future internet

MDPI

Article

Authorship Attribution of Social Media and Literary Russian-Language Texts Using Machine Learning Methods and Feature Selection

Anastasia Fedotova, Aleksandr Romanov, Anna Kurtukova * and Alexander Shelupanov

Department of Security, Tomsk State University of Control Systems and Radioelectronics, 634050 Tomsk, Russia; afedotowaa@icloud.com (A.F.); alexx.romanov@gmail.com (A.R.); saa@tusur.ru (A.S.)
* Correspondence: av.kurtukova@gmail.com

Abstract: Authorship attribution is one of the important fields of natural language processing (NLP). Its popularity is due to the relevance of implementing solutions for information security, as well as copyright protection, various linguistic studies, in particular, researches of social networks. The article is a continuation of the series of studies aimed at the identification of the Russian-language text's author and reducing the required text volume. The focus of the study was aimed at the attribution of textual data created as a product of human online activity. The effectiveness of the models was evaluated on the two Russian-language datasets: literary texts and short comments from users of social networks. Classical machine learning (ML) algorithms, popular neural networks (NN) architectures, and their hybrids, including convolutional neural network (CNN), networks with long short-term memory (LSTM), Bidirectional Encoder Representations from Transformers (BERT), and fastText, that have not been used in previous studies, were applied to solve the problem. A particular experiment was devoted to the selection of informative features using genetic algorithms (GA) and evaluation of the classifier trained on the optimal feature space. Using fastText or a combination of support vector machine (SVM) with GA reduced the time costs by half in comparison with deep NNs with comparable accuracy. The average accuracy for literary texts was 80.4% using SVM combined with GA, 82.3% using deep NNs, and 82.1% using fastText. For social media comments, results were 66.3%, 73.2%, and 68.1%, respectively.

Keywords: authorship identification; natural language processing; machine learning; deep neural networks; fastText; support vector machine; genetic algorithms

Citation: Fedotova, A.; Romanov, A.; Kurtukova, A.; Shelupanov, A. Authorship Attribution of Social Media and Literary Russian-Language Texts Using Machine Learning Methods and Feature Selection. *Future Internet* **2022**, *14*, 4. https://doi.org/10.3390/fi14010004

Academic Editor: Bennett Kleinberg

Received: 25 November 2021
Accepted: 21 December 2021
Published: 22 December 2021

1. Introduction

In the XXI century, the Internet has become a communication space of the information society. Everyone has an opportunity to express his opinions, share thoughts, and receive feedback from followers. The variety of available online texts (e.g., emails, blogs, correspondence in social networks, and messengers) and possibility to write them anonymously indicate a wide range of applications of methods for determining the author of the text [1]. The number of search results of the query "Authorship attribution" in Google Scholar for 2021 is 23,000, for 2017–2021—60,000. These facts indicate the interest of researchers in this issue and characterize this task as relevant. In the regard, the publication of materials on behalf of famous persons from hacked accounts deserves special mention. The text sent by an attacker on behalf of a celebrity is capable to become viral and quotable immediate. Additionally, in the case of the documents containing calls for actions prohibited by law, these publications can negatively influence the public mood. Online media are also actively used by attackers. The fact that usually, users are not required to provide true information about themselves—name, age, gender, and address [2] allows attackers to send and provide anonymous dissemination of antisocial information, threats, and advocacy of terrorism, online fraudulent activities. Establishing the identity of the criminal becomes a non-trivial task, but the authorship attribution methods allow identifying the creator of the text.

Nowadays, authorship attribution results are applied in forensic, information security, computational linguistics, and copyright protection. Authorship attribution can be categorized into four groups:

1. Authorship identification. The task comes down to the multi-class single-labeled text classification problem. In the classic version, the closed attribution problem is considered, that is, the true author of the disputed text is present in a set of candidate authors. For more complicated cases, authorship identification techniques are used to solve open attribution problems. The attribution system should establish the absence or presence of the true author in the list of candidate authors and determine the true author.
2. Authorship verification. The authorship verification task comes down to the problem of one-class classification. The essence of the task is solving the question of whether two documents were written by the same person or not.
3. Authorship clustering. This is the most difficult task when there are many texts and it is necessary to group them by the author, but there is no information about the number of candidate authors.
4. Authorship profiling. Classification by additional author's characteristics such as gender, age, educational level, etc.

Novel trends of authorship attribution are associated with the identification of the author of the texts that were distorted by anonymization methods [3], as well as the distinction between natural and generated texts [4].

The study considers the problem of closed attribution. The task is set as follows: there are text fragments belonging to a finite set of authors; the authorship of some fragments is established; other anonymous texts belong to one of the candidates, but whose exactly is unknown. It is necessary to determine the belonging of the disputed fragments to the true author. In this case, the authorship identification problem comes down to the multiclass classification. The set of authors will be a set of classes, and texts whose authors are known will be the training set. The goal is to classify disputed texts with the best accuracy.

The scientific novelty of the research lies in the application of authorship attribution methods that were not used for Russian-language texts earlier: fastText, the combination of support vector machine (SVM) with genetic algorithm (GA) for feature selection, and comparison of these methods with convolutional neural network (CNN), networks with long short-term memory (LSTM), their hybrids, bidirectional encoder representations from Transformers (BERT), *K*-Nearest Neighbors (KNN), decision tree (DT), random forest (RF), logistic regression (LR), and Naïve Bayes (NB). It should be noted that these methods were used not only for literary texts but also for short comments of users of social networks. Previously, these methods have not been applied to short Russian-language texts.

2. Related Works

There are many studies devoted to establishing authorship of a natural text and source code, the gender and age of the author, determining the sentiment of the text for forensic and social science purposes [5,6]. The early paper [1] provides a detailed review of 2015–2020 studies aimed at determining the author of a text, including approaches based on deep neural networks (NN), classical machine learning (ML) methods, and aspect analysis.

This section examines the novel studies for Russian and other languages, as well as the works devoted to feature selection. The majority of the publications used various features of the writing style [7], including lexical, syntactic, structural, and specific ones to the genre and subject of a document. Moreover, the features were used both separately and in the form of a single vector.

Classical ML methods imply the use of a set of informative features. Deep expert knowledge and special methods of feature filtering are required to obtain the set of features. This fact complicates the problem. There are two main solutions. Firstly, deep neural networks (NN) are capable of finding informative features and dependencies in data automatically. Secondly, using the feature selection methods, including genetic algorithms

(GA) [8] to select informative features. As such, it is possible to obtain an optimal subset of all possible features for classification.

2.1. Related Works on Classical Machine Learning Methods and Deep Neural Networks for Authorship Attribution

The purpose of the study [9] was to determine the authorship of Ukrainian journalistic-style texts using NN. The texts of three authors published in the Ukrainian Week and Weekly Mirror Ukraine during 2015–2019 (50 texts per author) were used to try out the method. Vectorization was an important part of the research. Among the vectorization methods, such as ASCII Converter, Simple Vectorizer, and Hashing Vectorizer, the last one proved to be the most efficient. 10-fold cross-validation was used to split the data into test and training sets. Results were presented only for binary classification using architectures [10, relu] [10, relu] [1, sigmoid]. Accuracy was 86%, 95%, and 96%, depending on the author.

In [10] presented a Topic Drift Model (TDM) capable of tracking the dynamics of changes in the author's style. It was also noted that this model is sensitive to the order of the words and chronology of a text. The author's model was based on the Gaussian Mixture Neural Topic Model (GMNTM). The main idea of TDM is to find similarities between words embeddings. The similarity was represented by the text's scalar product. Topics of texts can change over time, so the topics of each author were represented by a sequence of vectors, where a bias was controlled by a similarity of vectors. The TDM model is also able to accommodate the fact that co-authors usually share common topics, so their vectors of interest are positively correlated. In addition to TDM, SVM, Latent Dirichlet Allocation + Hellinger (LDA-H), and Time-aware Feature Sampling (TFS) were used for comparison. PAN'11 emails data (9337 documents by 72 different authors), IMDb62 movie reviews (62,000 movie reviews by 62 users), and blog data (678,161 blog posts by 19,320 authors from blogger.com) were used as datasets. All texts were written in English. For PAN'11 dataset, 90% of data were used for training and 10% were used for validation. For IMDb62 and Blog datasets, 80% of documents of each author were used as the training data, 10% of documents of each author as the validation data, and the remaining 10% for testing. For IMDb62 dataset, the best accuracy increased from 92.1% (LDA-H) to 93.8% (TDM), on the PAN'11 dataset, the best accuracies were 51.6% and 51.2%. For the Blog dataset, the best accuracy was provided by TDM (32%). All methods showed low accuracy (no more than 32%) on the Blog dataset since this dataset contains many classes, which complicates classification.

The authors of [11] proposed two algorithms: General Impostors (GI) and Ranking-Based Impostors (RBI). The study used all datasets released in PAN-2014 and PAN-2015 shared tasks. As a result, the dataset contained cross-topic (similar and mixed), cross-genre (essay, reviews, novels, articles) and cross-language (Dutch, English, German, Greek, Spanish languages) texts (1661 texts in total). To extract features, the authors used clustering based on the expectation-maximization algorithm [12]. The aggregation function used in GI and RBI was selected for each dataset. In almost all cases RBI was more effective than GI. AUC scores for PBI were 70.9–97.5%, for GI were 66.1–78.5%. It is also noted that RBI reached its maximum performance at 2700 words (45% of the average length), while BI performed best with a maximum length (about 6000 words). RBI demonstrated better results than BI for relatively short texts, while PBI was better for long texts.

The Authorship Verification task was solved by the authors of [13] using an improved method based on implicitly defined features. The essence of the method lied in the fact that any sequence of characters in the text could be a potential feature of an author's style. The disadvantage was a strong dependence on a topic of a text, which lead to a low-quality classification. To solve this problem, the authors used the POSNoise (POS-Tag-based Noise smoothing) preprocessing method, which effectively masks thematic content in the given text. Data for research consisted of 6320 texts written by 3097 authors from seven well-known datasets: Gutenberg Corpus, ACL Anthology Corpus, Perverted Justice Corpus,

The Telegraph Corpus, The Apricity Corpus, and Reddit Corpus. Empirical estimation of the TextDistortion method [14] showed that integration with POSNoise led to better results in 34 of 42 cases with an increase in accuracy to 10%.

In [15] the authors proposed the supervised machine learning framework incorporating stylometric features—an approach based on the study of changes in writing styles between 50 authors. The accuracy improves significantly due to the common use of certain linguistic stylometric features with text. The final corpus consisted of 50 authors, each having 100 texts/news from the RCV1 (Reuters Corpus Volume 1) dataset. Accuracy of 81.6% and Kappa statistic of 0.81 were achieved on the holdout test set using LibLINEAR SVM. Based on obtaining results, the most effective combination of features was: clubbing stylometric meta-features with textual features such as bigrams, part of speech (POS) bigrams, and word/POS pairs.

The authors of the article [16] applied authorship identification to prevent the spread of malicious messages on social media. The proposed model for extracting informative features used XGBoost as a preprocessor. XGBoost ranked texts using Multi-Criteria Decision-Making methods to build a classification model. Dataset consisted of 16,124 tweets of 280 characters from 20 Twitter users. The authors noted that data from any social network could be used instead of Twitter. According to the proposed method, *F*-measure was 94.3%.

The dataset for the study [17] consisted of 500 tweets for each of the 34 authors who met certain criteria. Raw data were collected using Nvivo software. The collected data were preprocessed to extract frequencies of 200 features. Jordan Recurrent Neural Network was used in this work. For N authors, an $N \times N$ network was trained for pairwise classification. These $N \times N$ experts were then organized into N special groups to combine the decisions of the $N \times N$ experts. Finally, it was observed that a large number of authors did not lead to a significant decrease of accuracy, and for any number of authors from 2 to 34, an accuracy of 66 to 88% was achieved.

SVM was used to determine authorship in [18]. Only the latest and popular tweets without advertisements from personal accounts were used. Finally, 20 tweets for each of the five authors were selected for the experiments. For each tweet were calculated the values of the following: unigram frequencies, average word length, sentence length, letter case, and punctuation frequencies. Then 16 features were randomly selected and used for classification using SVM. The accuracy of classification was calculated individually for each author. The average result for the five authors was 54%, the maximum was 75%, and the minimum—30%.

The authors of [19] conducted research of language-independent open set authorship verification in several representative Indo-European languages. A specialty of this work was a selection of one set of features for four different languages: 513 English texts, 280 Greek texts, 400 Spanish texts, and 158 Dutch texts. Text examples of Greek, Spanish, and Dutch were taken from the PAN2015 corpus, for English—from the PAN2014. A 90% accuracy was achieved using the classical ML methods: KNN, and SVM-SMO with the feature selection method SVM-RFE. A final set consisted of 26 stylometric features. Performances of all classification models were measured by 10-fold cross-validation using standard ML metrics: accuracy, precision, recall, and AUC-ROC. The results were improved from 90% to 94% by the MultiboostAB ensemble method.

In [20] identification of the author was provided using CNN. As a part of the study, the authors developed their own dataset. The dataset included 400 scientific publications written by 20 authors in the field of ML. The model proposed by the authors, at the top level, included a multi-layer CNN, which either calculated the probability distribution for the entire text (authorship verification task), or the average probability distribution for individual sentences (multiple candidate authors). At the lower level, texts were reduced to a length of 128 characters. Then Word2Vec and Glove were used. The authors' model was compared with SVM, KNN, and Multilayer Perceptron. For these models, n-grams with n ranging from 1 to 5 words and from 1 to 8 characters were used. The results showed that the multi-label CNN (word2vec+glove) achieved the highest accuracy of 65.3%. SVM

(with parameters sgd, squared hinge loss, l2, binary 1-g)—45.3%, MLP (doc2vec)—40%, KNN (3-g)—52%.

The topic of authorship identification is one of the most popular tasks at the PAN conference [21]. In 2021, the researchers were offered two datasets of different sizes, containing texts of authentically known authors. All texts were obtained from fanfiction.net. A special feature of PAN-2021 was open attribution. The datasets consisted of pairs of fragments from two different fanfics. Each pair was assigned a unique identifier, distinguishing between pairs with the same author and pairs with different authors. Participants were also provided information about the fandom (i.e., thematic category) for each text in the pair. The best result for the larger dataset was obtained by team Boenninghoff21 [22], with an accuracy of 95.45%. For the smaller dataset, team Weerasinghe21 [23] achieved an accuracy of 92.8%. The boenninghoff21 team proposed a hybrid neural-probabilistic framework. The authors finalized their own system presented at PAN-2020 for the 2021 competition. The updates were aimed to reduce sensitivities to topical variations and to improve the system calibration using the uncertainty adaptation layer. In addition, a significant addition to the system was the introduction of an out-of-distribution detector (O2D2) for defining non-responses. The Weerasinghe21 team used POS tagging and POS tag chunking to prepare the text for feature extraction. TF-IDF values for character *n*-grams, POS-Tag trigrams, 31 special characters, frequencies of 851 common English words, the average number of characters per token, and distribution of word-lengths were used as text features. A Stochastic Gradient Descent training algorithm was used to store the complete feature matrix in memory. LR was used as a classifier.

Most modern works on authorship attribution are devoted to an analysis of English texts. However, every year more and more studies appear aimed at analyzing other languages.

The purpose of the work [24] was the verification of the author of opinion pieces in Estonian and extraction unique to each author's stylistic features. The dataset contained 1474 opinion pieces (editorials, columns, etc.) written by 318 authors. The informative features were: word case, case of the first letter of the word, presence of digits in the word, POS tag frequencies, and verb type. In the first step, all texts encoded the matches of the patterns as bit vectors, where a true bit indicates a match. On the next calculation, the Matthews correlation between all resulting binary vectors was provided. Since the problem was set as a one-class learning problem, a One-Class SVM was chosen as the classifier. As a result, the proposed method had a precision of 74%.

Maciej Baj and Tomasz Walkowiak devoted their study [25] to determining the authorship of Polish texts using stylometric features. The authors considered various feature generation methods (grammatical classes frequency, methods based on statistical features, and based on common word appearance). The generated feature sets included the most common and rarest words of the dataset, POS tag frequencies, as well as statistics of character numbers, words, and sentences for each text. The authors noted that Polish is a highly inflected language. Due to that fact, lemmas of words were used. Nine classification methods, including ridge regression, multilayer perceptron, KNN, RF, passive-aggressive, elastic net learned by stochastic gradient descent, Rocchio classifier, NB for multinomial and multivariate Bernoulli models were used. Proposed methods were tested for 105 novels and 1058 short fragments written by five Polish authors. Best accuracies were 79% and 100% for volumetric and short texts, respectively. The most accurate classifiers were stochastic gradient descent in the case of volumetric texts, and ridge regression and NB for multinomial models in the case of text fragments.

The authors of [26] solved the problem aimed to determine the gender and age of the authors of Lithuanian texts. LSTM and CNN applied on the top of Lithuanian neural word embeddings were used. The dataset included Lithuanian parliamentary text transcripts, representing speeches and debates by the Lithuanian Seimas members. Texts with a length from 100 to 300 words were considered. For the age determination problem, the subset of 25,439 texts by 6 authors was used. For gender determination, 10,000 texts by two authors were used. Experiments were held with 6 balanced datasets of 100, 300, 500, 1000, 2000, and

5000 texts (i.e., instances) in each class. It was noted that LSTM worked more efficiently with small datasets, while CNN, in contrast, with large. The best accuracies of 32% and 61% for age and gender, respectively, were achieved using CNN with the largest datasets of 5000 instances in each class.

An open-set attribution of Lithuanian Internet Comments was discussed in [27]. The authors developed the recommendation system that returns a list of alleged authors and their corresponding probabilities for further analysis by experts. A simple Winner-Takes-All (WTA) metric was used to evaluate models. The dataset included 200 authors and 200 texts. Frequency distributions of functional words, unigrams, bigrams, prefixes, abbreviations were chosen as features. One-class SVM was chosen as the classifier. For each author, 80% of the text was used as a training set and 20% as a test set. The One-class SVM results were used to generate a list of possible authors for each disputed text. The resulting list was ranked by the probability of the true author's position in the rank of the suspected authors. The process was repeated for each of the 200 authors for evaluation. For a list of 40 possible authors, the list precision accuracy was 80%, and for 80 authors was 90%.

The main idea of the work [28] was finding an answer to the question "Did Radu Albala find a sequel to Matei Karadzhale's novel 'Sub pecetea tainei', or did he write the corresponding sequel himself?" using authorship verification methods. Firstly, the texts of Radu Albala and Mateiu Caragiale were classified using SVM. Further, for the same texts, clustering with a distance of ranks was used to build dendrograms. The dendrograms demonstrated the splitting of Mateiu's and Albala's works into two distinct groups. An additional experiment was also conducted to test if Albala tried to write in Mateiu's style. For this, an ad-hoc experiment was carried out: the last part of the novel "Sub pecetea tainei" was combined with the beginning of the "In deal, pe Militari" novel. Then the resulting text was used for reclassification and clustering. As a result, the authors of the article found that Albala wrote the first part of the novel "In deal, pe Militari" with Mateiu, but then Albala's participation in writing decreased. The final part of the novel was written only by Mateiu.

The article [29] was the first in a series of studies devoted to determining the authorship of poems written in Czech, German, Spanish, and English. SVM was trained on the formed feature set in two versions: using only the frequencies of the most popular words and *n*-grams of symbols, or adding characteristics of rhyme and syllable of the text to the feature set. The authors noted that the text's rhyme could be used as well as stylometric text features. Combining the rhyme with other features into a single set improved the result for all datasets by 6–8%.

As such, authorship attribution methods are being actively researched not only for the most common languages such as English but also for other languages. It should be noted that for the Ukrainian, Greek, Spanish, Dutch, Polish, Romanian, Estonian, and Lithuanian languages discussed above, in the common part of works the same informative features were used. But the authors of these works also noted the peculiarities of each language, e.g., the high inflection of the Polish language, that differentiate languages analysis.

2.2. Related Works on Identification the Author of a Russian-Language Text

However, not all methods showing excellent results for English and other languages can demonstrate high accuracy for Russian-language texts due to some peculiarities of the Russian: part-of-speech ambiguity, a large number of idioms and speech turns, a variety of synonyms, the flectivness, and complex morphology. These and other language constructions should be considered by a researcher analyzing the Russian-language text.

Tatiana Litvinova et al. [30] evaluated the effectiveness of using different types of character *n*-grams in the authorship identification task. Russian-language messages from the KavkazChat forum were used as the dataset. KavkazChat is a Russian-language forum with a focus on jihad in the North Caucasus. KavkazChat contains 699,981 posts written by 7125 members in 2003–2012. Neutral and extremism topics, written by 10 users, communicating on these topics, were selected for the study. The final dataset consisted of

32–374 messages per user. As features were calculated frequency distribution for affix, word, and punctuation trigrams. The classification was carried out using LinearSVC and 10-fold cross-validation. The results showed that character *n*-grams and affix *n*-grams are especially successful in attribution. The best results were obtained using all of the mentioned features: 41.9% for both topics, 48.8% for neutral topics, and 46.2% for extremist topics.

The study [31] was devoted to authorship identification on the database of 30 authors of 1506 Russian-language texts written in the XVIII-XXI centuries. To solve the problem several approaches were applied: RF, LR, SVM. The authors examined texts at three linguistic levels: lexical, morphological, and syntactic. The best model was doc2vec with LR—98% accuracy. Accuracy using only syntax features and LR was 89%. The simple morphology model was, on average, 10% less accurate than the model based on syntax features. The same trend was observed using morphology—syntax was 14% better on average.

The authors of [32] proposed their own specially designed datasetы—RusIdioStyle. The peculiarity of the dataset was using mostly short texts on different topics written by 125 authors. For authorship identification, a scheme of pairwise classification was used in the case that this method is close to the real forensic examination. In cross-topic tasks, semantic coherence features were introduced to supplement well-established *n*-gram features. Distance-based measures were compared with ML algorithms (SVM, KNN). The authors identified 17 out of 300 features as the most significant. Among them were frequency distributions of six punctuation marks, five words, four symbols, and two parts of speech. The results of the experiment confirmed the authors' hypothesis that for short texts distance-based measures work better than ML methods. Moreover, the results of the pairwise classification show that in complex cross-thematic scenarios the best results were obtained for features that did not depend on the topic of the text. In the pairwise comparison, the obtained accuracy was from 60.1% to 87.5%.

The Russian language has many unique characteristics that distinguish Russian from others: specific accents, complex word formation, borrowings from ancient Slavic languages. As a result, it is incorrect to use the set of informative features which provide high accuracy for other languages. In addition to the language, the size in characters and the theme of the text impose some limitations on the method.

2.3. Related Works on Using FastText for Authorship Attribution

In NLP tasks, much attention is paid to the quality of word representation. The fastText library in the implementation of Facebook [33] is a significant step in the evolution of vector semantic models and ML in text processing. However, it has not been applied to determine the authorship of short Russian-language texts yet, and it was decided to pay special attention to this model in this study. The main advantage of fastText is the processing speed in comparison with other models. Skip-gram negative sampling is used to model vector representations of words. Some languages (including Russian) contain compound words. Therefore, a subword model is added to the basic model, allowing the representation of words in the form of *n*-grams. Hashing is used to fix the dimension of features.

To solve the problem of identifying the author of a Bengali text, the authors of the study [34] applied the approach based on fastText due to the rapid training of the model. The dataset was obtained from three online public blogs. The total dataset consisted of 3 authors. 300 texts per author were used as the training dataset and 50 were used for testing. *N*-grams from one to five, as well as various combinations of them, were used as feature sets. Using fastText, an accuracy of 82.4% was achieved, surpassing the NB classifier's best accuracy of 69%, but inferior to SVM's 85%. Using *n*-grams of length 5, the training time of SVM was 2700 ms, of NB was 2235 ms, and of fastText was only 147 ms.

The article [35] solved the problem of short texts authorship identification. The authors hypothesized that such research could be useful for improving chatbots and personal assistants. The study compared the effectiveness of LR training on TF-IDF text representation and fastText using the dataset consisted of subtitles of 236 episodes of the

TV series "Friends". All punctuation marks, monosyllabic sentences, and replicas uttered by two or more actors at the same time, were removed. Train and test sets were obtained by 80:20 splitting. For quality estimation, 5-folds cross-validation was performed. Experiments were carried out for 2 to 30 word samples. The accuracy of TF-IDF was about 5% better than fastText's. The best accuracy for TF-IDF was obtained with 25 words—36% and for fastText—31%.

2.4. Related Works on Feature Selection

In most of the analyzed studies, various features were used both in the form of an aggregated vector of all features and as separate sets. The most commonly used features are bigrams and trigrams of symbols and words, functional words, the most frequent words of the language, POS tag frequencies, punctuation marks, distribution of word length, and sentence length. However, not all of them are effective. The set of selected features can be informative, uninformative, and redundant. Uninformative and redundant features are useless for classification. In addition, such features could reduce the effectiveness of classification due to the large dimension of the feature space, called "the curse of dimensionality". The purpose of feature selection is to obtain a subset of informative features and exclude uninformative and redundant ones to improve classification accuracy.

Feature selection for classification is used in many tasks: handwritten signature verification [36]; evaluation biometrics identification ability by the number of participants, their gender, and age [37]; for cardiovascular disease diagnostics [38] and other fields. For forensic purposes, it is important to ensure the quality of the evidence. As such, feature selection is used to choose multiple informative features, and improve the efficiency of author identification. Feature selection is a primary method to provide convincing and reliable evidence to support authorship attribution results.

Selection method is performed using GAs or metaheuristics. The latter is divided into two main groups: filter approaches and wrapper approaches. The advantages of filter approaches are their independence on learning algorithms, rapidity, and versatility. In addition, wrapper approaches permit the achievement of higher accuracy, but in a more time.

In the approach proposed by the authors [39], a correlation-based filter feature selection method was used to filter out uninformative features. A particle swarm optimization based (PSO) wrapper method was proposed for selecting informative features after filtering. Experiments were conducted on the two datasets: Blog (12 authors, 200 Chinese texts per author, 583 features) and E-mail (15 authors, 100 English texts per author, 386 features). Both datasets were randomly divided into a training set (70%) and a test set (30%). KNN was used with $k = 5$ and 10-fold cross-validation to evaluate the performance of a selected subset of features. In all PSO-based algorithms, the population size was 30, the maximum number of iterations was 100, the inertia weight w was 0.7298, and the acceleration constants were 1.49618. The particles in the swarm were initialized randomly. Author's method allowed to increase an accuracy from 45.9% to 76.2% for the Blog dataset, from 56.7% to 81.3% for the Email dataset. For the Blog dataset 93 out of 586 attributes were selected as informative, for Email—25 out of 386.

In [40] was considered the problem of binary classification. This paper described the application of PSO. PSO selected a specific classifier for each author completely automatically. In addition, PSO also selected preprocessing and feature selection methods. Two datasets were used: MX-PO (353 texts), and CCAT (5000 texts). The best result (96.6% accuracy) was obtained for the author Karl-Penhaul. The NB classifier was built in 104 out of the 3400 features. The lowest accuracy of 14.8% was obtained for the author Peter-Humphrey, whose texts were used with normalization for preprocessing and SVM.

3. Methods Used for Attribution

Numerous studies have proven the ability of SVM in identifying the authorship of literary texts [1,6,41]. ML algorithms most often require structured data, while deep NN

are capable of analyzing text sequences and selecting informative features automatically. By this fact, deep NNs, in particular such models as LSTM and CNN [42,43], have been successfully applied in NLP areas. NLP library fastText from Facebook Research is of separate notice, since a real breakthrough in the development of vector semantic models and ML in text processing.

3.1. Classical Machine Learning Methods

Practice shows that using simpler methods proven is more justified than novel approaches in different cases. As such, in our last work [1] SVM's accuracy was comparable to the more modern methods of deep learning, while SVM was training much faster. Therefore, it was decided to expand the list of classical methods and test SVM, LR, NB, DT, RF, KNN in authorship identification.

The advantage of these methods is the clarity of a decision-making process, in contrast to NNs, which is represented as a black box. The results of classical methods can be logically justified, which is important in forensics and other fields, where the persuasiveness of the evidence is more essential than a bit higher accuracy.

3.1.1. Support Vector Machine

SVM is a supervised learning algorithm. Its goal is to find the hyperplane equation to separate data the most optimally. This process occurs by maximizing the margin between the nearest points of different classes in the space—support vectors.

The efficiency of this algorithm is achieved due to the kernel transformation, which is responsible for reflecting the data into a space where the hyperplane separating the classes will be linear.

In this approach, the kernel can be any positively defined symmetric function of two variables. Such definiteness is necessary for the Lagrange function performing the optimization to be bounded below. When the optimization problem is correctly defined, the classifying function is constructed.

3.1.2. K-Nearest Neighbors Algorithm

KNN belongs to the class of nonparametric methods, i.e., KNN does not require assumptions about the statistical distribution of the training set. Based on this, classification models based on KNN will also be nonparametric. This means that the structure of the model has not been set strictly initially, but is determined by the data.

During classification, a new object, whose label has not been set, is presented. For this object, k nearest (by some metric, e.g., Euclidean distance) pre-classified objects are determined. Class, which most of the k nearest examples belong to, is chosen for a classified object.

The class of a new object, which is not included in the training set, is determined by weighted voting. The idea of weighted voting is based on the "penalty" for a class—it is the sum of values that inverses to the square of the distances from the sample of the j-th class of the object being classified. As such, the object is assigned a class with the maximal "penalty" value. This also reduces the probability that the classes receive the same number of votes.

The choice of the parameter k is important for obtaining correct classification results. A small value entails overfitting. With large values of the parameter, the noise level of the classification results decreases, but the severity of the class boundaries decreases. According to this fact, it is reasonable to choose k as an odd number. This reason avoids equality of votes when determining the class for a new observation.

3.1.3. Logistic Regression

LR is a special case of linear regression. The probability of accepting one or another value of the dependent variable based on the dependent variable, which includes the

classes, and the set of independent variables (the feature vector). The choice of the class depends on the calculated probabilities.

The classification algorithm is based on the weights of features, a value of a decision threshold, and a scalar product of the feature space of an object by the vector of weights. The task of training a linear classifier is to adjust the vector of weights according to the feature space. For this purpose, in LR, the problem of minimizing empirical risk with a loss function is solved. After finding the weights, it becomes possible to estimate the posterior probabilities of the object belonging to the classes.

3.1.4. Naive Bayes Classifier

NB classifiers are a group of simple probabilistic classifiers based on Bayes' theorem with strict (naive) independence assumptions. The assumption lies in using a probability approximation, which is the product of conditional probabilities of all words from a given object. That is, it is assumed that the probabilities of words are not related to each other, which is a completely incorrect assumption for natural language. To determine the most probable class, NB uses the estimate of the posterior maximum—it is necessary to find the probability of a set of presenting classes and to select the class with the maximum probability.

3.1.5. Decision Trees

DT classification creates a kind of graph of data distribution. At each node of a tree, a question about the importance of a particular feature is asked, and depending on the values of the features, the solution moves along the branches of the tree and falls into a certain class. The operation ends when the stop condition is reached, which is specified in the function parameters (e.g., the maximum tree depth). The advantages of the algorithm are simple interpretation and high operating speed.

3.1.6. Random Forest

RF is an ensemble of a set of DTs, which reduces the chance of overfitting and increases accuracy in comparison with a single DT. The belonging of an object to a certain class is determined as a result of aggregating the responses of the set of trees. The trees are trained independently on different subsets, which makes this algorithm very convenient for use in distributed computing systems.

3.2. Deep Neural Networks

A deep NN approach allows a network to find various informative features including implicit independence and therefore, the reduction of the manual feature space formation, and eliminates the need for expert knowledge in linguistics. Identification of implicit informative features makes it possible to determine the authorship even in the case of deliberate distortion of the text due to anonymization [3].

3.2.1. LSTM and BiLSTM

LSTM is an improved recurrent NN (RNN). Its modification solves the vanishing gradient problem of the classical RNN. This is achieved due to the fact that the semantic weights of this model are the same for all time steps in the error backpropagation. This allows the LSTM to analyze various time dependencies, including long-term ones.

LSTM is based on gates that are trained to find specific features in the data. In LSTM there are the forget, update, and reset gates. Gates include an activation layer and element-wise multiplication operations, thereby filtering the output information. An important part of the process is the memory state because the input context is stored there. Memory state changes depending on the need to add or remove information. The change in this state occurs due to the forget gate. If its value is 0, then the previous state is forgotten, and if 1, then the previous state is saved.

BiLSTM is an analog of unidirectional LSTM cells. However, BiLSTM has the advantage of being able to analyze not only the previous elements in the sequence but also the future ones, in contrast with its prototype.

3.2.2. CNN

CNNs are fundamental algorithms for solving computer vision problems. This is due to the ability of such NNs to recognize features of any dimension. This feature is extremely useful in text mining [44–46].

The basic principle of CNN is the work of filters that recognize certain features of the data. Moving through the text, the filter determines whether the necessary characteristic is present in a specific part of the text. A convolution operation is performed to obtain the result. This operation is a sum of the products of filter elements and a matrix of input signals.

In the general case, the formation of the hidden layer output feature map occurs due to the operation of kernel convolution with the feature map of the previous layer and their shift by a coefficient corresponding to the feature map. The CNN operation is reduced to the parallel analysis of n-grams, where n is determined by the convolution filter size.

3.2.3. Hybrid Neural Network Models

In addition to the standard deep NN architectures described above, various combinations of architectures are often used, for example, a combination of numerous LSTM layers consecutively or CNN with a gradual decrease in the number of filters in order to find more general patterns.

Hybrid NN's often perform better than using networks individually [47,48]. This is due to the disadvantages of one network which can be compensated with the advantages of another.

So, considering a hybrid NN consisting of CNN and LSTM, the following can be mentioned about its parts: CNN is good at extracting local information, but poorly describes contextual information. In turn, LSTM can extract contextual dependencies, which improves the efficiency of classification, but the training time of such model is long. By combining these networks, the responsibility for extracting local features of the text can be assigned to CNN, and LSTM will store temporary information and extract contextual text dependencies. The hybrid model can work better than separated networks. This article considers combinations of CNN + CNN, LSTM + CNN, and CNN + LSTM, which showed excellent results in the related problem of determining the author of the software source code [5]. It is worth noting that the popular modern architectures CNN with attention and Transformers in the previous study [1] proved to be less accurate and more time-consuming, so they were not considered in this study.

3.2.4. BERT

Another deep architecture that demonstrates high efficiency in NLP is Bidirectional Encoder Representations from Transformers (BERT). It combines the advantages of CNN with Self-attention and Transformer that discussed in the previous work [1], and at the same time, BERT allows for higher accuracy in related tasks [49].

BERT is a deep NN based on the composition of Transformer encoders. Each layer of the encoder involves two-way attention. Due to this, BERT considers the context on both sides of the token, which means that BERT more accurately determines the token's semantic meaning.

For the Russian language, there are two pre-trained BERT models: Rubert and Multi-Bert. These models show different results depending on the problem being solved. Therefore, both of them were tested in the study.

3.3. FastText

The fastText classifier is based on the simplest NN with one hidden layer. The Bag-of-Words input is passed to the first layer and converted to word embeddings. The resulting embeddings are averaged and reduced to one single embedding applicable to all input data. The resulting vector is passed through the classifier with the Softmax activation function to calculate the final probabilities.

4. Experiments Setup

Collection and preprocessing of data are an important part of the research. ML models, in particular, deep architectures, are very sensitive to the quality and volume of data. For this purpose, the author's dataset was collected. The dataset includes numerous works by Russian classics and comments of social network users.

In contrast to deep NNs, classical ML algorithms are not able to select informative features for decision-making automatically. Therefore, another factor influencing the results of experiments is the formation of feature space. Due to this fact, a separate experiment is devoted to the comparison of empirical and heuristic approaches to determining the set of informative features.

The main difficulty in the case of deep NNs is not the formation of the feature space but in the selection of hyperparameters that control the training process. Even minimal changes in these parameters can have a serious impact on the result. Therefore, the hyperparameters for the experimental models have been selected based on the authors' previous research experience [5].

In the Figure 1 is presented the IDEF0 diagram illustrating the process of the methodology.

Further in the article, more detailed information about the datasets, feature space, and approaches to its formation, as well as hyperparameters for training deep NNs are presented.

4.1. Datasets Description

The problem of identifying the author of the text was solved on the two datasets. The first includes the texts of Russian classics, the second—short comments from social network users. The choice of such data is driven by the cultural property and possibility of comparing the results with other researchers in the case of the Russian writers and the closeness to the real forensic tasks in the case of social media due to the small number and length of texts.

The dataset of classic writers includes 1100 literary texts in Russian created by 100 authors. All texts are collected from the Internet library [50]. A more detailed description of the dataset is presented in Table 1.

The second dataset includes short comments from users of the social network VK. The main differences from the first set are the length of the text and the conversational style. The last aspect is manifested in a presence of emoticons, a large number of exclamation and question marks, obscene language, and spelling errors. Texts with a length of at least 50 characters were selected for the training set. A detailed description of the dataset is shown in Table 2.

4.2. Text Preprocessing and Encoding

The purpose of preprocessing is to remove noise and redundant information from the dataset and to convert natural language text into a format understandable to the classifier.

In this study, a text preprocessing stage includes the following standard practices:

1. Converting all letters to lowercase;
2. Removal of stop-words;
3. Removal of digits (numbers) and special characters;
4. Whitespaces formatting.

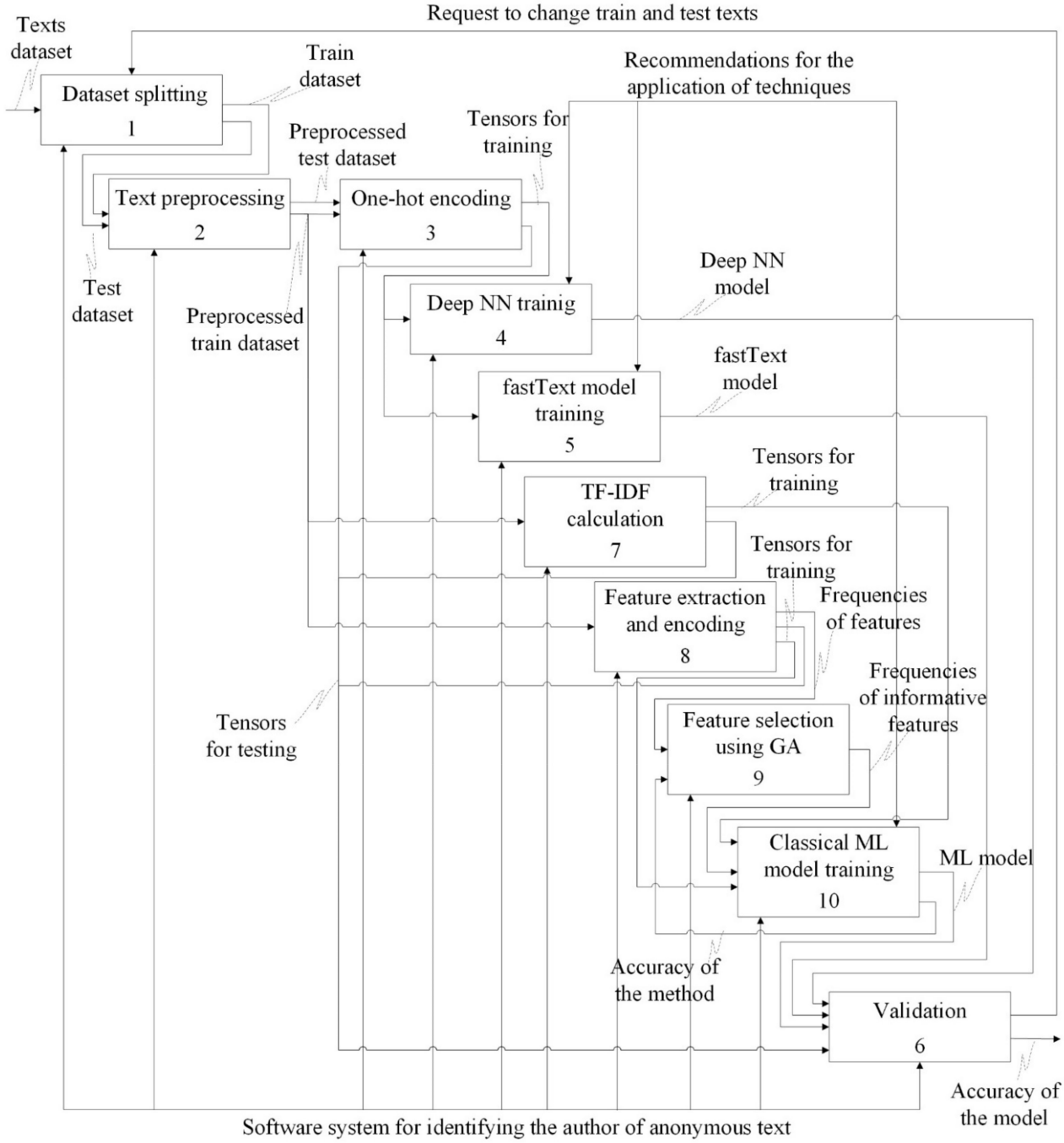


Figure 1. The methodology of the study.

Table 1. Description of the literary texts' dataset.

Dataset Characteristic	Value of the Characteristic
Number of authors	100
Number of texts	1100
Dataset size, symbols	375,618,852
Dataset size, words	62,603,142
Dataset size, sentences	5,216,929
The average length of text, symbols	973,342
The average length of sentence, words	14.3
Maximum number of texts per author	20
Minimal number of texts per author	9

Table 2. Description of the short texts' dataset.

Dataset Characteristic	Value of the Characteristic
Number of authors	3075
Number of texts	202,892
Dataset size, symbols	30,652,109
Dataset size, words	4,708,619
The average length of text, symbols	151.1
The average length of text, words	23.7
The average number of texts per author	115.37

A feature vector is constructed based on the processed text. According to the formed feature space, the classifier is able to identify common dependencies and correlations in all objects (texts) belonging to a given class (author), including texts that have not yet been used for training.

A set of n features measured for each text is fixed during the formation of the feature space. In this case, all n features are numeric, so the feature description of a text a the numeric vector of dimension n. Elements of the vector are frequencies of punctuation marks, parts of speech, unigrams, bigrams, and trigrams of symbols, the most popular words of the Russian language (based on the frequency dictionary [51]).

The values of the resulting vector will always be in the range [0, 1]. However, it is obvious that the letters, especially vowels, are found in the text much more often than bigrams, trigrams of symbols, etc. To bring the features to a common scale without losing information about differences in ranges, minimax normalization was used.

Texts are encoded using One-Hot Encoding to work with NN. The principle of this method is to convert categorical variables into a binary vector. The vector consists of zeros at the positions of all features, except one. The position corresponding to the numerical value of the feature is taken by one. The method is used to transform input data of NN and classical ML methods.

4.3. Parameters of Methods

We tested all combinations of hyperparameters' values and selected combinations with maximal classification accuracy. Due to a large number of experimental samples, only final values of the hyperparameters are presented below. Training parameters of ML models were determined empirically, based on the experience of previous studies [1,52]:

- For SVM training was used the sequential optimization method. The kernel was linear. The regularization parameter was 1, and the acceptable error rate is 0.00001. Normalization and compression heuristics were included as additional options.
- For KNN, different values of the parameter k were used: 3, 5, 7, 15, 25.
- To train LR were chosen: the liblinear optimization algorithm, regularization parameter 1, stopping criteria tolerance 1×10^{-4}, and limit number of 100 iterations.
- For DT training, gini was used as the partitioning quality function, and the maximum tree depth was 8.
- For RF training, 5, 15, 25, 35, and 50 decision trees were used.

When training deep NNs, an embedding layer was used as an input layer with an output size equal to 300. The spatial dropout method was used with the parameter 0.3 on the next layer. The activation function of the output layer was the logistic function for the multidimensional case (Softmax). Loss function was categorical cross-entropy, optimization algorithm—adaptive moment estimation (Adam), metric was accuracy. Training parameters were selected based on the research experience in the field of text analysis [53,54]:

- 128 filters for LSTM and Bidirectional LSTM were chosen. Dropout and recurrent dropout were equal to 0.3 in both cases. Rectified linear unit (ReLU) was selected as an activation function.

- Number of convolution filters for CNN was 1024, GlobalMaxPooling was chosen as a pooling layer. For the CNN with CNN hybrid, a network with a number of 512 convolution filters was also involved. To prevent overfitting, spatial dropout value 0.2 was used. The activation function was similar to LSTM.
- Hyperparameters for LSTM with CNN and CNN with LSTM hybrids were: the number of convolution filters—256, number of recurrent filters—128, and kernel size was 3. The activation function was selected as a ReLU; The dropout was carried out similarly to LSTM and BiLSTM. The activation function was similar to CNN.
- For fastText, the number of *n*-grams was 2–4. The learning rate parameter was defined as 0.6, the dimension for short texts was 50, for long texts—500. As a loss function, 'ova' (Softmax loss for multi-label classification) was used. The maximum number of allocated memory segments was 2,000,000. The rest of the parameters were default.
- When training BERT, the tokenizers "bert-base-multilingual-case" and "rubert-base-cased" were used. A ReLU was used as an activation function for hidden layers, Softmax as an activation function for the output layer, Adam as an optimization algorithm. For regularization, dropout (0.1) was chosen. The learning rate (lr) was 4×10^{-5}. The number of epochs was 5.

5. Results

This section shows results and training time for all described models on two datasets. The social media dataset was split into training and test samples in the proportion of 80:20, respectively. In literary texts, three texts of each author were used for training, and one text for the test. A cross-validation procedure was used in all cases. To evaluate the classification quality, accuracy was calculated as the proportion of the classifier's correct decisions to all.

5.1. Results Obtained on Literary Texts

In the previous article [1], authors concluded that 20,000 characters were sufficient to establish authorship. In this study, the volume of texts was reduced to 15,000 characters in order to complicate the task. All experiments were provided on the literary corpus using this volume. In addition, it is necessary to understand whether the proposed improvements and previously unused classifiers have an effect.

Tables 3 and 4 show the accuracy for cases of 2, 5, 10, 20, and 50 authors for ML models trained on feature space and TF-IDF, respectively, and the average accuracy for each model.

Table 3. Results of author identification using ML trained on the feature space.

Number of Authors	Accuracy of Models, %					
	SVM	**LR**	**NB**	**DT**	**RF**	**KNN**
2	**95.4 ± 1.7**	95.1 ± 4.4	91.9 ± 3.4	95.1 ± 1.1	97.4 ± 1.7	94.0 ± 2.6
5	**94.6 ± 2.1**	92.8 ± 3.1	87.1 ± 3.6	92.4 ± 2.3	92.1 ± 3.7	81.1 ± 2.1
10	**81.9 ± 4.6**	74.2 ± 5.1	74.4 ± 5.6	84.2 ± 4.4	67.6 ± 2.6	79.9 ± 3.3
20	**63.3 ± 4.8**	61.9 ± 5.1	58.3 ± 4.5	52.2 ± 2.3	62.2 ± 3.7	51.9 ± 5.2
50	37.7 ± 6.3	34.7 ± 7.1	29.8 ± 5.6	17.8 ± 2.7	35.9 ± 2.5	**41.9 ± 4.0**
Avg. accuracy	**74.7 ± 3.9**	71.2 ± 4.9	68.3 ± 4.5	68.7 ± 2.6	70.6 ± 2.8	69.8 ± 3.4

Table 4. Results of author identification using ML trained on the TF-IDF.

Number of Authors	Accuracy of Models, %					
	SVM	**LR**	**NB**	**DT**	**RF**	**KNN**
2	92.4 ± 1.1	**93.2 ± 2.3**	85.7 ± 2.7	90.1 ± 1.1	92.5 ± 2.2	86.9 ± 3.3
5	84.5 ± 3.4	82.2 ± 4.3	72.7 ± 4.8	81.1 ± 3.2	**88.5 ± 3.1**	78.0 ± 2.6
10	72.2 ± 5.6	68.7 ± 5.5	59.4 ± 4.9	**75.3 ± 2.2**	70.6 ± 1.1	71.4 ± 4.9
20	55.2 ± 4.2	52.3± 4.8	49.3 ± 5.3	33.6 ± 3.1	**59.0 ± 3.4**	57.4 ± 2.6
50	33.2 ± 4.8	27.7 ± 3.3	22.8 ± 4.1	16.1 ± 5.1	31.4 ± 4.1	**40.2 ± 3.4**
Avg. accuracy	67.5 ± 3.9	64.8 ± 4.1	57.9 ± 4.6	59.3 ± 2.9	**68.4 ± 2.9**	66.8 ± 3.5

Table 5 shows the accuracy of author identification using NNs for the same datasets and authors.

Table 5. Results of author identification using NNs.

Number of Authors	Accuracy of Models, %								
	LSTM	**BiLSTM**	**CNN**	**CNN + LSTM**	**LSTM + CNN**	**CNN + CNN**	**fastText**	**RuBERT**	**MultiBERT**
2	94.3 ± 5.5	95.5 ± 4.5	97.1 ± 3.6	94.5 ± 6.0	98.5 ± 5.6	**98.8 ± 4.1**	98.2 ± 4.5	95.2 ± 2.6	93.6 ± 2.8
5	86.7 ± 6.3	82.5 ± 4.8	**95.9 ± 2.8**	86.9 ± 4.8	95.5 ± 3.9	94.7 ± 3.4	95.0 ± 3.7	90.4 ± 4.1	89.8 ± 3.9
10	75.4 ± 3.3	70.2 ± 5.3	81.3 ± 5.9	78.2 ± 5.0	82.2 ± 5.4	86.5 ± 6.1	**92.2 ± 6.3**	84.3 ± 3.3	81.8 ± 3.5
20	63.9 ± 5.7	58.7 ± 4.9	71.2 ± 5.8	65.8 ± 3.2	62.2 ± 5.8	**72.8 ± 4.4**	69.9 ± 4.3	67.4 ± 4.1	64.9 ± 3.1
50	44.2 ± 6.1	41.1 ± 5.1	51.1 ± 5.1	55.0 ± 5.2	41.3 ± 4.5	**56.9 ± 4.2**	54.8 ± 6.2	52.2 ± 3.6	46.1 ± 4.0
Avg. accuracy	72.9 ± 5.4	69.6 ± 5.1	79.3 ± 4.8	76.1 ± 4.9	75.9 ± 5.1	**82.3 ± 4.8**	82.1 ± 6.0	77.9 ± 3.5	75.2 ± 3.5

Training times of all models obtained on the dataset of 50 authors are presented in Table 6.

Table 6. Training time on the dataset of 50 authors.

Training Time on Feature Vector, Sec.								
SVM	**LR**	**NB**	**DT**	**RF**	**KNN**			
1582	1082	677	714	1243	1134			
Training Time on TF-IDF, Sec.								
SVM	**LR**	**NB**	**DT**	**RF**	**KNN**			
1823	1418	746	1371	2334	2871			
Training Time of Neural Networks, Sec.								
LSTM	**BiLSTM**	**CNN**	**CNN + LSTM**	**LSTM + CNN**	**CNN + CNN**	**fastText**	**RuBERT**	**MultiBERT**
58,133	65,284	43,191	52,638	50,452	50,679	26,723	48,634	49,629

For voluminous texts, despite the limit of 15,000 characters, using classical ML methods trained on the formed feature space is effective. For datasets of 2, 5, and 10 authors, SVM, RF, and KNN achieve results comparable to deep NN's. The results obtained are based on these reasons: the volume of text fragments is sufficient to determine the author's writing style; completeness of the set of features selected for the identification; training on carefully selected experimentally parameters of ML models; the ability of SVM to work with a large feature space and solving of various complexity levels problems due to a high degree of flexibility; reduce the number of errors due to maximization of the margin of the separating hyperplane in the case of SVM.

In all considered experiments, fastText is inferior in accuracy by no more than 3%, and in the case of 10 authors, this method surpasses the rest of the models. In addition, the fastText learning is on average 51% faster than for other deep NN's. CNN and hybrid

networks, which include the convolutional network, train much faster than LSTM, BiLSTM, and BERT. The high speed of training is achieved by purely parallelizing the convolution process for each map, inverse convolution when the error propagates over the network.

5.2. Results Obtained on the Social Media Texts Dataset

Tables 7 and 8 show the accuracy for datasets of 2, 5, 10, 20, and 50 authors for ML models trained on feature space and TF-IDF, respectively, and the average accuracy for each model. For KNN and RF methods, the results presented only for k = 25 and 35 trees, respectively, due to the fact that classification accuracy is maximal with them.

Table 7. Results of author identification using ML trained on the feature space.

Number of Authors	Accuracy of Models,%					
	SVM	**LR**	**NB**	**DT**	**RF**	**KNN**
2	**72.2 ± 4.0**	67.1 ± 3.1	62.9 ± 2.3	69.1 ± 2.1	71.2 ± 3.9	68.1 ± 4.2
5	**69.9 ± 3.5**	59.5 ± 4.2	58.6 ± 2.6	43.5 ± 2.1	56.1 ± 2.8	65.4 ± 3.7
10	**66.3 ± 3.8**	48.2 ± 2.9	45.9 ± 3.5	24.2 ± 3.6	37.6 ± 2.7	61.9 ± 4.0
20	**55.3 ± 3.1**	34.3 ± 3.4	38.8 ± 4.1	19.9 ± 4.1	32.2 ± 1.7	43.9 ± 4.1
50	32.1 ± 3.9	28.6 ± 3.6	27.1 ± 3.3	15.9 ± 3.3	25.9 ± 2.4	**33.6 ± 3.4**
Avg. accuracy	59.2 ± 3.6	47.6 ± 3.4	46.8 ± 3.2	34.5 ± 3.0	44.6 ± 2.7	54.6 ± 3.9

Table 8. Results of author identification using ML trained on the TF-IDF.

Number of Authors	Accuracy of Models, %					
	SVM	**LR**	**NB**	**DT**	**RF**	**KNN**
2	61.1 ± 3.1	69.4 ± 5.2	**70.8 ± 1.0**	69.1 ± 2.1	68.4 ± 2.9	57.9 ± 1.7
5	54.4 ± 6.0	58.1 ± 4.4	**66.1 ± 7.3**	43.5 ± 2.1	60.4 ± 3.2	54.2 ± 3.1
10	39.4 ± 3.9	44.7 ± 0.7	**49.6 ± 5.4**	24.2 ± 3.6	42.6 ± 1.9	45.3 ± 1.9
20	32.0 ± 1.0	36.1 ± 2.2	**46.1 ± 2.3**	15.4 ± 3.6	33.6 ± 2.2	40.1 ± 2.3
50	17.3 ± 2.6	24.8 ± 3.1	**34.1 ± 5.0**	11.0 ± 4.8	23.5 ± 1.9	32.7 ± 2.45
Avg. accuracy	40.9 ± 3.3	46.6 ± 3.1	53.3 ± 4.2	32.5 ± 3.2	45.7 ± 2.4	46.1 ± 2.3

Table 9 shows the accuracy of author identification using NNs for the same datasets and authors.

Table 9. Results of author identification using NNs.

Number of Authors	Accuracy of Models, %								
	LSTM	**BiLSTM**	**CNN**	**CNN + LSTM**	**LSTM+CNN**	**CNN + CNN**	**fastText**	**RuBERT**	**MultiBERT**
2	93.0 ± 1.9	94.6 ± 2.1	**95.6 ± 2.0**	95.5 ± 3.5	92.3 ± 2.1	91.3 ± 2.4	94.0 ± 1.2	93.3 ± 2.1	90.2 ± 1.9
5	89.7 ± 1.9	92.5 ± 2.4	93.3 ± 1.5	**90.9 ± 2.2**	90.2 ± 1.0	89.2 ± 2.2	87.2 ± 2.2	88.6 ± 1.8	87.1 ± 2.2
10	73.0 ± 2.8	71.3 ± 2.4	72.4 ± 2.7	**77.1 ± 3.9**	64.2 ± 3.3	76.6 ± 4.5	76.1 ± 3.5	76.6 ± 3.2	69.5 ± 3.0
20	68.8 ± 2.4	59.3 ± 1.3	67.9 ± 3.3	62.2 ± 3.4	61.3 ± 3.2	**73.7 ± 2.5**	68.4 ± 2.3	66.8 ± 3.3	63.4 ± 2.7
50	50.1 ± 2.6	49.6 ± 3.9	48.8 ± 3.6	47.3 ± 1.9	47.4 ± 2.8	50.2 ± 1.4	**55.6 ± 2.8**	50.0 ± 2.9	47.1 ± 2.8
Avg. accuracy	74.9 ± 2.3	73.5 ± 2.4	75.6 ± 2.6	74.6 ± 3.0	71.1 ± 2.5	76.2 ± 2.6	**76.3 ± 2.4**	75.0 ± 2.7	71.5 ± 2.5

The most time-consuming is the case of the classification of 50 authors. Therefore, Table 10 presents the time taken to train all models on the dataset of 50 authors.

Table 10. Training time on the dataset of 50 authors.

Training Time on Feature Vector, Sec.								
SVM	**LR**		**NB**	**DT**		**RF**		**KNN**
589	**397**		**308**	**236**		804		604
Training Time on TF-IDF, Sec.								
SVM	**LR**		**NB**	**DT**		**RF**		**KNN**
717	584		**372**	416		1393		955
Training Time of Neural Networks, Sec.								
LSTM	**BiLSTM**	**CNN**	**CNN + LSTM**	**LSTM + CNN**	**CNN + CNN**	**fastText**	**RuBERT**	**MultiBERT**
30,190	32,980	25,380	28,397	26,467	25,874	15,926	26,547	27,117

In contrast to the classification of literary texts, the results obtained allow us to conclude that the classical ML methods are ineffective using the formed feature space. This is because the comments are short. The content of the comments reflects the emotions of the author about a commented post, so short statements and sentences prevail in the dataset. Due to this, the text volume is too small to obtain individual characteristics of the author even in a carefully formed feature space. SVM with experimentally chosen features trained on feature space achieves a maximum accuracy of 72% for two authors, while deep NNs are able to classify with an accuracy of 96% for the same task. This fact is explained by the ability of deep NNs to select implicit informative features automatically. The use of TF-IDF instead of feature vector gave an advantage only for two models (LR and NB). The accuracy obtained of all models in the cases of 20 and 50 authors is significantly inferior to the results obtained for literary texts. FastText outperforms LSTM+CNN and BERT models in accuracy for all sets of authors considered and learns 39% faster on average. In addition, for 2 and 10 authors, the accuracy of fastText is higher than BiLSTM, and in the case of 50 authors, fastText outperforms all other models. FastText outperforms all deep neural networks considered by the learning rate by 42% on average.

6. Feature Selection Using Genetic Algorithm

There can be numerous features that identify the personality of the author: preferred words, local speech features, length of sentences, use of turns of speech, vocabulary. However, a change in these parameters leads to a change in the frequency characteristics of the text. Due to this, the question of determining the set of the most informative features, as well as the exclusion of redundant features appears.

Genetic algorithms for feature selection are used to select an optimal subset from the general set of used features. In addition to the increase of the accuracy due to the removal of redundant and uninformative vector elements, this solution reduces the dimensionality of the feature space and accelerates the training of the model.

In total, there are three operators in GA: selection, crossing, and mutation. The rate of each operator belongs to a value from 0 to 1. Where 0 is the complete exclusion of the operator from an algorithm, 1 is the maximum possible work of the operator.

Selection is a must-used operator for selecting subsets of features for further work of the algorithm. Selection can be either random or conditional. Individuals (features) that have the maximum value of the "fitness" function are defined on the selected subset. The selected individuals "reproduce" the next generation using mutation and crossing operations. The number of generations is determined based on the choice of crossing and mutation rates. As the number of generations increases, the probability of finding a global optimum—the population where genes are the most adapted—increases too.

The stopping criterion can be the threshold value of the classification accuracy, finding a suboptimal or global optimum, exhaustion of the algorithm's running time, or a given number of calls to the target function.

The feature vector is binary: 1 corresponds to the inclusion of a feature into the set, and 0 means the exclusion. Since 1168 features were previously used, a complete enumeration requires considering 2^{1168} subsets. GA's goal is not to consider all variants but to select several subsets for a given number of features. Feature selection is implemented by embedding GA into the classifier, where the maximum accuracy or the minimum loss is used as the "suitability" of features.

In this study, GA was used in tandem with SVM. The reason lies in the fact that in the majority of cases SVM demonstrated the best accuracy among the classical ML methods. There is no need to use GA together with deep NNs, due to the ability of such architectures to select informative features automatically. In addition, the SVM training time is ten times smaller in comparison with deep NNs. As such, it is hypothesized to improve the accuracy of SVM classification. GA is defined by the following parameters:

- population size: 200;
- crossover ratio: 0.5;
- mutation rate: 0.2;
- number of populations: 20.

Experiments were provided to obtain 50, 100, 200, 300, 400, 500 informative features from the original set of 1168 elements. The experimental results for short social media texts are presented in Table 11, for literary texts—in Table 12.

Table 11. Results of GA for social media texts dataset.

Number of Authors	Number of Features						
	1168	**500**	**400**	**300**	**200**	**100**	**50**
2	72.2 ± 4.0	75.3 ± 5.2	**80.3 ± 3.3**	75.2 ± 4.7	67.2 ± 4.5	64.9 ± 3.8	65.3 ± 5.5
5	69.9 ± 3.5	70.4 ± 2.5	**77.1 ± 2.8**	72.4 ± 1.9	63.8 ± 3.9	59.0 ± 4.2	49.8 ± 3.9
10	66.3 ± 3.8	67.8 ± 3.7	**71.9 ± 2.6**	66.6 ± 3.1	60.2 ± 3.8	57.2 ± 3.9	47.2 ± 2.1
20	55.4 ± 3.1	62.4 ± 2.9	**64.8 ± 3.0**	59.3 ± 2.8	52.9 ± 4.1	49.4 ± 4.2	43.7 ± 3.7
50	32.1 ± 3.9	35.1 ± 6.3	**37.3 ± 4.1**	33.5 ± 2.5	27.4 ± 4.3	26.8 ± 3.0	22.4 ± 4.0
Avg. accuracy	59.2 ± 3.4	62.2 ± 4.1	**66.3 ± 3.2**	61.4 ± 3.0	54.3 ± 4.1	51.5 ± 3.8	45.7 ± 3.8

Table 12. Results of GA for literary texts dataset.

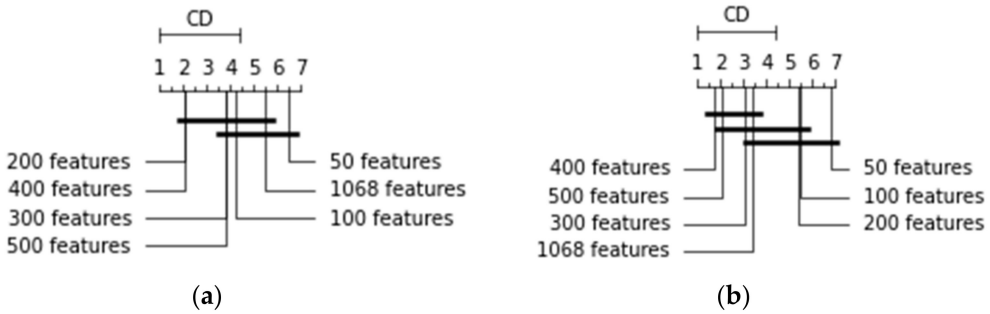
Number of Authors	Number of Features						
	1168	**500**	**400**	**300**	**200**	**100**	**50**
2	95.4 ± 1.7	96.1 ± 2.2	**98.6 ± 2.7**	94.5 ± 1.9	96.2 ± 1.6	98.3 ± 3.9	95.9 ± 3.1
5	94.6 ± 2.1	94.7 ± 3.3	**97.5 ± 3.1**	90.6 ± 1.8	95.0 ± 3.5	97.4 ± 1.9	94.8 ± 2.0
10	81.9 ± 4.6	83.7 ± 4.1	**88.0 ± 2.9**	82.1 ± 1.2	87.1 ± 3.4	85.9 ± 3.8	84.3 ± 3.2
20	63.3 ± 4.8	69.1 ± 2.5	**73.7 ± 3.3**	70.7 ± 2.4	72.9 ± 2.8	63.2 ± 2.5	60.7 ± 2.6
50	37.7 ± 6.3	40.2 ± 2.0.	**44.4 ± 2.6**	40.0 ± 3.7	42.4 ± 4.2	38.1 ± 3.7	33.8 ± 2.3
Avg. accuracy	74.7 ± 3.9	76.8 ± 2.9	**80.4 ± 2.9**	75.6 ± 2.2	78.3 ± 3.1	76.6 ± 3.2	73.9 ± 2.6

Based on the presented results, the decrease in the number of features by more than half (400) not only does not reduce the classification accuracy but also makes it possible to improve the result on both datasets. The accuracy obtained for 200 features exceeds the original in the case of literary texts and is comparable with original accuracy for short texts. 100 and 50 features are not sufficient to identify the author in all cases. The set of 400 features, on which the maximum accuracy was achieved, contains the frequency distributions of six punctuation marks, eight parts of speech, 165 words from the frequency dictionary, and character-level *n*-grams (20 unigrams, 107 bigrams, 98 trigrams).

To check if there is a statistically significant difference between results obtained by SVM trained using the different number of informative features selected by the GA, a

rank-based non-parametric Friedman and Némenyi post-hoc tests were applied to different cross-validation folds results. Friedman and Némenyi tests are suggested to use in the field of ML. These tests were performed for the most difficult of the considered cases—the case of 50 authors. The null hypothesis was that the difference between the results obtained on different numbers of features is only random. An alternative hypothesis was that there is a statistically significant difference between the results. The ***p***-value was 0.017 and 0.0007 for literary texts and comments from social networks, respectively. Since these values are less than 0.05, the null hypothesis can be rejected.

The effectiveness of the methods significantly differs if the respective mean ranks differ by at least a value of critical difference. To evaluate the difference, the Némenyi post-hoc test was applied after rejecting the null hypothesis of the Friedman test. The Némenyi post-hoc test is intended to detect different groups of data. The point of the Némenyi test is to do pair-wise performance tests. The results are presented as a Demšar significance diagram (Figure 2)**.** These diagrams help to visualize significant differences for each pair of used methods. In the case of the difference of mean ranks between two methods being smaller than the automatically calculated critical difference value, their performance difference not significant and on the figure presented as a horizontal line.



(**a**) (**b**)

Figure 2. Demšar significance diagram for literary texts (**a**) and for social media texts (**b**).

According to the diagram, reducing the training set to 50–100 informative features had a negative impact on the accuracy of the literary texts' author identification. The same result was achieved using all feature space (i.e., SVM without GA). The sets of 200–400 features allowed achieving comparable classification accuracy. For short texts, the best option was to use the SVM trained on 400 features obtained by GA, the worst—to limit the number of features to 50.

7. Limitations of the Proposed Methodology

When choosing a method for identification of the text's author, researchers should consider the volume of texts, the number of samples in the dataset, and the nature of texts.

Generalizing the obtained results, recommendations on the use of authorship identification methods were established:

1. The training dataset should include only texts in the author's writing style. It is recommended to remove the non-authors material from the text.
2. The author of each training text should be known for certain. If this condition is not met, the text should be excluded from the training set.
3. No less than three texts with a length of 15,000 and 50 texts with a length of 50 characters should be used for the training sets of literary and social media texts, respectively. In both cases, an increase in the number of texts or their lengths has a positive effect on the accuracy of the author's identification.
4. The specifics of the problem should be considered. In the case of short texts and/or limited resources, the classical ML methods with GAs or fastText should be used. In the case of the possibility of deliberate distortion of the text or an attack such as anonymization, deep NNs are more suitable due to the ability to automatically identify informative features of the author's style.

5. The proposed methodology is intended to solve the only authorship identification of a Russian-language text for a closed-set case.
6. And the last one, the fewer number of candidate authors, the higher the accuracy of author identification techniques. For the developed technique, this limit is ten and five authors, respectively, for literary and short texts datasets.
7. The most critical limitation for solving real-world scenarios is the lack of training data. However, the minimum amount of data required for the technique can be reduced. In future works, it is planned to apply confidence metrics and calibration curves, which will allow reducing the threshold even more. This will apply the presented methodology even on a small amount of training data without losing accuracy.

8. Discussion and Conclusions

The article considers classifiers for the identification of the author of a Russian-language text, such as classical ML methods (SVM, LR, NB, DT, RF, KNN), neural networks (CNN, LSTM, BiLSTM, RuBERT, MultiBERT, and fastText), and combinations of neural networks' architectures (CNN + LSTM, LSTM + CNN).

To train the models, two own datasets containing voluminous works of Russian classics and short comments from VK users were used.

The methods implemented in the work show results comparable and superior to those obtained by other researchers. For the classical methods of machine learning, the classification was carried out both on the formed feature space of the text and using TF-IDF. In both cases, the classification of short texts fails to achieve results comparable with deep neural networks. For all-number of disputed authors, the range of accuracy varies from 2 to 30%. The reason for this fact lies in the insufficient length of monosyllabic statements and sentences needed for the formation of a vector describing writing style.

When classifying datasets of 2, 5, and 10 authors of literary texts, the SVM, RF, and KNN methods are not inferior to deep NNs, reaching an accuracy of 97%. The obtained accuracy indicates a sufficient volume of text fragments (15,000 characters) for reliable classification based on the frequencies of character *n*-grams, as well as the ability of SVM to work with a large feature space.

In order to improve the quality of SVM classification, the selection of informative features using GA was carried out. In the selection process, the task was set to maximize the objective function, defined as SVM's accuracy. From the original set of 1168 features, subsets of 500, 400, 300, 200, 100, and 50 features were selected according to the value of the objective function. This solution allows not only to identify informative features but also to eliminate redundant ones that complicate the classification. Vectors of 50 features do not improve the classification on either of the two datasets. In the case of literary texts, training on subsets of 50 and 100 features improves the result for 2, 5, and 10 authors, and in the case of 20 and 50, there is no increase in accuracy. For both sets of text data, SVM training on the set of 400 features selected by GA allows achieving up to a 10% increase in accuracy for all datasets. This fact makes it possible to improve the learning rate of the process of training, reduce the load on computing resources and eliminate the redundancy of a feature set.

Based on the experiments and analysis of works devoted to the selection of informative features, several properties that characterize informative features can be noted:

1. Unconsciousness. In the case of choosing a feature that is poorly controlled by the author's consciousness, its deliberate distortion becomes less likely.
2. Immutability. The value of the feature is constant within a certain limited range for one author. Such features make it possible to distinguish between two or more authors with a similar writing style or someone, who is trying to imitate the style.

Deep neural networks, in contrast with SVM, can automatically identify implicit informative features for classification. Accuracy of more than 98% was obtained when training CNN on the dataset of literary texts. This result is higher than the accuracy of the SVM trained on a selected set of features with the maximum result in the entire

series of experiments. The accuracy of LSTM, including bidirectional ones, as well as their combinations with CNNs, achieves high accuracy for all datasets, but their training time in almost all cases exceeds the time spent on training SVM and other classical ML methods.

The optimal variant is fastText, the learning rate of which is on average 51% faster than for considered deep NNs, and the accuracy is below the maximum for all models by no more than 3%.

When choosing a method for identification of the text's author, researchers should consider the volume of texts, the number of samples in the dataset, and the nature of texts. In the case of short texts and/or limited resources, the classical ML methods with GAs or fastText should be used. In the case of the possibility of deliberate distortion of the text or an attack such as anonymization, deep NNs are more suitable due to the ability to automatically identify implicit features of the author's style.

In further works, a number of experiments are planned with hybrid models based on BERT and deep neural networks, as well as with ensembles of classifiers that include the most efficient models. It is planned to apply confidence metrics and calibration curves to reduce the required amount of data. Evaluation of the results will be carried out on an extended set of data—in addition to short messages from social users. networks and literary texts, fanfiction (amateur writings of authors) based on popular literary works will be used. It is also planned to conduct a series of experiments aimed at solving open set authorship attribution for both fictional and short texts of users of social networks.

Author Contributions: Supervision, A.R., A.S.; writing—original draft, A.F., A.R.; writing—review and editing, A.R., A.K.; conceptualization, A.K., A.R., A.F.; methodology, A.R., A.F.; software, A.F.; validation, A.F., A.K.; formal analysis, A.R., A.F.; resources, A.S.; data curation, A.S., A.R.; project administration, A.R.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Higher Education of Russia, Government Order for 2020–2022, project no. FEWM-2020-0037 (TUSUR).

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the reason that all literary texts were obtained from public sources, and social media texts were obtained from public forums of social networks. The comments have been anonymized for research.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data supporting reported results including links to publicly archived datasets and analysis code. Available online: https://github.com/afedotowaa/authorship_attribution/ (accessed on 20 December 2021).

Acknowledgments: The authors express their gratitude to the editor and reviewers for their work and valuable comments on the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Romanov, A.; Kurtukova, A.; Shelupanov, A.; Fedotova, A.; Goncharov, V. Authorship Identification of a Russian-Language Text Using Support Vector Machine and Deep Neural Networks. *Future Internet* **2021**, *13*, 3. [CrossRef]
2. Romanov, A.S.; Kurtukova, A.V.; Sobolev, A.A.; Shelupanov, A.A.; Fedotova, A.M. Determining the Age of the Author of the Text Based on Deep Neural Network Models. *Information* **2020**, *11*, 589. [CrossRef]
3. Romanov, A.; Kurtukova, A.; Fedotova, A.; Meshcheryakov, R. Natural Text Anonymization Using Universal Transformer with a Self-attention. In Proceedings of the III International Conference on Language Engineering and Applied Linguistics (PRLEAL-2019), Saint Petersburg, Russia, 27 November 2019; pp. 22–37.
4. Shumskaya, A.O. Method of the artificial texts identification based on the calculation of the belonging measure to the invariants. *Inform. Autom.* **2016**, *49*, 104–121. [CrossRef]
5. Kurtukova, A.; Romanov, A.; Shelupanov, A. Source Code Authorship Identification Using Deep Neural Networks. *Symmetry* **2020**, *12*, 2044. [CrossRef]
6. Romanov, A.S.; Vasilieva, M.I.; Kurtukova, A.V.; Meshcheryakov, R.V. Sentiment Analysis of Text Using Machine Learning Techniques. In Proceedings of the 2nd International Conference "R. Piotrowski's Readings LE & AL'2017", Saint Petersburg, Russia, 27 November 2017; pp. 86–95.

7. Khomenko, A.; Baranova, Y.; Romanov, A.; Zadvornov, K. Linguistic Modeling as a Basis for Creating Authorship Attribution Software. In Proceedings of the Computational Linguistics and Intellectual Technologies "Dialogue", Moscow, Russia, 16–19 June 2021; pp. 1063–1074.
8. Varela, P.; Justino, E.; Oliveira, L.S. Selecting syntactic attributes for authorship attribution. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 167–172.
9. Lupei, M.; Mitsa, A.; Repariuk, V.; Sharkan, V. Identification of authorship of Ukrainian-language texts of journalistic style using neural networks. *East.-Eur. J. Enterp. Technol.* **2020**, *1*, 30–36. [CrossRef]
10. Yang, M.; Chen, X.; Tu, W.; Lu, Z.; Zhu, J.; Qu, Q. A topic drift model for authorship attribution. *Neurocomputing* **2018**, *273*, 133–140. [CrossRef]
11. Potha, N.; Stamatatos, E. Improved algorithms for extrinsic author verification. *Knowl. Inf. Syst.* **2020**, *62*, 1903–1921. [CrossRef]
12. Dempster, A.; Laird, N.; Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–22.
13. Halvani, O.; Graner, L. POSNoise: An Effective Countermeasure Against Topic Biases in Authorship Analysis. In Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021; pp. 1–12.
14. Bevendorff, J.; Hagen, M.; Stein, B.; Potthast, M. Bias Analysis and Mitigation in the Evaluation of Authorship Verification. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; pp. 6301–6306.
15. Radhakrishnan, R.; Penstein, C. Machine Learning Framework for Authorship Identification from Texts. *arXiv* **2019**, arXiv:1912.10204.
16. Alterkav, S.; Erbay, H. Novel authorship verification model for social media accounts compromised by a human. *Multimed. Tools Appl.* **2021**, *80*, 13575–13591. [CrossRef]
17. Demir, N.; Can, M. Authorship Authentication of Short Messages from Social Networks Machines. *Southeast Eur. J. Soft Comput.* **2018**, *7*. [CrossRef]
18. Demir, N. Authorship Authentication for Twitter Messages Using Support Vector Machine. *Southeast Eur. J. Soft Comput.* **2016**, *5*. [CrossRef]
19. Adamovic, S. Automated language-independent authorship verification (for Indo-European languages). *J. Assoc. Inf. Sci. Technol.* **2019**, *70*, 858–871. [CrossRef]
20. Boumber, D.; Zhang, Y.; Mukherjee, A. Experiments with convolutional neural networks for multi-label authorship attribution. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
21. PAN: Shared Tasks. Available online: https://pan.webis.de/shared-tasks.html (accessed on 18 November 2021).
22. Boenninghoff, B.; Nickel, R.M.; Kolossa, D. O2D2: Out-of-distribution detector to capture undecidable trials in authorship verification. *arXiv* **2021**, arXiv:2106.15825.
23. Weerasinghe, J.; Singh, R.; Greenstadt, R. Feature vector difference based authorship verification for open-world settings. In Proceedings of the CEUR Workshop 2021, Bucharest, Romania, 21–24 September 2021; Volume 2936, pp. 2201–2207.
24. Petmanson, T. Authorship verification of opinion pieces in Estonian. *Eest. Raken. Uhin. Aastaraam.* **2014**, *10*, 259–267. [CrossRef]
25. Baj, M.; Walkowiak, T. Computer Based Stylometric Analysis of Texts in Polish Language. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 11–15 June 2017; pp. 3–12.
26. Kapočiūtė-Dzikicnė, J.; Damaševičius, R. Lithuanian Author Profiling with the Deep Learning. In Proceedings of the 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), Poznań, Poland, 9–12 September 2018; pp. 169–172.
27. Venckauskas, A.; Karpavicius, A.; Damaševičius, R.; Marcinkevičius, R.; Kapočiūte-Dzikiené, J.; Napoli, C. Open class authorship attribution of lithuanian internet comments using one-class classifier. In Proceedings of the 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 3–6 September 2017; pp. 373–382.
28. Dinu, L.P.; Popescu, M.; Dinu, A. Authorship Identification of Romanian Texts with Controversial Paternity. In Proceedings of the International Conference on Language Resources and Evaluation, Marrakech, Morocco, 26 May–1 June 2008.
29. Plecháč, P.; Bobenhausen, K.; Hammerich, B. Versification and authorship attribution. A pilot study on Czech, German, Spanish, and English poetry. *Studia Metr. Poet.* **2019**, *5*, 29–54. [CrossRef]
30. Litvinova, T.; Litvinova, O.; Panicheva, P. Authorship attribution of Russian forum posts with different types of *n*-gram features. In Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval, Tokushima, Japan, 28–30 June 2019; pp. 9–14.
31. Pimonova, E.; Durandin, O.; Malafeev, A. *Authorship Attribution in Russian with New High-Performing and Fully Interpretable Morpho-Syntactic Features //International Conference on Analysis of Images, Social Networks and Texts*; Springer: Cham, Switzerland, 2019; Chapter 193–204.
32. Panicheva, P.; Litvinova, T. Authorship attribution in Russian in real-world forensics scenario. In Proceedings of the International Conference on Statistical Language and Speech Processing; Springer: Cham, Switzerland, 2019; pp. 299–310.
33. FastText: Library for Efficient Text Classification and Representation Learning. Available online: https://fasttext.cc/ (accessed on 18 November 2021).

34. Chowdhury, H.; Imon, M.; Islam, M. Authorship Attribution in Bengali Literature Using fastText's Hierarchical Classifier. In Proceedings of the 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 13–15 September 2018; pp. 102–106.
35. Van Tussenbroek, T. Who said that? Comparing Performance of TF-IDF and fastText to Identify Authorship of Short Sentences. Bachelor's Thesis, Delft University of Technology, Delft, The Netherlands, 2020.
36. Hodashinsky, I.; Hancer, E.; Sarin, K.; Slezkin, A. A wrapper metaheuristic framework for handwritten signature verification. *Soft Comput.* **2021**, *25*, 8665–8681.
37. Svetlakov, M.; Hodashinsky, I.; Slezkin, A. Gender, Age and Number of Participants Effects on Identification Ability of EEG-based Shallow Classifiers. In Proceedings of the 2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russia, 13–14 May 2021; pp. 0350–0353.
38. Hodashinsky, I. Fuzzy classifiers in cardiovascular disease diagnostics. *Sib. J. Clin. Exp. Med.* **2020**, *35*, 22–31. [CrossRef]
39. Ma, J.; Xue, B.; Zhang, M. A Hybrid Filter-Wrapper Feature Selection Approach for Authorship Attribution. *Int. J. Innov. Comput. Inf. Control.* **2019**, *15*, 1989–2006.
40. Escalante, H.; Montes, M.; Villaseñor, L. Particle swarm model selection for authorship verification. In Proceedings of the Iberoamerican Congress on Pattern Recognition; Springer: Berlin/Heidelberg, Germany, 2009; pp. 563–570.
41. Martín-del-Campo-Rodríguez, C. Authorship Attribution through Punctuation *n*-grams and Averaged Combination of SVM. In Proceedings of the CLEF, Lugano, Switzerland, 9–12 September 2019.
42. Hitschler, J.; Van Den Berg, E.; Rehbein, I. Authorship attribution with convolutional neural networks and POS-eliding. In Proceedings of the Workshop on Stylistic Variation, Copenhagen, Denmark, 8 September 2017; pp. 53–58.
43. Huang, W.; Su, R.; Iwaihara, M. Contribution of improved character embedding and latent posting styles to authorship attribution of short texts. In Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data; Springer: Cham, Switzerland, 2020; pp. 261–269.
44. Xing, L.; Qiao, Y. Deepwriter: A multi-stream deep CNN for text-independent writer identification. In Proceedings of the 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 584–589.
45. Zhong, Z.; Sun, L.; Huo, Q. An anchor-free region proposal network for Faster R-CNN-based text detection approaches. *J. Doc. Anal. Recognit.* **2019**, *22*, 315–327. [CrossRef]
46. Yu, Y.; Wang, C.; Gu, X.; Li, J. A novel deep learning-based method for damage identification of smart building structures. *Struct. Health Monit.* **2019**, *18*, 143–163. [CrossRef]
47. Breuel, T. High Performance Text Recognition Using a Hybrid Convolutional-lstm Implementation. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 11–16.
48. Library of Maxim Moshkov. Available online: http://www.lib.ru/ (accessed on 18 November 2021).
49. Guo, Q.; Qiu, X.; Liu, P.; Xue, X.; Zhang, Z. Multi-Scale Self-Attention for Text Classification. *arXiv* **2019**, arXiv:1912.00544. [CrossRef]
50. Sharov's Russian Frequency Dictionary. Available online: http://www.slovorod.ru/freq-sharov/index.html (accessed on 18 November 2021).
51. Ruder, S.; Ghaffari, P.; Breslin, J. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv* **2016**, arXiv:1609.06686.
52. Akimushkin, C.; Amancio, D.; Oliveira, O., Jr. On the role of words in the network structure of texts: Application to authorship attribution. *Phys. A Stat. Mech. Its Appl.* **2018**, *495*, 49–58. [CrossRef]
53. Evert, S. Understanding and explaining Delta measures for authorship attribution. *Digit. Scholarsh. Humanit.* **2017**, *32*, ii4–ii16. [CrossRef]
54. Britt, C.; Rocque, M.; Zimmerman, G. The analysis of bounded count data in criminology. *J. Quant. Criminol.* **2018**, *34*, 591–607. [CrossRef]