Received: 15 August 2016 | Revised: 16 November 2016 | Accepted: 17 November 2016

DOI: 10.1002/dac.3259

WILEY

RESEARCH ARTICLE

Authorship verification using deep belief network systems

Marcelo Luiz Brocardo[1] | Issa Traore[1] | Isaac Woungang[2] | Mohammad S. Obaidat[3]

[1]Dept. of Electrical, & Computer Engineering, University of Victoria, BC, Canada
[2]Dept. of Computer Science, Ryerson University, Toronto, ON, Canada
[3]Dept. of Computer and Information Science, Fordham University, New York, USA

Correspondence
Isaac Woungang, Dept. of Computer Science, Ryerson University, 245 Church Street, Toronto, Ontario, M5B2K3, Canada.
Email: iwoungan@scs.ryerson.ca

Summary

This paper explores the use of deep belief networks for authorship verification model applicable for continuous authentication (CA). The proposed approach uses Gaussian units in the visible layer to model real-valued data on the basis of a Gaussian-Bernoulli deep belief network. The lexical, syntactic, and application-specific features are explored, leading to the proposal of a method to merge a pair of features into a single one. The CA is simulated by decomposing an online document into a sequence of short texts over which the CA decisions happen. The experimental evaluation of the proposed method uses block sizes of 140, 280, 500 characters, on the basis of the Twitter and Enron e-mail corpuses. Promising results are obtained, which consist of an equal error rate varying from 8.21% to 16.73%. Using relatively smaller forgery samples, an equal error rate varying from 5.48% to 12.3% is also obtained for different block sizes.

KEYWORDS

authorship verification, continuous authentication, Gaussian-Bernoulli deep belief network, stylometry

1 | INTRODUCTION

Continuous authentication (CA) is a reinforcement of the traditional static authentication which protects against session hijacking. The CA consists of repeating the authentication process from time to time. The strength of a CA system depends on the adequacy/strength of the underlying authentication modalities. In the literature, CA using biometric modalities have been extensively studied,[1,2] but CA using stylometry is yet to be widely explored.[3] Stylometric analysis involves establishing a document's authorship on the basis of sample writing styles. This process can be classified in 3 different categories: authorship identification, authorship characterization, and authorship verification.[4]

Authorship identification assigns an author for a document from a group of possible authors. Authorship verification checks if a document was written or not by a specific person. Finally, authorship characterization (or profiling) determines the characteristics (e.g., race, sex, and age) of the author of an anonymous document. It is well known that authorship verification is the most relevant to CA because the user identity verification is a key element of any authentication system. Among the above 3 forms of stylometric analysis, authorship verification is the most relevant to CA, as user identity verification is central to any authentication system. Similar to forensic authorship verification, authentication consists of verifying the similarity between a text and the users' profile (or model) at the login time (this is a 1-to-1 identity matching). While authorship attribution/identification and authorship characterization using stylometry have been widely investigated in the literature, limited work exists in authorship verification, particularly, stylometry-based authorship verification for online documents (e.g., e-mails and tweets), which poses significant challenges because of the unstructured nature of such documents.[4]

A key requirement of CA is that repeated authentication decisions should occur over a short period or a short text or messages. Stylometric analysis using short messages is a challenging issue because of the limited amount of information available for decision-making purpose. Likewise, most stylometric analysis approaches thus far proposed in the literature use relatively large document sizes and are unsuitable for CA. The threat of forgery is another issue to be care about when using stylometry for CA because an adversary might be able to reproduce many of the stylometric features if he had access to writing samples of a user. Therefore, it is important

to develop new mechanisms to integrate in the authentication system that can mitigate the forgery attacks. In an attempt to address those shortcomings, new stylometric features and robust classifiers are explored in this paper.

The shallow-structured architectures of machine learning have been widely used for authorship analysis of electronic documents.[3] Such architecture refers to a classifier with only 1 or 2 layers responsible for classifying the features into a problem-specific class. Examples of such classifiers are Naïve Bayes and hidden Markov model. In the work of Koppel and Schler,[5] it was proven that shallow architectures can be effective in solving many stylometric analysis problems. However, the approaches proposed so far still face significant challenges in identifying an author when the number of possible authors grows or when the size of text for analysis decreases.[6] Deep belief networks (DBNs) have emerged as alternatives to shallow machine learning techniques.[7]

In this paper, a stylometry-based authorship verification model on the basis of the Gaussian-Bernoulli DBN is presented, which uses Gaussian units in the visible layer to model real-valued data. To our knowledge, this is the first attempt to use DBN for stylometry-based authorship analysis. A new set of stylometric features is introduced on the basis of the *n*-gram analysis, and a method to merge a pair of random features into a single feature is proposed. Our proposed model is evaluated using a micromessages dataset on the basis of Twitter feeds and the Enron e-mails dataset, on the basis of the following performance metrics: (1) false acceptance rate (FAR), consists of measuring the probability of falsely recognize someone as a genuine person; (2) false rejection rate (FRR), consists of measuring the probability of rejecting a genuine person; and (3) equal error rate (EER), consists of determining the operating point where the FRR and FAR have a similar value. In addition, as part of this research, a forgery dataset has been created by collecting the simulated attacks against the profiles of 10 users, and different block sizes including 140, 280, and 500 characters have been tested on the above-mentioned Enron and Twitter datasets, yielding an EER ranging from 8.21% to 16.73%.

The rest of the paper is organized as follows. In Section 2, some related work are discussed. Section 3 describes the feature space. In Section 4, our classification model is presented. Section 5 is devoted to the experimental setup. In Section 6, the experimental results are presented. Finally, Section 7 concludes the paper.

2 | RELATED WORK

Authorship analysis using stylometry has so far been studied primarily for the purpose of forensic analysis.[5] Various works in the literature have deal with authorship verification, either from a similarity detection issue standpoint or as a 1- or 2-class problem, where the proposed models mostly rely on shallow machine learning architectures for classification.[3,5] Examples of such shallow classifiers used in stylometry-based authorship verification models include k-NN, Naïve Bayes, decision tree, Markov chains, support vector machine (SVM), and logistic regression, to name a few.

Koppel and Schler[5] introduced a technique to quantify the dissimilarity between a text from a suspect and other users (i.e., the imposters). The considered dataset is composed of 10 authors, where 21 English books are split into blocks of 500 words. Their scheme produced an overall accuracy of 95.7% when analyzing the feature set composed by the 250 most frequent words.

Iqbal et al[6] experimented not only with variants of SVM including SVM with sequential minimum optimization and SVM with RBF kernel but also with linear regression, Bayesian network, and discriminative multinomial naive Bayes classifiers. Their feature set included lexical, syntactic, idiosyncratic (grammatical and spelling mistakes), and content-specific features. Their scheme has been evaluated using the Enron e-mail corpus, yielding an EER ranging from 17.1% to 22.4%.

Canales et al[8] introduced a method that combines stylometry and keystroke dynamics analysis for authenticating the online test takers. In their scheme, the k-NN algorithm is used for classification purpose. The experimental evaluation of their scheme involved 40 students with sample document sizes ranging between 1710 and 70 300 characters, yielding respectively (FRR = 20.25%, FAR = 4.18%) and (FRR = 93.46%, FRR = 4.84%) as performances when using separately the keystroke and the stylometry. The combination of both types of features yielded an EER of 30%.

Chen et al[9] proposed to measure the similarity between e-mail messages by mining frequent patterns. In their proposed model, the basic feature set includes lexical, syntactic, content specific, and structural features, and principal component analysis, k-NN and SVM are used as classifiers. The experimental evaluation of their scheme involved 40 authors and used a subset of the Enron dataset, yielding 84% and 89% as classification accuracy rates for 10 and 15 short e-mails, respectively.

Koppel and Winter[10] proposed an unsupervised method for authorship verification that uses a dataset consisting of 500 blog pairs. Their method consists of transforming the authorship problem from a 1-class to a multiclass classification problem by adding additional authors from external sources (such as the Web). Fragments or chunks of blogs consisting of 500 words are analyzed, and the features of the 100 000 most frequent character 4-grams are extracted. The experimental evaluation of their scheme yields a classification rate of 87.4% for the blog dataset.

In the work of Brocardo et al,[11] authorship verification, using short messages, has been investigated with focus on shallow machine learning architectures. More specifically, a combination of logistic regression and SVM (so-called SVM-LR method) has been introduced for classification purpose. The proposed approach was evaluated using shorter

messages that consist of blocks of texts of 140, 280, and 500 characters on the basis of twitter feeds and on Enron e-mails, yielding an EER ranging from 9.18% to 21.45% for different configurations. In this paper, we improve these performances by investigating a new model on the basis of a deep machine learning framework.

3 | FEATURE SPACE

In our framework, a document is decomposed into a sequence of short texts over which CA decisions happen. Each block of text b is represented as a set of features $X = (x_1, x_2, \dots, x_n)$. Each feature x_i is normalized in a scale between 0 and 1 using a *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. Our feature set includes some basic features inspired from the existing literature, and new features derived from novel n-gram analysis and features merging techniques proposed in this work.

To reduce the large feature space and eliminate redundant features that could be created by the merging process, the information gain and mutual information (MI) have been computed and analyzed for feature selection.

3.1 | Basic features

In this study, we have divided the features into 3 subsets: lexical, syntactic, and application specific features.

Lexical features. These features are meant to indicate the preference of a user for a certain group of words or symbols. They can be extracted by dividing the text into tokens, where a token can be a word or a character. Word-level features may include the average sentence length of words, the frequency of short and long words, the average word length, and the most frequently used words per author.[8] Vocabulary richness is computed by counting the number of words that occur only once or twice, which are called *hapax legomena* and *dis legomena*, respectively. The relevant character-level features include the frequency of characters comprehending upper case, lower case, vowels, white spaces, alphabets (A-Z) digits, special characters, and the writer's mood expressed in the form of icons and symbols. Another type of lexical features that has proven to be efficient in capturing the writing style is the n-grams count.[5] An N-gram is a token formed by a contiguous sequence of characters or words, which is tolerant to typos including grammatical errors and misuse of punctuations. We have derived new features corresponding to character *5*-grams and *6*-grams, and we also devise a new method to determine the character n-grams (see Section 3.2). We have also created a vector with the most frequent words *2*-grams and *3*-grams.

Syntactic features. These features are context independent and can be used to capture the author's style across different subjects. They are categorized in punctuation and part of speech (POS).[5] Punctuation is a rule that can be used to define the boundaries and identify the meaning by splitting a paragraph into sentences, which themselves can be split into tokens.[6] Punctuation includes single quotes, commas, periods, colons, semicolons, question marks, exclamation marks, and uncommon marks on the basis of the unicode format (e.g., †, ⋮, … ∴). The POS feature consists of tagging a word on the basis of its context, and it can be classified as verbs, nouns, pronouns, adjectives, adverbs, prepositions, conjunctions, and interjections.[9] The weakness of this type of features is that POS is language-dependent because it relies on a language parser and also could produce some noise because of the unavoidable errors made by the parser. In this work, we have use a list of functional words*.

Application specific features. These are meant to capture the overall characteristics of the organization and to format of a text. They can be categorized at the message level or paragraph level, or even according to the technical structure of the document.[13] In this paper, we have extracted only the features related to the paragraph structure because our focus is on short messages. Paragraph-level features include the number of sentences per block of text, the average number of characters, words, and sentences in a block of text, and the average number of sentences beginning with upper and lower cases.

3.2 | *N*-gram model

In our proposed n-gram model, we measure the degree of similarity between a block b of characters and the profile of a user U, for details see our previous work in the work of Brocardo et al.[14] We analyze whether or not a specific n-gram is present and compute new features by defining 2 similarity metrics: $r_U(b, m)$ the real-valued similarity and $d_U(b)$ the binary similarity. For determining the n-gram, 2 modes have been considered: the unique n-grams mode denoted by $m = 0$ and the all n-grams mode denoted by $m = 1$, where m is a binary variable. All n-grams with a frequency equal or higher than a given number f are considered. Only 5-grams and 6-grams are considered, as well as 2 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, the number of new features created from the n-gram model is 2 (for f) × 2 (for m) × 2 (n-gram types) × 2 (for r_U and d_U) = 16. The values of $r_U(b,m)$ are discretized using

*The list of functional words used in our work to tag the syntactic words is obtained from http://www.sequencepublishing.com/academic.html[12]

the entropy-based discretization method proposed in the work of Fayyad and Irani.[15] This method consists of creating a set of contiguous intervals and finding a cut- or split-point that divides the range into a number of intervals.

3.3 | Features merging

Zhou et al[16] achieved a significant performance improvement in generative tasks (such as minimizing reconstruction error) and discriminative tasks (such as minimizing supervised loss function) by merging the features that are similar. The cosine distance is used to find the most similar feature pairs for merging, and a linear combination is applied to generate the new features. In this paper, a new method is proposed to merge a pair of features into a single one that considers only the information gain as selection criterion.

Let $X = [x_1, x_2, \dots, x_n]$ be an n-dimensional feature vector that describes our feature space. Let $S = \{X_1, X_2, \dots, X_m\}$ be the set of training samples for a given user. Each training sample corresponds to a vector of feature values $X_j = [x_{ij}]_{1 \leqslant i \leqslant n}$, where x_{ij} is the value of the feature x_i for the sample X_j. The information entropy $H(x_i)$ of feature x_i is defined as

$$H(x_i) = -\sum_{j=1}^{m} p(x_{ij}) log_2 p(x_{ij}) \tag{1}$$

where $p(x_{ij})$ denotes the probability mass function of x_{ij}.

Given a variable y with samples $(y_1, \dots, y_M)$, the (conditional) entropy of x_i is obtained as

$$H(x_i|y) = -\sum_{j=1}^{m} \sum_{k=1}^{M} p(x_{ij}, y_k) log_2 p(x_{ij}|y_k) \tag{2}$$

Assuming that the dataset has 2 classes (negative and positive), the IG for a feature x_i with respect to a class is computed as

$$IG(Class, x_i) = H(Class) - H(Class|x_i) \tag{3}$$

Since some features have different ranges of values, the selected features are preprocessed before being merged. The preprocessing consists of normalizing the feature values in a range from 0 to 1, and discretizing the numeric feature values into binary values (0 and 1) using the approach described in the work of Fayyad and Irani.[15] The new features created after completing the merging process are also normalized between 0 and 1 and then added to the features list. Given 2 features x and y, let $P_y(x)$ denote the following ratio:

$$P_y(x) = \frac{IG(x)}{IG(x) + IG(y).} \tag{4}$$

Let x_i and x_k be 2 features to be merged in a new one called x_r. The merging consists of computing the values of features x_r from the training samples. The merged values are computed as

$$x_{rj} = P_{x_k}(x_i) \times x_{ij} + P_{x_i}(x_k) \times x_{kj} \tag{5}$$

The decision to keep the new feature is made by comparing the corresponding information gain $IG(x_r)$ against $IG(x_i)$ and $IG(x_j)$, respectively. The new feature x_r is added to the feature set if and only if $Max(IG(x_i), IG(x_k)) < IG(x_r)$. In this case, feature x_r is added to the feature set while features x_i and x_k are removed from the set. The above process is repeated for all features by comparing 2 features at a time.

3.4 | Feature selection

Our final feature set includes the basic features, the character n-grams, and new features generated from the merging process. At this stage, one needs to deal with the problem of redundant features. An ideal feature is expected to have high correlation with a class and low correlation with any other features. On the basis of this, a correlation between a class and a feature is measured by computing the IG and the correlation between a pair of features by computing the MI. Given 2 features x_i and x_k, their MI is given by

$$MI(x_i, x_k) = H(x_i) - H(x_i|x_k) \tag{6}$$

By computing the MI for pairs of features, the features with zero information gain and high correlation are identified and removed for each user. Also, a feature is removed when the MI is higher than 95%.[17] At the end, each user is equipped with a subset of features that is specific to his/her individual profile.

4 | CLASSIFICATION MODEL

We approach the authorship verification as a classification task composed by 2 classes: one made of positive samples from the author and the other made of negative samples originated from the other authors. In the training phase, a profile for individual users is generated given a feature set and a training set of positive and negative blocks of short texts. The dataset is balanced by oversampling the minority class,[18] i.e., the positive samples. To authenticate a user, the monitored block of text is matched against the profile for the claimed identity, and the individual metrics are computed.

4.1 | Deep belief network classification

For classification, we use a generative model that consists of multiple stacked levels of a neural network named Gaussian-Bernoulli DBN. The model is composed by 1 layer of Gaussian-Bernoulli restricted Boltzmann machine (RBM), followed by a stack of RBMs, and a top layer with a shallow classifier. The *RBM*[19] is a generative stochastic network that learns the probability distribution over its set of inputs. It is composed by a layer of n visible (*input*) neurons $v = [v_1, \dots, v_n]$ and a layer of m_1 hidden neurons $h =$

$[h_1, \ldots, h_{m_1}]$. It also allows a connection between the visible and hidden units only, and there is no connection between the units from the same layer.

The standard (Bernoulli-Bernoulli) RBM has binary-valued stochastic neurons in the visible and hidden units as well as a joint configuration (v, h) defined for an energy function $E(v, h)$.[7] Training an RBM consists of minimizing the energy of the network by updating the weights and biases. The *Gaussian-Bernoulli RBM* allows modeling real-valued data in the RBM by transforming the data into binary values using Gaussian units in the visible layer.[7] It has real values in its visible layer and binary values in its hidden layer. On the other hand, the *Gaussian-Bernoulli DBN* is a probabilistic generative model composed of a single layer of Gaussian-Bernoulli RBM and multiple layers of RBMs followed by a softmax layer as shown in Figure 1. The training is semisupervised and is performed in 2 phases: a *pretraining* phase and a *fine-tuning* phase.

The pretraining phase uses unsupervised learning, and it is performed incrementally layer by layer. Likewise, the first layer of the Gaussian-Bernoulli RBM receives the real-valued input. The layer is trained for several epochs, and the activation probabilities from the hidden units are used as the visible data input for the layer up (RBM_1). The same process is repeated for the next layers, propagating upward the transformed data.

A softmax is added on top of the last RBM layer, and the input of the softmax is considered as the output of $h^{(l)}$, which is the last hidden layer. The model is fined tuned through a supervised training phase where the weights are adjusted considering the *inputs* and the desired outputs on the basis of the labeled training data. This is performed using a supervised gradient descent of the negative log-likelihood cost function.

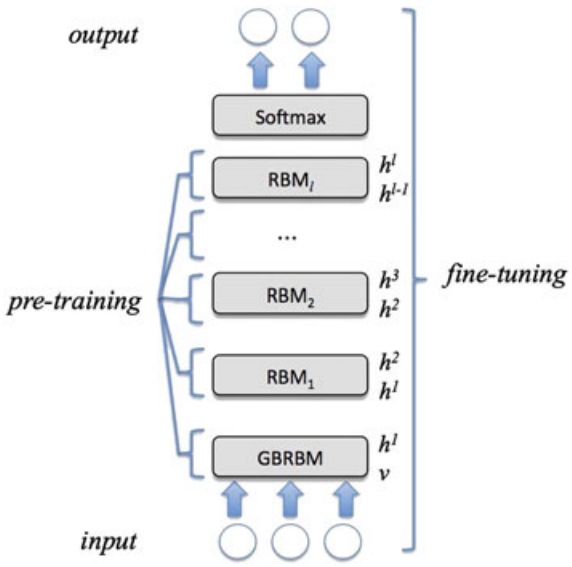


FIGURE 1 Gaussian-Bernoulli Deep Belief Network structure composed by 1 layer of Gaussian-Bernoulli restricted Boltzmann machine (RBM), l layers of RBMs, and a softmax layer on top of the last layer (RBM_l)

4.2 | Model settings and implementation

We have implemented a Java program to perform the feature extraction, feature selection, and data preprocessing. Our Gaussian-Bernoulli DBN classifier is implemented in python with Theano on GPU[20] by modifying the original DBN source code from the Deep Learning Tutorial.[21] Our proposed classifier involved 3 hidden layers consisting of 1 Gaussian-Bernoulli RBM layer and 2 Bernoulli-Bernoulli RBM layers. The softmax is implemented using a logistic regression classifier.

The input layer consists of n real-valued features varying from 0 to 1 as required by the Gaussian-Bernoulli RBM. The inputs of the layer above are binary values as defined in the Bernoulli-Bernoulli RBM. The number of hidden units varies from one author to another. Inspired from the work carried by Sarikaya et al,[22] 3 hidden layers are used in our experiments. We have also adopted a linear shape for our network, where the number of hidden units per layer decreases when the number of layers increases. Therefore, after trying different layer decompositions using the sample data, we have used 75%, 50%, and 25% of the initial feature space for the hidden units that prevail in the first, second, and third layers. Other parameters include a supervised (resp. unsupervised) learning rate of 0.01 (resp. 0.001, mini-batches size equal to the number of self samples 45, 90, or 180, and 100 pretraining epoch.[†])

In our model, the number of epochs in the fine-tuning phase is variable because the number of features ($input(x)$) varies from one author to another. Thus, one author may need fewer epochs to model the training data while another author may need more epochs. To avoid overfitting, the appropriate number of epochs is described as follows:

1. Define a variable ve for the validation error and set ve to a desired target value. The validation error corresponds to the percentage of incorrectly classified training samples.
2. The initial epoch e is set to 50.
3. Calculate the current validation error after performing e epochs.
4. If the current validation error is higher than ve, increment e by 50 epochs.
5. Perform Steps 3 and 4 until the current validation error is lower than ve.
6. Stop the fine-tuning if the current validation error is lower than 2% or e is higher than or equal 1000 epochs.
7. Calculate the metrics for the testing dataset when the fine-tuning phase is over.

5 | EXPERIMENTAL SETUP

This section describes our experimental procedures and evaluation method. The experiment results obtained using our approach are also discussed and contrasted against alternative baselines.

[†]An epoch is a complete learning cycle.

5.1 | Datasets

Three different datasets are used, namely, an e-mail corpus, a micro messages dataset on the basis of twitter feeds, and a forgery dataset.

- *E-mail dataset*. The Enron corpus[‡] is a large set of e-mail messages from Enron, a USA-based company. In this work, we have used a version of this database, which contains more than 200 000 messages (e-mails in plaintext) belonging to 150 users with an average of 757 messages per user.
- *Micro messages dataset*. Twitter is a microblogging service that allows authors to post messages called "tweets." Each tweet is limited to 140 characters and sometimes expresses opinions about different topics. Other particularities of tweets include the following: (1) the use of emoticons to express sentiments; (2) the use of URL shorteners to refer to some external sources; (3) the use of a tag retweet in front of a tweet indicating that the user is repeating or reposting the same tweet; (4) the use of a hashtag "#" to mark and organize tweets according to some topics or categories; (5) the use of the symbol "@ < user > " to link a tweet to a Twitter profile whose user name is "user." The dataset[§] used in this study contains 100 English users and 3194 twitter messages on average, with 301 100 characters per author.[3] All tweets in this dataset were posted before November 6, 2013, inclusive.
- *Impostors dataset*. To test our proposed framework against forgery, we created a novel forgery dataset. The dataset consists of tweets collected from 10 volunteers—with 7 men and 3 women—with ages varying from 23 to 50 years. In the experiment, each volunteer generates forgeries tweets trying to impersonate a specific author. We have selected randomly sample tweets from 10 authors, considered as legal users from the Twitter dataset. Volunteers samples were collected through a simple form consisting of 2 sections. In the first section, tweets from a specific legal user were made available. This allowed simulating a scenario where an adversary has access to legitimate writing samples. In the second section, the participant needed to write 3 or 4 tweets trying to copy the writing style of a legal user. The size of the tweets sample should have at least 350 characters. A "submit" button was used to send the sample to the database when completed. The form was sent by e-mail and made available online via a web page. Our survey was implemented using the Google Forms platform. The experiment run over a period of 30 days, where the volunteers received in different days a new form with different legal user information. We had no control over the way volunteers wrote their tweets, and the collected data consisted of an average of 4253 characters per volunteer spread over 10 attacks.

[‡]It is available at http://www.cs.cmu.edu/~enron/.
[§]The dataset is available at http://www.uvic.ca/engineering/ece/isot/datasets/.

5.2 | Data preprocessing

The data were preprocessed to normalize some e-mail and tweets particularities.[23] In the Enron corpus, we used only the body of the messages from the e-mails found in the folders "sent" and "sent items" for each user. We removed all duplicate e-mails. Similarly, we removed all e-mails that contain tables with numbers when the average number of digits per total number of characters was higher than 25%. We also removed the reply texts when present and replaced e-mail and web addresses by meta tags "e-mail" and "http," respectively.

We removed all retweet posts and all duplicated tweets from the Twitter and forgery corpuses. Hashtag symbols such as "#word" and the following word were replaced by a meta tag "#hash"; @ < user > reference was replaced by meta tag "@cite"; web addresses were replaced by meta tag "http." We also removed all messages that contain 1 or more of the following unicode blocks: Arabic, Bengali, Cherokee, CJK-unified-ideographs, Cyrillic, Devanagari, Greek, Hangul-syllables, Hebrew, Hiragana, and Malaya-lam.

In both datasets, we replaced currency by a meta tag "$XX," percentage by a meta tag "XX%," date by a meta tag "date," hours by a meta tag "time," numbers by a meta tag "numb," phone number by a meta tag "phone," file name by a meta tag "file," and the information between tags (" < information > ") by a meta tag "TAG." Finally, the document was normalized to printable ASCII, all characters were converted to lower case, and the white space was normalized. To simulate the CA, we created a long text by grouping all the messages from the same author and divided it into blocks of characters.

In our cleaned Enron corpus, we kept only 76 from 150 users that had more than 50 instances (blocks) and 500 characters per instance. The number of users in the micro messages and forgery corpuses were kept to 100 and 10, respectively.

5.3 | Evaluation method

Authentication consists of computing the similarity of a sample against the profile corresponding to the claimed identity and comparing the obtained score S against some threshold Th. If the score is equal or exceeds Th, the sample is accepted and considered as *genuine*. Otherwise, it is rejected and classified as originated from an *impostor*.

During the above classification process, samples may be wrongly accepted as genuine or rejected (ie. from imposters). In this context, the accuracy of the biometric system is evaluated primarily for false rejection (FR) and false acceptance (FA). The FR occurs when the system rejects a legitimate user, and the FA occurs when the system accepts an impostor as a legitimate user. The evaluation involves the use of the 10-fold cross validation. At every validation round, the dataset was sorted randomly, and 90% of the data have been allocated for training purpose and 10% for testing purpose. We then recorded the average of all the validation results obtained for all the rounds.

During the enrolment mode in each round of the cross validation, a reference profile is generated for each user. For a user U, this profile is based on a training set consisting of samples from that user (referred to as *positive samples*) and samples (referred to as negative samples) from other users considered as impostors. From these samples, a vector of features was extracted on the basis of which the merging and selection processes were applied.

The verification mode is a 1-to-1 matching process, which consists of comparing a sample against the enrolled user profile. The FR was computed by comparing the test samples of each user U against his/her own profile. The FRR was obtained as the ratio between the number of FRs and the total number of trials. On the other hand, the FA was computed by comparing for each user U all the negative test samples against his/her profile. The FAR was obtained as the ratio between the number of FAs and the total number of trials. The overall FRR and FAR were obtained by averaging the individual measures over the entire user population. Finally, the EER was then derived as the point where the FRR and FAR values match. For our proposed model, the confidence interval is calculated on the basis of the method proposed in the work of Bengio and Mariethoz,[24] by approximating the distribution of the number of errors to a normal distribution with a standard deviation σ.

6 | EXPERIMENTAL RESULTS

6.1 | Using the micro messages corpus

Using the Twitter dataset involving 100 authors, we conducted initially a series of experiments to evaluate our proposed approach, then performed further experiments to compare our approach against some baseline methods. The results are captured in Table 1. We started our evaluation by testing a block size of 280 characters, then reduced this subsequently to 140 characters per block, with 50, 100, and 200 blocks per users.

For each test, we calculated the EER for the optimal *ve* limit. It can be observed that the best result on the Twitter dataset is achieved with a block size of 280 characters and 100 blocks per user with EER of 10.08%. With this configuration, we obtain HTER = 10.06% with a standard deviation $\sigma = 0.0503$. Around this HTER, the calculated confidence interval for different confidence levels are listed in Table 2.

TABLE 1 Authorship verification using the Twitter dataset

Block size	Blocks per user	*ve*	EER %
140	100	9.1	16.73
	200	8.8	16.58
280	50	19.0	12.61
	100	7.0	10.08

In this table, the equal error rate (EER) is obtained for the Gaussian-Bernoulli deep belief network classifier using the Twitter dataset involving 100 authors. In the pretraining phase, 100 is used as epoch, and 0.001 is used as learning rate. In the fine-tuning phase, 0.01 is used as learning rate.

TABLE 2 Margin of error (E) for the confidence interval for HTER Performance

	E		
δ, %	Enron	Twitter	Forgery
90	0.4123	0.2262	0.2083
95	0.4352	0.2388	0.2198
99	0.4535	0.2489	0.2291

The HTER confidence interval for block size of 500 characters on the Enron dataset and 280 characters on the Twitter and Forgery datasets; δ is the confidence level.

Our baseline experiments consisted of different tests to assess the impact of the classifier, the n-gram model, the features merging, and feature selection approaches involved in our approach. These experiments were conducted using the Twitter dataset involving 100 authors, with a block size of 280 characters and 100 blocks per user, and the WEKA (Waikato Environment for Knowledge Analysis)[¶] toolkit. We used Java to implement the proposed framework.

For comparison purpose, a study of the classifier using a linear kernel-based SVM[25] is conducted. We then replaced the proposed deep learning classifier with SVM, and we kept our default feature model consisting of the basic features, the n-gram model, and the features, resulting from features merging. In this case, the feature selection was achieved by considering a positive information gain. Using this configuration, we achieved an EER of 12.34%, which is higher than the EER obtained using our approach (10.08%). In subsequent baseline experiments, we have used DBN as classifier with a validation error *ve* of 7.0 (corresponding to the best results shown in Table 1).

To investigate the impact of the proposed n-gram model, we substituted it by a traditional n-gram model while another features were kept the same. This traditional model is a vector made of the frequencies of the space-free character n-grams in a document.[10] In our previous work,[3] we showed that character n-grams, and in particular *4*-grams, were more effective than alternatives. Since our dataset is composed by short texts, we chose only the 20 000 most common 4-grams to create our baseline feature set. The results of our experiments showed an EER of 12.01% as illustrated in Figure 2, which is an indication that our n-gram model performs better than the traditional model.

To test the effect of our feature merging process, we used the default feature set without feature merging while keeping the rest of the default configuration the same. The experiment yielded an EER of 10.67%, as shown in Figure 2. This shows an increase in EER when the feature merging is not used.

We have also investigated the impact of the feature selection method described in Section 3.4. We performed experiments

[¶]The WEKA is vailable at http://weka.wikispaces.com.

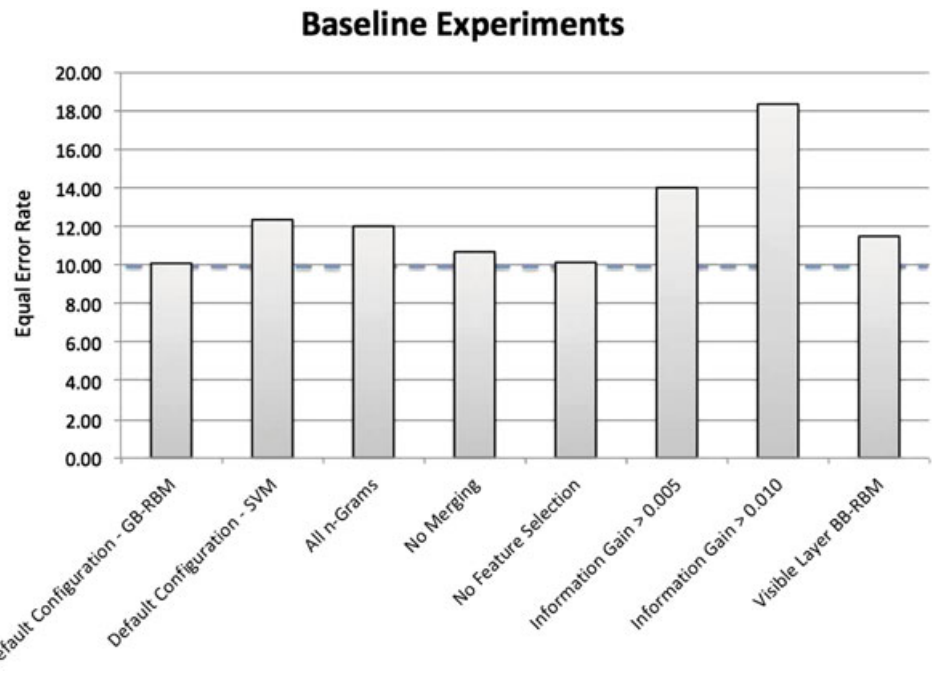


FIGURE 2 Baseline experiments comparing our proposed approach with support vector machine as a classifier, and comparing the impact of the *n*-gram model, feature merging process, feature selection method, and the effect of Bernoulli-Bernoulli restricted Boltzmann machine (RBM) versus Gaussian-Bernoulli RBM as a visible layer in the deep belief network classifier

by setting the information gain to be greater than 0.005 and 0.010, yielding an EER of 13.99% and 18.34%, respectively, as shown in Figure 2. This shows an increase in EER as the information gain threshold increases. When the feature selection was omitted, we obtained as performance a EER of 10.13%, which seems to indicate that DBN did not benefit a lot in accuracy from the feature selection. However, feature selection is still beneficial in reduction in the processing time due reduction in the number of features. On average, 80% reduction in the number of features is achieved when feature selection takes place compared to when it is omitted. Finally, we have analyzed the effect of modeling DBN using real-valued data versus binary data, and we have kept the default configuration. Experiments when using Bernoulli-Bernoulli RBM and Gaussian-Bernoulli RBM for the visible layer yielded an EER of 11.48% and 10.08%, respectively.

The above baseline experimental results indicate that our proposed approach performs well compared to the alternative ones. In the remaining of this section, the results obtained by evaluating our approach using other datasets, namely, the forgery and Enron E-mail corpuses, are presented.

6.2 | Using the e-mail corpus

Our experiments using the Enron corpus was performed with a block size of 500 characters and 50 blocks or instances per user. We performed initially an experiment by using SVM with linear kernel as a classifier, yielding an EER of 11.09%. Our next set of experiments used DBN as a classifier. Figure 3 shows the relationship between FA and FR rates for different values of *ve* varying from 0 to 50. The optimal performance was achieved when setting the *ve* limit to 15, yielding an FRR of 8.24%, an FAR of 8.20%, and an EER of 8.21%. The HTER was found to be 8.22% with a standard deviation $\sigma = 0.0648$. The confidence interval around an HTER is $HTER \pm E$, where E is the margin of error. Table 2 lists the margin of error at different confidence levels δ for the above performance value.

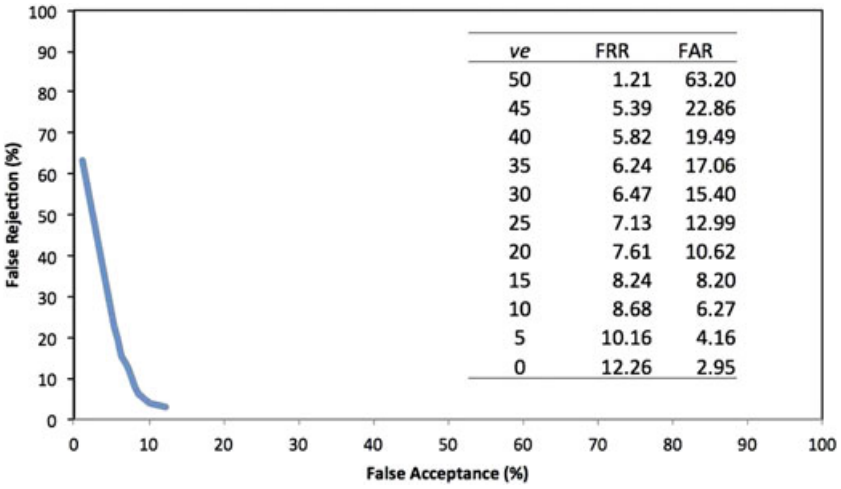
6.3 | Using the forgery corpus

For each of the 10 legal users, we have calculated the FRR by evaluating their own test samples against their profiles. Then, we have calculated the FAR by testing the 10 forgery samples of each legal user against their profile. The EER values were calculated considering the *ve* presented in Table 1. Table 3 shows the obtained EER performance for 2 different block sizes, 280 and 140 characters, which are 5.48% and 12.30%, respectively. The HTER for block size of 280 characters was calculated as 6.68%, with a standard deviation $\sigma = 0.0485$. The corresponding confidence intervals for different confidence levels are shown in Table 2.

TABLE 3 Authorship verification using the Forgery dataset

Block size	Blocks per user	*ve*	EER, %
140	100	9.1	12.30
280	100	7.0	5.48

Experiment results on the Forgery dataset involving 10 forgery attempts against 10 authors profiles.



ve	FRR	FAR
50	1.21	63.20
45	5.39	22.86
40	5.82	19.49
35	6.24	17.06
30	6.47	15.40
25	7.13	12.99
20	7.61	10.62
15	8.24	8.20
10	8.68	6.27
5	10.16	4.16
0	12.26	2.95

FIGURE 3 Receiver operating characteristic curve for the Gaussian-Bernoulli deep belief network classifier on the Enron corpus and sample performance values for different *ve*

TABLE 4 Comparative performances, block sizes, and population sizes for authorship verification studies

Reference	Dataset	Sample size	Block size	Number of features	Classification technique	Accuracy[a] (%)	EER (%)
Canales et al[8]	Custom	40	1710-70300 ch	L(62), Sy(20)	k-NN	—	30
Chen et al[9]	Enron	25-40	30-50 w	L(40), Sy(76), Se(25), A(9)	SVM	83.90 - 88.31	
Iqbal et al[6]	Custom	8	628-1342 w	L(292)	—	17.1 - 22.4	
Koppel and Schler[5]	Custom	10	500 w	L (250)	SVM	95.70	—
Brocardo et al[11]	Enron	87	250-500 ch	L(*n*-gram)	Supervised	—	18.90 - 14.35
Brocardo et al[3]	Enron	76	500 ch	L(91), Sy(251), A(7)	SVM	—	12.42
Brocardo and Traore[26]	Enron	76	500 ch	L(537), Sy(362), A(7)	SVM-LR and LR	—	9.98-9.18
	Twitter	100	140 ch			—	18.37-16.74
	Twitter	100	280 ch			—	13.27-11.83
Current	Enron	76	500 ch	L(703), Sy(362), A(7), Merging	G-B DBN	—	8.21
	Twitter	100	140 ch			—	16.58
	Twitter	100	280 ch			—	10.08

The percentage of correctly matched authors in the testing set. A, application; L, lexical; ch, character; Se, semantic; Sy, syntactic; w, word; G-B DBN, Gaussian-Bernoulli DBN.

These results suggest that the forgery attack has limited impact on the accuracy of the proposed method. On the other hand, it can be noted that the error rates achieved for the forgery dataset are lower than the rates obtained in the previous experiments. Intuitively, such difference in performance can be explained by the forgery dataset is much smaller than the dataset used previously. The literature shows that stylometric experiments on small number of users tend to achieve better results.

6.4 | Discussion

The above experimental evaluation have assessed the ability of our proposed approach to address 3 key challenges related to CA: reducing the authentication delay, high verification accuracy, and the ability to withstand forgery. We simulated the short authentication delays by investigating short blocks of a text. The accuracy achieved on different datasets with block sizes of 500, 280, and 140 characters are very encouraging and much better than the results obtained thus far in the literature as shown in Table 4. The block sizes investigated in our work (i.e., 500 to 140 characters) are shorter than the messages block used so far by other researchers for identity verification. Sanderson and Guenter[27] have investigated similar message size by splitting a long text in chunks of 500 characters.

Accuracy is traditionally measured using Type 1 error, corresponding to the FRR, and Type 2 error, corresponding to the FAR. However, some previous studies calculated only the true match rate, as shown in Table 4. Therefore, the Type 2 error could be derived by calculating $FAR = 1 - accuracy$.

Another important problem investigated in our work is the impact of forgery attacks on the proposed approach. Our performance results obtained are very encouraging. However, our forgery study involved a limited number of attack instances (only 10) on 10 different user profiles. In future, more data should be collected and analyzed to confirm these results.

Compared to the existing literature,[3] it can be argued that the use of a machine learning method on the basis of deep structure, in particular DBN, can help enhancing the accuracy of the authorship verification using stylometry. In the early

stages of this work, we investigated the standard DBN, which has binary neurons only. Our approach was to normalize each input variable to binary values and run the DBN classifier. However, the obtained results did not improve when compared with our previous work[3] using a shallow structure. To strengthen the accuracy, we have replaced the Bernoulli-Bernoulli RBM layer with a Gaussian-Bernoulli RBM layer, where Gaussian units are used in the visible layer to model real-valued data. Also, the Gaussian units work better when the feature values follow a Gaussian distribution.

7 | CONCLUSION

In this paper, a novel framework for carrying CA on the basis of stylometric analysis was proposed, which introduces new stylometric features on the basis of *n*-gram analysis and features merging. It also uses for the first time a deep machine learning technique for the classification of the stylometric profiles. We have also investigated 3 main challenges faced by any CA system, namely, short authentication delay, authentication accuracy, and resilience to forgery. Our experimental evaluation results consist of an ERR of 8.21% and 10.08% for block sizes of 500 and 280 characters, respectively. When using a relatively small forgery dataset, our results yield an EER varying from 5.48% to 12.3% for different block sizes. Although our experiments used only English-based datasets, our model can further be applied to different languages with a slight adjustment to the feature selection technique, especially for language-dependent features such as functional words. We also intend to investigate how the accuracy can be improved by decreasing the EER. Finally, it would be desirable to investigate the resilience of our approach to forgery by expanding the dataset used and to evaluate our proposed method using other published corpora.

REFERENCES

1. Traore I, Woungang I, Nakkabi Y, et al. Dynamic sample size detection in learning command line sequence for continuous authentication. *IEEE Trans Syst Man Cybern Part B Cybern*. 2012;42(5): 1343–1356.
2. Ahmed A, Traore I. Biometric recognition based on free-text keystroke dynamics. *IEEE Trans Cybern*. 2014;44(4): 458–472.
3. Brocardo ML, Traore I, Woungang I. Toward a framework for continuous authentication using stylometry. *Proc. of IEEE AINA2014*, Victoria, BC, Canada; May 2014:106–115.
4. Stamatatos E, Koppel M. Plagiarism and authorship analysis: introduction to the special issue. *Lang Resour Eval*. 2011;45: 1–4.
5. Koppel M, Schler J. Authorship verification as a one-class classification problem. *Proc. of 21st ACM Intl Conference on Machine Learning (ICML)*, Banff, Alberta, Canada; 2004:62–69.
6. Iqbal F, Binsalleeh H, Fung BC, Debbabi M. Mining writeprints from anonymous e-mails for forensic investigation. *Digital Invest*. 2010;7(1-2): 56–64.
7. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Sci*. 2006;313(5786): 504–507.
8. Canales O, Monaco V, Murphy T, et al. A stylometry system for authenticating students taking online tests. *Proc. of Student-Faculty Research Day*. CSIS, Pace University, New York; May 6, 2011:B4.1–B4.6.
9. Chen X, Hao P, Chandramouli R, Subbalakshmi KP. Authorship similarity detection from e-mail messages. Proc. of the 7th Intl. conference on Machine learning and data mining in pattern recognition (MLDM), New York, USA; Sept 3, Aug. 30, 2011:375–386.
10. Koppel M, Winter Y. Determining if two documents are written by the same author. *J Assoc Inf Sci Technol*. 2014;65(1): 178–187.
11. Brocardo ML, Traore I, Saad S, Woungang I. Authorship verification for short messages using stylometry. *Proc. of IEEE CITS 2013*, Piraeus-Athens; May 7-8, 2013:1–6.
12. TheSage's english dictionary and thesaurus sequence publishing. (Available from: http://www.sequencepublishing.com/academic.html) [Accessed on July 26, 2016].
13. Abbasi A, Chen H. Applying authorship analysis to extremist-group web forum messages. *IEEE Intell Sys*. 2005;20(5): 67–75.
14. Brocardo ML, Traore I, Saad S, Woungang I. Authorship verification of e-mail and tweet messages applied for continuous authentication. *J Comput Syst Sci*. 2015;8.8:1429–1440.
15. Fayyad UM, Irani KB. Multi-interval discretization of continuous-valued attributes for classification learning. *Proc. of 13th Intl. Joint Conference on Artificial Intelligence*, vol. 2. Morgan Kaufmann Publishers, Chambery, France; 1993:1022–1027.
16. Zhou G, Sohn K, Lee H. Online incremental feature learning with de-noising auto-encoders. *Proc. of Intl. Conference on Artificial Intelligence and Statistics*, Palma, Canary Islands:1453–1461.
17. Zaffalon M, Hutter M. Robust feature selection by mutual information distributions. *Proc. of the 18th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., Edmonton, Alberta, Canada; 2002:577–584.
18. Barandela R, Valdovinos RM, Sanchez JS, Ferri FJ. The imbalanced training sample problem: under or over sampling? *Structural, Syntactic & Statistical Pattern Recognition*. Springer, New York, USA; 2004:806–814.
19. Smolensky P. Information processing in dynamical systems: foundations of harmony theory. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. MIT Press, Cambridge, MA, USA; 1986:194–281.
20. Bergstra J, Breuleux O, Bastien F, et al. Theano: a CPU and GPU math expression compiler. *Proc. of the 9th Python in Science Conference (SciPy)*, Austin, Texas, USA; June 2010:1–7.
21. Deep learning tutorial. DeepLearning 0.1 Documentation. (Available from: http://deeplearning.net/tutorial) [Accessed on July 26, 2016].
22. Sarikaya R, Hinton GE, Ramabhadran B. Deep belief nets for natural language call-routing. *Proc. of IEEE Intl. Conference in Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic; May 22-27, 2011:5680–5683.
23. Deng WW, Peng H. Research on a naive Bayesian based short message filtering system. *Proc. of IEEE Intl. Conference on Machine Learning and Cybernetics*, Dalian, China; August 13-16, 2006:1233–1237.
24. Bengio S, Mariethoz J. A statistical significance test for person authentication. *ODYS-2004*, Toledo, Spain; May 31-June 3, 2004:237–244.
25. Kim W, Stankovic MS, Johansson KH, Kim HJ. A distributed support vector machine learning over wireless sensor networks. *IEEE Trans Cybern*. 2015;45(11): 2599–2611.
26. Brocardo ML, Traore I. Continuous authentication using micro-messages. *Proc. of 12th Annual Intl. Conference on Privacy, Security and Trust (PST)*, Toronto, Canada; 2014:1–8.
27. Sanderson C, Guenter S. Short text authorship attribution via sequence kernels, Markov chains and author unmasking: an investigation. *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA:482–491.

How to cite this article: Brocardo ML, Traore I, Woungang I, Obaidat MS. Authorship verification using deep belief network systems. *Int J Commun Syst*. 2017;30:e3259. https://doi.org/10.1002/dac.3259