Continuous Authentication using Micro-Messages

Marcelo Luiz Brocardo, Issa Traore
Department of Electrical and Computer Engineering
University of Victoria - UVIC
Victoria, British Columbia, Canada
{marcelo.brocardo, itraore}@ece.uvic.ca

***Abstract*—Authorship verification consists of checking whether a target document was written or not by a specific individual. In this paper, we study the problem of authorship verification for Continuous Authentication (CA) purposes. Different from traditional authorship verification that focuses on long texts, we tackle the use of micro-messages. Shorter authentication delay (i.e. smaller data sample) is essential to reduce the window size of the re-authentication period in CA. We explored lexical, syntactic, and application specific features. We investigated two different classification schemes: on one hand Logistic Regression (LR) and on the other hand an hybrid classifier combining Support Vector Machine (SVM) and LR. Experimental evaluation based on the Enron email dataset involving 76 authors and Twitter dataset involving 100 authors yield very promising results consisting of Equal Error Rates (EER) of 9.18% and 11.83%, respectively.**

***Keywords*—Continuous authentication; biometrics systems; authorship verification; stylometry; short message verification.**

I. Introduction

Static authentication where user identity is checked once at login time can be circumvented no matter how strong the authentication mechanism is. Through attacks such as man-in-the-middle and its variants, an authenticated session can be hijacked later after the initial login process has been completed. In the last decade, continuous authentication (CA) using biometrics has emerged as a possible remedy against session hijacking. CA consists of testing the authenticity of the user repeatedly throughout the authenticated session as data becomes available.

CA is expected to be carried out unobtrusively, due to its repetitive nature, which means that the authentication information must be collectible without any active involvement of the user and without using any special purpose hardware devices (e.g. biometric readers).

Emerging behavioural biometric technologies such as mouse dynamics and keystroke dynamics biometrics are good candidates for CA because data can be collected passively using standard computing devices (e.g. mouse and keyboard) throughout a session without any knowledge of the user [1]. We believe that developing continuous authentication approach based on authorship analysis can achieve the same purpose and will contribute to improving the verification accuracy while maintaining acceptable authentication delays. The linguistic characteristics used to identify the author of a text is referred to as stylometry [2], [3]. Stylometry analysis consists of inferring the authorship of a document by extracting and analyzing the writing styles or stylometric features from a document content.

A number of studies on stylometry have been published [4]–[7]. These papers typically target three different problems, including, authorship attribution or identification, authorship verification, and authorship profiling or characterization. Authorship attribution consists of determining the most likely author of a target document among a list of known individuals. Authorship verification consists of checking whether a target document was written or not by a specific individual. Authorship profiling or characterization consists of determining the characteristics (e.g. gender, age, and race) of the author of an anonymous document.

While the existing work have shown promising results when extracting writing styles from long documents, their accuracy was considerably low when analyzing short documents. Short documents analysis is challenging because they are poorly structured and use informal language, as opposed to literary documents.

In this paper, we compare sample writing of an individual against the model or profile associated with the identity claimed by that individual at login time (i.e. 1-to-1 identity matching), which is very similar to authorship verification. The use of micro-messages (i.e. smaller data sample) allows shorter authentication delay, which is essential to reduce the window of vulnerability of the system. We identify a set of new features including lexical, syntactic, and application specific features. For classification, we investigate in this paper an hybrid classifier that combines two different traditional classifiers, namely, Support Vector Machine (SVM) and Logistic Regression (LR). We evaluate our model using the Enron emails dataset and a micro-messages dataset based on Twitter feeds with 76 and 100 authors respectively. To our knowledge, it is the first time that Twitter dataset is used for authorship verification. We evaluate experimentally our approach by computing the following performance metrics:

- False Acceptance Rate (FAR): measures the likelihood that the system will fail to recognize the genuine person;
- False Rejection Rate (FRR): measures the likelihood that the system may falsely recognize someone as the genuine person;
- Equal Error Rate (ERR): corresponds to the operating point where FAR and FRR have the same value.

Our evaluation yields an EER of 9.18% using Enron dataset as a test and 11.83% using the micro-message corpus from Twitter as a test.

The rest of the paper is structured as follows. Section II

summarizes and discusses related works. Section III introduces our proposed approach. Section IV presents our experimental evaluation by describing the underlying methodology and discussing the obtained results. Section V discusses the strengths and shortcomings of our approach and outlines the ground for future works. Section VI makes some concluding remarks.

II. Related Work

Among the few studies available on authorship verification, are works by Koppel et al. [6], Iqbal et al. [30], Canales et al. [5].

Koppel et al. proposed an authorship verification method named "unmasking" where an attempt is made to quantify the dissimilarity between the sample document produced by the suspect and that of other users (i.e. imposters) [6]. The experimental evaluation, however, shows that the proposed approach can provide trustable results only for documents of at least 500 words long, which is not realistic in the case of online verification.

Iqbal et al. studied email authorship verification by extracting 292 different features and analyzing these features using different classification and regression algorithms [30]. Experimental evaluation of the proposed approach using the Enron e-mail corpus yielded EER ranging from 17.1% to 22.4%.

Canales et al. extracted keystroke dynamics and stylistic features from sample exam documents for the purpose of authenticating online test takers [5]. The extracted features consisting of timing features for keystroke and 82 stylistic features were analyzed using a K-Nearest neighbor (KNN) classifier. Experimental evaluation involving 40 students with sample document size between 1710 to 70,300 characters yielded (FRR=20.25%, FAR=4.18%) and (FRR= 93.46%, FRR=4.84%) when using separately keystroke and stylometry, respectively. The combination of both types of features yielded EER= 30%.

III. Proposed Approach

In this section, we briefly summarize our previous work and present our approach by discussing feature selection and describing our classification model.

A. Previous Work

We investigated in previous work the possibility of using stylometry for authorship verification for short online messages [28]. The technique was based on a combination of supervised learning and *n-gram* analysis. The experimental evaluation was conducted using the Enron email dataset and yielded an EER of 14.35% for 87 users for message blocks of 500 characters. In follow-up experiments we extended the feature set and used SVM for classification. Experimental evaluation of the approach yielded an EER of 12.42% with 76 authors [29].

The differences between the current paper and our previous work concern the classsification method, the feature set, and the evaluation dataset. While we used previously only SVM for classification, we investigate in this work LR and an hybrid of LR and SVM. Furthermore, we identify and define several new features and apply not only Information Gain (IG) to decide which features to use, but also use Mutual Information (MI) metric to discard highly correlated features. Finally, we evaluate the approach proposed in this paper using shorter messages (i.e. micro messages), consisting of blocks of text of 140, 280 and 500 characters. Furthermore, in addition to the Enron dataset, we use a new dataset in the form of twitter feeds.

B. Approach Overview

Figure 1 shows our proposed framework. The blocks "sources" and "preprocessing" are discussed in detail in Section IV. In a general overview of our approach, we decompose an online document into consecutive blocks of short texts over which (continuous) authentication decisions happen. We extract for each block a set of basic features by computing the frequency and average of lexical, syntactic and application specific characteristics of documents. In addition, we compute new features based on *n-gram* analysis, and use information gain and mutual information techniques for feature selection. In order to balance the dataset, we define a weight for the instances based on the proportion of positive and negative training samples.

Our classification model consists of a collection of profiles generated separately for individual users. Our proposed system operates in two modes: enrollment and verification. Based on sample training data, the enrollment process computes the behavioral profile of the user using machine learning clasification. We investigate in this work an hybrid classifiers combining SVM and LR.

C. Basic Features

Feature extraction consists of quantifying the writing style of a document in order to create a profile that represents the style of its author.

In this study we selected an initial set of basic features by combining lexical characters, lexical words, syntactic and application specific characteristics. For each block of text, we extract all features represented as a vector of features values. We normalize the feature values to range between 0 and 1 using *maximum normalization* scheme, in which case a given feature value will be replaced by its ratio over the maximum value for the same feature over the training set.

Lexical features are related to the words or vocabulary of a language and operate at the character or word level [31]. Character level features are obtained by measuring the frequency of characters, which include letters (uppercase and lowercase), white-space, vowels, and special characters (e.g. '$', '%', '(', ')', '{', '}', etc.). Also, in online messages it is common to express a writer's mood in form of icons, which can be used as a stylistic marker. An icon can be written in a text form, for instance, ":-)" , ":o)", or in unicode characters, for instance, ☺ - ☹. We categorized 126 text based emotion icons in 38 different categories (e.g. smiley, laughing, very happy, frown, angry, crying, etc.) with an average of 3.31 icons to describe each emotion. In a unicode character, the range of emoticons vary from code 1F600 to 1F64F with 80 different possible icons. In addition, some authors use miscellaneous symbols in their message, for instance, ☋ or ♫. The range of these symbols in unicode characters is from

TABLE I. COMPARATIVE PERFORMANCES, BLOCK SIZES AND, POPULATION SIZES FOR STYLOMETRY STUDIES.

Category	Ref	Sample Population Size	Block Size	Number of Features	Technique	Accuracy* (%)	EER (%)
Attribution	[4]	100 **	277 words	L(25065), Sy(2766), A(128)	Karhunen-Loeve (K-L) - (Principal Component Analysis)	83.10	- -
	[8]	10	200 words	L(1), Sy(10)	Discriminant Function Analysis (DFA)	95.70	- -
	[9]	2 - 4	60,000 words	Se (many)	Synonym-based features through statistical classification	93.8 - 97.8	- -
	[10]	3	20 sentences	L(28820), Sy(4117), Se(1896)	SVM	87.63	- -
	[11]	3 **	200 words	400 features including lexical, syntactic, and structural features	SVM and C4.5 (Decision Tree)	69 -83	- -
	[12]	87	287 words	L(not specified the quantity, but include top-k word frequencies, 4-grams characters), Sy(8)	Logic Fuzzy	50 - 60	- -
	[13]	3 - 10 **	200 words	L(82), Sy(311), A(26)	Expectation Maximization (EM), k-means, and bisecting k-means	80 - 90	- -
	[14]	4 - 20 **	300 words	L(105), Sy(159), Se(10), A(28)	Frequent pattern	69.75 - 88.37	- -
	[15]	6 - 10 **	200 words	Not specify the quantity, but include L, Sy, A	Frequent pattern	77 - 80.5	- -
	[2]	20	169 words	L(87), Sy(158), Se(11), A(14)	SVM	99.01	- -
	[16]	20	600 words	Sy(171)	Prediction by Partial Matching (PPM)	84.30	- -
	[7]	50	500 characters	L(several)	Hidden Markov Model (HMM)	- -	8.08 - 30.88
	[17]	10,000	500 words	L(n-gram)	k-NN (cosine similarity)	46	- -
	[18]	100,000	335 words	L(95) , Sy(1093)	k-NN, Naive Bayes (NB), and SVM	20	- -
Characterization	[19]	5	76 words	L(79), Sy (262), Se(15), A(62)	C4.5 decision tree and SVM***	90.1 - 97	- -
	[20]	108 **	50 - 200 words	L(130), Sy(402) , A(13)	SVM, Bayesian logistic regression, and AdaBoost decision tree	73 - 82.23	- -
	[21]	114 **	50 - 200 words	L(130), Sy(402), A(13)	Decision Tree, SVM	80.08 - 82.20	- -
	[22]	325	50 - 200 words	L(69), Sy(122), A(30)	SVM	70.20	- -
	[14]	4 - 20 **	300 words	L(105), Sy(159), A(15), Se(23)	Frequent Pattern	39.13% - 60.44%	- -
	[23]	100	300 words	L(89), Sy(119), A(3)	k-NN, NB, Patient rule induction method, SVM	39.0 - 99.70	- -
	[24]	10 - 40	450 words	L (several - unigram, bigram, and trigram)	Probabilistic Context-Free Grammar (PCFG)	68.3 - 91.5	- -
Verification	[5]	40	1710 - 70300 characters	L(62), Sy(20)	k-NN	- -	30
	[25]	25 - 40 **	30 - 50 words	L(40), Sy(76), Se(25), A(9)	SVM	83.90 - 88.31	
	[26]	8	628 - 1342 words	L(100K), Sy(900K)	Weighted Probability Distribution Voting (WPDV)	- -	3
	[6]	10	500 words	L(250)	SVM	95.70	- -
	[27]	29	2400 words	L(40)	Linear Discriminant Analysis (LDA)	- -	22
	[28]	87 **	250 - 500 character	L(n-gram)	Supervised	- -	18.90 - 14.35
	[29]	76 **	500 character	L(91), Sy(251), A(7)	SVM	- -	12.42

* The accuracy is measured by the percentage of correctly matched authors in the testing set.
** Used Enron dataset for testing.
(L) = Lexical, (Sy) = Syntactic, (Se) = Semantic, (A) = Application

2600 to 26FF with 256 different symbols. We calculated the average of each symbol by group. Other lexical features are obtained by extracting *n*-grams from a text. *N*-grams are tokens formed by a contiguous sequence of *n* items. Character *n*-grams have been shown to be efficient [32]–[35] and the most frequent *n*-grams constitute the most important feature for stylistic purposes. Importantly, *n*-grams are relatively tolerant to typos [34]. However, instead of using the most frequent *n*-grams present in a text as a feature, we derive 16 new features corresponding to *5*-grams and *6*-grams. More details about our *n-gram* model are available in subsection III-D.

For word level features, we calculated the number of words per sample of text, average of short words (1-3 characters), average of long words (more than 6 characters), and the ratio of characters by words, among other features [5], [34]. We extract the fifty most frequently used words per author and create a unique list of words or dictionary [36]. The feature extraction consists of calculating the frequency of the dictionary words from a text. Other group of lexical features is the vocabulary richness. We measure the vocabulary richness by quantifying the number of *hapax legomenon* and *hapax dis legomenon*, referring to a word which occurs only once or twice in a text, respectively.

Syntactic Features are related to the grammatical structure or rules used to construct a sentence in a language. The part-of-speech tagging (POS) allows categorizing the tokens (words) according to their content or function in the context. Content word consists of the lexical meaning of a word and is classified as a noun, verb, adverb, or adjective. Functional words express a grammatical relationship (syntactic) between words (i.e. articles, a preposition, a conjunction) [11], [20], [37], [38]. Our functional words are topic-independent and include conjunction, interrogative, preposition, interjection, and pronouns. Following Iqbal et al. [30], we also included a set of punctuation marks. Basic punctuation includes single quotes, commas, periods, colons, semi-colons, question marks,

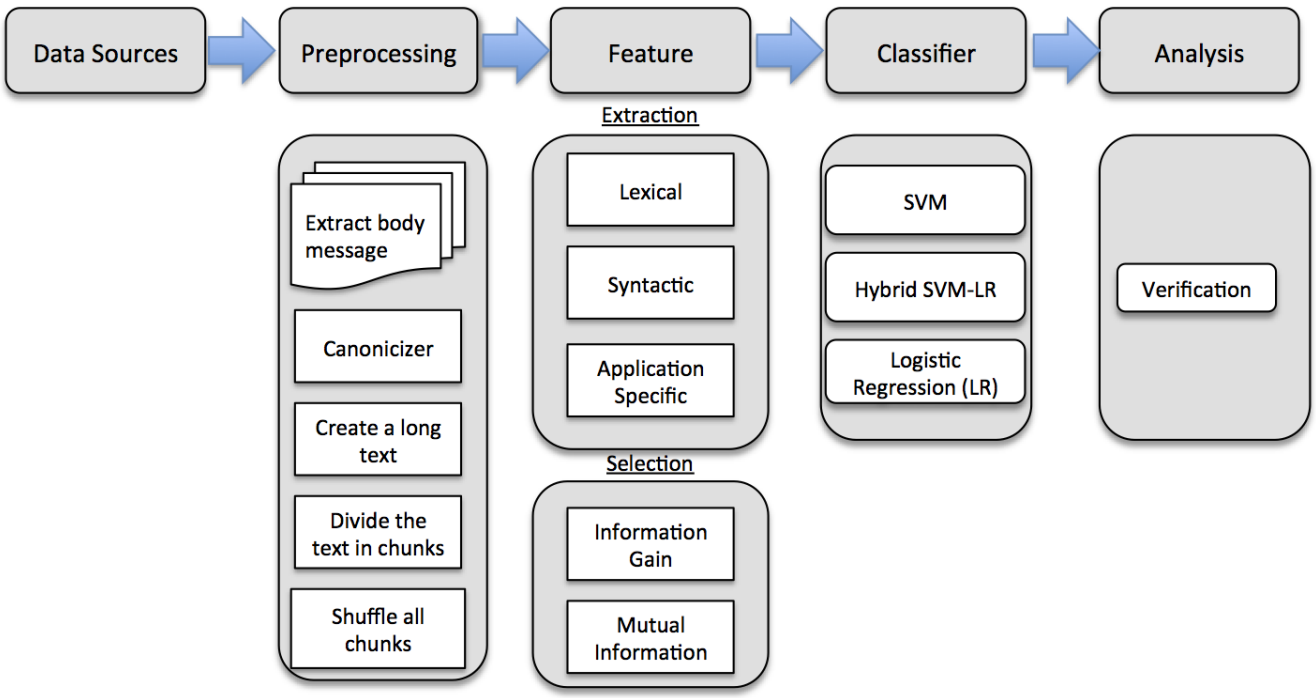

Fig. 1. Block diagram of our proposed framework. Sources and Preprocessing is discussed in Section IV.

and exclamation marks. However, authors are using uncommon punctuations such as as †, ⁝, and ∴. Therefore, we created a subgroup named general punctuation with 112 different symbols, based on the unicode format with code ranging from 2000 to 206F.

Application specific features consist of structural and content-specific characteristics [11], [25], [39], [40]. Structural characteristics are categorized at the message-level, paragraph-level or according to the technical structure of the document [4]. Since we are analyzing short messages from e-mails and twitter posts, we extracted only features related to the paragraph level. These features include the number of sentences and paragraphs per block of text, the average of characters, words, and sentences in a block of text, and the average of sentences beginning with upper and lower case.

D. N-gram Model

Given a user U, we divide her training data into two subsets, denoted $T(f)_1^U$ and T_2^U. Let $N(T(f)_1^U)$ denote the set of all unique *n-grams* occurring in $T(f)_1^U$ with frequency f.

Let m denote a binary variable (i.e., $m \in \{1, 0\}$) that represents the mode of calculation of the *n-grams*.

Given a block b, let $N_m(b)$ denote the following:

$$\begin{cases} \text{- the set of all unique } n\text{-grams occurring in } b, \text{ if } m = 0 \\ \text{- the set of all } n\text{-grams occurring in } b, \text{ otherwise.} \end{cases} \tag{1}$$

We define the similarity $r_U(b)$ between a sample data block b and the profile of user U, where the similarity varies between 0 and 1, as the percentage of unique *n-grams* shared by block b and training set $T(f)_1^U$, giving[1]:

$$r_U(b) = \frac{|N_m(b) \cap N(T(f)_1^U)|}{|N_m(b)|} \tag{2}$$

We also define a binary similarity metric, denoted $d_U(b)$ and referred to as *decision*, which captures the closeness of a block b to the profile of user U, as follows:

$$\begin{cases} d_U(b) = 1 \text{ if } |r_U(b)| \geq \epsilon_U \\ d_U(b) = 0, \text{otherwise} \end{cases} \tag{3}$$

where ϵ_U is a user-specific threshold derived from the training data.

We derive the value of ϵ_U for user U using a supervised learning technique outlined by $Algorithm$ 1. Given a user U, we divide the training subset T_2^U into p blocks of characters of equal size: $b(m)_1^U, ..., b(m)_p^U$. Our model approximates the actual (but unknown) distribution of the ratios $(r_U(b_1^U), ..., r_U(b_p^U))$ (extracted from T_2^U) by computing the sample mean denoted μ_U and the sample variance σ_U^2 during the training. In the algorithm, the threshold is initialized (i.e. $\epsilon_U = \mu_U - (\sigma_U/2)$), and then varied incrementally by minimizing the difference between FRR and FAR values for the user, the goal being to obtain an operating point that is as close as possible to the EER.

For each test block b, we derive 2 new features corresponding to $r_U(b)$ and $d_U(b)$. In this study, we consider only 5-grams and 6-grams, and cover two different values for the frequency f (i.e. $f = 1$ and $f = 2$) and for the mode of calculation of the n-grams (i.e. $m = 0$ and $m = 1$). Therefore,

[1] $|X|$ denote the cardinality of set X.

Features			Characteristics	Total
Lexical	Character	Frequency	Number of characters (C)	1
			Number of lower character/C	1
			Number of upper characters/C	1
			Number of white-space characters/C	1
			Total number of vowels (V)/C	1
			Vowels (a, e, i, o, u) / V	5
			Alphabets (A-Z) / C	26
			Number of special characters (S) /C	1
			Special Characters (%,&,etc.) / S	13
		n-grams	Character 5 and 6-grams (Ru and Decision)	16
		Icon	Text based icon (8 groups)	126
			Unicode - emoticons	80
			Unicode - miscellaneous symbols	256
				528
	Word	Total number of words (N)		1
		Average sentence length in terms of words /N		10
		Frequency	Words longer than 6 characters/N	1
			Total number of short words (1-3 characters)/N	1
			Average word length	1
			Average syllable per word	1
			Ratio of characters in words to N	1
			Replaced words / N	6
			The 50 most frequent words per author	50
		Vocabulary richness	Hapax legomena	1
			Hapax dislegomena	1
			Vocabulary richness (total different words/N)	1
				75
Syntactic		Total number of punctuation (P)		1
		Frequency	single quotes, commas, periods, colons, semi-colons, question marks, exclamation marks divided by P	8
			Unicode - General punctuation	112
		Functional words	Total number of conjunction, interrogative, preposition, interjection, and pronouns each one divide by N	5
			Ratio of functional word divide by the respective total word group	236
				362
Application-specific	Structural	Paragraph-Level	Total number of sentences	1
			Total number of paragraphs	1
			Average of characters, words and sentences in a block of text	3
			Average of sentences beginning with upper case	1
			Average of sentences beginning with lower case	1
				7

Fig. 2. List of stylometry features used in our work.

the number of new features created from the above *n-gram* model is 2 (for f) ×2 (for m) ×2 (for *n-gram* types) ×2 (for r_U and d_U) = 16.

E. Features Selection

In order to evaluate the worth of an attribute with the highest discrimination and to get pure classes, we apply in this work a ranking strategy based on the Information Gain (IG) for deciding which features to use and applied Mutual Information (MI) to discard highly correlated features. In fact, the processes to obtain IG and MI are very similar, the difference being the parameters of the equation.

Let $H(X)$ denote the information entropy, which is a measure of the uncertainty in a random variable X; $H(X)$ is defined:

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i) \tag{4}$$

where $p(x_i)$ denote the probability mass function of x_i.

```
/* U a user for whom the threshold is being calculated */
/* I_1,...,I_m: a set of other users (I_k ≠ U)          */
/* ε_U: threshold computed for user U                   */
Input: Training data for U, I_1, ..., I_m
Output: ε_U
begin
    up ← false;
    down ← false;
    δ ← 1;
    ε_U ← μ_U − (σ_U/2);
    while δ > 0.0001 do
        /* Calculating FAR and FRR for user U       */
        FRR_U, FAR_U = calculate(U, I_1, ..., I_m, ε_U, γ);
            /* Minimizing the difference between FAR and
               FRR                                  */
        if (FRR_U − FAR_U) > 0 then
            down ← true;
            ε_U ← ε_U − δ;
        end
        if (FAR_U − FRR_U) > 0 then
            up ← true;
            ε_U ← ε_U + δ;
        end
        if (up & down) then
            up ← false;
            down ← false;
            δ ← δ/10;
        end
    end
    return ε_U;
end
```

Algorithm 1: Threshold calculation for a given user.

Let $H(X|Y)$ denote the conditional entropy given a variable X after the observation of the variable Y. $H(X|Y)$ is defined as:

$$H(X|Y) = -\sum_{i=1}^{N}\sum_{j=1}^{M} p(x_i, y_j) log_2 p(x_i|y_j) \tag{5}$$

We calculate the IG of a feature individually (without considering other features) by analyzing only its classes and attribute values. Prior to computing IG and MI, it is necessary to discretize the numeric feature values into binary values (0 and 1). The discretization process consists of finding a cut-point or split-point that divides the range into two intervals, one interval being less or equal than the cut-point while the other is greater [41]. We use the entropy-based discretization method proposed by Fayyad and Irani [42], which is a supervised discretization method and has been known to achieve some of the best performances in the literature.

Suppose that the dataset is composed by positive and negative instances with a feature named $Attr_a$. The IG is calculated by analyzing only the values present in $Attr_a$, values from $Attr_b$ are not considered. The IG is the difference between $H(Class)$ and $H(Class|Attr)$, defined as follows:

$$IG(Class, Attr) = H(Class) - H(Class|Attr) \tag{6}$$

The MI is calculated by computing the entropy of $Attr_a$ and the conditional entropy $H(Attr_a|Attr_b)$, as follows:

$$MI(Attr_a, Attr_b) = H(Attr_a) - H(Attr_a|Attr_b) \tag{7}$$

F. Classification Method

We investigate an hybrid classifier that combines two popular classifiers namely LR and SVM.

LR classifier: it is a well-known and efficient probabilistic statistical classification model [43]. LR is applied for binary classification ($y \in \{0,1\}$) and Multinomial LR is applied for multi-class problems ($y \in \{0,1,2,...n\}$). The LR prediction is based on the logistic function that is a common sigmoid function with equation:

$$f(x) = \frac{1}{1+e^{-x}} \tag{8}$$

The logistic regression predicts whether a feature vector x belongs to a class y_i. The result of the logistic function is always between zero and one ($0 \leq f(x) \leq 1$).

SVM Classifier: it is a supervised learning method used for classification and regression analysis that performs a non-probabilistic binary linear classification [44]. SVM is based on the idea of mapping the original finite-dimensional space X into a much higher-dimensional space F. SVM constructs a hyperplane separating points of the two classes and finds a dividing straight line (optimal hyperplane). The decision boundary is the maximum-margin hyperplane or the largest minimum distance to the training examples. The hyperplane can be modeled using only the samples on the margin, called the support vectors, and training an SVM consists of identifying the support vectors within the training samples.

Assume that $s_i \in X$ are the support vectors, each associated with a class label $y_j \in \{+1,-1\}$ (for positive and negative examples, respectively). Given an unlabeled sample $x \in X$, classification consists of predicting the corresponding label y_j. This is performed using a decision function as follows.

$$f(x) = \sum_i y_i \alpha_i K(x, s_i) + b \tag{9}$$

Where α_i are Lagrangian multipliers, and K is a kernel function that measures the similarity or distance between the unlabeled sample x and the support vector s_i. The kernel function $K(x, s_i)$ maps the sample space X into a high-dimensional feature space F. Examples of kernel functions include linear, polynomial, Gaussian, and hyperbolic tangent [44].

Hybrid SVM-LR Classifier: Although SVM is a non-probabilistic classifier, probability estimates can be obtained by integrating SVM with logistic regression into a more robust classifier [45], [46]. The output of the SVM ($f(x)$) is submitted to a logistic function, defined as:

$$P(x) = \frac{1}{1+e^{-f(x)}} \tag{10}$$

IV. Experimental Evaluation

A. Dataset

In order to validate our system, we performed experiments on a real-life dataset from Enron e-mail corpus[2] and a micro-messages corpus from Twitter[3].

Enron was an energy company (located in Houston, Texas) that was bankrupt in 2001 due to white collar fraud. The e-mails of Enron's employees were made public by the Federal Energy Regulatory Commission during the fraud investigation. The e-mail dataset contains more than 200 thousands messages from about 150 users. The average number of words per e-mail is 200. The e-mails are plain texts and cover various topics ranging from business communications to technical reports and personal chats.

Twitter is a microblogging service that allows authors to post up to 140 characters. Twitter has over 200 million active users worldwide, posting 9,100 tweets per second. Registered users can read and post tweets, reply to a tweet, send private messages and re-tweet a message, while unregistered users can only read them. Also, a registered user can follow and be followed by other users. One of the Twitter datasets available for research is the Text Retrieval Conference (TREC) 2011 dataset, which has approximately 16 million tweets. However, the quantity of messages written by the same author is very small and insufficient to run our proposed stylometry experiments, which need at least 28,000 characters per author. Therefore, we decided to create our own dataset by crawling messages of authors from Twitter. Firstly, we need to choose an author with several messages. So, we used a list of the UK's most influential tweeters written by Ian Burrell (The Independent newspaper). His methodology to choose the people included help from the social media monitoring group, PeerIndex, with additional input from a panel of experts. We randomly selected 100 names from the 2011[4] and 2012[5] lists and crawled their Twitter accounts. We created a sample from all tweets posted before October 22nd, 2013 (inclusive). Our dataset contains on average 3,194 twitter messages with 301,100 characters per author. The Twitter terms of services forbids third-parties from redistributing Twitter Content [6]. Third-parties are allowed to distribute a set of tweet identifiers (tweet IDs and user IDs). A researcher could use the Twitter REST API to download each tweet in JavaScript Object Notation (JSON) format or to crawl raw HTML pages from the twitter.com site. Although the JSON structure provides several information, we used only the content from the "text" field in our experiments, which characterize the authorship of a message.

B. Data Preprocessing

In order to obtain the same structural data and improve classification accuracy, we performed several preprocessing steps to the data as follows:

[2]available at http://www.cs.cmu.edu/~enron/

[3]Available at http://www.uvic.ca/engineering/ece/isot/datasets/

[4]Available at http://www.independent.co.uk/news/people/news/the-full-list-the-twitter-100-2215529.html

[5]Available at http://www.independent.co.uk/news/people/news/the-twitter-100-the-full-ataglance-list-7467920.html

[6]https://dev.twitter.com/terms/api-terms

- For e-mail messages:
  - E-mails from the folders "sent" and "sent items" within each user's folder were selected, with all duplicate e-mails removed;
  - JavaMail API was used to parse each e-mail and extract the body of the message;
  - Remove messages where the average of digit per total of character was higher than 25%;
  - Strip replies;
  - Replace e-mail address for a single e-mail word;
  - Replace htpd address for a single http word;
- For micro messages:
  - All non-English messages and all Re-Tweet (RT) posts were deleted. The RT information can be retrieved from the JSON structure, but the RT flag is false when the author write something before the RT. So, we performed an extra filtering by analyzing the content of the message and keeping only the text before the RT flag;
  - Based on the unicode characters, we deleted messages that contain one or more of the following unicode blocks: ARABIC, CYRILLIC, DEVANAGARI, HANGUL-SYLLABLES, BENGALI, HEBREW, MALAYALAM, GREEK, HIRAGANA, CHEROKEE, CJK-UNIFIED-IDEOGRAPHS;
  - Replace pound sign or hashtag symbol "#word" and the following word for a meta tag "#hash". Hashtag is used before a relevant keyword in order to categorize the topic, allowing a topic to be searched easily;
  - Replace @username by a meta tag "@cite". The @ sign followed by a username link to a Twitter profile;
- Since different texts must be logically equivalent, (i.e. must have the same canonical form), the following filters were applied:
  - Replace phone number by a single phone word;
  - Replace currency by $XX;
  - Replace percentage by XX%;
  - Replace information between tags ("<information>") by a single TAG word;
  - Replace date by a meta tag date;
  - Replace time by a meta tag time;
  - Delete content when contains the following information: Date:, Time:, Location:;
  - Replace numbers by a meta tag "numb";
  - Replace information among quotes ("information") by a meta tag quote;
  - Normalize the document to printable ASCII;
  - Convert the document to lowercase characters;
  - Strip white space.

Our proposed approach is a two-class classification problem. The first class is composed by positive sample from the author, whereas the negative class is composed by all samples from other authors. Thereby, negative class has more samples than the positive class, generating imbalance class distribution. Our approach to deal with this situation is to assign a weight to the negative class corresponding to the ratio between the total number of positive samples and the total number of negative samples.

C. Evaluation Method

We implemented our proposed approach in Java and used a popular machine learning toolkit named WEKA (Waikato Environment for Knowledge Analysis)[7] [47]. We used a SVM learner called Sequential Minimal Optimization (SMO) and a Logistic Regression learner called SimpleLogistic in WEKA [48], [49].

After the preprocessing phase, the Enron dataset was reduced from 150 authors to 76 authors to ensure that only users with 50 instances and 500 characters per instance were involved in our analysis. The number of users in the Twitter dataset remained 100.

In order to simulate CA, all messages per author were grouped creating a long text or stream of characters that was divided into blocks. CA occurs by performing authentication decisions repetitively over consecutive blocks of data captured during a session. We performed our tests with a block size of 140, 280, and 500 characters on average and 50, 100, and 200 blocks per author.

To evaluate our authorship verification scheme, we employed the 10-fold cross-validation approach. We randomly sorted the dataset, and allocated in each (validation) round 90% of the dataset for training and the remaining 10% for testing. For each user U, we computed a corresponding profile by using their training data and training data from other users considered as impostors. Ten rounds of cross-validation were performed using different partition of the dataset. In each round, we computed the FRR for user U by comparing each of the instances from her test data against her profile. A false rejection (FR) is counted when the system rejects one of these instances. The FAR is computed by comparing each of the test instances from the other users (i.e. the impostors) against the profile of user U. A false acceptance (FA) occurs when the system categorizes any of these instances as belonging to user U. By repeating the above process for each of the users, we compute the overall FAR and FRR by averaging the individual measures. Finally, we computed the EER, which corresponds to the operating point where FAR and FRR have the same value.

D. Results

We studied and compared the performance of the SVM classifier alone against the performance of the hybrid SVM-LR classifier and LR.

In order to test the effect of the SVM kernel, we conducted a set of experiments using Twitter dataset involving 100 authors, a block size of 280 characters and 100 blocks per user. Feature selection uses only information gain approach. Our first experiment uses a linear kernel, which yields EER of 18.86% and 15.34% for pure SVM and hybrid SVM-LR, respectively. Subsequent experiments focused only on the hybrid SVM-LR classifier, since it outperforms pure SVM

[7] Available at http://weka.wikispaces.com

by about 3%. Experiments using polynomial kernel degree 3 and degree 5, and also Gaussian kernel yield (for the hybrid classifier) EER varying from 19.20% to 46.01%, as shown in Table II. From the above results, we can conclude that the hyperplane separating positive from negative data is linear. Therefore, the next experiments were run with linear kernel only.

TABLE II. EER OBTAINED BY VARYING THE TYPE OF SVM KERNELS

SVM Kernel	ERR %
Linear (Pure SVM)	18.86
Linear (Hybrid)	**15.34**
Polynomial 3 (Hybrid)	19.20
Polynomial 5 (Hybrid)	28.47
Gaussian (Hybrid)	46.01

Comparison among different SVM kernels based on the Twitter dataset involving 100 authors with block size of 280 characters and 100 blocks per user. In these experiments only Information Gain was used as a feature selection technique.

Our second set of experiments extend the feature selection by adding the mutual information selection approach and varying the block size and the number of blocks per user. Table III shows our results based on Twitter dataset involving 100 authors with block size of 140 characters and 100 blocks per user, yielding ERR of 21.45% and 19.05%, using SVM and hybrid SVM-LR classifiers, respectively. Increasing the training set and test set size affect the accuracy. For instance, when increasing the number of blocks per user to 200, we obtain ERR of 18.37% and 16.74%, for SVM and hybrid SVM-LR classifiers, respectively.

Also, using a block size of 280 characters and 50 blocks per user, our results reach ERR of 17.83% and 16.16% for hybrid SVM-LR and LR classifiers, respectively. Using 100 blocks per user, we obtain ERR of 13.27% and 11.83% for hybrid SVM-LR and LR classifiers, respectively.

Furthermore, our results demonstrate that combining Information Gain and Mutual Information feature selections achieves significant improvement over our baseline system. As a result, the impact of feature selection (IG + MI) had an improvement of 3% again, when compared to the best results using SVM with block size of 280 characters and 100 blocks per user.

TABLE III. AUTHORSHIP VERIFICATION USING TWITTER DATASET

Block Size	Blocks per user	SVM-LR %	LR %
140	100	21.45	19.05
	200	18.37	16.74
280	50	17.83	16.16
	100	13.27	**11.83**

EER for SVM and hybrid SVM-LR using Twitter dataset involving 100 authors and varying the size of the block and the number of blocks per author. Feature selection was performed by Information Gain and Mutual Information approaches.

Our third set of experiments was based on Enron dataset involving 76 authors with block size of 500 characters and 50 blocks per author. Feature selection was performed using Information Gain and Mutual Information approaches. Table IV shows our results for SVM and LR classifiers, where ERR of 9.98% and 9.18% were obtained, respectively.

We also examine the classification speed for hybrid SVM-LR and LR. Table V illustrates the processing time measured in seconds for different experiments and classifiers. The column "Train" shows the processing time required to train the profile of a single author. The performance is computed by varying the number of features and training samples. In fact, the number of features has the most significant impact on the performance. For instance, using SVM classifier, the required time to train a single user with 3,420 training samples and 242 features was 1.30 seconds. On the other hand when the number of features decrease to 147 and the number of training samples increases to 4,500, the overall time decreases to 1.16 seconds. Furthermore, results demonstrate that LR requires substantially more processing time to train a classifier than hybrid SVM-LR; on average LR is 22 times slower than hybrid SVM-LR. All experimental tests were performed on a Dell C6100 computer with twelve 2.66-GHz Xeon x5650 cores and 24 GB of RAM. The experiments were run in a serial job and used only one core at a time.

TABLE IV. AUTHORSHIP VERIFICATION USING ENRON DATASET

Block Size	Blocks per user	SVM-LR	LR
500	50	9.98	**9.18**

Authorship Verification with 76 authors, block size of 500 characters, and 50 blocks per author. Feature selection was performed by Information Gain and Mutual Information approach. LR classifier outperforms hybrid SVM-LR.

V. DISCUSSIONS

The results obtained in this work are promising compared with the performance of other existing authorship verification approaches. For example, one of the best results among the few existing papers on authorship verification by Canales et al. achieved an EER of 30% [5].

Furthermore, we improve accuracy while reducing authentication delay (message block size). Smaller block size means shorter authentication delay, which is important for CA. Our attempt to reduce the authentication delay (block size from 500 to 280 characters) results in an increase of EER by about 2.7 percentage points, ranging from 9.18% to 11.83%, because block size and number of blocks per user are loosely related to each other. A reasonable explanation is that 280 characters has fewer stylistic information than 500 characters. However, the results could be improved by increasing the number of blocks per authors. To our knowledge, it is the first time that a Twitter dataset was used for authorship verification tasks. Table VI summarizes the performances, block sizes, and population sizes of this study. More work still needs to be done in order to reach a more robust CA system.

The bottom line is that SVM with linear kernel produced best EER accuracy than polynomial or Gaussian. In addition, a hybrid SVM with LR improves the prediction accuracy of the SVM classifier. However, the LR classifier outperforms the hybrid SVM-LR classifier.

VI. CONCLUSION

Continuous authentication requires high accuracy, low authentication delay and ability to withstand forgery. In this paper, we investigated the use of authorship verification as a method for continuous authentication targeting the first two requirements for a robust CA. Our feature set consists of lexical,

TABLE V. PROCESSING TIME MEASURED IN SECONDS

Dataset	Number of blocks / Block size	Training Samples	Test samples	Features	SVM-LR		LR	
					Train	1 Fold	Train	1 Fold
Enron (76)	50/500	3420	380	242	1.30	304.09	28.25	2632.21
Twitter (100)	50/140	4500	500	147	1.16	324.33	12.07	2577.38
	50/280	4500	500	290	1.49	578.10	25.32	4833.40
	100/140	9000	1000	211	3.10	1161.92	51.14	8340.87
	100/280	9000	1000	452	4.65	1837.21	149.61	14533.69
	200/140	18000	2000	220	5.36	3064.92	114.69	21389.95

TABLE VI. COMPARATIVE PERFORMANCES, BLOCK SIZES AND, POPULATION SIZES FOR THE CURRENT RESULTS IN AUTHORSHIP VERIFICATION.

Sample Population Size	Block Size	Number of Features	Technique	EER (%)
76 *	500 character	L(537+), Sy(362+), A(7)	SVM-LR and LR	9.98 - 9.18
100 **	140 character	L(537+), Sy(362+), A(7)	SVM-LR and LR	18.37 - 16.74
100 **	280 character	L(537+), Sy(362+), A(7)	SVM-LR and LR	13.27 - 11.83

* Used Enron dataset for testing.
** Used Twitter dataset for testing.
(L) = Lexical, (Sy) = Syntactic, (A) = Application

syntactic, and application specific features. In order to select the best set of features to represent individual user profile, we computed and analyzed the information gain. In addition, we applied mutual information feature selection in order to discard features that are highly correlated. Experiment results based on Enron email dataset indicate the authorship verification performance improved from previous studies, where the ERR dropped from 12.42% to 9.18%. Also, results based on the Twitter dataset with block size of 140 and 280 characters showed a promising EER of 16.74% and 11.83%, respectively. Results show that linear regression classifier outperforms SVM classifier. However, SVM is faster than logistic regression for training and classification tasks.

Although our performance results are encouraging, we still need to improve them significantly to be fully usable for continuous authentication. We plan in our future work to improve the obtained EER by investigating new machine learning techniques such as Deep Belief Network classifiers, which have been shown powerful analysis techniques in handwriting and visual detection of objects.

Another important issue that we need to address to achieve a robust CA system is to assess and strengthen the approach against forgeries. Stylometry analysis can be the target of automated attacks, also referred to as generative attacks, where high quality forgeries can be generated automatically using a small set of genuine samples [50]. An adversary having access to writing samples of a user may be able to effectively reproduce many of the existing stylometric features. It is essential to integrate specific mechanisms in the authentication system that would mitigate forgery attacks. Furthermore, our evaluation methodology and dataset must be upgraded by generating and incorporating forgery samples. We intend to tackle these issues in our future work.

VII. ACKNOWLEDGMENTS

This research has been enabled by the use of computing resources provided by WestGrid and Compute/Calcul Canada. The research is funded by a Vanier scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC) and CNPq scholarship (Brazil). Also, we are grateful to Dr. Aaron Gulliver for his insights regarding Information Theory.

REFERENCES

[1] I. Traore and A. A. E. Ahmed, Eds., *Continuous Authentication Using Biometrics: Continuous Authentication Using Biometrics: Data, Models, and Metrics*. IGI Global, 2012.

[2] J. Li, R. Zheng, and H. Chen, "From fingerprint to writeprint," *Commun. ACM*, vol. 49, pp. 76–82, April 2006.

[3] J. L. Hilton, *On verifying wordprint studies: Book of Mormon authorship*, ser. Reprint (Foundation for Ancient Research and Mormon Studies). F.A.R.M.S., 1991.

[4] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Trans. Inf. Syst.*, vol. 26, pp. 7:1–7:29, April 2008.

[5] O. Canales, V. Monaco, T. Murphy, E. Zych, J. Stewart, C. T. A. Castro, O. Sotoye, L. Torres, and G. Truley, "A stylometry system for authenticating students taking online tests," P. of Student-Faculty Research Day, Ed., CSIS. Pace University, May 6 2011.

[6] M. Koppel and J. Schler, "Authorship verification as a one-class classification problem," in *Proceedings of the 21st international conference on Machine learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 62–.

[7] C. Sanderson and S. Guenter, "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*, ser. EMNLP '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 482–491.

[8] C. E. Chaski, "Who's at the keyboard: Authorship attribution in digital evidence investigations," *International Journal of Digital Evidence*, vol. 4, no. 1, Spring 2005.

[9] J. H. Clark and C. J. Hannon, "A classifier system for author recognition using synonym-based features," in *Proceedings of the 6th Mexican international conference on Advances in artificial intelligence*, ser. MICAI'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 839–849.

[10] M. Gamon, "Linguistic correlates of style: authorship classification with deep linguistic analysis features," in *Proceedings of the 20th international conference on Computational Linguistics*, ser. COLING '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004.

[11] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem, "Towards an integrated e-mail forensic analysis framework," *Digital Investigation*, vol. 5, no. 3-4, pp. 124 – 137, 2009.

[12] N. Homem and J. Carvalho, "Authorship identification and author fuzzy fingerprints," in *Fuzzy Information Processing Society (NAFIPS), 2011 Annual Meeting of the North American*, march 2011, pp. 1 –6.

[13] F. Iqbal, H. Binsalleeh, B. C. Fung, and M. Debbabi, "Mining writeprints from anonymous e-mails for forensic investigation," *Digital Investigation*, vol. 7, no. 1-2, pp. 56 – 64, 2010.

[14] ——, "A unified data mining solution for authorship analysis in anonymous textual communications," *Information Sciences*, 2011.

[15] F. Iqbal, R. Hadjidj, B. C. Fung, and M. Debbabi, "A novel approach of mining write-prints for authorship attribution in e-mail forensics," *Digital Investigation*, vol. 5, Supplement, no. 0, pp. S42 – S51, 2008, the Proceedings of the Eighth Annual DFRWS Conference.

[16] D. Pavelec, L. Oliveira, E. Justino, F. Neto, and L. Batista, "Author identification using compression models," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, july 2009, pp. 936 –940.

[17] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resour. Eval.*, vol. 45, pp. 83–94, March 2010.

[18] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale author identification," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 300–314.

[19] A. Abbasi and H. Chen, "Applying authorship analysis to extremist-group web forum messages," *IEEE Intelligent Systems*, vol. 20, pp. 67–75, September 2005.

[20] N. Cheng, R. Chandramouli, and K. Subbalakshmi, "Author gender identification from text," *Digital Investigation*, vol. 8, no. 1, pp. 78 – 88, 2011.

[21] N. Cheng, X. Chen, R. Chandramouli, and K. Subbalakshmi, "Gender identification from e-mails," in *Computational Intelligence and Data Mining, 2009. CIDM '09. IEEE Symposium on*, 30 2009-april 2 2009, pp. 154 –158.

[22] M. Corney, O. de Vel, A. Anderson, and G. Mohay, "Gender-preferential text mining of e-mail discourse," in *Proceedings of the 18th Annual Computer Security Applications Conference*, 2002, pp. 282 – 289.

[23] T. Kucukyilmaz, B. B. Cambazoglu, C. Aykanat, and F. Can, "Chat mining: Predicting user and message attributes in computer-mediated communication," *Information Processing Management*, vol. 44, no. 4, pp. 1448 – 1466, 2008.

[24] K. G. Ruchita Sarawgi and Y. Choi, "Gender attribution: tracing stylometric evidence beyond topic and genre," in *Proceedings of the 15th Conference on Computational Natural Language Learning*, ser. CoNLL '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 78–86.

[25] X. Chen, P. Hao, R. Chandramouli, and K. P. Subbalakshmi, "Authorship similarity detection from email messages," in *Proceedings of the 7th international conference on Machine learning and data mining in pattern recognition*, ser. MLDM'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 375–386.

[26] H. V. Halteren, "Author verification by linguistic profiling: An exploration of the parameter space," *ACM Trans. Speech Lang. Process.*, vol. 4, pp. 1:1–1:17, February 2007.

[27] I. Krsul and E. H. Spafford, "Authorship analysis: identifying the author of a program," *Computers and Security*, vol. 16, no. 3, pp. 233 – 257, 1997.

[28] M. L. Brocardo, I. Traore, S. Saad, and I. Woungang, "Authorship verification for short messages using stylometry," in *In Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS)*. Piraeus-Athens, Greece, May 2013, pp. 1–6.

[29] M. L. Brocardo, I. Traore, and I. Woungang, "Toward a framework for continuous authentication using stylometry," in *The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA-2014)*, Victoria, Canada, May 2014.

[30] F. Iqbal, L. A. Khan, B. C. M. Fung, and M. Debbabi, "E-mail authorship verification for forensic investigation," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10. New York, NY, USA: ACM, 2010, pp. 1591–1598.

[31] S. M. Alzahrani, N. Salim, and A. Abraham, "Understanding plagiarism linguistic patterns, textual features, and detection methods," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.

[32] B. Kjell, W. Woods, and O. Frieder, "Discrimination of authorship using visualization," *Information Processing and Management*, vol. 30, no. 1, pp. 141 – 150, 1994.

[33] F. Peng, D. Schuurmans, S. Wang, and V. Keselj, "Language independent authorship attribution using character level language models," in *Proceedings of the 10th Conference on European. Chapter of the Association for Computational Linguistics - Volume 1*, ser. EACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 267–274.

[34] E. Stamatatos, "A survey of modern authorship attribution methods," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, pp. 538–556, March 2009.

[35] P. Juola, "Authorship attribution for electronic documents," in *Advances in Digital Forensics II*, ser. IFIP Advances in Information and Communication. Springer New York, 2006, vol. 222, pp. 119–130.

[36] H. Baayen, H. van Halteren, and F. Tweedie, "Outside the cave of shadows: using syntactic annotation to enhance authorship attribution," *Literary and Linguistic Computing*, vol. 11, no. 3, pp. 121–132, 1996.

[37] P. Juola and R. H. Baayen, "A controlled-corpus experiment in authorship identification by cross-entropy," *Literary and Linguistic Computing*, vol. 20, no. Suppl, pp. 59–67, 2005.

[38] Y. Zhao and J. Zobel, "Searching with style: authorship attribution in classic literature," in *Proceedings of the thirtieth Australasian conference on Computer science - Volume 62*, ser. ACSC '07. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2007, pp. 59–68.

[39] O. de Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *SIGMOD RECORD*, vol. 30, pp. 55–64, 2001.

[40] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *J. Am. Soc. Inf. Sci. Technol.*, vol. 57, pp. 378–393, February 2006.

[41] S. Kotsiantis and D. Kanellopoulos, "Discretization techniques: A recent survey," *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 47–58, 2006.

[42] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuousvalued attributes for classification learning," in *Thirteenth International Joint Conference on Articial Intelligence*, vol. 2. Morgan Kaufmann Publishers, 1993, pp. 1022–1027.

[43] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[44] V. Vapnik, *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1982.

[45] G. Wahba, G. Wahba, X. Lin, X. Lin, F. Gao, F. Gao, D. Xiang, D. Xiang, R. Klein, and B. Klein, "The bias-variance tradeoff and the randomized gacv," in *Advances in Neural Information Processing Systems*. MIT Press, 1999, pp. 620–626.

[46] Y.-c. I. Chang, "Boosting svm classifiers with logistic regression," Institute of Statistical Science - Academia Sinica, Tech. Rep., 03 2003.

[47] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical machine learning tools and techniques with java implementations," 1999.

[48] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Mach. Learn.*, vol. 59, no. 1-2, pp. 161–205, May 2005.

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

[50] L. Ballard, "Biometric authentication revisited: Understanding the impact of wolves in sheep's clothing," in *In Proceedings of the 15 th Annual Usenix Security Symposium*, 2006, pp. 29–41.