future internet

MDPI

Article

Authorship Identification of a Russian-Language Text Using Support Vector Machine and Deep Neural Networks

Aleksandr Romanov [1], Anna Kurtukova [1,*], Alexander Shelupanov [1], Anastasia Fedotova [1] and Valery Goncharov [2]

[1] Department of Security, Tomsk State University of Control Systems and Radioelectronics, 634050 Tomsk, Russia; alexx.romanov@gmail.com (A.R.); saa@tusur.ru (A.S.); afedotowaa@icloud.com (A.F.)
[2] Department of Automation and Robotics, The National Research Tomsk Polytechnic University, 634050 Tomsk, Russia; gvi@tpu.ru
* Correspondence: av.kurtukova@gmail.com

Abstract: The article explores approaches to determining the author of a natural language text and the advantages and disadvantages of these approaches. The importance of the considered problem is due to the active digitalization of society and reassignment of most parts of the life activities online. Text authorship methods are particularly useful for information security and forensics. For example, such methods can be used to identify authors of suicide notes, and other texts are subjected to forensic examinations. Another area of application is plagiarism detection. Plagiarism detection is a relevant issue both for the field of intellectual property protection in the digital space and for the educational process. The article describes identifying the author of the Russian-language text using support vector machine (SVM) and deep neural network architectures (long short-term memory (LSTM), convolutional neural networks (CNN) with attention, Transformer). The results show that all the considered algorithms are suitable for solving the authorship identification problem, but SVM shows the best accuracy. The average accuracy of SVM reaches 96%. This is due to thoroughly chosen parameters and feature space, which includes statistical and semantic features (including those extracted as a result of an aspect analysis). Deep neural networks are inferior to SVM in accuracy and reach only 93%. The study also includes an evaluation of the impact of attacks on the method on models' accuracy. Experiments show that the SVM-based methods are unstable to deliberate text anonymization. In comparison, the loss in accuracy of deep neural networks does not exceed 20%. Transformer architecture is the most effective for anonymized texts and allows 81% accuracy to be achieved.

Keywords: authorship; text mining; machine learning; attribution; neural networks; deep learning; forensic intelligence

Citation: 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. https://doi.org/10.3390/fi13010003

Received: 10 December 2020
Accepted: 23 December 2020
Published: 25 December 2020

1. Introduction

It is now known that it is possible to determine the individual characteristics of the author on the basis of the writing style, since each text has a specific linguistic personality [1].

The topic of attribution overlaps with information security [2–5]. With the constant increase in volume of transmitted and received documents, there are many opportunities for the illegal use of personal data. An example is a type of fraud in which an attacker sends an employee of an organization an email on behalf of a manager asking them to perform a specific action (e.g., to divulge confidential information of the organization or to transfer funds). In addition, quite often there are situations related to hacking the victim's social media accounts and sending messages on the victim's behalf. One solution to this kind of problem is to compare the writing style of the suspicious texts with others for which it is certain that they were written by the person. As a result of the comparison, it is possible to determine the author. Establishing general differences in the documents based on the writing style is most relevant if there are no other data that would allow the author to be identified.

One type of violation in cyberspace is a copyright infringement and related rights of a text, which can be expressed, for example, by claiming a text by another author for material gain or attempting to pass off the authorship of a created text as the authorship of another person. The effectiveness of intellectual property protection in the digital space is determined by an ability to resist such violations and threats of their occurrence. Authorship identification methods allow determining such infringements and establishing the identity of the text creator.

Interest in the topic is also due to a growth in the volume of text data, the evolution of technology, and social networks. Thus, automatic identification of authorship is a growing area of research, which is also important in the fields of forensic science and marketing.

In this article, we solve the problem of identifying the author of a Russian-language text using a support vector machine and deep neural networks. Literary texts written by Russian-speaking writers were used as input data. The article includes an overview of related works, the statement of the text authorship problem, a detailed description of approaches to solving the authorship identification problem, an impact evaluation of attacks on the developed approaches, and a discussion of the results obtained.

2. Related Works

An excellent overview of articles up to 2010 is presented in [1]. However, since then, methods based on deep neural networks (NN) have become more and more popular, replacing classical methods of machine learning. For example, the topic of author identification is considered annually at the PAN conference [6]. As part of the conference, researchers were offered two datasets of different sizes, containing texts by well-known authors.

The authors of [7] emphasize that they have proposed an approach that takes into account only the topic-independent features of a writing style. Guided by this idea, the authors chose several features such as the frequency of punctuation marks, highlighting the last word in a sentence, consideration of all existing categories of functional words, abbreviations and contractions, verb tenses, and adverbs of time and place. An ensemble of classifiers was used in the work. Each of them accepts or rejects the supposed authorship. Research is distinguished by the application of an approach that, in general, is aimed at recognizing a person based on his behavior. Here, Equal Error Rate (EER) has been applied as the thresholding mechanism. Essentially, the EER corresponds to the point on the curve where the false acceptance rate is equal to the false rejection rate. The results are 80% and 78% accuracy for the large and small datasets, respectively. The results of the approach allowed the authors to take third place among all the submitted works.

In [8], stylometric features were extracted for each pair of documents. The absolute difference between the feature vectors was used as input data for the classifier. Logistic regression was used for a small dataset, and a NN was used for a large one. These models achieved 86% and 90% accuracy for small and large datasets, respectively. As a result, the authors of the study took second place.

The work that achieved the best result in the competition [9] presents the combination of NN with statistical modeling. Research is aimed at studying pseudo metrics that represent a variable-length text in the form of a fixed-size feature vector. To estimate the Bayesian factor in the studied metric space, a probability layer was added. The ADHOMINEM system [10] was designed to transmit the association of selected tokens into a two-level bi-directional long short-term memory (LSTM) network with an attention mechanism. Using additional attention levels made it possible to visualize words and sentences that were marked by the system as “very significant”. It was also found that using the sliding window method instead of dividing a text into sentences significantly improves results. The proposed method showed excellent overall performance, surpassing all other systems in the PAN 2020 competition on both datasets. The accuracy was 94% for the large dataset and 90% for the small one.

The authors of [11] took into account the syntactic structure of a sentence when determining the author of a text, highlighting two components of the self-supervised

network: lexical and syntactic sub-network, which took a sequence of words and their corresponding structural labels as input data. The lexical sub-network was used to code a sequence of words in a sentence, while the syntactic subnetwork was used to code selected labels, e.g., parts of speech. The proposed model was trained on the publicly available LAMBADA dataset, which contains 2662 texts of 16 different genres in English. The consideration of the syntactic structure made it possible to eliminate the need for semantic analysis. The resulting accuracy was 92.4%.

The work in [12] provides an overview of the methods for establishing authorship with the possibility of their subsequent application in the field of forensic research on social networks. According to the authors, in forensic sciences, there is a significant need for new attribution algorithms that can take context into account when processing multimodal data. Such algorithms should overcome the problem of a lack of information about all candidate authors during training. Functional words have been chosen as a feature, as they are quite likely to appear even in small samples and can therefore be particularly effective for analyzing social networks. Combinations of different sets of *n*-grams at symbol and word level with *n*-grams at the part-of-speech level were investigated. An accuracy of 70% was obtained for 50 authors.

The main idea of the study [13] is to modify the approach to establishing authorship by combining it with pre-trained language models. The corpus of texts consisted of essays by 21 undergraduate students written in five formats (essay, email, blog post, interview, and correspondence). The method is based on a recurrent neural network (RNN) operating at the symbol level and a multiheaded classifier. In cross-thematic authorship determination, the results were 67–91%, depending on the subject, and in cross-genre, 77–89%, depending on the genre.

The essence of [14] is to research document vectors based on *n*-grams. Experiments were conducted on a cross-thematic corpus containing some articles from 1999 to 2009 published in the English newspaper *The Guardian*. Articles by 13 authors were collected and grouped into five topics. To avoid overlapping, those articles for which content included more than one category were discarded. The results show that the method is superior to linear models based on *n*-gram symbols. To train the Doc2vec model, the authors used a third-party library called GENSIM 3. The best results were achieved on texts of large sizes. Accuracy for different categories ranged from 90.48 to 96.77%.

In [15], an ensemble approach that combines predictions made by three independent classifiers is presented. The method based on variable-length *n*-gram models and polynomial logistic regression and used to select the highest likelihood prediction among the three models. Two evaluation experiments were conducted: using the PAN-CLEF 2018 test dataset (93% accuracy) and a new corpus of lyrics in English and Portuguese (52% accuracy). The results demonstrate that the proposed approach is effective for fiction texts but not for lyrics.

The research conducted in [16] used the support vector machine (SVM). Parameters for defining the writing style were highlighted at different levels of the text. The authors demonstrated that more complex parameters are capable of extracting the stylometric elements presented in the texts. However, they are most efficiently used in combination with simpler and more understandable *n*-grams. In this case, they improve the result. The dataset included 20 samples in four different languages (English, French, Italian, and Spanish). Thus, five samples from 500 to 1000 words in each language were used. The challenge was to assign each document in the set of unknown documents to a candidate author from the problem set. The results were 77.7% for Italian, 73% for Spanish, 68.4% for French, and 55.6% for English.

Authorship identification methods are used not only for literary texts but also to determine plagiarism in scientific works. For example, [17] presents a system for resolving the ambiguity of authorship of articles in English using Russian-language data sources. Such a solution can improve the search results for articles by a specific author and the calculation of the citation index. The link.springer.com database was used as the initial

repository of publications, and the eLIBRARY.ru scientific electronic library was used to obtain reliable information about authors and their articles. To assess the quality of the comparison, experiments were carried out on the data of employees of the A.P. Yershov Institute of Informatic Systems. The sample included 25 employees, whose publications are contained in the link.springer.com system. To calculate the similarity rate of natural language texts, they were presented as vectors in multidimensional space. To construct a vector representation of texts, a bag-of-words algorithm was used with the term frequency–inverse document frequency (TF-IDF) measure. Stop-words were preliminarily removed from the texts, and stemming of words was carried out. Experiments were also provided on the vectorization of natural language texts using the word2vec. The average percentage of the number of publications of authors recognized by the system was 79%, while the number of publications that did not belong to the author but were assigned to his group was close to zero. The approaches used in the system are applicable for disambiguating authorship of publications from various bibliographic databases. The implemented system showed a result of 92%.

There were only a few works that achieved a high level of author identification in Arabic texts. In [18], the Technique for Order Preferences by Similarity to Ideal Solution (TOPSIS) was used to select the basic classifier of the ensemble. More than 300 stylometric parameters were extracted as attribution features. The AdaBoost and Bagging methods were applied to the dataset in Arabic. Texts were taken from six sources. Corpora included both short and long texts by three hundred authors writing in various genres and styles. The final accuracy was 83%.

A new area of research is attribution, which uses not only human-written texts but also texts obtained using generation [19]. Several recently proposed language models have demonstrated an amazing ability to generate texts that are difficult to distinguish from those written by humans. In [20], a study of the problem of authorship attribution is proposed in two versions: determining the authorship of two alternative human–machine texts and determining the method that generated the text. One human-written text and eight machine-generated texts (CTRL, GPT, GPT2, GROVER, XLM, XL-NET, PPLM, FAIR) were used. Most generators still produce texts that significantly differ from texts written by humans, which makes it easier to solve the problem. However, the texts generated by GPT2, GROVER, and FAIR are of significantly better quality than the rest, which often confuses classifiers. For these tasks, convolutional neural networks (CNN) were used, since the CNN architecture is better suited to reflect the characteristics of each author. In addition, the authors improved the implementation of the CNN using *n*-gram words and part-of-speech (PoS) tags. The result in the "human-machine" category ranges from 81% to 97%, depending on the generator, and, for determining the generation method, 98%.

The author of [21] presented the software product StylometRy, which allows the identification of the author of a disputed text. Texts were presented in the form of a bag-of-words model. Naive Bayesian classifier, *k*-nearest method, and logistic regression were chosen as classifiers, and pronouns were used as linguistic features. The models were checked in L. Tolstoy, M. Gorky, and A. Chekhov texts. The minimum text volume for analysis was 5500 words. The accuracy of the model for texts over 150,000 characters was in the range of 60–100% (average 87%).

The scientific work [22] describes the features of four styles of the Russian language —scientific, official, literary, and journalistic. The parameters selected for texts analysis were: the ratio of the number of verbs, nouns, adjectives, pronouns, particles, and interjections to the number of words in the text, the number of "noun + noun" constructions, the number of "verb + noun" constructions, the average word length, and the average sentence length. Decision trees were used for classification. The accuracy of the analysis of 65 texts of each style was 88%. The highest accuracy was achieved when classifying official and literary texts, and the lowest was achieved for journalistic texts.

The authors of [23] present the analysis and application of various NNs architectures (RNN, LSTM, CNN, bi-directional LSTM). The study was conducted based on three datasets

in Russian (Habrahabr blog—30 authors, average text length 2000 words; vk.com—50 and 100 authors, average text length 100 words; Echo.msk.ru—50 and 100 authors, average text length 2000 words). The best results were achieved by CNN (87% for Habrahabr blog, 59% and 53% for 50 and 100 authors with vk.com, respectively). Character's trigrams performed significantly better for short texts from social networks, while for longer texts, both trigram and tetragram representations achieved almost the same accuracy (84% for trigrams, 87% for tetragram representations).

The object of research study [24] is journalistic articles from Russian pre-revolutionary magazines. The information system Statistical Methods of Literary Texts Analysis (SMALT) has been developed to calculate various linguistic and statistical features (distribution of parts of speech, average word and sentence length, vocabulary diversity index). Decision trees were used to determine the authorship. The resulting accuracy was 56%.

The problem of authorship attribution of short texts obtained from Twitter was considered in scientific work [25]. Authors proposed a method of learning text representations using a joint implementation of words and character n-grams as input to the NNs. Authors used an additional feature set with 10 elements: text length, number of usernames, topics, emoticons, URLs, numeric expressions, time expressions, date expressions, polarity level, and subjectivity level. Two series of comparative experiments were provided to test using CNN and LSTM. The method achieved an accuracy of 83.6% on the corpus containing 50 authors.

The authors of [26] applied integrated syntactic graphs (ISGs) to the task of automatic authorship attribution. ISGs allow for combining different levels of language description into a single structure. Textual patterns were extracted based on features obtained from the shortest path walks over integrated syntactic graphs. The analysis was provided on lexical, morphological, syntactic, and semantic levels. Stanford dependency parser and WordNet taxonomy were applied in order to obtain the parse trees of the sentences. The feature vectors extracted from the ISGs can be used for building syntactic *n*-grams by introducing them into machine learning methods or as representative vectors of a document collection. Authors showed that these patterns, used as features, allow determining the author of a text with a precision of 68% for the C10 corpus and also performed experiments for the PAN'13 corpus, obtaining a precision of 83.3%.

An approach based on joint implementation of words, *n*-grams, and the latent Dirichlet allocation (LDA) was presented in [27]. The LDA-based approach allows the processing of sparse data and volumetric texts, giving a more accurate representation. The described approach is an unsupervised computational methodology that is able to take into account the heterogeneity of the dataset, a variety of text styles, and also the specificity of the Urdu language. The considered approach was tested on 6000 texts written by 15 authors in Urdu. The improved sqrt-cosine similarity was used as a classifier. As a result, an accuracy of 92.89% was achieved.

The idea of encoding the syntax parse tree of a sentence into a learnable distributed representation is proposed in [28]. An embedding vector is created for each word in the sentence, encoding the corresponding path in the syntax tree for the word. The one-to-one correspondence between syntax-embedding vectors and words (hence their embedding vectors) in a sentence makes it easy to integrate obtained representation into the word-level Natural Language Processing (NLP) model. The demonstrated approach has been tested using CNN. The model consists of five types of layers: syntax-level feature embedding, content-level feature embedding, convolution, max pooling, and softmax. The accuracy obtained on the datasets was 88.2%, 81%, 96.16%, 64.1%, and 56.73% on five benchmarking datasets (CCAT10, CCAT50, IMDB62, Blogs10, and Blogs50, respectively).

The authors of [29] combined widely known features of texts (verbs tenses frequency, verbs frequency in a sentence, verbs usage frequency, commas frequency in a sentence, sentence length frequency, words usage frequency, words length frequency, characters *n*-gram frequency) and genetic algorithm to find the optimal weight distribution. The genetic algorithm is configured with a mutation probability of 0.2 using a Gaussian convolution on the values with a standard deviation of 0.3 and evolved over 1000 generations. The method

was tested on the Gutenberg Dataset, consisting of 3036 texts written by 142 authors. The method is implemented using Stanford CoreNLP, stemming, PoS tagging, and genetic algorithm. The obtained accuracy was 86.8%.

There is no generally accepted opinion regarding the set of text features that provides the best result. In most works, text features such as bigrams and trigrams of symbols and words, functional words, the most frequent words in the language, the distribution of words in parts of speech, punctuation marks, and the distribution of word length and sentence length have proven to be effective. It is incorrect to judge the accuracy of the methods applied to the Russian language based on the results of research in the English language or any other languages because of the specific structure of each language. The choice of approach depends on the text language, the authorship identification method, and the accuracy of the available analysis methods. Particularly, the peculiarity of the Russian language in comparison with English, for which most of the results are presented, is its flexibility and, consequently, more complex word formation and a high degree of morphological and syntactic homonymy, which makes it difficult to use some features useful for the English language. The problems of genre, sample representativeness, and dataset size also limit the implementation of some approaches.

Investigations aimed at finding a method with high separating ability with a large number of possible authors are not always useful when solving real-life tasks. It is necessary to continue further research aimed at finding new methods or improving/combining existing methods of identifying the author, as well as conducting experiments aimed at finding features that allow accurately dividing the styles of authors of Russian-language texts. By using these features, it will be possible to work with small samples.

3. Problem Statement

We define the identification of the text author as the process of determining the author based on a set of general and specific features of the text that formed the author's style.

The problem of identifying the author of the text with a limited set of alternatives is formulated as follows. There are the set of texts $\mathbf{T} = \{t_1, \ldots, t_k\}$ and the set of authors $\mathbf{A} = \{a_1, \ldots, a_l\}$. For a certain subset of texts $\mathbf{T}' = \{t_1, \ldots, t_m\} \subseteq \mathbf{T}$, the authors are known; i.e., there are the set of text–author pairs $\mathbf{D} = \left\{(t_i, a_j)\right\}_{i=1}^{m}$. It is necessary to determine which author from set $\mathbf{A}$ is the true author of the remaining texts (anonymous or disputed) $\mathbf{T}'' = \{t_{m+1}, \ldots, t_k\} \subseteq \mathbf{T}$.

In this statement, the author's identification problem can be considered as a multi-label classification task. In this case, set $\mathbf{A}$ is the set of predefined classes and their labels, set $\mathbf{D}$ is the set of training samples, and objects to be classified are included in the set $\mathbf{T}''$. The goal is to develop a classifier that solves the problem—finding the objective function $\mathbf{F} : \mathbf{T} \times \mathbf{A} \to [-1, 1]$, which assigns some text from the set $\mathbf{T}$ to its true author. The function value is described as the degree to which the object belongs to the class, where 1 corresponds to the completely positive solution, while -1, on the contrary, is a negative one.

4. Methods for Determining the Author of a Natural Language Text

Early research [1] was aimed at evaluating the accuracy and the speed of classifiers based on machine learning algorithms. Then, the best results in all parameters were demonstrated by the SVM classifier. However, over the past 10 years, many solutions based on deep NNs appeared in the field of NLP: RNN and CNN for multi-label text categorization, category text generation, and learning word dependencies, and hybrid networks for aspect-based sentiment analysis. These solutions significantly exceed the effectiveness of traditional algorithms. As of 2020, LSTM, CNN with self-attention, and Transformer [30,31] are the models that successfully solve related text analysis problems. Thus, the purpose of the study was to compare SVM with modern classification methods based on deep NN. The enumerated models, their mathematical apparatuses, as well as the techniques of their application to the task of authorship attribution are described below.

4.1. Support Vector Machine

The SVM classifier is similar to the classical perceptron. Application of its kernel transformations allows training radial basis function network and perceptron with a sigmoidal activation function, the weights of which are determined by solving a quadratic programming problem with linear constraints, while training a standard NN implies solving the problem of non-convex minimization without restrictions. In addition, SVM allows working directly with a high-dimensional vector space without preliminary analysis and also without manually selecting the number of neurons in the hidden layer.

The main difference between SVM and deep-learning models is that SVM is unable to find unobvious informative features in text that have not been pre-processed. Therefore, it is necessary to first extract such features from the text.

Let us denote the set of letters of the alphabet, numbers, and separators $\mathbf{A} = \left\{a_1, a_2, \ldots, a_{|A|}\right\}$, the set of possible morphemes $\mathbf{M} = \left\{m_1, m_2, \ldots, m_{|M|}\right\}$, the language dictionary $\mathbf{W} = \left\{w_1, w_2, \ldots, w_{|W|}\right\}$, the set of phrases $\mathbf{C} = \left\{c_1, c_2, \ldots, c_{|C|}\right\}$, the set of sentences $\mathbf{S} = \left\{s_1, s_2, \ldots, s_{|S|}\right\}$, and the set of paragraphs $\mathbf{P} = \left\{p_1, p_2, \ldots, p_{|P|}\right\}$. Then, the text T can be represented as sequences of elements as follows:

$$T = \left\{a_j^i\right\}_{i=1}^{N_a} = \left\{m_j^i\right\}_{i=1}^{N_m} = \left\{w_j^i\right\}_{i=1}^{N_v} = \left\{c_j^i\right\}_{i=1}^{N_c} = \left\{s_j^i\right\}_{i=1}^{N_s} = \left\{p_j^i\right\}_{i=1}^{N_p}, \quad (1)$$

where $a_j^i \in \mathbf{A}$, $m_j^i \in \mathbf{M}$, $w_j^i \in \mathbf{W}$, $c_j^i \in \mathbf{C}$, $s_j^i \in \mathbf{S}$, $p_j^i \in \mathbf{P}$; N_a, N_m, N_p, N_w, N_c, N_s—the number of characters, morphemes, words, phrases, sentences, paragraphs in the text.

Thus, the SVM feature space can be described as vectors of features that reflect the properties of text elements: $\{a'_1, \ldots, a'_k\}$ for symbols, $\{m'_1, \ldots, m'_l\}$ for morphemes, $\{w'_1, \ldots, w'_n\}$ for words, $\{c'_1, \ldots, c'_r\}$ for phrases, $\{s'_1, \ldots, s'_t\}$ for sentences, and $\{p'_1, \ldots, p'_u\}$ for paragraphs.

In the study, when classifying with SVM, informative features are used as an unordered collection as inputs of the SVM. The frequencies of single text's elements are used as follows:

$$t_k = \left\{ \begin{array}{l} 1 \Leftrightarrow w_i^j \in \mathbf{W} \\ 0 \Leftrightarrow w_i^j \notin \mathbf{W} \end{array} , j = \overline{1, n_i}, k = \overline{1, |\mathbf{W}|} \right\}, \quad (2)$$

In addition, the texts elements sequences of some length (n-grams) or a limited number of them from the dictionary are used as follows:

$$f(a_i, \ldots, a_{i+n-1}) = \frac{C(a_i, \ldots, a_{i+n-1})}{L}, \quad (3)$$

$$P(a_i | a_{i-n+1} \ldots a_{i-1}) = \frac{C(a_{i-n+1}, \ldots, a_i)}{C(a_{i-n+1}, \ldots, a_{i-1})}, \quad (4)$$

where L—total number of counted n-grams; k—threshold value; *f()*—relative frequency of the element in the text; a—the symbol; *P()*—the probability of the element appearing in the text; n—the length of the n-gram.

It should be noted that for texts of small volumes, it is supposed to use frequencies smoothed by the methods of Laplace (5), Good-Turing (6), and Katz (7), which makes it possible to estimate the probabilities of non-occurring events:

$$P_{ADD}(a_i, \ldots, a_{i+n-1}) = \frac{1 + C(a_i, \ldots, a_{i+n-1})}{\mathbf{W} + \sum\limits_i C(a_i, \ldots, a_{i+n-1})}, \quad (5)$$

where P_{ADD}—estimates of Laplace; **W**—the language dictionary; *C()*—the number of occurrences of the element in the text.

$$P^*_{GT} = \frac{C*}{N}, P^*_{GT} = \frac{N_1}{N}C* = (C+1)\frac{N_{C+1}}{N_C}, \quad (6)$$

where P_{GT}—estimates of Laplace; N—the total number of the considered elements of the text; N_C—the number of text elements encountered exactly C times; C*—discounted Good Turing estimate.

$$P_{KATZ}(a_i|a_{i-n+1},\ldots,a_{i-1}) = \begin{cases} P*(a_i|a_{i-n+1},\ldots,a_{i-1}), & if\ C(a_{i-n+1},\ldots,a_i) > k \\ \alpha(a_{i-n+1},\ldots,a_{i-1})P_{KATZ}(a_i|a_{i-n+2},\ldots,a_{i-1}), & if\ 1 \le C(a_{i-n+1},\ldots,a_i) \le k. \end{cases}, \quad (7)$$

where t_k—the fact of the existence of the j-th word of the i-th text in the dictionary **W**; P_{KATZ}—estimates of Katz; $\alpha()$—weight coefficient.

In the process of authorship attribution of natural language text using classical machine learning methods, not only standard feature sets can be used; features obtained as a result of solving related tasks such as determining the author's gender and age, the level of the author's education, the sentiment of the text, etc. can also be used. However, as a part of this study, aspect-oriented analysis was also used for informative features extraction. Such a type of analysis involves understanding the meaning of a text by identifying aspect terms or categories. Thus, it becomes possible to extract keywords and opinions related to aspects.

There are two well-known approaches to implementing aspect analysis: statistical and linguistic. The statistical approach is performed as an extraction of aspects, determination of the threshold value for them, and selection such aspects, the values of which are indicated above the given threshold. The linguistic approach takes into account the syntactic structure of the sentence and searches for aspects by patterns.

We decided to use a combination of these methods. Aspects chosen were nouns and noun phrases (statistical approach), and the syntactic structure of the sentence was determined based on the dependencies between words (linguistic approach).

Multi-layered NN, consisting of fully connected layers, was implemented to extract aspects. The following training parameters were used:

- Optimization algorithm—adaptive moment estimation (Adam);
- Regularization procedure—dropout (0.3);
- Loss function—Binary cross-entropy;
- Hidden layers activation function—ReLU;
- Function of activation of the output layer—Sigmoid.

The principle of operation of SVM is to construct a hyperplane in the space of high-dimensional features in such a way that the gap between the support vectors (the extreme points of the two classes) is maximized. The mapping of the original data onto space with the linear separating surface is performed using a kernel transformation:

$$\big(\Phi(x), \Phi(x')\big) = k\big(x, x'\big), \quad (8)$$

where $(\Phi(x), \Phi(x'))$ is the inner product between the sample being recognized and the training samples, and k is some mapping of the original space onto the space with the inner product (the space of dimension sufficient for linear separability).

Then the function performing the classification looks like this:

$$f(x) = \left\{ \sum_{i=1}^{l} \alpha_i y_i k(x_i, x) \right\} + b, \quad (9)$$

where α is the optimal coefficient, k is the kernel function, y is the label of class, b is the parameter that ensures the fulfillment of the second Karush-Kuhn-Tucker condition for all input samples corresponding to Lagrange multipliers that are not on the boundaries.

The optimal coefficient α is determined by maximizing the objective function:

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i, x_j), \tag{10}$$

where the maximization condition:

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{11}$$

in the positive quadrant $0 \leq \alpha_i \leq C$, $i = \overline{1, l}$.

The regularization parameter C determines the ratio between the number of errors in the training set and the size of the gap.

4.2. Deep Neural Networks

A distinctive feature of deep NNs is their ability to analyze a text sequence and extract informative features by itself. In some studies, texts should be accepted by the model unchanged [1]. However, in solving the problem of determining the author of a natural language text, preliminary preparation is an important stage.

The purpose of preprocessing is to cleaning the dataset from noise and redundant information. Within the framework of the study, the following actions were taken to clean up the texts:

- Converting text to lowercase;
- Removing stop-words;
- Removing special characters;
- Removing digits;
- White space formatting.

The data obtained from the results of preprocessing must be converted into a vector-understandable NN. For this purpose, it was decided to use word embeddings—a text representation, where words having a similar meaning are defined by vectors close to each other in hyperspace. The received word representations are fed to the inputs of the deep NN.

4.2.1. Long Short-Term Memory

LSTM is a successful modification of the classical RNN, which avoids the problem of vanishing or exploding gradients. This is due to the fact that the semantic weights of the LSTM model are the same for all time steps during error backpropagation. Therefore, the signal becomes too weak (exponentially decreases) or too strong (exponentially increases). This is the problem that LSTM solves.

The LSTM model contains the following elements:

- Forget Gate "f"—an NN with sigmoid;
- Candidate layer "C'"—an NN with Tanh;
- Input Gate "I"—an NN with sigmoid;
- Output Gate "O"—an NN with sigmoid;
- Hidden state "H"—an vector;
- Memory state "C"—an vector.

Then the time step t is considered. The input to the LSTM cell is the current input vector $\mathbf{X}_t$, the previous hidden state H_{t-1}, and the previous memory state C_{t-1}. The cell outputs are the current hidden state H_t and the current memory state $C_{t.}$ The following formulas are used to calculate outputs:

$$f_t = \sigma(\mathbf{X}_t * \mathbf{U_f} + H_{t-1} * \mathbf{W_f}), \tag{12}$$

$$\overline{C_t} = tanh(\mathbf{X}_t * \mathbf{U_c} + H_{t-1} * \mathbf{W_c}), \tag{13}$$

$$I_t = \sigma(\mathbf{X}_t * \mathbf{U_i} + H_t * \mathbf{W_i}), \tag{14}$$

$$O_t = \sigma(\mathbf{X}_t * \mathbf{U_o} + H_{t-1} * \mathbf{W_o}), \tag{15}$$

where $\mathbf{X}_t$—the input vector; H_{t-1}—the hidden state of the previous cell; C_{t-1}—the memory state of the previous cell; H_t—the hidden state of the current cell; C_t—the memory state of the current cell at time t; **W**, **U** are the weight vectors for the forget gate *f()*, the gate of candidates, i.e., an input and output gates; σ—sigmoidal function; *tanh*—tangential function.

The most important role is the state of memory $\overline{C_t}$. It is the state in which the input context is stored. It changes dynamically depending on the need to add or remove information. If the value of the forget gate is 0, then the previous state is completely forgotten; if equal to 1, then it is completely transferred to the cell. With the current state of C_t memory, a new one can be calculated:

$$C_t = f_t * C_{t-1} + I_t * \overline{C}_t. \tag{16}$$

Then it is necessary to calculate the output from the hidden state H at time t. It will be based on memory state:

$$H_t = O_t * tanh(C_t), \tag{17}$$

Received C_t and H_t are transferred to the next time step, and the process is repeated.

4.2.2. CNN with Attention

CNN consists of many convolutional layers and subsampling layers. Each convolutional layer uses filters with input and output dimensions D_{in} and D_{out}. The layer is parameterized by the four-dimensional nuclear tensor **W** of the measurement and the displacement vector D_{out}—b_{out}. Therefore, the output value for some word q:

$$Y_q = \sum_{\Delta} \mathbf{X}_{q+\Delta} \mathbf{W}_q + b, \tag{18}$$

where Δ—kernel change.

The main difference between the attention mechanism and CNN is that the new meaning of a word is determined by every second word of the sentence, since the receptive field of attention includes the full context and not just a grid of nearby words.

The attention mechanism takes as input a token feature matrix, query vectors, and several key-value pairs. Each of the vectors is transformed by a trainable linear transform, and then the inner product query vectors are calculated with each key in turn. The result is run through Softmax, and with the weights obtained from Softmax, all vectors values are summed into a single vector. As a result of applying the attention mechanism, a matrix is obtained where the vectors contain information about the value of the corresponding tokens in the context of other tokens.

4.2.3. Transformer

The mechanism of attention in its pure form can lose information and complicate the convergence, and therefore a solution is required to this problem. Therefore, it was decided to also try its more complex modification—a transformer.

The transformer consists of an encoder and a multi-head attention mechanism. Some of the transformer layers are fully connected, and part of a shortcut is connected. A mandatory component of the architecture is multi-head attention, which allows each input vector to interact with other tokens using the attention mechanism. The study uses a common combination of multi-head attention, a residual layer, and a fully connected layer. The depth of the model is created by repeating this combination 6 times.

A distinctive feature of multi-head attention is that there are several attention mechanisms and they are trained in parallel. The final result is concatenated, passed through the training linear transformation once again, and goes to the output. Formally, it can be described as follows. The attention layer is determined by the size of the key/query

D_k, the number of heads N_h, the size of the head D_h, and the output D_{out}. The layer is parametrized with the key matrix, the query matrix $\mathbf{W}^{x}{}_{qry}$, and the value matrix $\mathbf{W}^{x}{}_{val}$ for each head, together with the protector matrix $\mathbf{W}_{out}$ used to assemble all the heads together. Attention for each head is calculated as:

$$A_q = \mathbf{X}_q : \mathbf{W}_{qry}\mathbf{W}^{T}_{key}\mathbf{X}^{T}_{k}. \quad (19)$$

The actual head value is calculated as:

$$H_q^{(h)} = \sum_{k' \in [W]x[H]} softmax(A_q^{(h)})_{k'}\mathbf{X}_{k'}\mathbf{W}^{(h)}_{val}. \quad (20)$$

And the output value is calculated as follows:

$$H_q = concat(H_q^{(1)}, \ldots, H_q^{(N_h)})\mathbf{W_{out}} + b_{out}, \quad (21)$$

where $\mathbf{X}$—output values, $\mathbf{W}_{key}$—the matrix of keys, T—the transposition operation, A_q—the attention value for a particular head, k—the key position, q—the query position, N_h—the number of heads, b_{out}—the bias coefficient of the measurement D_{out}.

5. Experiment Setup and Results

About 45 groups of different features of text were used to train the SVM classifier [1]. Vectors ranging in size from 33 to 5000 features were used, including characteristics of different levels of text analysis:

- Lexical (punctuation, special symbols, lexicon, slang words, dialectic, archaisms);
- Morphological (lemmas, morphemes, grammar classes);
- Syntactic (complexity, position of words, completeness, sentiments);
- Structural (headings, fragmentation, citation, links, design, mention of location);
- Content-specific (keywords, emoticons, acronyms and abbreviations, foreign words);
- Idiosyncratic stylistic features (spelling and grammatical errors, anomalies);
- Document metadata (steganography, data structures).

Even a carefully selected feature space does not guarantee high model efficiency, but equally important are the training parameters of the SVM model. In an early study [1], the following parameters were identified as the most appropriate:

- Learning algorithm—sequential optimization method;
- Kernel—sigmoid;
- Regularization parameter C = 1;
- Acceptable error level—0.00001;
- Normalization—included;
- Compression heuristic—included.

As stated earlier, deep NNs do not need a predetermined set of informative text features, as they are able to search for them on their own. However, these models are also extremely sensitive to learning parameters. These parameters have been selected based on the results of model experiments for related tasks [32,33]:

- Optimization algorithm—adaptive moment estimation (Adam);
- Regularization procedure—dropout (0.2);
- Loss function—cross-entropy;
- Hidden layer activation function—rectified linear unit (ReLU);
- Output layer activation function—logistic function for multi-dimensional case (Softmax).

A large number of data are required to train models. For this purpose, the corpus was collected from the Moshkov library [34]. The corpus includes 2086 texts written by 500 Russian authors. The minimum size of each text was 100,000 symbols.

As part of experiments with models, the number of training examples varied with needs in solving real-life authorship identification tasks (including when the training data

are limited). Therefore, the texts were divided into fragments ranging from 1000 to 100,000 characters (~ 200–20,000 words). We used three training examples for each author and one for testing.

Table 1 shows the accuracy of the SVM model for datasets of 2, 5, 10, and 50 candidate authors. Table 2 shows the results of applying SVM trained on statistical features and extracted aspects. Cross-validation for 10-folds was used as a procedure for evaluating the effectiveness of the models.

Table 1. Average accuracy of author identification using SVM.

The Length of Text, Symbols	2 Authors	5 Authors	10 Authors	50 Authors
1000	0.9	0.71	0.64	0.49
5000	0.91	0.79	0.77	0.54
10,000	0.93	0.85	0.81	0.59
20,000	0.98	0.97	0.94	0.78
40,000	0.99	0.99	0.97	0.82
60,000	0.99	0.97	0.97	0.89
80,000	0.99	0.98	0.98	0.93
100,000	1	0.99	0.99	0.95
Average accuracy	0.96	0.91	0.88	0.75

Table 2. Average accuracy of author identification using SVM with extracted aspects.

The Length of Text, Symbols	2 Authors	5 Authors	10 Authors	50 Authors
1000	0.92	0.74	0.68	0.53
5000	0.93	0.81	0.79	0.58
10,000	0.95	0.87	0.85	0.62
20,000	0.99	0.98	0.96	0.81
40,000	0.99	0.99	0.98	0.84
60,000	0.99	0.99	0.98	0.91
80,000	1	0.99	0.99	0.95
100,000	1	0.99	0.99	0.97
Average accuracy	0.97	0.92	0.90	0.78

It should be noted that the results presented in Tables 1 and 2 were obtained by joint application of SVM and the Laplace smoothing method, which gives a slight increase in accuracy (from 0.01 to 0.07) on small sample sizes. Experiments have also shown that the Good-Turing and Katz smoothing methods negatively affect the quality of identification, with an average accuracy 0.04–0.11 lower when using them.

Table 3 shows the accuracy of determining the author using the LSTM for datasets of similar size and obtained by 10-fold cross-validation, while Table 4 shows the CNN with Attention and Table 5, the Transformer.

Table 3. Average accuracy of author identification using LSTM.

The Length of Text, Symbols	2 Authors	5 Authors	10 Authors	50 Authors
1000	0.68	0.51	0.4	0.23
5000	0.75	0.53	0.45	0.3
10,000	0.82	0.59	0.49	0.37
20,000	0.88	0.64	0.55	0.41
40,000	0.91	0.73	0.58	0.46
60,000	0.95	0.77	0.64	0.56
80,000	0.97	0.82	0.68	0.62
100,000	0.98	0.89	0.74	0.66
Average accuracy	0.87	0.69	0.57	0.45

Table 4. Average accuracy of author identification using CNN with attention

The Length of Text, Symbols	2 Authors	5 Authors	10 Authors	50 Authors
1000	0.84	0.61	0.59	0.36
5000	0.89	0.69	0.68	0.4
10,000	0.92	0.75	0.76	0.5
20,000	0.95	0.78	0.79	0.56
40,000	0.96	0.83	0.85	0.62
60,000	0.96	0.88	0.86	0.68
80,000	0.99	0.93	0.91	0.73
100,000	0.99	0.95	0.95	0.78
Average accuracy	0.94	0.80	0.79	0.58

Table 5. Average accuracy of author identification using Transformer

The Length of Text, Symbols	2 Authors	5 Authors	10 Authors	50 Authors
1000	0.86	0.64	0.6	0.34
5000	0.88	0.68	0.69	0.4
10,000	0.91	0.74	0.77	0.52
20,000	0.94	0.8	0.81	0.59
40,000	0.95	0.86	0.83	0.66
60,000	0.95	0.88	0.86	0.72
80,000	0.98	0.94	0.92	0.76
100,000	0.98	0.95	0.96	0.8
Average accuracy	0.93	0.81	0.80	0.60

Obtained results allow one to form a conclusion about the special effectiveness of SVM trained on accurately selected parameters and features. The approach based on SVM demonstrates superior accuracy to modern deep NNs architectures, regardless of the number of the samples and their volume. It should also be noted that the SVM classifier is able to learn on large volumes of data 10 times faster than deep NNs architectures. The average training time for SVM was 0.25 machine-hours, while deep models were trained for an average of 50 machine-hours.

6. Attacks on the Method

SVM classifier showed excellent results in determining the author of a natural-language text. However, keep in mind that the above experiments were not complicated by deliberate modifications aimed at text anonymization. Anonymization may have a negative impact on the accuracy of authorship identification. This hypothesis was confirmed by an early study [35]. A text anonymization technique was proposed based on a fast correlation filter, dictionary synonymizing, and a universal transformer model with a self-attention mechanism. The results of the study showed that decision-making accuracy can be reduced by almost 50% due to the proposed method of anonymization, keeping the text in readable and understandable form for humans.

As part of the work, it was decided to evaluate the described anonymization technique on the developed approaches. The results are presented in Table 6. The results of the experiments confirm that deep models are much more resistant to the anonymization technique than the SVM classifier. This is due to their ability to extract unobvious features that are not controlled by the author on a conscious level, while SVM operates on the basis of pre-defined features manually found by experts, and therefore text may be exposed to deliberate confusion by anonymization techniques. It should be noted that in such cases, SVM with aspect analysis shows a bit higher accuracy than SVM without it.

Table 6. Average accuracy of author identification using Transformer.

Number of Authors	Model	Accuracy before Anonymization	Accuracy after Anonymization
10	SVM	0.97	0.46
	SVM (with aspects)	0.98	0.52
	LSTM	0.84	0.66
	CNN with attention	0.93	0.78
	Transformer	0.94	0.81
30	SVM	0.95	0.42
	SVM (with aspects)	0.97	0.49
	LSTM	0.72	0.63
	CNN with attention	0.91	0.71
	Transformer	0.93	0.74
50	SVM	0.93	0.39
	SVM (with aspects)	0.95	0.44
	LSTM	0.68	0.59
	CNN with attention	0.75	0.68
	Transformer	0.77	0.69

7. Discussion and Conclusions

During the course of the research, the authors analyzed modern approaches to determining the author of a natural-language text, implemented approaches of authorship attribution based on SVM and deep NNs architectures, evaluated the developed approaches on different numbers of authors and volumes of texts, and evaluated the resistance of the approaches to anonymization techniques. The results obtained allow us to draw several conclusions.

Firstly, despite the great popularity of deep NNs architectures, they are inferior to the traditional SVM machine learning algorithms in accuracy by more than 10% on average. This is due to the fact that NNs require more data for learning than SVM to extract informative features from the text. However, when solving real-life authorship identification tasks, the number of data could be not enough for accurate decision-making by the NN.

Secondly, the SVM classification is based on an accurately found set of features manually formed by experts. Such informative features are also obvious for anonymization techniques and therefore can be removed or significantly corrupted. Thus, to solve the problem of identification of the author of a natural language text, both the SVM-based approach and deep models proposed by authors are equally suitable. However, when choosing an approach, the researched data and available technical resources should be objectively evaluated. In the case of a lack of resources, an SVM approach should be used. If there are traces of use anonymization in the text, despite the longer processing time, deep NNs architectures are recommended because they can find both the obvious and unobvious dependences in the text.

Thirdly, when using SVM, we recommended using five of the most informative features of the author's style that may improve the authorship identification process: unigrams and trigrams of Russian letters, high-frequency words, punctuation marks, and distribution of words among parts of speech.

Finally, based on the results obtained, as well as on the experience of earlier research, the authors identified the important criteria to obtain accurate results when identifying the author of a natural language text:

1. Author's personality: the informative features extracted from the text should contain all important information about the writing style. In this case, the authorship attribution system will be able to distinguish between the same or different authors.
2. Invariance: the writing style's characteristics should be stable for certain reasons, e.g., author's mood, emotional state, and the subject of the text.

3. Stability: the writing style may be influenced by the imitation of another author's writing style or by deliberately distorting the author's own style for other reasons. The chosen approach should be resistant to such actions.
4. Adaptability to the style of text: the author adapts to the specificities of the selected style of text to follow it. Adaptability to the style of text leads to significant changes in the characteristics of the author's writing style. In addition, when writing official documents, many people use ready templates and just fill in their own data. As a result, it is quite problematic to identify the similarity of official documents and, for example, messages in social networks written by one person.
5. Distinguishing ability: the selected informative features of the text should be significantly different for various authors, greater than the possible difference between the texts written by the same author. Selecting a single parameter that clearly separates two authors is problematic. Therefore, it should be a complex set of features from different levels of text that are not controlled by the author at the conscious level. In this case, the probability of wrong identification for different authors is reduced.

Author Contributions: Supervision, A.S.; writing—original draft, A.K., A.F.; writing—review and editing, A.R., V.G., A.S.; conceptualization, A.K., V.G., A.S.; methodology, A.K., A.R.; software, A.K., A.F.; validation, A.K., A.R., A.R.; formal analysis, A.K., A.F.; resources, A.S.; data curation, A.K., 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).

Informed Consent Statement: Not applicable.

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.S.; Shelupanov, A.A.; Meshcheryakov, R.V. Development and Research of Mathematical Models, Methods and Software Tools of Information Processes in the Identification of the Author of the Text, Tomsk: V-Spektr, 2011.
2. Kurtukova, A.; Romanov, A.; Fedotova, A. De-Anonymization of the Author of the Source Code Using Machine Learning Algorithms. In Proceedings of the 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia, 21–27 October 2019; pp. 612–617.
3. Kurtukova, A.V.; Romanov, A.S. Identification author of source code by machine learning methods. *SPIIRAS Proc.* **2019**, *18*, 741–765. [CrossRef]
4. Rakhmanenko, I.A.; Shelupanov, A.A.; Kostyuchenko, E.Y. Automatic text-independent speaker verification using convolutional deep belief network. *Comput. Opt.* **2020**, *44*, 596–605. [CrossRef]
5. Kostyuchenko, E.Y.; Viktorovich, I.; Renko, B.; Shelupanov, A.A. User Identification by the Free-Text Keystroke Dynamics. In Proceedings of the 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC), Vladivostok, Russia, 18–25 August 2018; pp. 1–4.
6. PAN: Shared Tasks. Available online: https://pan.webis.de/shared-tasks.html (accessed on 18 November 2020).
7. Halvani, O.; Graner, L.; Regev, R. Cross-domain authorship verification based on topic agnostic features. In Proceedings of the Working Notes of CLEF, Thessaloniki, Greece, 22–25 September 2020.
8. Feature Vector Difference Based Neural Network and Logistic Regression Models for Authorship Verification. Available online: https://pan.webis.de/downloads/publications/slides/weerasinghe_2020.pdf (accessed on 18 November 2020).
9. Boenninghoff, B. Deep bayes factor scoring for authorship verification. *arXiv* **2020**, arXiv:2008.10105.
10. Boenninghoff, B.; Hessler, S.; Kolossa, D.; Nickel, R.M. Explainable Authorship Verification in Social Media via Attention-based Similarity Learning. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019.
11. Jafariakinabad, F.; Hua, K.A. A Self–Supervised Representation Learning of Sentence Structure for Authorship Attribution. *arXiv* **2020**, arXiv:2010.06786.
12. Mamgain, S.; Balabantaray, R.C.; Das, A.K. Author Profiling: Prediction of Gender and Language Variety from Document. In Proceedings of the 2019 International Conference on Information Technology (ICIT), Bhubaneswar, India, 19–21 December 2019; pp. 473–477.

13. Barlas, G.; Stamatatos, E. Cross-Domain Authorship Attribution Using Pre-Trained Language Models. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Neos Marmaras, Greece, 5–7 June 2020; pp. 255–266.
14. Gomez-Adorno, H. Document embeddings learned on various types of n-grams for cross-topic authorship attribution. *Computing* **2018**, *100*, 741–756. [CrossRef]
15. Custodio, J.E.; Paraboni, I. An ensemble approach to cross-domain authorship attribution. In Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages, Lugano, Switzerland, 9–12 September 2019; pp. 201–212.
16. Bartelds, M.; de Vries, W. Improving Cross-domain Authorship Attribution by Combining Lexical and Syntactic Features. In Proceedings of the CLEF (Working Notes), Lugano, Switzerland, 9–12 September 2019; Volume 24.
17. Isachenko, V.V.; Apanovich, Z.V. System of analysis and visualization for cross-language identification of authors of scientific publications. *NSU Vestnik Inf. Technol.* **2018**, *16*, 29–60. [CrossRef]
18. El Bakly, A.H.; Darwish, N.R.; Hefny, H.A. Using Ontology for Revealing Authorship Attribution of Arabic Text. *Int. J. Eng. Adv. Technol. (IJEAT)* **2020**, *4*, 143–151.
19. Iskhakova, A.O. Method and Software for Determining Artificially Created Texts. Available online: https://tusur.ru/ru/nauka-i-innovatsii/podgotovka-kadrov-vysshey-nauchnoy-kvalifikatsii/ob-yavleniya-o-zaschitah-dissertatsiy/dissertatsiya-metod-i-programmnoe-sredstvo-opredeleniya-iskusstvenno-sozdannyh-tekstov (accessed on 18 November 2020).
20. Uchendu, A. Authorship Attribution for Neural Text Generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020; pp. 8384–8395. Available online: http://www.cs.iit.edu/~{}kshu/files/emnlp20.pdf (accessed on 25 December 2020).
21. Chashchin, S.V. Application of "supervised" machine learning methods for text attribution: Individual approaches and intermediate results in identifying authors of Russian-language texts. *Probl. Criminol. Forensic Sci. Forensic Exam.* **2018**, *1*, 139–147.
22. Dubovik, A.R. Automatic determination of the stylistic affiliation of texts by their statistical parameters. *Comput. Linguist. Comput. Ontol.* **2017**, *1*, 29–45.
23. Dmitrin, Y.V. Comparison of deep neural network architectures for authorship attribution of Russian social media texts. In Proceedings of the Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference Dialogue, 2018; Available online: http://www.dialog-21.ru/media/4560/_-dialog2018scopus.pdf (accessed on 25 December 2020).
24. Kulakov, K.A. Attribution of texts using mathematical methods and computer technologies. *Digit. Technol. Educ. Sci. Soc.* **2019**, *3*, 121–125.
25. 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, Tianjing, China, 12–14 August 2020; pp. 261–269.
26. Gómez-Adorno, H.; Sidorov, G.; Pinto, D.; Vilariño, D.; Gelbukh, A. Automatic authorship detection using textual patterns extracted from integrated syntactic graphs. *Sensors* **2016**, *16*, 1374. [CrossRef] [PubMed]
27. Anwar, W.; Bajwa, I.S.; Choudhary, M.A.; Ramzan, S. An empirical study on forensic analysis of urdu text using LDA-based authorship attribution. *IEEE Access* **2018**, *7*, 3224–3234. [CrossRef]
28. Zhang, R.; Hu, Z.; Guo, H.; Mao, Y. Syntax encoding with application in authorship attribution. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2742–2753.
29. Keyrouz, Y.; Fonlupt, C.; Robilliard, D.; Mezher, D. Evolving a Weighted Combination of Text Similarities for Authorship Attribution. In Proceedings of the International Conference on Artificial Evolution (Evolution Artificielle), Mulhouse, France, 29–30 October 2018; pp. 13–27.
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Gomez, A.N.J.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
31. Chang, W.-C.; Yu, H.-F.; Zhong, K.; Yang, Y.; Dhillon, I. Taming Pretrained Transformers for Extreme Multi-label Text Classification. *arXiv* **2019**, arXiv:1905.02331.
32. Kurtukova, A.; Romanov, A.; Shelupanov, A. Source Code Authorship Identification Using Deep Neural Networks. *Symmetry* **2020**, *12*, 2044. [CrossRef]
33. 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]
34. Moshkov's Library. Available online: http://lib.ru/ (accessed on 18 November 2020).
35. 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, Saint Petersburg, Russia, 27 November 2019; pp. 22–37.