Motivation

- It enable attribution of successful attacks from code left behind on an infected system, or aid in resolving copyright, copyleft, and plagiarism issues in the programming fields.
- Code attribution may be helpful in a forensic context, such as detection of ghostwriting, a form of plagiarism, and investigation of copyright disputes.
- It might also give us clues about the identity of malware authors.
- Programmer De-Anonymization.
- Ghostwriting Detection.
- Software Forensic.
- Copyright Investigation.
- Authorship Verification.

Process

- Three types of features syntactic,layout-based and lexical.
- First, we use syntactic features for code stylometry. Extracting such features requires parsing of incomplete source code using a fuzzy parser to generate an abstract syntax tree.
- A bagging (portmanteau of “bootstrap aggregating”) classifier - random forest was used to attribute programmers to source code.
- Code stylometry feature set (CSFS).
- Lexical and layout features are obtained from source code while the syntactic features can only be obtained from AST’s.
- The AST node bigrams are the most discriminating features of all.
- Classification.
- Random Forest Classification.
- IG-CSFS.
- Scaling.
- Training Data and Features.
- Relaxed Classification.

Features

Lexical Features:

- wordUnigramTF → Term frequency of word unigrams in source code
- ln(numkeyword/length) → log of number of occurrences of keyword divided by file length in character.
- ln(numTernary/length)
- ln(numTokens/length)
- ln(numComments/length)
- ln(numLiterals/length)
- ln(numFunctions/length)
- ln(numMacros/length)
- nesting Depth → highest degree to which control statements and loops are nested within each other.
- Branching factor → branching factor of the tree formed by converting code blocks of files into nodes.
- AvgParams → the average number of parameters among all functions.
- StdDevNumParams → standard deviation of number of parameters.
- AvgLineLength → average length of each line
- stdDevLineLength

Layout Features:

- ln(numTabs/length)
- ln(numSpaces/length)
- ln(numEmptyLines/length)
- whiteSpaceRatio → ratio between white space and non white space character.
- NewLineBeforeOpenBrace
- tabsLeadLines

Syntactic Features:

- MaxDepthASTNode → maximum depth of an AST node.
- ASTNodeBigramsTF → term frequency AST node bigrams
- ASTNodeTypesTF → term frequencyof 58 possible AST node type excluding leaves
- ASTNodeTypeTFIDF → term frequency inverse document frequency of 58 possible AST node type excluding leaves.
- ASTNodeTypeAvgDep → average depth of 58 possible AST node excluding leaves.
- CppKeywords → terms frequency of 84 c++ keywords.
- CodeInASTLeavesTF → term frequency of code unigrams in AST leaves.
- CodeInASTLeavesTFIDF → term frequency inverse document frequency of code unigrams in AST leaves.
- CodeInASTLeavesAvgDep → average depth of code unigrams in AST leaves.