if $\delta(x, y) < \delta(x, z)$, x is considered to be *closer to* y than to z. The authors use normalized dot-product (cosine distance), defined as:

$$\delta(M, F) = \frac{M \cdot F}{\|M\|\|F\|} = \frac{\sum_{i=1}^{n} m_i f_i}{\sqrt{\sum_{i=1}^{n} m_i^2}\sqrt{\sum_{i=1}^{n} f_i^2}}$$

as it is shown effective for stylometry (Noecker and Juola, 2009) and efficient for large-scale datasets. Note that the authors define *closer to* using $>$ instead of $<$, which is consistent with cosine distance (where 1 is perfect match). However, we use $<$ as the more intuitive direction (according to which a smaller distance means better match), and adjust cosine distance δ in the equation above to $1 - \delta$.

The threshold t is set such that we determine that D is written by A when $\delta(M, F) < t$. Ideally, it is empirically determined by analysis of the average δ between the author's training documents. However, although mentioned in their paper, they evaluate their method using a hard coded threshold that does not undertake these author-wise δ's into account (which V^a does, as seen next).

6.2.2 V_σ, V^a, V^a_σ: Sigma Verification

We apply two adjustments to V:

1) V_σ: Per-Feature SD Normalization – The first improvement is based on the variance of the author's writing style. If an author has rather unvaried style, we want a tighter bound for verification, whereas for a more varied style we can loosen the model to be more accepting. To do so we use the standard deviation of an author, denoted SD, on a per-feature basis. For each author, we determine the SD of all of his features. When computing distance between an author and a document, we divide each feature-distance by its SD, so if the SD is smaller, A and D move closer together, otherwise they move farther apart. This idea is applied in (Araujo et al., 2005) for authentication through typing biometrics, however, to the best of our knowledge, this is the first use of this method for stylometric verification.

2) V^a: Per-Author Threshold Normalization – The second improvement we offer is to adjust the verification threshold t on a per-author basis, based on the average pairwise distance between all of the author's documents, denoted δ_A. V does not take this into account and instead uses a hard threshold. Using δ_A to determine the threshold is, intuitively, an improvement because it accounts for how spread out the documents of an author are. This allows the model to relax if the author has a more varied style. Similarly to V, this "varying" threshold is still applied by setting a fixed threshold t across all authors, determined empirically over the training set; however for V^a every author-document distance measurement δ is adjusted by *subtracting* δ_A prior to being compared with t, thus allowing per-author thresholds but still requires the user to set *only* one fixed threshold value.

Tab. 1 details the differences in distance calculations and threshold test among V, V_σ and V^a. We denote $\delta_{D,A}$ as the overall distance measured by some distance metric δ between the feature vector of document D and the centroid vector of author A across all his documents, denoted $C(A)$. In addition we denote the respective feature level representation of δ as follows: $\delta_{D,A} = \Delta(D_i, C(A)_i)_{i=1}^{n}$, where n is the number of features (dimension of D and $C(A)$). Finally, we define $\sigma(A)$ as the standard deviation vector of author A's features, and δ_A as the pairwise distance between all of A's documents.

Note that using V^a may derive nonintuitive thresholds (e.g. negative thresholds when using cosine distance, which normally produces values in $[0, 1]$). However this is only to adjust to the distance shift from $\delta_{D,A}$ (used in V) by δ_A to $\delta_{D,A} - \delta_A$ used by V^a, i.e. it is a byproduct of the per-author threshold normalization.

Distance \ Test	$\delta < t$	$\delta - \delta_A < t$
$\delta_{D,A} = \Delta(D_i, C(A)_i)_{i=1}^n$	V	V^a
$\delta^\sigma_{D,A} = \Delta(\frac{D_i}{\sigma(A)_i}, \frac{C(A)_i}{\sigma(A)_i})_{i=1}^n$	V_σ	V^a_σ

Table 1: Differences in distance calculation and t-threshold test for V, V_σ and V^a.

6.3 Standalone Verification: Evaluation

We evaluate the methods above on the the EBG corpus, and the blog corpus as control. The evaluation is done by examining false positive and false negative error rates, visualized via ROC curves. The EBG corpus is evaluated only on the non-adversarial documents, and the blog corpus is evaluated entirely. The evaluation is done using 10-fold cross-validation with the $\langle 500, 2\rangle$-*chars* feature set, where author models are built using the training documents. In each fold, every test document is tested against every one of the authors models, including its own (which is obviously trained on other documents of the author).

ROC curves for evaluation of V, V_σ and V^a_σ are illustrated in Fig. 2 and Fig. 3 for the EBG and the blog corpora, respectively.

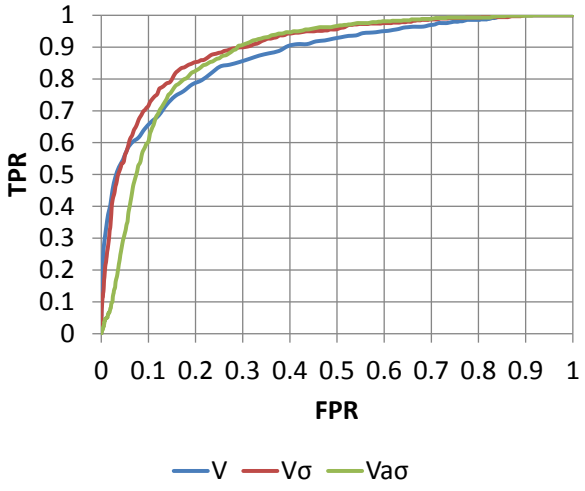

Figure 2: ROC curves for V, V_σ and V^a_σ evaluation on the EBG corpus.

On the EBG corpus V_σ significantly outperforms V (and can be seen clearly from $FP = 0.05$ and above), at a confidence level of $p\text{-}val < 0.01$. V seems to outperform V^a_σ up to $FP = 0.114$, at which on V^a_σ outperforms V, at a confidence level of $p\text{-}val < 0.01$. However, these results change for the blog control corpus, where V significantly outperforms both V_σ and V^a_σ. One way to explain the differences is by the characteristics of the corpora: the EBG is a "cleaner" and more stylistically consistent corpus, consisting of all English formal writing samples (essays written originally for business or academic purposes), whereas the blog dataset contains less structured and formal language, which may reduce the distinguishable effects of style variance normalization. This is supported also by the increased performance for EBG compared to the blog corpus (larger AUC). Clearly the results suggest that there is no one method preferable over the other, and selecting a verifier for a problem should rely on empirical testing over a stylistically similar training data.

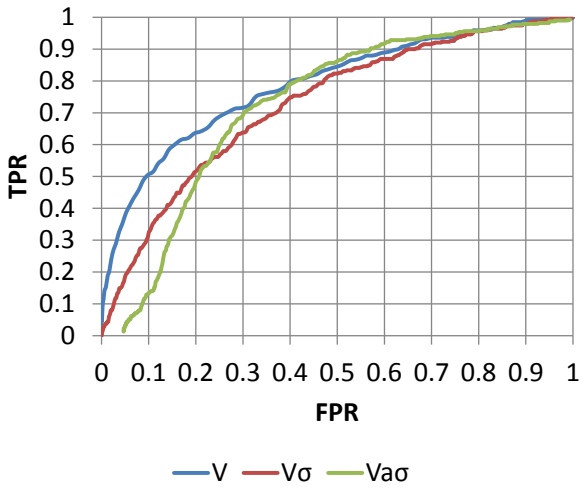

Figure 3: ROC curves for V, V_σ and V^a_σ evaluation on the blog corpus.

As for the effect of adding the per-author threshold adjustments, for both corpora V_σ outperforms V^a_σ on low FPR until they intersect (at $FP = 0.27$ and $FP = 0.22$ for the EBG and blog corpora, respectively), at which point V^a_σ begins to outperform V_σ. These properties allow various verification approaches to be used per need, dependent on false positive and false negative error rates constraints the problem in hand may impose.

7 Classify-Verify

The main novelty and contribution of this paper is introducing the *Classify-Verify* method. This method

is an abstaining classifier (Chow, 1970), i.e. a classifier that refrains from classifications in certain cases to reduce misclassifications. *Classify-Verify* combines classification with verification, to expand closed-world authorship problems to open-world, by essentially adding another class: "unknown". Another aspect of the novelty of our approach is the utilization of abstaining classification methods to upgrade from closed-world to open-world, where we evaluate how methods for thwarting wrongly classified instances apply on misclassifications that originate in being outside the assumed suspect set, rather than simply missing the true suspect.

First, closed-world classification is applied on the document in question, D, and the author suspect set $\mathcal{A} = \{A_1, ..., A_n\}$ (with their sample documents). Then, the output of the classifier $A_i \in \mathcal{A}$, is given to the verifier to determine the final output. Feeding only the classifier result into the verifier utilizes the high accuracy attainable by classifiers, which outperform verifiers in closed-world settings, thus focusing the verifier only on the top choice author in $\mathcal{A}$ (or least worse choice of all). The verifier determines whether to accept A_i or reject by returning $\perp$, based on a verification threshold t. *Classify-Verify* is essentially a classifier over the suspect set $\mathcal{A} \cup \{\perp\}$.

The threshold t selection process can be automated with respect to varying expected portions of *in-set* and *not-in-set* documents. We denote the likelihood of D's author being in $\mathcal{A}$, the expected *in-set* documents fraction, as $p = Pr[A_D \in \mathcal{A}]$ (making the likelihood of the expected *not-in-set* documents $1 - p$). In addition, we use the notation p-$\langle measure \rangle$, which refers to that measure's weighted average with respect to p. For instance, p-$F1$ is the weighted F1-score, weighted over F1-scores of p expected *in-set* documents and $1 - p$ expected *not-in-set* documents. t can thus be determined in several ways:

1) Manual: The threshold t can be manually set by the user. The threshold determines the sensitivity of the verifier, so setting t manually allows adjusting it from strict to relaxed, where the stricter it is, the less likely it is to accept the classifier's output. This allows tuning the algorithm to different settings, imposing the desired rigidity (expressed in limiting false positive and false negative error rates).

2) p-Induced Threshold: The threshold can be set empirically over the training set to the one that maximizes the target measurement, e.g. F1-score, in an automated process. In case p is given, the algorithm will apply cross-validation on the training data alone using the range of all relevant manually set thresholds (at some preset decimal precision advancements) and will choose the threshold that yields the best target measurement. This is essentially applying *Classify-Verify* recursively on the training data one level deeper with a range of manual thresholds. The relevant threshold search range is determined automatically by the minimum and maximum distances observed in the *verify* phase of *Classify-Verify*.

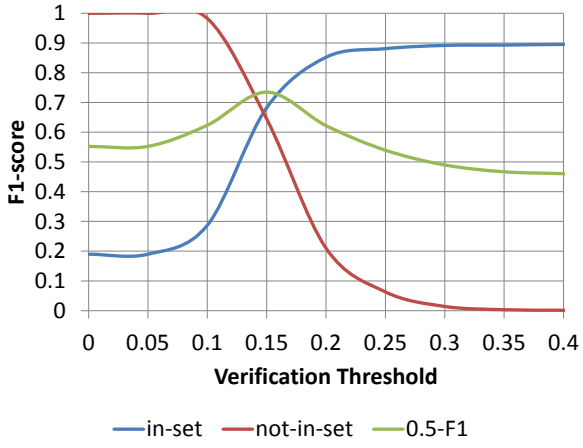

Figure 4: 0.5-$F1$ results for *Classify-Verify* on EBG using SMO SVM and V_σ with varying manually-set thresholds.

Fig. 4 illustrates the automated threshold tuning process on the EBG corpus for $p = 0.5$ (i.e. 50% expected *in-set* documents). In this example, *Classify-Verify* uses SMO SVM as classifier, V_σ as verifier and manually-set thresholds. The target measurement is F1-score, i.e. the automated process would have chosen the threshold that maximizes 0.5-$F1$ (here at $t = 0.15$). The base assumption is that the threshold that maximizes the target measurement on the training set is also the best choice for the testing phase.

An observation that should be noted is that max 0.5-$F1$ outperforms the F1-score at the intersection of the *in-set* and *not-in-set* curves. This is due to calculating 0.5-$F1$ in a "micro" rather than a "macro" fashion: the confusion matrices [4] of *in-set* and *not-in-set* are aggregated in a weighted sum and only then is the weighted 0.5-$F1$ calculated, across all authors (and $\perp$), rather than weighing the *in-set* and *not-in-set* F1-scores in a simple weighted average.

[4] A table in which each column represents the documents in a predicted class and each row represents the documents in an actual class

3) *in-set/not-in-set*-Robust: If the expected *in-set* and *not-in-set* documents proportion is unknown, we can apply the same idea of the previously described threshold. If we examine the *Classify-Verify* F1-score curve for some p along a range of thresholds, as p increases, it favors smaller (more accepting) thresholds, therefore the curve behaves differently for different values of p; however, all curves intersect at one t – at which the *in-set* and *not-in-set* curves intersect. This is illustrated in Fig. 5, which presents the same results as Fig. 4, only for varying values of p (0.1 to 0.9, with steps of 0.1).

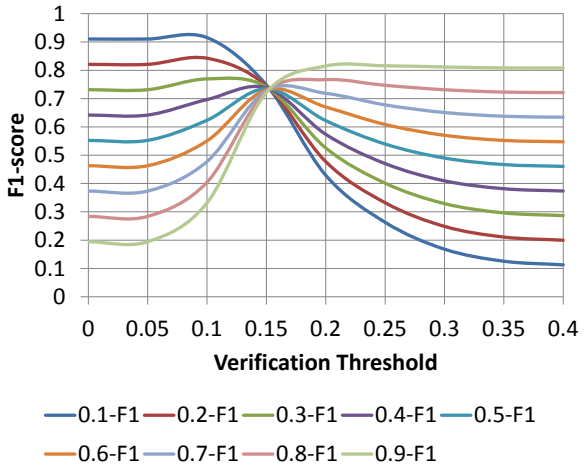

Figure 5: p-$F1$ results for *Classify-Verify* on EBG using SMO SVM and V_σ with varying manually-set thresholds and varying values of p.

This can be utilized to automatically obtain a threshold *robust* for any value of p by taking thresholds that minimize the difference between p-$F1$ and q-$F1$ curves, for any arbitrary $p, q \in (0, 1)$ (for simplicity we use 0.3 and 0.7; minimizing the difference is an approximation of the true intersection). As illustrated in Fig. 5, the robust threshold does not guarantee the highest measurement; it does, however, guarantee a relatively high expected value of that measure, independent of p, and thus robust for any open-world settings. We denote this measurement as p-$\langle measure \rangle_R$ (for *Robust*), e.g. p-$F1_R$.

Finally, the entire *Classify-Verify* algorithm is described in Alg. 1, and the flow of the algorithm is illustrated in Fig. 6.

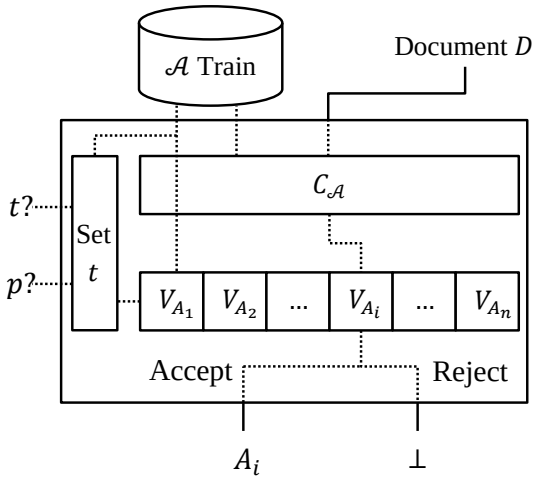

Figure 6: The flow of the *Classify-Verify* method on a test document D and a suspect set $\mathcal{A}$, with optional threshold t and *in-set* portion p.

8 Evaluation and Results

8.1 Evaluation Methodology

8.1.1 Main

In our main experiment we evaluate the *Classify-Verify* method on the EBG corpus, excluding the adversarial documents, and the blog corpus as control. We evaluate the datasets in two settings: when the authors of the documents under test are in the set of suspects (*in-set*), and when they are not (*not-in-set*).

Each classification over n authors $\mathcal{A} = \{A_1, ..., A_n\}$ can result with one of n+1 outputs: an author $A_i \in \mathcal{A}$ or $\perp$ (meaning: "unknown"). Therefore when the verifier accepts, the final result is the author A_i chosen by the classifier, and when it rejects, the final result is $\perp$.

In the evaluation process we credit the *Classify-Verify* algorithm when the verification step thwarts misclassifications in *in-set* settings. For instance, if D is written by A, classified as B but the verifier replaces B with $\perp$, we consider the result as true. This approach for abstaining classifiers (Herbei and Wegkamp, 2006) relies on the fact that we would rather be truly told "unknown" than get a wrong author.

We evaluate the overall performance with 10-folds cross validation. For each fold experiment, with 9 of the folds as training set and 1 fold as test set, we evaluate every test document twice: once as *in-set* and once as *not-in-set*.

Algorithm 1 *Classify-Verify*

Input: Document D, suspect author set $\mathcal{A} = \{A_1, ..., A_n\}$, target measurement μ
Optional: in-set portion p, manual threshold t
Output: A_D if $A_D \in \mathcal{A}$, and $\perp$ otherwise
$C_{\mathcal{A}} \leftarrow$ classifier trained on $\mathcal{A}$
$\mathcal{V}_{\mathcal{A}} = \{V_{A_1}, ..., V_{A_n}\} \leftarrow$ verifiers trained on $\mathcal{A}$
if t, p not set **then**
  $t \leftarrow$ threshold maximizing p-μ_R of *Classify-Verify* cross-validation on $\mathcal{A}$
else if t not set **then**
  $t \leftarrow$ threshold maximizing p-μ of *Classify-Verify* cross-validation on $\mathcal{A}$
end if
$A \leftarrow C_{\mathcal{A}}(D)$
if $V_A(D, t) = True$ **then**
  return A
else
  return $\perp$
end if

For the *classification* phase of *Classify-Verify* over n authors, we train n $(n-1)$-class classifiers, where each classifier C_i is trained on all authors except A_i. A test document by some author A_i is then classified as *in-set* using one of the $n-1$ classifiers that were trained on A_i training data; for simplicity we choose C_{i+1} (and C_1 for A_n). For the *not-in-set* classification, we simply use the classifier *not* trained on A_i, i.e. C_i.

For the *verification* phase of *Classify-Verify*, we evaluate several methods: one standalone method (V_σ for the EBG corpus and V for the blog corpus), *Gap-Conf*, P_1 and P_1-P_2-*Diff*. We use V_σ for EBG and V for the blog corpus, as these methods outperform the other standalone methods evaluated per corpus, as discussed in Sec. 6.

The more the verifiers reject, the higher the precision is (as bad classifications are thrown away), however recall decreases (as good classifications are thrown away as well), and vice-versa – higher acceptance increases recall but decreases precision. Therefore we measure overall performance using F1-score, since it provides a balanced measurement of precision and recall:

$$F1\text{-}score = 2 \times \frac{precision \times recall}{precision + recall}$$

$$precision = \frac{tp}{tp + fp}, recall = \frac{tp}{tp + fn}$$

We use the two automatic verification threshold selection methods discussed in Sec. 7: for the scenario in which the proportion of *in-set* and *not-in-set* is known with *in-set* proportion $p = 0.5$ (thus the *not-in-set* proportion is $1 - p$), we use the p-induces threshold that maximizes F1-score on the training set; for the scenario in which p is unknown, we use the robust threshold configured as described in Sec. 7. In order to calculate the F1-score of evaluating the test set, we combine the confusion matrices produced by the *in-set* and *not-in-set* evaluations in a p-weighted average matrix, from which weighted F1-score is calculated. We denote p-induced F1-scores as p-$F1$, and robust threshold induced F1-scores evaluated at some p as p-$F1_R$.

The threshold optimization phase of the *Classify-Verify* method discussed in Sec. 7 is done using 9-fold cross validation with the same experimental settings as the main experiment. Since F1-score is used to evaluate the overall performance, it is also used as the target measurement to maximize in the automatic threshold optimization phase. When p is known, the threshold that maximizes p-$F1$ is selected, and when it is unknown, the robust threshold is selected as the one for which the F1-score of different p's intersect (arbitrarily set to 0.3-$F1$ and 0.7-$F1$).

As baseline we compare F1-scores with 10-fold cross validation results of *closed-world* classification using the underlying classifier, SMO SVM with the $\langle 500, 2 \rangle$-*chars* feature set. We denote p-*Base* as the baseline F1-score of the closed-world classifier where the *in-set* proportion is p. It follows that 1-*Base* is the performance in pure closed-world settings (i.e. only *in-set* documents), and for any $p \in [0, 1]$, p-$Base = p \cdot 1$-*Base* (since for the *not-in-set* documents, the classifier is always wrong).

8.1.2 Adversarial Settings

To evaluate the *Classify-Verify* method in adversarial settings, we train our models on the non-adversarial documents in the EBG corpus, and test them on the imitation and obfuscation attack documents to measure how well *Classify-Verify* thwarts attacks (by returning $\perp$ instead of a wrong author). In this context, $\perp$ can be considered as either "unknown" or "possible-attack". We measure 0.5-$F1$, i.e. how well *Classify-Verify* performs on attack documents in an open-world scenario, where the verification threshold is set independent of a possible attack, tuned only to maximize performance on expected *in-set* and *not-in-set* document portions of 50% each.

As a baseline we compare results with standard classification using SMO SVM with the $\langle 500, 2 \rangle$-*chars*

feature set.

8.2 Results

Tab. 2 summarizes the different F1-score measurements terminology used throughout this section.

Term	Meaning
p	Portion of *in-set* documents
p-$F1$	F1-score of *Classify-Verify* for p portion of *in-set* documents
p-$F1_R$	F1-score of *Classify-Verify* for p portion of *in-set* documents, using robust thresholds
p-*Base*	F1-score of closed-world classifier for p portion of *in-set* documents 1-*Base* means pure closed-world settings For each $p \in [0,1]$, p-*Base* $= p \cdot$ 1-*Base*

Table 2: F1-score references terminology.

8.2.1 Main

For the EBG corpus, the baseline closed-world classifier attains 1-*Base* of 0.928 in perfect *in-set* settings, which follows that 0.5-*Base* = 0.464. As for the blog corpus, 1-*Base* = 0.64 which follows that 0.5-*Base* = 0.32. Fig. 7 illustrates 0.5-$F1$ results of the *Classify-Verify* method on the EBG corpus, where the authors are equally likely to be *in-set* or *not-in-set* ($p = 0.5$) and verification thresholds are automatically selected to maximize 0.5-$F1$. A similar illustration of 0.5-$F1$ results for the blog corpus is shown in Fig. 8.

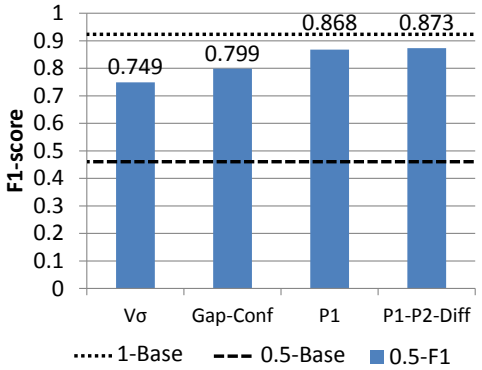

Figure 7: 0.5-$F1$ results of the *Classify-Verify* method on the EBG corpus, where the expected portion of *in-set* and *not-in-set* documents is equal (50%). *Classify-Verify* attains 0.5-$F1$ that outperforms 0.5-*Base* and even close to 1-*Base*.

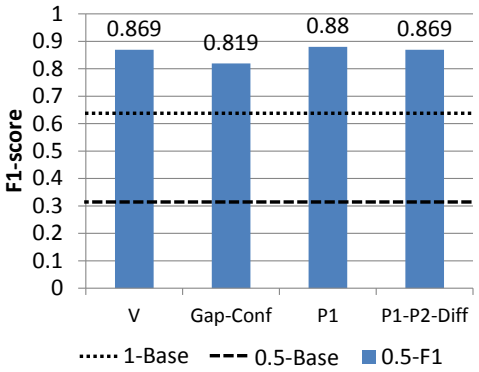

Figure 8: 0.5-$F1$ results of the *Classify-Verify* method on the blog corpus, where the expected portion of *in-set* and *not-in-set* documents is equal (50%). *Classify-Verify* attains 0.5-$F1$ that outperforms both 0.5-*Base and* 1-*Base*.

For both EBG and the blog corpora, *Classify-Verify* 0.5-$F1$ results significantly outperform 0.5-*Base* (the dashed lines), using any of the underlying verification methods, at a confidence level of p-$val < 0.01$.

Furthermore, the results are not only better than the obviously bad 0.5-*Base*, but produce similar results to 1-*Base*, giving overall 0.5-$F1$ in open-world settings up to ≈ 0.87. For the EBG corpus, moving to open-world settings only slightly decreases F1-score compared to the closed-world classifier performance in closed-world settings (the dotted line), which is a reasonable penalty for upgrading to open-world settings. However, on the blog corpus, where the initial 1-*Base* is low (at 0.64), *Classify-Verify* manages to both upgrade to open-world settings *and* outperform 1-*Base*. These results suggest that out of the *in-set* documents, many misclassifications were thwarted by the underlying verifiers, leading to an overall increase in F1-score.

Next, we evaluate the robust threshold selection scheme. In this scenario, the portion of *in-set* documents p is unknown in advance. Fig. 9 illustrates p-$F1_R$ results for the EBG corpus, where different p scenarios are "thrown" at the *Classify-Verify* classifier that uses robust verification thresholds. Similar illustration for the blog corpus are illustrated in Fig. 10.

In the robust thresholds scenario using the EBG corpus, *Classify-Verify* still significantly outperforms the respective closed-world classifier (p-*Base* results) for $p < 0.7$ with any of the underlying verifiers, at a

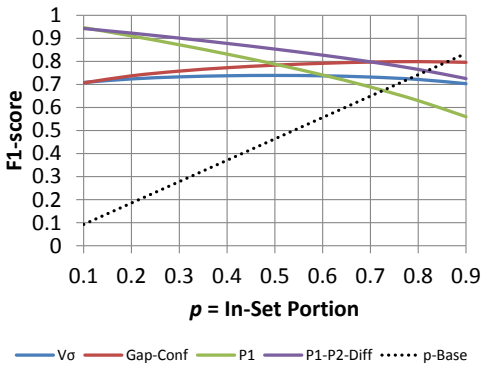

Figure 9: p-$F1_R$ results of the *Classify-Verify* method on the EBG corpus, where the expected portion of *in-set* documents p varies from 10% to 90% and is assumed to be unknown. Robust p-independent thresholds are used for the underlying verifiers. *Classify-Verify* attains p-$F1_R$ that outperforms the respective p-*Base*.

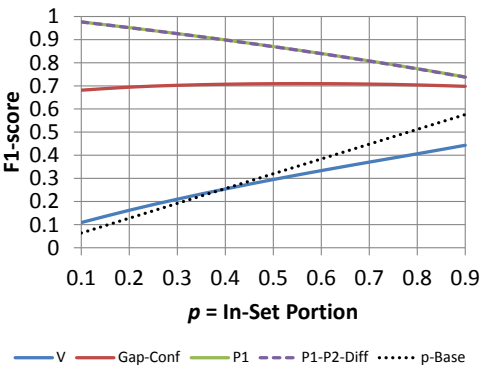

Figure 10: p-$F1_R$ results of the *Classify-Verify* method on the blog corpus, where the expected portion of *in-set* documents p varies from 10% to 90% and is assumed to be unknown. Robust p-independent thresholds are used for the underlying verifiers. *Classify-Verify* attains p-$F1_R$ that outperforms the respective p-*Base*.

confidence level of p-$val < 0.01$. For the blog corpus, *Classify-Verify* significantly outperforms p-*Base* using any of the classifier-induced verifiers for all p, at a confidence level of p-$val < 0.01$.

Moreover, the robust threshold selection hypothesis holds true, and for both corpora all methods (with the exception of V on the blog corpus) manage to guarantee a high F1-score, at ≈ 0.7 and above, for almost all values of p. For EBG, at $p \geq 0.7$ the *in-set* portion is large enough so that the overall p-*Base* becomes similar to p-$F1_R$. For the blog corpus, using V fails and performs similar to 0.5-*Base*.

Of all verification methods, P_1-P_2-*Diff* is proven the most preferable verifier to use, since it consistently outperforms the other methods across almost all values of p for *both* corpora, which implies it is robust to domain variation.

8.2.2 Adversarial Settings

Evaluated on the EBG corpus under imitation and obfuscation attacks, the baseline closed-world classifier attains F1-scores of 0 and 0.044 for the imitation and obfuscation attack documents, respectively. These results mean that the closed-world classifier is highly vulnerable to these types of attacks. Fig. 11 illustrates F1-scores for *Classify-Verify* on the attack documents. Note that all attack documents are written by *in-set* authors, and thus addressed as *in-set* documents.

The results suggest that *Classify-Verify* successfully manages to thwart the majority of the attacks, with up to 0.874 and 0.826 F1-scores for the obfuscation and imitation attacks, respectively. These results are very close to the deception detection results reported in (Afroz et al., 2012), with F1-scores of 0.85 for obfuscation and 0.895 for imitation attacks. A major difference is that here these results are obtained in open-world settings, with threshold configuration that does *not* take inside-attacks under consideration. Moreover, as opposed to the methods applied in (Afroz et al., 2012), no attack documents were used as training data.

Interestingly, the results above are obtained for a standard $p = 0.5$ open-world scenario, without possible attacks in mind, yet the overall results are harmed little to not at all, depending on the underlying verifier. For instance, when using *Gap-Conf*, 0.5-$F1$ is at 0.799 in non-attack scenarios, and F1-scores are 0.784-0.826 when under attack.

9 Discussion

In this section we discuss interesting observations from the results.

In Sec. 6 we examined two families of verifiers, classifier-induced and standalone, later used by *Classify-Verify*. The results suggest that classifier-induced verifiers consistently outperform the standalone ones; however this trend may be limited to large datasets with many suspect authors in the underlying classifier, like those evaluated in this paper,

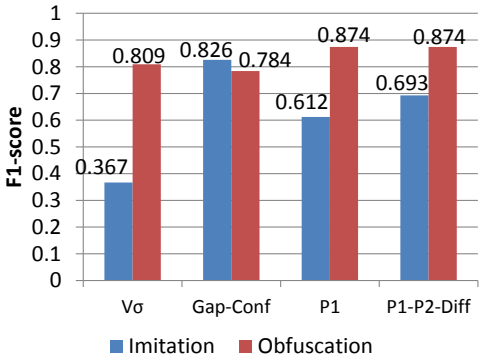

Figure 11: F1-score results of the *Classify-Verify* method on the EBG corpus under imitation and obfuscation attacks, where the expected portion of *in-set* and *not-in-set* documents is equal (50%). *Classify-Verify* successfully thwarts most of the attacks.

on which classifier-induced verifications rely. It may be the case that on small author sets, standalone verifiers will perform better, and this direction should be considered in future work. Moreover, the standalone verifiers presented here provide a reasonable accuracy, which can be used in pure one-class settings, where no training data exists except that of the true author (a scenario in which classifier-induced methods are useless).

The *Classify-Verify* 0.5-$F1$ results on both the EBG and the blog corpora illustrated in Fig. 7 and Fig. 8 suggest that using P_1 or P_1-P_2-*Diff* as the underlying verification method provide domain-independent results, at 0.5-$F1$ of ≈ 0.87. The superiority of P_1-P_2-*Diff* is emphasized by the p-$F1_R$ results illustrated in Fig. 9 and Fig. 10, where p-$F1_R$ over 0.7 is obtained for both corpora, independent of p. Therefore P_1-P_2-*Diff* is proven as a robust, domain and *in-set*/*not-in-set* proportion independent verification method to be used with *Classify-Verify*.

Finally, *Classify-Verify* is shown effective in adversarial settings, where it outperformed the traditional closed-world classifier, without the requirement of training on adversarial data, like required in (Afroz et al., 2012). Furthermore, no special threshold tuning is needed to achieve this protection, i.e. we can use the standard threshold selection schemes for non-adversarial settings and still thwart most attacks. It follows that results in adversarial settings can potentially be improved, if p is tuned not to the likelihood of *in-set* documents, but to the likelihood of an attack.

10 Conclusion

From a forensics perspective, the possibility of authors outside the suspect set makes the use of closed-world classifiers unreliable. In addition, whether linguistic authorship attribution can handle open-world scenarios has important privacy implications for both the authors of anonymous texts and those likely to be falsely implicated by faults in such systems. This research shows that when the closed-world assumption is violated, traditional stylometric approaches fail ungracefully.

The *Classify-Verify* method proposed in this work is not only able to handle open-world settings where the author of a document may not be in the training set, but can also improve results in closed-world settings, by abstaining from low-confidence classification decisions. Furthermore, this method is able to filter out attacks, as demonstrated on the adversarial samples in the EBG corpus. In all these settings, the method is able to replace wrong assertions with more honest and useful statements of "unknown."

We conclude that the *Classify-Verify* method is preferable over the standard underlying closed-world classifier. This statement is true regardless of the expected *in-set*/*not-in-set* ratio of the data, and in adversarial settings as well. Since the *Classify-Verify* algorithm is general, it can be applied with any set of stylometric classifiers and verifiers. In the forensics context, when satisfactorily-rigorous techniques are used, the *Classify-Verify* method is an essential tool for forensic analysts facing the often case of open-world problems.

We propose several directions for future work:

1) Other classification-related forensics and privacy applications. An important characteristic of the *Classify-Verify* algorithm is its generality, which makes it applicable to other problems. For instance, it can be used for behavioral biometrics, like in authentication systems that may have a set of potential users (e.g. employees in an office) but should consider outside attacks; the Active Linguistic Authentication dataset (Juola et al., 2013) is a perfect candidate dataset for such application.

2) Fusion of verification methods. Having various verification methodologies can be used to combine decisions, for instance using the Chair-Varshney optimal fusion rule (Chair and Varshney, 1986), into a centralized verification method that outperforms its single components. This approach is motivated

by (Ali and Pazzani, 1995), which shows that greater reduction in error rates is achieved when the verifiers are distinctly different – here to be applied by using both standalone and classifier-induced verifiers.

3) Utilization of *Classify-Verify* for scalability. In cases where the analyst is faced with a closed-world yet large problem, *Classify-Verify* may be used to apply a divide-and-conquer approach, to increase the potential accuracy of the classifier in-hand. Instead of building a model trained over all suspect authors, many small *Classify-Verify* models can be trained. Each test document will be classified by all models, and all suspects for which their model returns $\perp$ are immediately omitted. All result authors returned by the other models will compete in a final round, by a *Classify-Verify* model (or simple closed-world one) trained on all of them. Since each phase of this problem formulation is much smaller than one big model, this approach is initially prone to higher success, and can potentially increase the accuracy attainable by forensic analysts in large problem domains.

References

[Abbasi and Chen, 2008] Abbasi, A. and Chen, H. (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Trans. Inf. Syst.*, 26(2):1–29.

[Afroz et al., 2012] Afroz, S., Brennan, M., and Greenstadt, R. (2012). Detecting hoaxes, frauds, and deception in writing style online. In *Proceedings of the 33rd conference on IEEE Symposium on Security and Privacy*. IEEE.

[Ali and Pazzani, 1995] Ali, K. M. and Pazzani, M. J. (1995). *On the link between error correlation and error reduction in decision tree ensembles*. Citeseer.

[Araujo et al., 2005] Araujo, L. C., Sucupira Jr, L. H., Lizarraga, M. G., Ling, L. L., and Yabu-Uti, J. B. (2005). User authentication through typing biometrics features. *Signal Processing, IEEE Transactions on*, 53(2):851–855.

[Brennan et al., 2012] Brennan, M., Afroz, S., and Greenstadt, R. (2012). Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Trans. Inf. Syst. Secur.*, 15(3):12:1–12:22.

[Brennan and Greenstadt, 2009] Brennan, M. and Greenstadt, R. (2009). Practical attacks against authorship recognition techniques. In *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference*.

[Burton et al., 2009] Burton, K., Java, A., and Soboroff, I. (2009). The icwsm 2009 spinn3r dataset. In *Proceedings of the Third Annual Conference on Weblogs and Social Media (ICWSM 2009), San Jose, CA*.

[Chair and Varshney, 1986] Chair, Z. and Varshney, P. (1986). Optimal data fusion in multiple sensor detection systems. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-22(1):98–101.

[Chaski, 2013] Chaski, C. E. (2013). Best practices and admissibility of forensic author identification. *JL & Pol'y*, 21:333–725.

[Chow, 1970] Chow, C. (1970). On optimum recognition error and reject tradeoff. *Information Theory, IEEE Transactions on*, 16(1):41–46.

[Clark, 2011] Clark, A. (2011). Forensic stylometric authorship analysis under the daubert standard. *Available at SSRN 2039824*.

[Clough, 2000] Clough, P. (2000). Plagiarism in natural and programming languages: an overview of current tools and technologies.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

[Herbei and Wegkamp, 2006] Herbei, R. and Wegkamp, M. H. (2006). Classification with reject option. *Canadian Journal of Statistics*, 34(4):709–721.

[Juola, 2004] Juola, P. (2004). Ad-hoc authorship attribution competition. In *Proc. 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities (ALLC/ACH 2004)*, Göteborg, Sweden.

[Juola, 2008] Juola, P. (2008). Authorship attribution. *Foundations and Trends in information Retrieval*, 1(3):233–334.

[Juola, 2013] Juola, P. (2013). Stylometry and immigration: A case study. *JL & Pol'y*, 21:287–725.

[Juola et al., 2013] Juola, P., Noecker, Jr., J., Stolerman, A., Ryan, M. V., Brennan, P., and Greenstadt, R. (2013). A dataset for active linguistic authentication. In *Proceedings of the Ninth Annual*

IFIP WG 11.9 International Conference on Digital Forensics, Orlando, Florida, USA. National Center for Forensic Science.

[Killourhy, 2012] Killourhy, K. S. (2012). A scientific understanding of keystroke dynamics. Technical report, DTIC Document.

[Koppel et al., 2011a] Koppel, M., Akiva, N., Dershowitz, I., and Dershowitz, N. (2011a). Unsupervised decomposition of a document into authorial components. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1356–1364. Association for Computational Linguistics.

[Koppel and Schler, 2004] Koppel, M. and Schler, J. (2004). Authorship verification as a one-class classification problem. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 62–, New York, NY, USA. ACM.

[Koppel et al., 2011b] Koppel, M., Schler, J., and Argamon, S. (2011b). Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94.

[Koppel et al., 2007] Koppel, M., Schler, J., and Bonchek-Dokow, E. (2007). Measuring differentiability: Unmasking pseudonymous authors. *J. Mach. Learn. Res.*, 8:1261–1276.

[Lee and Cho, 2007] Lee, H.-j. and Cho, S. (2007). Retraining a keystroke dynamics-based authenticator with impostor patterns. *Computers & Security*, 26(4):300–310.

[Manevitz and Yousef, 2007] Manevitz, L. and Yousef, M. (2007). One-class document classification via neural networks. *Neurocomputing*, 70(7):1466–1481.

[McDonald et al., 2012] McDonald, A., Afroz, S., Caliskan, A., Stolerman, A., and Greenstadt, R. (2012). Use fewer instances of the letter "i": Toward writing style anonymization. In *Privacy Enhancing Technologies Symposium (PETS)*.

[Meyer zu Eissen et al., 2007] Meyer zu Eissen, S., Stein, B., and Kulig, M. (2007). Plagiarism detection without reference collections. In Decker, R. and Lenz, H.-J., editors, *Advances in Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 359–366. Springer Berlin Heidelberg.

[Narayanan et al., 2012] Narayanan, A., Paskov, H., Gong, N., Bethencourt, J., Stefanov, E., Shin, R., and Song, D. (2012). On the feasibility of internet-scale author identification. In *Proceedings of the 33rd conference on IEEE Symposium on Security and Privacy*. IEEE.

[Noecker and Juola, 2009] Noecker, Jr., J. and Juola, P. (2009). Cosine distance nearest-neighbor classification for authorship attribution. In *Digital Humanities 2009*, College Park, MD.

[Noecker and Ryan, 2012] Noecker, Jr., J. and Ryan, M. (2012). Distractorless authorship verification. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

[Paskov, 2010] Paskov, H. S. (2010). *A regularization framework for active learning from imbalanced data*. PhD thesis, Massachusetts Institute of Technology.

[Pietraszek, 2005] Pietraszek, T. (2005). Optimizing abstaining classifiers using roc analysis. In *Proceedings of the 22nd international conference on Machine learning*, pages 665–672. ACM.

[Platt, 1998] Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In Schoelkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

[Potthast et al., 2011] Potthast, M. et al. (2011). Pan 2011 lab: Uncovering plagiarism, authorship, and social software misuse. Conference CFP at `http://pan.webis.de/`.

[Sanderson and Guenter, 2006] Sanderson, C. and Guenter, S. (2006). Short text authorship attribution via sequence kernels, markov chains and author unmasking: an investigation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 482–491, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Schölkopf et al., 2001] Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471.

[Sommer and Paxson, 2010] Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In

Security and Privacy (SP), 2010 IEEE Symposium on, pages 305–316. IEEE.

[Sorio et al., 2010] Sorio, E., Bartoli, A., Davanzo, G., and Medvet, E. (2010). Open world classification of printed invoices. In *Proceedings of the 10th ACM symposium on Document engineering*, pages 187–190. ACM.

[Tax, 2001] Tax, D. M. (2001). *One-class classification*. PhD thesis.

[van Halteren, 2004] van Halteren, H. (2004). Linguistic profiling for authorship recognition and verification. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 199–206, Barcelona, Spain.

[Wayman et al., 2009] Wayman, J., Orlans, N., Hu, Q., Goodman, F., Ulrich, A., and Valencia, V. (2009). Technology assessment for the state of the art biometrics excellence roadmap. MITRE Technical Report Vol. 2, FBI.