Large-Scale and Language-Oblivious Code Authorship Identification

Mohammed Abuhamad *Inha University, Incheon, South Korea*
Tamer AbuHmed *Inha University, Incheon, South Korea*
Aziz Mohaisen *University of Central Florida, Orlando, USA*
DaeHun Nyang *Inha University, Incheon, South Korea*

CCS '18: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security Pages 101-114.
Toronto, Canada — October 15 - 19, 2018
ISBN: 978-1-4503-5693-0
doi>10.1145/3243734.3243738
link: https://dl.acm.org/citation.cfm?id=3243738

Abstract:
Efficient extraction of code authorship attributes is key for successful identification. However, the extraction of such attributes is very challenging, due to various programming language specifics, the limited number of available code samples per author, and the average code lines per file, among others. To this end, this work proposes a Deep Learning-based Code Authorship Identification System (DL-CAIS) for code authorship attribution that facilitates large-scale, language-oblivious, and obfuscation-resilient code authorship identification. The deep learning architecture adopted in this work includes TF-IDF-based deep representation using multiple Recurrent Neural Network (RNN) layers and fully-connected layers dedicated to authorship attribution learning. The deep representation then feeds into a random forest classifier for scalability to de-anonymize the author. Comprehensive experiments are conducted to evaluate DL-CAIS over the entire Google Code Jam (GCJ) dataset across all years (from 2008 to 2016) and over real-world code samples from 1987 public repositories on GitHub. The results of our work show the high accuracy despite requiring a smaller number of files per author. Namely, we achieve an accuracy of 96% when experimenting with 1,600 authors for GCJ, and 94.38% for the real-world dataset for 745 C programmers. Our system also allows us to identify 8,903 authors, the largest-scale dataset used by far, with an accuracy of 92.3%. Moreover, our technique is resilient to language-specifics, and thus it can identify authors of four programming languages (e.g. C, C++, Java, and Python), and authors writing in mixed languages (e.g. Java/C++, Python/C++). Finally, our system is resistant to sophisticated obfuscation (e.g. using C Tigress) with an accuracy of 93.42% for a set of 120 authors.

DL-CAIS: Deep Learning-based Code Authorship Identification System

Mohammed Abuhamad
University of Central Florida
abuhamad@knights.ucf.edu

Tamer AbuHmed
Inha University
tamer@inha.ac.kr

Aziz Mohaisen
University of Central Florida
mohaisen@ucf.edu

DaeHun Nyang
Inha University
nyang@inha.ac.kr

***Abstract*—Code authorship identification is useful in many software forensics contexts. Successful code authorship identification relies on efficient extraction of authorship attributes. This work proposes *DL-CAIS*, a Deep Learning-based Code Authorship Identification System, for large-scale, language-oblivious, and obfuscation-resilient code authorship identification. The proposed system includes learning TF-IDF-based deep representations of code authorship attributions using recurrent neural network. The deep representations are used to construct a random forest classifier for scalable and robust de-anonymization of programmers. We evaluate *DL-CAIS* using the entire Google Code Jam (GCJ) dataset across all years (from 2008 to 2016) and using public real-world code repositories from GitHub. The results show that the proposed system achieves an accuracy of 92.3% for identifying 8,903 authors for GCJ and 94.38% for the real-world dataset for 745 C programmers. Moreover, the results show that *DL-CAIS* is resilient to language-specifics, temporal effects, and obfuscation.**

I. Introduction

Source code authorship identification is the process of code writer identification by associating a programmer to a given code based on the programmer's distinctive stylometric features. Authorship identification for textual documents is a well-established field that has attracted big attention. Identifying programmers of source code can be more difficult and different from authorship identification of natural language text. This basic difficulties are driven from the inherent inflexibility of the written code expressions established by the syntax rules of compilers. Recently, a growing attention has been given to provide robust and scalable authorship identification for software. Being able to identify code authors is both a risk and a desirable feature. On the one hand, code authorship identification poses a privacy risk for programmers who wish to remain anonymous, including contributors to open-source projects, activists, and programmers who conduct programming activities on the side. On the other hand, code authorship identification is useful for software forensics and security analysts, especially for identifying malicious code programmers. Moreover, authorship identification of source code is helpful with plagiarism detection [1], authorship disputes [4], copyright infringement [2], and code integrity investigations [3].

NDSS '16, 21-24 February 2016, San Diego, CA, USA

http://dx.doi.org/10.14722/ndss.2016.23xxx

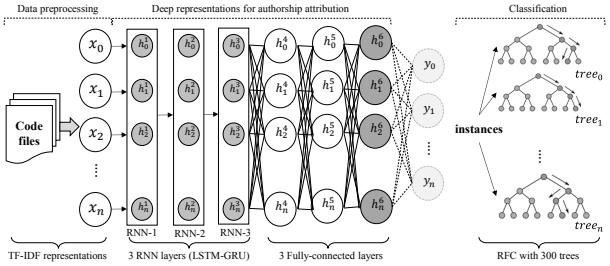
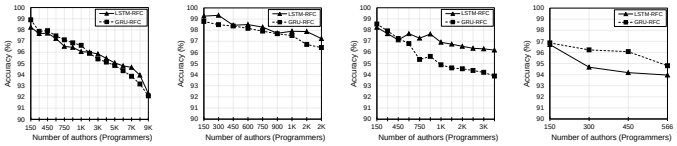


Fig. 1: A high-level illustration of the Deep Learning-based Code Authorship Identification System (DL-CAIS).

The problem of code author identification is challenging, and faces several obstacles that prevent the development of practical identification mechanisms. First, programming "style" of programmers continuously evolves over time. Second, the programming style of programmers varies from language to another. Third, while it is sometimes possible to obtain the source code of programs, sometimes it is not, and the source code is occasionally obfuscated by automatic tools, preventing their recognition.

This work contributes to code authorship identification in multiple directions as follows: First, we design a feature learning architecture using recurrent neural network (RNN) to enable the extraction of high quality and distinctive code authorship attributes. The deep representations of code authorship attributes are learned by feeding code samples presented by the TF-IDF (Term Frequency-Inverse Document Frequency) to the RNN architecture. Thus, our approach does not require a prior knowledge of any specific programming language. Second, we experimentally conduct a large scale code authorship identification and demonstrate that our technique can handle a large number of programmers (8,903 programmers) while maintaining a high accuracy (92.3%). Third, we show that our approach is oblivious to language specifics when using a dataset of authors writing in multiple languages. We based our assessment on an analysis over four individual programming languages (namely, C++, C, Java, and Python). Fourth, we investigate the effect of obfuscation methods on the authorship identification and show that our approach is resilient to both simple off-the-shelf obfuscators, such as Stunnix, and more sophisticated obfuscators, such as Tigress. Finally, we examine our approach on real-world datasets and achieve 95.21% and 94.38% of accuracy for datasets of 142 C++ programmers and 745 C programmers, respectively.



(a) C++ (b) Java (c) Python (d) C

Fig. 2: Accuracy of authorship identification of programmers with seven sample code files per programmer in C++. Java, Python, and C languages.

TABLE I: The accuracy of authorship identification using models trained on data from 2014 and tested on data from 2015 and 2016

	# Authors	LSTM-RFC	GRU-RFC
C++	292	97.65	96.43
Python	44	100	100
Java	50	100	100

II. DL-CAIS: Deep Learning-based Code Authorship Identification System

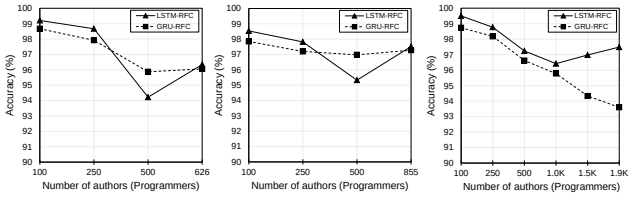
Our approach for large-scale code authorship identification has three phases: preprocessing, representation through learning, and classification. To identify authors, we need a scalable classifier that can accommodate a large number of programmers. However, the deep learning architecture alone does not give us a good accuracy (e.g., 86.2% accuracy for 1,000 programmers). Instead of using the softmax classifier of the deep learning architecture, we use RFC for the classification, and by providing the deep representation of TF-IDF as an input. RFC is known to be scalable, and our target dataset has more than 8,000 authors (or classes) to be identified. Such a large dataset can benefit from the capability of RFC. Our authorship identifier is built by feeding a TF-IDF-based deep representation extracted by RNN and then classifying the representation by RFC. This hybrid approach allows us to take advantage of both deep representation's distinguishing attribute extraction capability and RFC's large scale classification capability.

III. Experiments and Results

Large-scale Authorship Identification. Figure 2 shows the results of *DL-CAIS* using the dataset of all programmers with seven code samples for four different programming languages. Figure 2(a) shows that the LSTM-RFC achieves an accuracy of 92.3% for 8,903 C++ programmers. Figure 2(b) shows that LSTM-RFC achieves an accuracy of 97.24% for 1,952 Java programmers. Figure 2(c) shows an accuracy of 96.2% when using LSTM-RFC for 3,458 Python programmers. Finally, Figure 2(d) shows the result for C programmers, where LSTM-RFC achieves an accuracy of 93.96% for 566 C programmers.

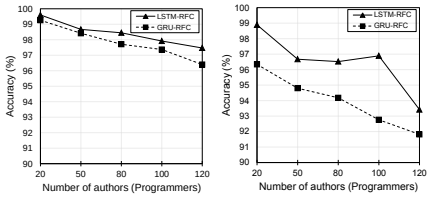
Effect of Temporal Changes. We trained our models (LSTM-RFC and GRU-RFC) on data from the year 2014 and used the data from 2015 and 2016 as a testing set. As a result, Table I shows that our approach of code authorship identification is resilient to temporal changes in the coding style of programmers as it achieves 100% accuracy for both Python and Java languages and 97.65% for the 292 C++ programmers.

Identification with Mixed Languages. Figure 3 shows the accuracy of our approach with three datasets: C++/C, C++/Java, and C++/Python. Figure 3(a) shows an accuracy of 96.34% for a dataset of 626 C++/C programmers with LSTM-RFC, and its accuracy of 97.52% when used with LSTM-RFC on 855 C++/Java programmers, as illustrated in Figure 3(b). For the C++/Python dataset, Figure 3(c) shows that our approach provides an accuracy of 97.49% for 1,879 programmers.



(a) C/C++ (b) Java/C++ (c) Python/C++

Fig. 3: The accuracy of the authorship identification of programmers with sample codes of two programming languages.



(a) C++ obfuscated with Stunnix. (b) C code obfuscated with Tigress.

Fig. 4: The accuracy of authorship identification with obfuscated source code.

Identification in Obfuscated Domain. Figure 4(a) shows the accuracy achieved using our approach on different Stunnix-obfuscated C++ datasets. Figure 4(b) shows the achieved accuracy on different Tigress-obfuscated C datasets ranging from 20 to 120 authors using two different RNN units. The results shows that our approach is resilient to different obfuscation.

IV. Conclusion

This work contributes to the extension of deep learning applications by utilizing deep representations in authorship attribution. In particular, we examined the learning process of large-scale code authorship attribution using RNN, a more efficient and resilient approach to language-specifics, number of code files available per author, and code obfuscation.

Acknowledgement. This work was supported by NRF-2016K1A1A2912757 (Global Research Lab Initiative), and a collaborative seed grant from the Florida Cybersecurity Center (FC2).

References

[1] S. Burrows, S. M. M. Tahaghoghi, and J. Zobel, "Efficient plagiarism detection for large code repositories," *Softw. Pract. Exper.*, vol. 37, no. 2, pp. 151–175, Feb. 2007.

[2] G. Frantzeskou, E. Stamatatos, S. Gritzalis, C. E. Chaski, and B. S. Howald, "Identifying authorship by byte-level n-grams: The source code author profile (scap) method," *International Journal of Digital Evidence*, vol. 6, no. 1, pp. 1–18, 2007.

[3] C. H. Malin, E. Casey, and J. M. Aquilina, *Malware forensics: investigating and analyzing malicious code*. Syngress, 2008.

[4] L. J. Wilcox, "Authorship: the coin of the realm, the source of complaints," *The Journal of the American Medical Association*, vol. 280, no. 3, pp. 216–217, 1998.

DL-CAIS: Deep Learning-based Code Authorship Identification System

Mohammed AbuHamad abuhamad@knights.ucf.edu
Tamer Abuhmed tamer@inha.ac.kr
Aziz Mohaisen mohaisen@ucf.edu
DaeHun Nyang nyang@inha.ac.kr

INTRODUCTION

Code Authorship Identification

- **Code authorship identification**: Source code authorship identification is the process of code writer identification by associating a programmer to a given code based on the programmer's distinctive stylometric features.
- **Code authorship identification**: advancement in code authorship identification research has enabled successful applications in forensic contexts including ghostwriting detection, copyright dispute settlements, and other code analysis applications.
- This work proposes DL-CAIS, a Deep Learning-based Code Authorship Identification System, for large-scale, language-oblivious, and obfuscation-resilient code authorship identification.

Challenges

- **Temporal effect:** programmers' style change and evolve.
- **Language-specifics:** programming languages have different features.
- **Obfuscation:** sometimes only obfuscated code is available.

DL-CAIS

- **Data collection and preprocessing**:
  - Dataset: Google Code Jam Competition (2008-2016) .
  - Four programming languages (c, C++, Java, and Python)
  - Presented in TF-IDF
- **Code authorship attribution**: using a deep learning (RNN) model high-quality authorship attribution are extracted.
- **Code authorship identification**: using Random Forest Classifier, the system achieved state-of-the-art results in different settings.

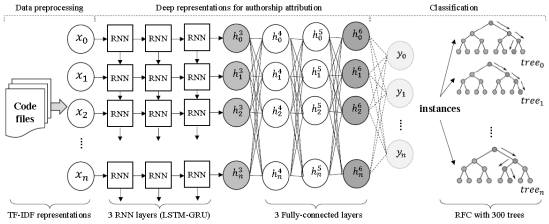
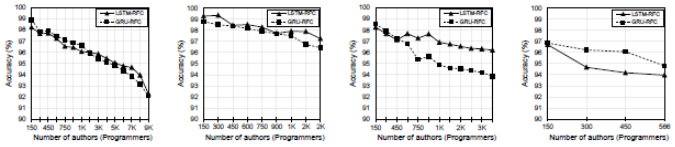


Figure 1: DL-CAIS: Deep Learning-based Code Authorship Identification System

EXPERIMENTS and RESULTS

- ***Large-scale code authorship identification:*** For four programming languages C, C++, Java and Python and with 7 samples per programmer.



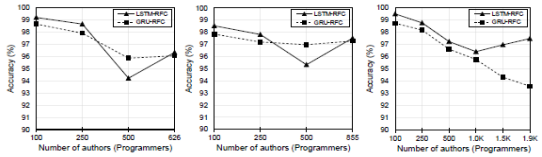
(a) C++ (b) Java (c) Python (d) C

Figure 2: Large-scale code authorship identification

- ***Effect of Temporal Changes:*** *There's an effect but our approach is resilient to such effects. The table* The accuracy of authorship identification using models trained on data from 2014 and tested on data from 2015 and 2016

	# Authors	LSTM-RFC	GRU-RFC
C++	292	97.65	96.43
Python	44	100	100
Java	50	100	100

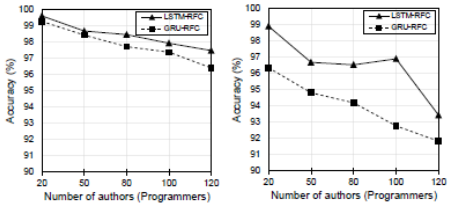
- ***Identification with Mixed Languages:*** The figure shows The accuracy of the authorship identification of programmers with sample codes of two programming languages.



(a) C/C++ (b) Java/C++ (c) Python/C++

Figure 3: Identification of authors writing in multiple languages

- ***Identification in Obfuscated Domain.*** Figure 4 shows the accuracy achieved using our approach on different obfuscated datasets with authors from 20 to 120 using two different RNN units. The results shows that our approach is resilient to different obfuscation.



(a) C++ obfuscated with Stunnix. (b) C code obfuscated with Tigress.

Figure 4: Identification in obfuscation domain using Stunnix and Tigress obfuscation tools

Conclusion and Future work

- This work was dedicated to facilitating the learning process of large-scale code authorship attribution using RNN, which is more efficient and more resilient to language-specifics, temporal effects, and code obfuscation.
- ***Future Work***
  - Binary code authorship identification.
  - Obfuscated binary code authorship identification.
  - Code multi-authors identification.
- ***Acknowledgement:*** This work was supported by NRF-2016K1A1A2912757 (Global Research Lab Initiative), and a collaborative seed grant from the Florida Cybersecurity Center (FC2).