Author Identification via a Distributed Neural-Evolutionary Hybrid (DiNEH)

Sarp Aykent
Department of Computer Science
& Software Engineering
Auburn University
Auburn, AL, USA
sza0112@auburn.edu

Gerry Dozier
Department of Computer Science
& Software Engineering
Auburn University
Auburn, AL, USA
doziegv@auburn.edu

***Abstract*—In this paper, we propose a non-traditional Genetic & Evolutionary Feature Selection (GEFeS) method for Author Identification. This method evolves a distributed feature vector for each author and it's therefore distributed. We refer to this new approach as a distributed neural evolutionary hybrid (DiNEH). We compare the performance of DiNEH with a number of well-known Authorship Attribution Techniques (AATs) from the literature. DiNEH was able to outperform all of the AATs on one dataset and was second on the second dataset by a narrow margin.**

***Keywords*—authorship attribution, feature selection, genetic algorithms, distributed systems, GEFeS**

I. Introduction

Over the years, the field of Evolutionary Computation (EC) has seen development in a wide variety of successful distributed evolutionary computations (DECs). To date, DECs can be classified as: domain-based [1]–[3], function-based [4], and/or variable-based [2]. In domain-based DECs, the overall population of candidate solutions, P, is distributed across k processors where each processor receives P/k candidate solutions (individuals). Function-based DECs distribute the functions of evolutionary operators and processes (e.g., Crossover, Function Evaluation, Selection, etc.) across k processors while variable-based DECs distribute the chromosome across k processors. In this paper, we present a distributed neuro-evolutionary hybrid (DiNEH) approach that combines domain-based, function-based, and variable-based DEC. We compared the DiNEH with other Authorship Identification methods in different datasets. Our results show that DiNEH is an effective method for Author Identification [5].

The remainder of the paper is as follows. In Section II, we introduced some related work, in Section III, we introduce DiNEH, in Section IV, we describe the datasets used in our experiments. In Section V, we describe our experiments. In Section VI, we provide our results and in Section VII, we present our conclusions and future work.

978-1-7281-6861-6/20/$31.00 ©2020 IEEE

II. Related Work

In the literature, a variety of methods have been developed and used for Author Identification [5]. Luyckx et al. [6] used memory-based learning and discovered that increasing the number of authors within a dataset has an adverse effect on Author Identification accuracy while increasing the number of writing instances per author as well as the size of each writing instance increases the identification rate. Baron [7] compared the following methods: Decision Trees (PART and C4.5), Random Forest, k-Nearest Neighbors, Multilayer Perceptrons, Radial Basis Function Networks, and Naive Bayesian Classifiers for identifying authors. Their research suggests that cross-validation can result in over-estimation for Authorship Attribution.

In [8]–[12], the authors used similarity-based methods that utilized the concept of relative and cross-entropy. In [13]–[15], Support Vector Machines were used. Jockers and M. Witten [14] introduced the concept of introduced Nearest Shrunken Centroids (NSC) and Regularized Discriminant Analysis (RDA).

A wide variety of features have been used for increasing the effectiveness of Author Identification [8], [13], [16]. Some of these include n-grams [8], [17], bag-of-words [13], stylometry [18], to name just a few. Khomytska and Teslyuk [16] introduced phonological level features. A more complete overview of Author Identification analysis can be found in [5].

Convolutional Neural Networks (CNNs) for Authorship Attribution tasks are showing promising results [19], [20]. Kim [19] proposed a CNN architecture, utilized multiple convolutional filter sizes (3, 4, and 5), dropout rate, and max-over-time pooling (Collobert et al. [21]), for sentence classification tasks. The proposed approach has different variations with a randomly initialized model, static model, non-static model, and multichannel model that combines static and non-static models. The models, except the random model, were performing similarly. The performances of the three models were competitive against the other methods used. Shrestha et al. [20] used a different approach to identify the writers of the short texts. The authors used a sequence of character n-grams for the input and analyzed sensitive n-grams that contribute to classification for an author using Salient sections [22]. The proposed approach performed better than the state-of-the-art approach by Schwartz et al. [23].

III. Author Identification via DiNEH

In this section, we introduce GEFeS (Genetic & Evolutionary Feature Selection) which evolves a single feature mask to reduce the number of the features needed for Author Identification. GEFeS also increases the identification accuracy. After GEFeS is introduced we will introduce DiNEH which evolves a feature mask for every author within a dataset. DiNEH not only has superior performance but is also scalable on datasets with a large number of authors.

A. Single GEFeS Method

The GEFeS is based on the steady-state genetic algorithm [24] that evolves a population of 20 candidate feature masks. The initial population of candidate feature masks is generated using a binary standard uniform distribution function. Hence, each feature mask generated in the initial population is anticipated to utilize half of the total number of features. In each generation, two parents are selected via binary tournament selection, and one offspring feature mask is created using a uniform crossover with a mutation rate of 5%. The worst feature mask in the population was replaced with the new offspring feature mask.

A Linear Support Vector Machine (LSVM) is used for Author Identification. We use LSVMs because of their ability to cope with a large number of inputs and their applications for Authorship Identification in the literature [13]. The Scikit-learn library was used for the LSVM with the LinearSVC module. GEFeS calculates the accuracy of the classification using leave-one-out cross-validation.

In order to calculate the accuracy of a candidate feature mask, each writing sample is classified one-by-one. Each writing sample of the dataset classified by the LSVM results in a decision vector that has a number associated with each author in the dataset. The author with the highest score in the decision vector is selected as the author of that writing sample.

Three different feature sets used were Character Unigrams, Stylometry, and Bag-of-Words. The features that were used for Stylometry were similar to [18] and shown in Table I.

B. DiNEH

Scalability is a big concern for Authorship Identification tasks as examined in [25]. We propose a distributed method for feature selection. DiNEH consists of a set of author cells (one author cell for each author). Figure 1 provides a diagram of an author cell. In Figure 1, one can see that an author cell is composed of six components. The first two components, self and non-self, represent the dataset as a whole. However, for an author cell, the self-set contains only a writing sample for the associated author while the non-self set represents the writing samples of all the other authors. Therefore, an author cell represents a two-class identification problem. Each author cell has an LSVM that is trained using leave-one-out. Finally,

TABLE I
Stylometry Feature Set

Category	Description	Count
Length	number of words/characters in post	2
Vocabulary richness	Yule's K and frequency of *hapax legomena*, *dis legomena*, etc.	12
Word shape	frequency of words with different combination of upper- and lower-case letters.	5
Word length	frequency of words that have 1-20 characters.	20
Letters	frequency of *a* to *z*, ignoring case	26
Digits	frequency of *0* to *9*	10
Punctuation	frequency of *.?!,;:()”-’*	11
Special Characters	frequency of other special characters “˜@#$ˆ&*_+=[]{}\\|/<>	22
Function words	frequency of words like ’a’, ’about’, ’after’ etc.	320
Total		428

Note: The syntactic category pairs were omitted and different function words were used from [18].

each author cell has a GEFeS which evolves a population of a candidate feature mask.

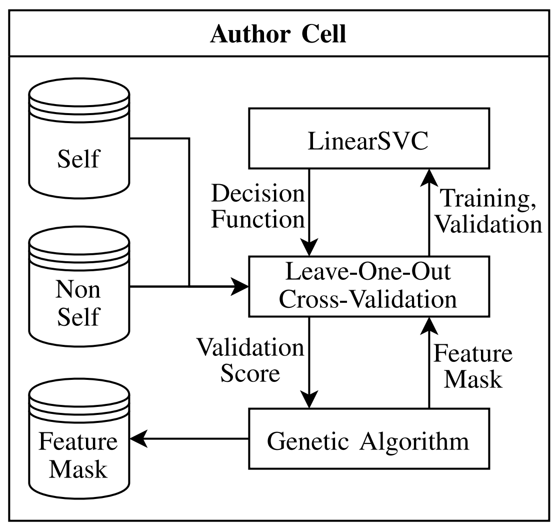

Fig. 1. Diagram of the feature selection process in author cells

DiNEH uses the author cell GEFeSs for evaluating the feature masks within the author cells. The fitness function for the GEFeS used in each author cell is as follows:

$$\sum_{i=1}^{|self|} f(LSVM(self_i)) \rightarrow f(x) = \begin{cases} x > 0 & |self| \\ x \leq 0 & x+1 \end{cases} \quad (1)$$

The classification of an author cell LSVM is performed in the same way with the single GEFeS approach. Each author cell classifies a writing sample using its feature mask. The resulting decision vector of the classification was a score for a given author cell.

DiNEH uses the same Character Unigrams and Stylometry feature sets introduced in as the single GEFeS mentioned earlier. As for the Bag-of-Words feature set, DiNEH utilizes author-based Bag-of-Words generated from each authors' writing sample independently. Thus, each author cell is trained with the Bag-of-Words associated with their writing samples only.

IV. Datasets

The datasets used in this paper are the subsets of the CASIS-1000 [26], [27] and Reuters_50_50 [28] dataset. The CASIS-1000 dataset composed of blog entries from 1,000 authors. There are four writing samples for each of the 1,000 authors. Hence, there are a total of 4,000 writing samples in this dataset. Each subset, namely CASIS-N, is the first n authors of the CASIS-1000 dataset. The Reuters_50_50 (C50) dataset is widely used for Authorship Attribution [29]. The C50 dataset composed of news articles of 50 authors. The dataset has 50 writing samples for each training and test set. Hence, there is a total of 2,500 writing samples for the training set and 2,500 writing samples for the test set. We used this dataset to compare the performance of DiNEH with other proposed approaches in [8]–[12].

Feature sizes for subsets of CASIS-1000 and C50 are shown in Table II for both author-based and global Bag-of-Words. It is clear that the difference between a single feature set and an average author feature set increases when the dataset gets larger.

TABLE II
Bag-of-Words Feature Sizes

Dataset	Single Feature Set[1]	Author Feature Sets[2]	
		Average	Maximum
CASIS-25	6,082	488.20	1,147
CASIS-50	9,984	523.16	1,403
CASIS-100	14,351	491.00	1,849
CASIS-1000	52,919	488.81	2,084
C50	35,202	2,177.09	4,539

The raw features collected from the writing samples are first preprocessed before the classification. The scikit-learn python library [30] is used for the preprocessing of the feature vectors produced by DiNEH. Term Frequency-Inverse Document Frequency (tf-idf) transformer and StandardScaler are first used to modify each feature vector, and then those feature vectors were normalized. The tf-idf transformer scales down the impact of tokens that occur very frequently in a given corpus. In order to standardize each feature, the StandardScaler subtracts the mean and then divides it by the variance per feature. Each feature vector is scaled into a unit vector using the normalization. This preprocessing process is utilized before using GEFeSs.

[1]Number of words used in the dataset.
[2]Number of words used by an author in the dataset.

V. Experiments

For Authorship Attribution, we first extracted the features described below from a set of writing samples. Then, we used preprocessing techniques that are explained in the preprocessing section. We conducted feature selection experiments with the Single GEFeS and DiNEH approaches on subsets CASIS-1000 dataset. Then, we compared DiNEH approach with the authorship attribution systems proposed in [8]–[12].

In Experiment I, the baselines for CASIS-25, 50, 100, and 1000 were computed using an LSVM without GEFeS for the Character Unigram, Stylometry, and Bag-of-Words features. In Experiments II, III, and IV, the performances of Single GEFeS and DiNEH were compared for CASIS-25, 50, and 100 datasets using Character Unigram and Stylometry[3].

In Experiment V, the performances of the four well-known author identification systems were compared with DiNEH methods on CASIS-25, 50, 100, and 1000 datasets. In Experiment VI, the performances of the four well-known author identification systems and six different CNNs were compared with DiNEH methods on the C50 dataset. The following CNN model variants were used for the comparison with DiNEH in Experiment VI.

- CNN$_{word_rand}$ [19]: All words are randomly initialized and then updated in the training.
- CNN$_{word_static}$ [19]: All words are initialized with a pre-trained word vector from word2vec. They are not updated during the training.
- CNN$_{word_non-static}$ [19]: Similar to word_static but pre-trained vectors are updated during the training.
- CNN$_{word_multichannel}$ [19]: This model is where word_static and word_non_static are treated as an individual channel. Only the word vectors in word_non_static are updated during the training.
- CNN$_{char_1}$ [20]: Similar to word_rand but character unigrams are used as input instead of words.
- CNN$_{char_2}$ [20]: Similar to word_rand but character bigrams are used as input instead of words.

VI. Results

The GEFeS approach was created to evaluate the initial population. In addition, 4,980 offspring feature masks were created, one per generation, for a total of 5,000 function evaluations. Let n be the number of writing samples. Each function evaluation calculated the fitness using leave-one-out cross-validation, and each sample was needed to be classified. Therefore, each classifier was trained with all writing samples excluding the test writing sample which was $(n-1)$. The

[3]The reason why Bag-of-Words was not used in Experiment II, III, and IV due to inability of Single GEFeS to scale as shown in Table II

computational effort for the Single GEFeS function evaluation ($\omega_{SingleGEFeS}$) is as follows, where n represents the number of writing samples:

$$\omega_{SingleGEFeS} = n^2 - n \quad (2)$$

The computational effort for DiNEH function evaluation (ω_{DiNEH}) was as follows, where n represents the number of writing samples and where m represents the number of writing instances per author:

$$\omega_{DiNEH} = \frac{n}{m} \times m \times (n-1) = n^2 - n \quad (3)$$

For the CASIS-1000 dataset $n = 4000$ and $m = 4$. For the C50 dataset $n = 2500$ and $m = 50$.

The computational efforts required for single-function evaluation in DiNEH and the Single GEFeS approach (Equations (2) and (3)) were equal. For a fair comparison, the same number of function evaluations were used for DiNEH and the Single GEFeS approach.

A. Experiment I

Table III shows the baseline results of the LSVMs without feature selection. The first column lists the dataset used for the Experiment I. The second, third and fourth columns list the baseline accuracies of the LSVMs using Character Unigrams, Stylometry, and Bag-of-Words respectively. In Table III, one can see that as the number of authors increases the accuracy decreases for all LSVMs. An unexpected result is that the Character Unigram LSVM outperformed the Stylometry LSVM for all four datasets. The Bag-of-Words LSVMs had the best overall performance on all four datasets.

TABLE III
PERFORMANCE BASELINE

Dataset	Unigram	Stylometry	Bag-of-Words
CASIS-25	65.00%	58.00%	96.00%
CASIS-50	51.00%	44.00%	90.00%
CASIS-100	46.00%	34.25%	84.50%
CASIS-1000	24.53%	19.65%	47.57%

B. Experiment II

Table IV shows the results of the LSVMs with Single GEFeS and DiNEH on CASIS-25 dataset 30 times with 5,000 function evaluations for each run. In Table IV, the first column lists the algorithm and feature set pair used in the experiment. The second column lists the equivalent classes of the methods that were determined by using ANOVA and Student's t-test. The third and fourth columns list the best and average accuracies, respectively. The fifth and sixth columns list the lowest and the average number of features used in the feature mask, respectively. One can see that DiNEH outperforms Single GEFeS in terms of an equivalence class, the best accuracy, average accuracy. DiNEH$_{uni}$ outperforms Single GEFeS$_{uni}$ in terms of the fewest number of features and the average number of features; however, the same conclusion was not observed for DiNEH$_{sty}$ and Single GEFeS$_{sty}$. In this case, Single GEFeS$_{sty}$ outperforms DiNEH$_{sty}$ in terms of the fewest number of features used while DiNEH$_{sty}$ outperforms Single GEFeS$_{sty}$ in terms of the average number of features used. In Table IV, DiNEH$_{sty}$ has the best overall performance in terms of accuracy.

TABLE IV
CASIS-25 RESULTS

Algorithm	EQ Class	Accuracy		Number of Features	
		Best	Average	Lowest	Average
Single GEFeS$_{uni}$	III	83.00%	80.70%	49.00	55.17
Single GEFeS$_{sty}$	III	83.00%	81.20%	205.00	219.37
DiNEH$_{uni}$	II	93.00%	92.10%	47.40	48.88
DiNEH$_{sty}$	I	99.00%	98.10%	211.64	213.49

C. Experiment III

Table V shows the results of the LSVMs with Single GEFeS and DiNEH on CASIS-50 dataset 30 times with 5000 function evaluations for each run. As similar to Experiment II, for Table V, the first column lists the algorithm and feature set pair used in the experiment. The second column lists the equivalent classes of the methods that were determined by using ANOVA and Student's t-test. The third and fourth columns list the best and average accuracies. The fifth and sixth columns list the lowest and the average number of features used in the feature mask. In Table V, one can see that DiNEH outperforms Single GEFeS in all performance measures (equivalence class, accuracy, number of features). The Single GEFeS$_{uni}$ outperformed Single GEFeS$_{sty}$ in terms of equivalence class and accuracy while DiNEH$_{sty}$ outperformed DiNEH$_{uni}$ in terms of equivalence class and accuracy. As in Experiment II, DiNEH$_{sty}$ outperformed the others in terms of equivalence class and accuracy.

TABLE V
CASIS-50 RESULTS

Algorithm	EQ Class	Accuracy		Number of Features	
		Best	Average	Lowest	Average
Single GEFeS$_{uni}$	III	66.00%	64.80%	53.00	59.93
Single GEFeS$_{sty}$	IV	62.00%	60.55%	214.00	230.93
DiNEH$_{uni}$	II	82.00%	80.90%	47.14	48.47
DiNEH$_{sty}$	I	95.50%	94.60%	210.38	213.53

D. Experiment IV

Table VI shows the results of the LSVMs with Single GEFeS and DiNEH on CASIS-100 dataset with 30 random runs with 5,000 function evaluations for each run. In Table VI, the first column lists the algorithm and feature set pair used in the experiment. The second column lists the equivalent classes of the methods that were determined by using ANOVA and Student's t-test. The third and fourth columns list the best and average accuracies, respectively. The fifth and sixth columns list the lowest and the average number of features used in the feature mask, respectively. In Table VI, as in Experiment III, DiNEH outperformed Single GEFeS in terms of an equivalence class, accuracy, and the number of features used. As we have noticed earlier, Single GEFeS$_{uni}$ once again outperformed Single GEFeS$_{sty}$ in terms of equivalence class and accuracy while DiNEH$_{sty}$ outperforms DiNEH$_{uni}$ in terms of equivalence class and accuracy. DiNEH$_{sty}$ had the best overall performance. Figure 2 provides the plot of the average accuracy for all 4 methods on the CASIS-25, 50, and 100 datasets.

TABLE VI
CASIS-100 RESULTS

Algorithm	EQ Class	Accuracy		Number of Features	
		Best	Average	Lowest	Average
Single GEFeS$_{uni}$	III	54.00%	53.34%	63.00	69.41
Single GEFeS$_{sty}$	IV	47.50%	46.04%	214.00	229.12
DiNEH$_{uni}$	II	78.00%	77.45%	48.34	48.81
DiNEH$_{sty}$	I	90.75%	87.85%	212.44	213.56

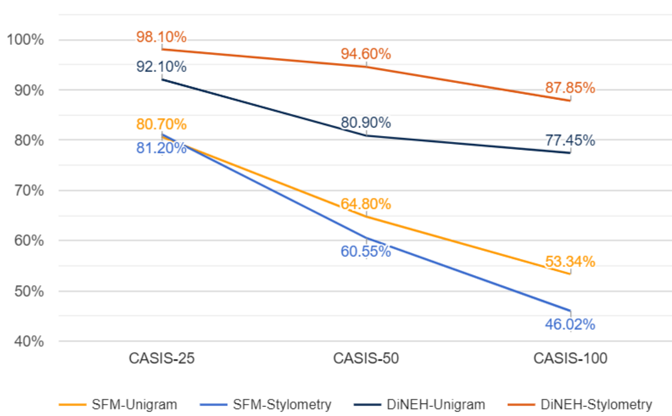

Fig. 2. Average accuracy of the Single GEFeS approach and DiNEH using Character Unigram and Stylometry feature sets.

E. Experiment V

In Table VII, we compared the DiNEH methods with four well-known author identification systems (AISs). The first column of Table VII lists the names of the methods that are being compared. Recorded in the second, third, fourth, and fifth columns are the accuracies of the five AISs and the average accuracies of DiNEHs on the CASIS-25, 50, 100, and 1000 datasets. As in Experiments II-IV, the DiNEHs was run for 30 times on the CASIS datasets. In terms of the AISs, Keselj has the best performance on all of the datasets. In terms of the DiNEHs, DiNEH$_{BoW}$ has the best performance on all of the datasets. Also, DiNEH$_{BoW}$ has the best performance overall on all of the datasets.

TABLE VII
PERFORMANCE COMPARISON

Algorithm	CASIS-25	CASIS-50	CASIS-100	CASIS-1000
Koppel [9]	87.00%	74.00%	69.25%	51.55%
Teahan [10]	89.00%	78.00%	65.75%	55.15%
Stamatatos [11]	10.00%	3.01%	3.00%	01.78%
Benedetto [12]	75.00%	65.00%	40.75%	28.00%
DiNEH$_{uni}$	92.10%	80.90%	77.45%	62.43%
DiNEH$_{sty}$	98.10%	94.80%	87.85%	79.40%
DiNEH$_{BoW}$	**99.00%**	**98.50%**	**95.75%**	**96.22%**

F. Experiment VI

In Table VIII, we compared DiNEH methods with four well-known author identification systems (AISs) and six different CNNs. The first and third columns of Table VIII list the names of the methods that are being compared. Recorded in the second, and fourth columns are the accuracies of the five AISs, six CNNs, and the average accuracies of DiNEHs on the C50 dataset. On the C50 dataset, the DiNEHs run a total of 30 times. In terms of the AISs, Teahan has the best performance on the C50 dataset. In terms of the DiNEHs, DiNEH$_{Baseline}$ has the best performance on the C50 dataset. In terms of the CNNs, CNN$_{word_multichannel}$ has the best performance on the C50 dataset. Teahan has the best performance followed by DiNEH$_{Baseline}$. In our analysis, we saw that reducing the number of features on the C50 dataset has an adversarial effect on the C50 dataset.

TABLE VIII
PERFORMANCE COMPARISON

Algorithm	C50
Koppel [9]	59.72%
Teahan [10]	**69.16%**
Stamatatos [11]	18.18%
Benedetto [12]	60.84%
CNN$_{word_rand}$ [19]	60.32%
CNN$_{word_static}$ [19]	65.60%
CNN$_{word_non_static}$ [19]	66.48%
CNN$_{word_multichannel}$ [19]	67.28%
CNN$_{char_1}$ [20]	60.60%
CNN$_{char_2}$ [20]	64.00%
DiNEH$_{uni}$	44.28%
DiNEH$_{sty}$	48.88%
DiNEH$_{BoW}$	64.45%
DiNEH$_{Baseline}$	68.76%

VII. CONCLUSIONS & FUTURE WORK

In this paper, we introduced a distributed neuro-evolutionary hybrid for Author Identification which was referred to as DiNEH. Our results show that DiNEH scales well to datasets with larger numbers of authors by evolving a distributed feature mask. We compared the DiNEH with four well-known AISs and six CNN variants. DiNEH had the best performance on all but one dataset (C50) both of which had 50 or fewer authors. Our future work will be devoted towards developing DiNEHs that combine Bag-of-Words, Stylometry, Character N-grams, and other types of features.

References

[1] T. C. Belding, "The distributed genetic algorithm revisited," *arXiv preprint adap-org/9504007*, 1995.

[2] M. A. Potter, K. A. De Jong, and J. J. Grefenstette, "A coevolutionary approach to learning sequential decision rules." in *ICGA*, 1995, pp. 366–372.

[3] R. Tanese, "Distributed genetic algorithms for function optimization," 1989.

[4] D. E. Brown, "A parallel genetic heuristic for the quadratic assignment problem," in *Proc. 3rd ICGA*, 1989, pp. 406–415.

[5] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, and D. Woodard, "Surveying stylometry techniques and applications," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 86, 2018.

[6] K. Luyckx and W. Daelemans, "The effect of author set size and data size in authorship attribution," *Literary and linguistic Computing*, vol. 26, no. 1, pp. 35–55, 2011.

[7] G. Baron, "Comparison of cross-validation and test sets approaches to evaluation of classifiers in authorship attribution domain," in *International Symposium on Computer and Information Sciences*. Springer, 2016, pp. 81–89.

[8] V. Kešelj, F. Peng, N. Cercone, and C. Thomas, "N-gram-based author profiles for authorship attribution," in *Proceedings of the conference pacific association for computational linguistics, PACLING*, vol. 3. sn, 2003, pp. 255–264.

[9] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resour. Eval.*, vol. 45, no. 1, pp. 83–94, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1007/s10579-009-9111-2

[10] W. J. Teahan and D. J. Harper, "Using compression-based language models for text categorization," in *Language modeling for information retrieval*. Springer, 2003, pp. 141–165.

[11] E. Stamatatos, "Author identification using imbalanced and limited training texts," in *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*. IEEE, 2007, pp. 237–241.

[12] D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping," *Physical Review Letters*, vol. 88, no. 4, p. 048702, 2002.

[13] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Applied intelligence*, vol. 19, no. 1-2, pp. 109–123, 2003.

[14] M. Jockers and D. M. Witten, "A comparative study of machine learning methods for authorship attribution," *LLC*, vol. 25, pp. 215–223, 05 2010.

[15] J. Gaston, M. Narayanan, G. Dozier, D. L. Cothran, C. Arms-Chavez, M. Rossi, M. C. King, and J. Xu, "Authorship attribution vs. adversarial authorship from a liwc and sentiment analysis perspective," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2018, pp. 920–927.

[16] I. Khomytska and V. Teslyuk, "Authorship attribution by differentiation of phonostatistical structures of styles," in *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, vol. 2. IEEE, 2018, pp. 5–8.

[17] U. Sapkota, S. Bethard, M. Montes, and T. Solorio, "Not all character n-grams are created equal: A study in authorship attribution," in *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2015, pp. 93–102.

[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 *2012 IEEE Symposium on Security and Privacy*, May 2012, pp. 300–314.

[19] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. [Online]. Available: https://www.aclweb.org/anthology/D14-1181

[20] P. Shrestha, S. Sierra, F. González, M. Montes, P. Rosso, and T. Solorio, "Convolutional neural networks for authorship attribution of short texts," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 669–674. [Online]. Available: https://www.aclweb.org/anthology/E17-2106

[21] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[22] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and understanding neural models in NLP," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 681–691. [Online]. Available: https://www.aclweb.org/anthology/N16-1082

[23] R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel, "Authorship attribution of micro-messages," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1880–1891.

[24] L. Davis, Ed., *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

[25] K. Luyckx, *Scalability issues in authorship attribution*. ASP/VUBPRESS/UPA, 2011.

[26] C. Faust, G. Dozier, J. Xu, and M. C. King, "Adversarial authorship, interactive evolutionary hill-climbing, and author caat-iii," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2017, pp. 1–8.

[27] N. Mack, J. Bowers, H. Williams, G. Dozier, and J. Shelton, "The best way to a strong defense is a strong offense: Mitigating deanonymization attacks via iterative language translation," *International Journal of Machine Learning and Computing*, vol. 5, no. 5, p. 409, 2015.

[28] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[29] J. Houvardas and E. Stamatatos, "N-gram feature selection for authorship identification," in *Artificial Intelligence: Methodology, Systems, and Applications*, J. Euzenat and J. Domingue, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 77–86.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.