ISSN 0361-7688, Programming and Computer Software, 2024, Vol. 50, No. 4, pp. 304–314. © Pleiades Publishing, Ltd., 2024.
Russian Text © The Author(s), 2024, published in Programmirovanie, 2024, Vol. 50, No. 4.

Analytical Review of Confidential Artificial Intelligence: Methods and Algorithms for Deployment in Cloud Computing

E. M. Shiriaev[a,*] (ORCID: 0000-0002-2359-1291), A. S. Nazarov[a,] (ORCID: 0000-0002-0109-6097), N. N. Kucherov[a,***] (ORCID: 0000-0003-0337-0093), and M. G. Babenko[a,b,****] (ORCID: 0000-0001-7066-0061)**

[a] *North Caucasus Federal University, Stavropol, 355017 Russia*

[b] *Ivannikov Institute for System Programming, Russian Academy of Sciences, Moscow, 109004 Russia*

**e-mail: eshiriaev@ncfu.ru*

***e-mail: anazarov@ncfu.ru*

****e-mail: nkucherov@ncfu.ru*

*****e-mail: mgbabenko@ncfu.ru*

Received February 21, 2024; revised March 18, 2024; accepted March 18, 2024

Abstract—The technologies of artificial intelligence and cloud computing systems have recently been actively developed and implemented. In this regard, the issue of their joint use, which has been topical for several years, has become more acute. The problem of data privacy preservation in cloud computing acquired the status of critical long before the necessity of their joint use with artificial intelligence, which made it even more complicated. This paper presents an overview of both the artificial intelligence and cloud computing techniques themselves, as well as methods to ensure data privacy. The review considers methods that utilize differentiated privacy; secret sharing schemes; homomorphic encryption; and hybrid methods. The conducted research has shown that each considered method has its pros and cons outlined in the paper, but there is no universal solution. It was found that theoretical models of hybrid methods based on secret sharing schemes and fully homomorphic encryption can significantly improve the confidentiality of data processing using artificial intelligence.

Keywords: cloud computing, artificial intelligence, neural network, secret sharing scheme, homomorphic encryption, residue number system

DOI: 10.1134/S0361768824700117

1. INTRODUCTION

Artificial intelligence (AI) technologies are becoming increasingly widespread in the industrial and everyday life of society. The growing popularity and rapid development of technologies lead to the complication of tasks solved with the help of AI, and, as a result, to the fact that information processing on a standard user device is performed critically inefficiently, which is unacceptable for the end user. In this case, the solution is the use of cloud technologies (CT), which in turn have already gained wide popularity and have developed their methodology. It is also worth noting that AI and CT are of great interest in the scientific community. For example, projects such as [1, 2], have gained popularity in the daily field of activity, while they are scientific projects in the field of AI. When AI and CT methods are used together, a number of both standard and specific problems arise related to reliability, security and confidentiality. Considering that of all the functionality implemented by CT, AI uses mainly cloud computing (CC), for AI using CT, all the problems characteristic of CC occur. CC security threats can generally be divided into two categories: external threats and internal threats. External threats include various attacks by adversaries aiming to steal information (for example, hacking) or to damage information (for example, DDOS attacks) [3]. Internal threats in general are a combination of all possible ways to compromise the security system from the inside. To combat the latter, for example, secret sharing schemes are used [4]. However, the highest level of security is achieved when using methods that allow you to process data in encrypted form. In this case, the probability of compromising the system is reduced to a minimum. A possible solution to the problem of confidential data processing is homomorphic encryption. This paper presents a study of modern AI privacy methods and algorithms used in practice and a predictive study of methods that may be used in the future.

The work consists of 4 sections. Section 2 discusses the main aspects related to AI and cloud technologies. Section 3 discusses approaches to building a privacy-preserving AI. Section 4 presents the results of the analytical review.

2. ARTIFICIAL INTELLIGENCE AND CLOUD TECHNOLOGIES

2.1. Artificial Intelligence

Artificial intelligence (AI) originally developed within the framework of intelligent systems and is still considered one of their components, namely the ability to perform creative functions. Initially, the need for intelligent systems was due to the automation of decision-making. That is, a certain reaction to certain events was expected from the system. With the development of computer technology, methods and algorithms, as well as methods for developing systems and applications, these reactions became more complex and more flexible.

Historically, the first AI methods were machine learning (ML) methods. ML is a class of AI methods whose task is not to solve a problem directly, but to find it through learning based on the analysis of many solutions to similar problems [5]. The ML problem statement can be defined as follows. There is some unknown relationship between the two sets. Only the precedents are known, i.e. pairs of these two sets, which are called a training sample. Based on these data, the task is to restore dependence, that is, to build an algorithm that will create a new pair with a given accuracy. In fact, ML is tasked with approximating a function, but not necessarily by another function, but by some kind of algorithm.

AI methods can also be divided according to the method of learning. For example, there are reinforcement learning models [6], among such models, genetic algorithms can be distinguished. Genetic algorithms are heuristic search algorithms used to solve optimization and modeling problems by random selection, combination and variation of initial parameters, which are based on methods similar to natural selection in nature [7]. There is also teaching without a teacher, such methods are used to solve, for example, the clustering problem [8]. However, the largest group of methods consists of methods using teaching with a teacher, where for a set of precedents (known pairs of input and output data) it is necessary to build an algorithm that returns the required solution [9].

Artificial neural networks or simply neural networks (NN) are a mathematical model that is based on the principle of functioning of networks of nerve cells of a living organism [10]. The development of computer technology has made it possible to create NN models of great complexity. In this context, several events can be identified that have allowed the expansion of the use of AI, this is the emergence of various hardware and graphics accelerators [11–13], as well as valve matrices [14–17] for AI, which allow you to solve more a wide range of tasks previously unavailable. The work of such NN models is carried out through the use of so-called deep learning (DL). In fact, DL is a learning process for multi-layered NN. Theoretically, 2–3-layer NN is sufficient to solve a wide range of problems, however, DL is often used to solve complex problems, which shows good results [18]. DL includes methods such as the limited Boltzmann machine. Convolutional neural networks are also built on the basis of DL, which use different forms of convolutions on different layers [19]. They are often used for pattern recognition [20]. One of the most popular modern AI technologies is GPT technology. Even at the beginning of the development of GPT models required large training samples, which take up tens of gigabytes at the pre-training stage, and given the fact that the number of users of models with GTP exceeds 100 million, the amount of resources consumed exceeds the resources available to one device, if we consider the average office computer. The effective operation of such technologies requires the use of distributed computing systems.

2.2. Cloud Computing

Let's take a closer look at cloud technologies (CT). In fact, CT and CC are synonymous, since any data processing in the cloud involves some kind of calculation. The situation is similar with cloud storage, since when processing information (for example, search) Certain calculations are also performed.

They are a development of the distributed computing model [21] with some exceptions. Distributed computing assumes the presence of parallelism in computing, namely the integration of computing resources into a parallel computing system. The implementation of such a system is also possible on a single physical device, for example, a server rack or a supercomputer [22]. The transition point between distributed computing and cloud computing can be considered grid computing [23]. The main difference between CC is its conceptualization [24]. Unlike distributed and grid computing, which are primarily a set of tools and methods for solving computationally complex problems, CC is primarily a service to provide opportunities for the implementation of elastic computing. It is customary to classify CC according to the types of models within which this service is provided:

• Software as a Service (SaaS) – this model involves using a cloud infrastructure to implement CC by providing the user with an application software package. The control and management of the infrastructure is carried out exclusively by the service provider [25];

• Platform as a Service (PaaS) – in this case, the user is presented with an infrastructure for hosting var-

ious basic software. These are usually tools for creating software and, for example, database management systems. The provider can also provide various environments for working with programming languages [26];

• Infrastructure as a Service (IaaS) – in this case, the provider provides the user with the most complete rights to use the cloud. Here, the user is offered a basic infrastructure within which he independently organizes the processes of managing computing resources, as well as building a network and storing data. The user also independently controls the operating systems that are deployed in the cloud. Another difference from the previous categories is that the user is given limited control over the network services of the cloud space allocated to him [27, 28].

Based on these three models, various hybrid ones are also distinguished, such as Data Base as a Service [29], Monitoring as a Service [30], etc., however, hybrid types, in fact, are divided in terms of customer needs and are rather aimed at narrow specialization of the application of a particular type of CC. Based on the above information, it is possible to highlight the advantages that CC provides for AI. As already mentioned, AI methods are quite computationally complex. When conducting research and developing applications and services related to AI, the researcher/developer faces difficulties related to limited computing resources. In the case of a researcher, in theory, you can use the distributed computing system of your institution, but not every institution has its own powerful computing system, or access to it is difficult. CC's allow a researcher/developer to rent computing resources from a vendor.

2.3. Cloud Computing and Artificial Intelligence

Let's consider the applicability of various CC models for AI. In general, AI can be deployed within any of the three service delivery models. However, there are several nuances, in the case of SaaS, the supplier, in addition to computing resources, also supplies AI, in this case AI is also a service. An example of such a service is MLaaS [31]. In the case of PaaS, the provider provides the user, in addition to computing power, with AI development tools. Various cloud services for software development can be included in this category. A striking example is Google Colab [32]. Despite the fact that this solution is not specialized, it still provides the user with opportunities for AI development and research. At the same time, there are solutions aimed specifically at working with AI [33]. IaaS and AI are quite difficult to distinguish into a specialized category, since in this case the service provider provides exclusively computing resources and the infrastructure connecting them. In this case, specialization in AI is achieved due to two factors: due to the fact that the hardware of the technical component of CC is selected in such a way that AI works as efficiently as possible; as well as due to the very position of the service provider [34]. In the cases considered, AI acts as an object, either being part of a service, or being an actual consumer of computing power, however, there are works that consider other possibilities for using AI in CC [35–38]. For example, AI can be used to build infrastructure and load balancing.

2.4. Data Privacy in Cloud Computing

If the projects, as in the cases [1, 2], are open, i.e. they do not initially contain confidential information, then the requirements for their security are low. However, any AI can be used in tasks related to the processing of confidential data. It could be medical information [39–44], banking [45–48], government [49, 50] and all kinds of personal data. In these cases, a situation arises in connection with which CC is often criticized both in society as a whole and in the scientific community. When processing any information in the cloud, both parties have access to it – the user and the service provider. In this case, the degree of data confidentiality is determined by an agreement between the user and the provider, and how this agreement is respected is the responsibility of the service provider. It is not uncommon for confidential data stored or processed by various service providers to be compromised. In this case, the user is forced to give preference to suppliers who have either proven themselves to be reliable in terms of ensuring the confidentiality of the processed data, or claim a low probability of compromise through the use of effective protection methods. This raises a number of questions. Which privacy method is effective for storing data? What methods are capable of providing the necessary level of confidentiality in the context of data processing using AI in general and Big Data in particular?

The purpose of this work is to answer the questions asked and provide the reader with the opportunity to familiarize themselves with current methods of ensuring the confidentiality of CC using AI.

For a more visual demonstration of the retrospective review of distributed computing technologies and AI, a table was compiled (Table 1). The above works are not presented in chronological order, Table 1 is designed to restore it. Note that some technologies were first introduced much earlier than they were developed and studied in full.

Analyzing the data in Table 1 and the overview presented above, it can be noted that theoretical AI models have been actively developing since the middle of the XX century, but practical models were developed in the late XX–early XXI centuries. The table also shows that the development of practical AI models

Table 1. Historical background on the technologies considered

Year	Distributed computing	Cloud technologies	Artificial intelligence
1943	–	–	NN concept [51], [52]
1954	–	–	The origin of genetic algorithms [53]
1959	–	–	Machine learning [54]
1962	The collective computing model [55]	–	–
1966	–	–	The emergence of language models [56]
1978	Principles of distribution of work between processors [57]	–	–
1980	–	–	Theoretical description DL [58]
1992	The origin of the GRID [59]	–	–
1996	GIMPS Integer Search Project [60]	–	–
1999	The SETI project based on BOINC [61]	–	–
2000	–	–	The beginning of the practical application of DL [62]
			Computer vision [63]
2006	–	The birth of the concept of cloud computing [64]	–
2008	–	Defining the concept of cloud computing as a service [65]	–
2009	–	Launching Google Apps [66]	–
2011	–	Standardization of SaaS, PaaS and IaaS as service models in CC [25], [26], [27]	–
2015	–	The development of foggy computing as the basis for the Internet of Things [67]	–
		Launching OpenFog [68]	
2018	–	–	GPT [69]

coincides with the late stages of the development of distributed computing (until the end of the 20th century) and the emergence of CC at the beginning of the 21st century. This coincidence is not accidental and is due to the fact that the development of distributed and cloud computing all this time has been on the path of aggregating more and more computing power, which was so lacking in large AI models. Thus, it is fair to say that distributed computing has made a significant contribution to the development of AI models, and CC allows us to develop and create modern computationally expensive AI models, the complexity of which is growing every day.

Given the above, it can be argued that the issue of CC privacy has the same importance for AI as for any CC "consumer," which AI essentially is. Next, the methods of ensuring the confidentiality of AI in CC will be considered.

3. MODERN METHODS AND ALGORITHMS FOR ENSURING AI PRIVACY

3.1. Differential Privacy

The concept of confidentiality is somewhat lax. Depending on the situation, it can be viewed from different angles. For example, from the point of view of medicine, it is enough that the data is anonymous, i.e., the medical history is depersonalized. In [70], a study of DL methods for ensuring soft confidentiality is carried out. The authors achieve their goal by introducing differential confidentiality, i.e. by mixing private depersonalized data together with synthesized data. The authors have demonstrated the high accuracy of this method. However, this method has a number of disadvantages: the basis of confidentiality is the introduction of additional noise and distances between points during gradient descent. This not only increases the computational complexity of calculations, but also

does not ensure proper data security, since if an attacker gets access to the data during processing, he will be able to remove unnecessary noise.

On the other hand, [71] considers a fundamentally different approach to ensuring data confidentiality from the above. The authors consider the possibility of training and using a neural network by a group of users without the need to disclose confidential data to each other. This possibility is provided by the feature of stochastic gradient descent, which allows it to be performed in parallel and asynchronously. The authors also claim that their solution allows participants to train the neural network independently on their own (confidential) sets by sharing subsets of key parameters. In general, this method allows you to ensure confidentiality among the group members, however, it turns out to be ineffective in the case of prior collusion of the participants or an external attack.

3.2. Secret Sharing Schemes

In the context of building confidential AI in CC, it is also necessary to consider secret sharing schemes (SSS) [4]. Here are briefly the main theoretical aspects related to SSS. The secret S is divided by dealer D between N participants in such a way that to decrypt any information, it will be necessary to combine the shares S_i of all participants in the scheme back into a secret, $i \in 1, 2, ..., N$, where N is the number of participants. If the secret is, for example, the key for the encryption scheme, the confidentiality of the system increases.

There are two types of secret separation schemes: full [4] and threshold [72]. The full ones assume that all the fractions of the secret are needed for recovery, the threshold ones mean that a certain number of fractions are needed, but not all. Threshold secret sharing schemes are used much more often in practice due to their flexibility. Threshold SSS allow you to reduce the computational load, while the threshold is set so that an attacker cannot master the required number of shares of the secret, or so that a preliminary collusion of the required number of participants cannot take place.

In [73], the authors propose a distributed DL when participants train a neural network using their confidential sets. The authors cite the results of a study in which it was possible to maintain confidentiality with distributed DL in the cloud with untrusted participants. The work demonstrates the functionality provided by SSS when working with AI, and the very fact of the possibility of confidential training. The authors apply Shamir's scheme [72]. During its existence, the scheme has proven its suitability from a security point of view. However, if we consider the application of SSS methods from the CC point of view, then security problems are also overlaid with reliability problems, additional corrective codes impose additional loads on the system. To offset the need for additional correction codes, SSS based on the residue number system (RNS) [74] can be used, for example, such as SSS Asmut-Bloom [75] and SSS Mignotte [76].

The paper [77] considers the organization of federated training of neural networks in cloud systems using SSS Asmut-Bloom to ensure confidentiality [78]. The authors also consider the SSS Mignotte, but there are works proving its unsuitability for security.

In [79], the authors propose a protocol for confidential data transmission in conditions of working with neural networks. The transmission protocol is based on SSS. The main focus of the article is on the speed of data processing while maintaining their confidentiality. The authors show the effectiveness of their solution in comparison with some methods of homomorphic encryption, which is also applicable when building confidential AI in CC.

3.3. Homomorphic Encryption

The first works on homomorphic encryption (HE) appeared several decades ago. It allows the processing of encrypted data without the need for a decryption operation. Homomorphic encryption schemes can be homomorphic in addition, multiplication, or both of these arithmetic operations simultaneously. This feature is typical for various asymmetric ciphers. For example, homomorphic addition is supported by schemes presented in [80, 81], and homomorphic multiplication by schemes from [82, 83]. It is worth noting that such encryption schemes are still being used, for example, in [84] the authors use a modified El Gamal scheme to train a neural network, based on the fact that the El Gamal scheme is homomorphic in multiplication. The authors use a linear approximation of the activation function. This method is quite narrowly focused and difficult to scale.

In 2009, Gentry developed a fully homomorphic encryption scheme (FHE) [85]. This scheme was not computationally efficient enough, but further research by its followers allowed to increase efficiency to a level sufficient for real practical application of FHE [86–91]. FHE allows you to perform both homomorphic addition and multiplication (however, it has a limit on the number of multiplications with one key), in addition, there are schemes that allow you to perform polynomial operations on ciphertexts [92].

Such a set of operations allows you to implement a large number of AI algorithms, which was quickly discovered by many AI researchers. Let's consider the methods they developed.

CryptoNets [93]. In this solution, the authors propose the use of GS in the operation of confidential cloud storage with the possibility of processing by neural networks. The main result achieved is the accuracy of the neural network for pattern recognition, equal to 99%.

In the case of SEALion [94], a solution based on TensorFlow [95] and SEAL [96] is proposed. TensorFlow implements calculations in tensors, SEAL itself GS. With the help of the developed solution, the authors implement convolutional neural networks (CNN), the method of support vectors is used as a learning method. The paper presents several NN models, the accuracy is determined based on the quality of digital image recognition. The authors show that their solutions are more effective than, for example, in [93]. A similar solution is TenSEAL [97], this library provides opportunities to work with ML, NN, and CNN in conjunction with GS. As a disadvantage, difficulties can be identified when learning from encrypted data, while the trained NN demonstrates performance and gives accurate results.

In [98], variations of confidential CNNs with GS are considered. The authors propose a CNN, which uses the integer scheme FHE – BGV [99]. The main focus is on the possibility of calculating the value of the ReLU activation function from ciphertexts obtained according to the BGV scheme. The authors defined three requirements for NN – accuracy, confidentiality and efficiency, the result of the work is a polynomial approximate ReLU function, which allowed to reduce the multiplicative depth, thereby increasing the efficiency of the network. This work can be considered quite important for several reasons: firstly, the very fact of the possibility of completely confidential calculations is shown; secondly, the possibility of building and using neural networks with FHE is shown.

In [100], the confidentiality-preserving DL is considered. The authors contrast their research with many of the works that were reviewed earlier, focusing on eliminating the shortcomings that were voiced above. The main result is based on the application of approximation methods for various activation functions, for example, such as ReLU and Softmax. To provide a large number of homomorphic multiplications in the CKKS scheme, the authors propose their modification of the bootstrapping procedure, which is based on the fact that the CKKS scheme [92] uses a system of residual classes to speed up arithmetic calculations. This result is interesting because it increases the speed of GS methods for specific tasks, whereas increasing the speed and applicability of GS in general is the main focus of research in this area at present.

GS allows you to solve the security problems of CC and CT in general by creating a transparent environment for free and confidential computing in the clouds. The analysis of GS methods in AI allows us to state that many researchers hold this opinion. The above FHE schemes are considered, which are more based on cryptographic lattices, however, there are FHE schemes that are built on SSS. This approach allows you to expand the capabilities of FHE in CT. Next, hybrid models that combine FHE and SSS methods are considered.

3.4. Hybrid Systems of Homomorphic Encryption and Secret Sharing Schemes

Considering the results in the direction of increasing AI privacy obtained through the use of SSS and FHE, the idea of creating a hybrid FHE-SSS system looks promising. Such a hybrid system will enhance the security of SSS by processing information in encrypted form using FHE in the space of a single local node. SSS-based security algorithms are used to transfer data between nodes. Thus, it becomes possible to adjust the balance, shifting the focus towards safety or productivity.

In [101], a solution was proposed that uses SSS and FHE to implement a confidentiality-preserving NN. The authors describe various algorithms used in this system, for example, security algorithms, key generation, SSS operation, etc., and also analyze in detail how NN works with this approach to security. The main focus is on demonstrating the effectiveness, safety, accuracy of the NN obtained and its comparison with other solutions, however, the data obtained are theoretical. The considered work is remarkable in that, despite the lack of specifics in terms of the FHE and SSS used, it shows the very possibility of implementing such a system and describes its characteristics, encouraging other researchers to engage in developments in this field.

4. ANALYSIS OF THE RESULTS OBTAINED

In the course of the study, three groups of methods for ensuring the confidentiality of artificial intelligence were analyzed (Table 2). The data presented below are based on information provided in the reviewed works.

Analyzing the results obtained (Table 2), we can summarize the following.

Differential confidentiality methods are primarily based on changes in the training sample. It is proposed to add unnecessary noise to the data, due to which the degree of confidentiality increases, while at the same time seriously reducing the efficiency of calculations. In lines 1 and 2, the values of computational complexity are highlighted in bold. In the first case, ε – means additional noise. In the second case, δ is the number of asynchronous nodes. Taking into account the level of confidentiality achieved in this case, additional security measures are required when transferring data. During processing, the degree of confidentiality is average – in order to steal data, an attacker will either need to filter the information from noise, or take control of several nodes in order to track the data flow. In addition, the need to apply corrective codes to avoid collisions or data loss will have an additional impact on the effectiveness of the system.

Table 2. Results of analytical review of methods

No.	Method	Computational complexity	Confidentiality of data transmission	Confidentiality of data processing	Ensuring reliability	The considered AI method
1	Modified gradient descent	$O(D^2N + (D + \varepsilon)^3)$	Low	Average	Absent	AC-GAN [70]
2	Asynchronous gradient descent	$O(D^2(N\delta) + D^3)$	Low	Average	Absent	CNN [71]
3	Shamir's SSS	$O(N^2)$	High	Low	Absent	DL [73]
4	SSS of Asmut-Bloom	$O(\log_2^2(N) + \log_2(N^2))$	High	Low	RNS Correction Codes	Federated learning [77], [79]
5	SSS Mignotte	$O(N^2)$	Low	Low	RNS Correction Codes	Federated learning [77]
6	BFV	$O(N\log N)$	High	High	RNS Correction Codes (in theory)	NN, CNN [93], [94]
7	BGV	$O(N\log N)$	High	High	Absent	NN, CNN [98]
8	CKKC	$O(N\log N)$	High	High	RNS Correction Codes (in theory)	NN, CNN, DL [93], [94], [97], [100]
9	Hybrid SSS-FHE	$O(N\log N)$	*High*	*High*	*RNS Correction Codes*	*NN*, *CNN*, *DL* [101]

In the case of SSS, RNS-based circuits can improve the reliability of the system by using the self-correcting properties of RNS. However, confidentiality during data processing must be ensured by additional encryption methods. Also, in this case, to intercept data, an attacker will need to take control of the threshold number of nodes. When building SSS, the threshold for secret recovery is calculated so that the time spent compromising the shares of the secret is greater than the time of data relevance.

FHE allows you to ensure complete confidentiality. However, it has a significant drawback. Despite the fact that the computational complexity of encryption is low, compared with other algorithms, the computational complexity of data processing is much higher, which the authors of the schemes do not hide.

The last of the categories considered were hybrid SSS-FHE methods. At the moment, they are represented only by theoretical models, however, analyzing them, it is possible to assess the possible characteristics of the system.

The conducted research has shown that different research groups are interested in developing an AI with a high level of confidentiality. However, at the moment there are no relevant methods to achieve this level. In the future, it is planned to study the most promising methods based on SSS-FHE from the point of view of ensuring confidentiality. In particular, it is planned to develop NN using the FHE CKKS scheme, as well as Asmut-Bloom SSS. Justifying the choice of these algorithms, it can be noted that among the many FHE schemes, it is CKKS that is of the greatest interest to researchers, and among the many SSS based on RNS, it is Asmut-Bloom SSS that has the best characteristics in terms of security. The use of RNS in CKKS will increase the efficiency of the solution, and in Asmut-Bloom's SSS – the reliability of data processing. An important part of the study will be to determine the accuracy of the calculation results, due to the integer nature of RNS, as well as the approximation error in CKKS.

CONCLUSIONS

In this paper, the methods of constructing a confidentiality-preserving AI in CC were investigated. At the first stage of the study, an analytical review of both AI and CC methods was conducted. Based on the results of the review, the criteria for AI safety in CC were determined. Next, the second stage of the analytical review was performed, namely, a review of AI privacy methods, on the basis of which 4 groups of methods were identified:

- differential privacy;
- secret sharing schemes;
- homomorphic encryption;
- hybrid methods based on SSS-FHE.

Based on the results of the study, the positive and negative sides of the considered methods were identified, an idea of the current state of the problem of

ensuring AI confidentiality in CC was formed, and approaches to its solution were outlined. An analytical review has shown that there is currently no relevant solution. Solutions that provide the highest level of confidentiality have low efficiency due to the need to perform complex calculations. FHE supports addition and multiplication operations on encrypted values, such operations as determining the sign of a number, division, and matrix operations are not fully implemented. SSS does not allow you to process information in encrypted form. Confidentiality is ensured by not disclosing the source of a part of the data set of one participant to others. Differential confidentiality ensures anonymity during the operation of the neural network, while it has no protection against data interception. A hybrid method based on SSS-FHE is theoretically established as a possible solution, but it requires detailed research.

In future works, SSS-FHE research will be carried out, namely, the development of a prototype and the study of its characteristics.

FUNDING

This work was supported by the Russian Science Foundation 19-71-10033, https://rscf.ru/project/19-71-10033/.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. Brown, T. et al., Language models are few-shot learners, *Adv. Neural Inf. Process. Syst.,* 2020, vol. 33, pp. 1877–1901.
2. *OpenAI, GPT-4 Technical Report,* March 27, 2023. https://doi.org/10.48550/arXiv.2303.0877
3. Douligeris, C. and Mitrokotsa, A., DDoS attacks and defense mechanisms: classification and state-of-the-art, *Comput. Networks,* 2004, vol. 44, no. 5, pp. 643–666.
4. Beimel, A., Secret-sharing schemes: a survey, in *Coding and Cryptology,* Chee, Y.M., Guo, Z., Ling, S., Shao, F., Tang, Y., Wang, H., and Xing, C., Eds., Berlin, Heidelberg: Springer, 2011, pp. 11–46. https://doi.org/10.1007/978-3-642-20901-7_2
5. Mahesh, B., Machine learning algorithms-a review, *Int. J. Sci. Res.,* 2020, vol. 9, no. 1, pp. 381–386.
6. Kaelbling, L.P., Littman, M.L., and Moore, A.W., Reinforcement learning: a survey, *J. Artif. Intellig. Res.,* 1996, vol. 4, pp. 237–285.
7. Srinivas, M. and Patnaik, L.M., Genetic algorithms: a survey, *Computer,* 1994, vol. 27, no. 6, pp. 17–26.
8. Spragins, J., Learning without a teacher, *IEEE Trans. Inf. Theory,* 1996, vol. 12, no. 2, pp. 223–230.
9. Liu, B., Supervised learning, in *Web Data Mining,* Berlin, Heidelberg: Springer, 2011, pp. 63–132. https://doi.org/10.1007/978-3-642-19460-3_3
10. Wang, S.-C., Artificial neural network, in *Interdisciplinary Computing in Java Programming,* Boston: MA: Springer US, 2003, pp. 81–100. https://doi.org/10.1007/978-1-4615-0377-4_5
11. Park, H. and Kim, S., Chapter three – hardware accelerator systems for artificial intelligence and machine learning, *Adv. Comput.,* 2021, vol. 122, pp. 51–95. https://doi.org/10.1016/bs.adcom.2020.11.005
12. Hwang, D.H., Han, C.Y., Oh, H.W., and Lee, S.E., ASimOV: a framework for simulation and optimization of an embedded AI accelerator, *Micromachines,* 2021, vol. 12, no. 7. https://doi.org/10.3390/mi12070838
13. Mishra, A., Yadav, P., and Kim, S., Artificial intelligence accelerators, in *Artificial Intelligence and Hardware Accelerators,* Mishra, A., Cha, J., Park, H., and Kim, S., Eds., Cham: Springer Int. Publ., 2023, pp. 1–52. https://doi.org/10.1007/978-3-031-22170-5_1
14. Carminati, M. and Scandurra, G., Impact and trends in embedding field programmable gate arrays and microcontrollers in scientific instrumentation, *Rev. Sci. Instrum.,* 2021, vol. 92, no. 9. https://pubs.aip.org/aip/rsi/article-abstract/92/9/091501/1030652.
15. Shawash, J. and Selviah, D.R., Real-time nonlinear parameter estimation using the Levenberg-Marquardt algorithm on field programmable gate arrays, *IEEE Trans. Ind. Electron. Control Instrum.,* 2012, vol. 60, no. 1, pp. 170–176.
16. Ruiz-Rosero, J., Ramirez-Gonzalez, G., and Khanna, R., Field programmable gate array applications-a scientometric review, *Computation,* 2019, vol. 7, no. 4, p. 63.
17. Mellit, A. and Kalogirou, S.A., MPPT-based artificial intelligence techniques for photovoltaic systems and its implementation into field programmable gate array chips: review of current status and future perspectives, *Energy,* 2014, vol. 70, pp. 1–21.
18. Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning,* MIT Press, 2016. https://books.google.com/books?hl=ru&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&dq=Deep+Learning&ots=MNV5aolzSS&sig=waXAS6C-_v-48H2qbW9rMFkEhFY.
19. Bouvrie, J., Notes on convolutional neural networks, 2006. http://web.mit.edu/jvb/www/papers/cnn_tutorial.pdf.
20. Rawat, W. and Wang, Z., Deep convolutional neural networks for image classification: a comprehensive review, *Neural Comput.,* 2017, vol. 29, no. 9, pp. 2352–2449.
21. Needham, R.M. and Herbert, A.J., *The Cambridge Distributed Computing System,* Cambridge, 1983.
22. Adiga, N.R. et al., An overview of the BlueGene/L supercomputer, *Proc. ACM/IEEE Conf. on Supercomputing, SC'02,* Baltimore, MD, 2002, p. 60. https://ieeexplore.ieee.org/abstract/document/1592896/.
23. Jacob, B., Brown, M., Fukui, K., and Trivedi, N., Introduction to grid computing, in *IBM Redbooks,* 2005, pp. 3–6.
24. Foster, I., Zhao, Y., Raicu, I., and Lu, S., Cloud computing and grid computing 360-degree compared, *Proc. IEEE Grid Computing Environments Workshop,* Austin,

TX, 2008, pp. 1–10. https://ieeexplore.ieee.org/abstract/document/4738445/?casa_token=TbNOHOEaljQAAAAA:j6MuEJKmrGL8iCvH-HzRnmI2k5UKn5y1w7hC4MNJanJXZPfiBC_XKLoTFsCImP1RYzyKfRKiCE0.

25. Cusumano, M., Cloud computing and SaaS as new computing platforms, *Commun. ACM,* 2010, vol. 53, no. 4, pp. 27–29. https://doi.org/10.1145/1721654.1721667
26. Rodero-Merino, L., Vaquero, L.M., Caron, E., Muresan, A., and Desprez, F., Building safe PaaS clouds: a survey on security in multitenant software platforms, *Comput. Secur.,* 2012, vol. 31, no. 1, pp. 96–108.
27. Bhardwaj, S., Jain, L., and Jain, S., Cloud computing: a study of infrastructure as a service (IAAS), *Int. J. Eng. Inf. Technol.,* 2010, vol. 2, no. 1, pp. 60–63.
28. Manvi, S.S. and Shyam, G.K., Resource management for infrastructure as a service (IAAS) in cloud computing: a survey, *J. Network Comput. Appl.,* 2014, vol. 41, pp. 424–440.
29. Lehner, W. and Sattler, K.-U., Database as a service (DBaaS), *Proc. IEEE 26th Int. Conf. on Data Engineering (ICDE 2010),* Long Beach, CA, 2010, pp. 1216–1217. https://ieeexplore.ieee.org/abstract/document/5447723/?casa_token=uaXogPZV0C0AAAAA:4Dg_40-GvhUsuHXFKUOgxZ_ZyGlCOqjcZtpRoK6UosB-k-_Wh5wAmJIBtHYRE9OLXZ1xwVKuLAE.
30. Meng, S. and Liu, L., Enhanced monitoring-as-a-service for effective cloud management, *IEEE Trans. Comput.,* 2012, vol. 62, no. 9, pp. 1705–1720.
31. Weng, Q., et al., {MLaaS} in the wild: workload analysis and scheduling in {Large-Scale} heterogeneous {GPU} clusters, *Proc. 19th USENIX Symp. on Networked Systems Design and Implementation (NSDI 22),* Renton, WA, 2022, pp. 945–960. https://www.usenix.org/conference/nsdi22/presentation/weng.
32. Bisong, E., Google colaboratory, in *Building Machine Learning and Deep Learning Models on Google Cloud Platform,* Berkeley: CA: Apress, 2019, pp. 59–64. https://doi.org/10.1007/978-1-4842-4470-8_7
33. H2O AI Cloud. https://h2o.ai/platform/ai-cloud/.
34. NVIDIA NGC | NVIDIA. https://www.nvidia.com/en-us/gpu-cloud/.
35. Tang, J., Artificial intelligence-based e-commerce platform based on SaaS and neural networks, *Proc. 4th IEEE Int. Conf. on Inventive Systems and Control (ICISC),* Seoul, 2020, pp. 421–424. https://ieeexplore.ieee.org/abstract/document/9171193/?casa_token=TmYwFdLDXq0AAAAA:8P5VVcZS_KWCXEnEm8xk2RPMV5kfWF27K9S9O9Z5fYh273EkseT7j0Jf7jZYAMOnPUX0l-5sCbs.
36. Yathiraju, N., Investigating the use of an artificial intelligence model in an ERP cloud-based system, *Int. J. Electr., Electron. Comput.,* 2022, vol. 7, no. 2, pp. 1–26.
37. Mishra, S. and Tripathi, A.R., AI business model: an integrative business approach, *J. Innov. Entrepreneur,* 2021, vol. 10, no. 1, p. 18. https://doi.org/10.1186/s13731-021-00157-5
38. Mishra, D. and Shekhar, S., Artificial intelligence candidate recruitment system using software as a service (SaaS) architecture, *Int. Res. J. Eng. Technol.,* 2018, vol. 05, no. 05, pp. 3804–3808.
39. Cadario, R., Longoni, C., and Morewedge, C.K., Understanding, explaining, and utilizing medical artificial intelligence, *Nat. Hum. Behav.,* 2021, vol. 5, no. 12, pp. 1636–1642.
40. Kim, M., Song, Y., Wang, S., Xia, Y., and Xiang, X., Secure logistic regression based on homomorphic encryption: design and evaluation, *JMIR Med. Inf.,* 2018, vol. 6, no. 2, p. e8805.
41. Klonoff, D.C., Fog computing and edge computing architectures for processing data from diabetes devices connected to the medical internet of things, *J. Diabetes Sci. Technol.,* 2017, vol. 11, no. 4, pp. 647–652.
42. Kocabas, O. and Soyata, T., Utilizing homomorphic encryption to implement secure and private medical cloud computing, *Proc. 8th IEEE Int. Conf. on Cloud Computing,* New York, 2015, pp. 540–547.
43. Liu, R., Rong, Y., and Peng, Z., A review of medical artificial intelligence, *Global Health J.,* 2020, vol. 4, no. 2, pp. 42–45.
44. Sun, X., Zhang, P., Sookhak, M., Yu, J., and Xie, W., Utilizing fully homomorphic encryption to implement secure medical computation in smart cities, *Pers. Ubiquitous Comput.,* 2017, vol. 21, no. 5, pp. 831–839.
45. Kaya, O., Schildbach, J., Ag, D.B., and Schneider, S., Artificial intelligence in banking, in *Artificial Intelligence,* 2019. https://www.dbresearch.com/PROD/RPS_EN-PROD/PROD0000000000495172/Artificial_intelligence_in_banking%3A_A_lever_for_pr.pdf.
46. Rahman, M., Ming, T.H., Baigh, T.A., and Sarker, M., Adoption of artificial intelligence in banking services: an empirical analysis, *Int. J. Emerging Markets,* 2021. https://www.emerald.com/insight/content/doi/10.1108/IJOEM-06-2020-0724/full/html.
47. Sadok, H., Sakka, F., and El Maknouzi, M.E.H., Artificial intelligence and bank credit analysis: a review, *Cogent Econ. Fin.,* 2022, vol. 10, no. 1, p. 2023262. https://doi.org/10.1080/23322039.2021.2023262
48. Smith, A. and Nobanee, H., Artificial intelligence: in banking a mini-review. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3539171.
49. Reis, J., Santo, P.E., and Melão, N., Artificial intelligence in government services: a systematic literature review, in *New Knowledge in Information Systems and Technologies,* Rocha, Á., Adeli, H., Reis, L.P., and Costanzo, S., Eds., Cham: Springer Int. Publ., 2019, vol. 930, pp. 341–259. https://doi.org/10.1007/978-3-030-16181-1_23
50. Valle-Cruz, D., Alejandro Ruvalcaba-Gomez, E., Sandoval-Almazan, R., Ignacio Criado, J., A review of artificial intelligence in government and its potential from a public policy perspective, in *Proc. 20th Annu. Int. Conf. on Digital Government Research,* Dubai: ACM, 2019, pp. 91–99. https://doi.org/10.1145/3325112.3325242
51. Pitts, W., The linear theory of neuron networks: the dynamic problem, *Bull. Math. Biophys.,* 1943, vol. 5, pp. 23–31.
52. Khare, S.S. and Gajbhiye, A.R., Literature review on application of artificial neural network (ANN) in oper-

ation of reservoirs, *Int. J. Comput. Eng. Res.,* 2013, vol. 3, no. 6, p. 63.

53. Seesing, A., Evotest: test case generation using genetic programming and software analysis, *Oper. Res.,* 1954, vol. 2, pp. 393–410.
54. Samuel, A.L., Machine learning, *Technol. Rev.,* 1959, vol. 62, no. 1, pp. 42–45.
55. Evreinov, Ė.V. and Kosarev, I., *Odnorodnye universal'nye vychislitel'nye sistemy vysokoi proizvoditel'nosti* (Uniform High Efficiency Computating Systems), Novosibirsk: Nauka, 1966. https://cir.nii.ac.jp/crid/1130282272859765760.
56. Gold, E.M., Language identification in the limit, *Inf. Control,* 1967, vol. 10, no. 5, pp. 447–474.
57. Glushkov, V.M., Computating system, 1996. https://elibrary.ru/item.asp?id=41074434.
58. Huang, X., Deep-learning based climate downscaling using the super-resolution method, *Preprint,* 1981. https://pdfs.semanticscholar.org/cf5c/3b29559ababba5a889444632e1c91d6b78fc.pdf.
59. Smarr, L. and Catlett, C.E., Metacomputing, in *Grid Computing,* Berman, F., Fox, G., and Hey, T., Eds., 1st ed., Wiley, 2003, pp. 825–835. https://doi.org/10.1002/0470867167.ch37
60. Buske, D. and Keith, S., GIMPS finds another prime!, *Math Horizons,* 2000, vol. 7, no. 4, pp. 19–21. https://doi.org/10.1080/10724117.2000.11975124
61. Anderson, D.P., Boinc: a system for public-resource computing and storage, *Proc. 5th IEEE/ACM Int. Workshop on Grid Computing,* Pittsburgh, PA, 2004, pp. 4–10. https://ieeexplore.ieee.org/abstract/document/1382809/?casa_token=cjAKtADFAKwAAAAA:-WGH_xmovZAUi-kr_PA-h3nXtuizBL829DPFlC0B6pbcCoApRKDCZLwFWxzfYdT0WauFC5c6EQw1
62. Du, T. and Shanker, V., Deep learning for natural language processing, *Brain Nerve,* 2019, vol. 71, no. 1, pp. 45–55.
63. Davies, E.R., *Machine Vision: Theory, Algorithms, Practicalities,* Elsevier, 2004. https://books.google.com/books?hl=ru&lr=&id=uYZ3vORugwC&oi=fnd&pg=PP1&dq=Machine+Vision+:+Theory,+Algorithms,+Practicalities&ots=QOl9U9_MBf&sig=w0poN6d3IGeXs4oacagO4MlnxYs.
64. Mell, P. and Grance, T., The NIST definition of cloud computing, *Natl. Inst. Stand. Technol. Spec. Publ.,* 2011, vol. 53, pp. 1–7.
65. Finkelstein, R., Analyzing trend of cloud computing and it's enablers using Gartner strategic technology, 2004. https://www.researchgate.net/profile/Amol-Adamuthe/publication/308747055_Analyzing_Trend_of_Cloud_Computing_and_it's_Enablers_using_Gartner_Strategic_Technology/links/59a929d3a6fdcc2398414d6f/Analyzing-Trend-of-Cloud-Computing-and-its-Enablers-using-Gartner-Strategic-Technology.pdf.
66. A history of cloud computing, *Computer Weekly*. https://www.computerweekly.com/feature/A-history-of-cloud-computing.
67. Dolui, K. and Datta, S.K., Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing, *Proc. IEEE Global Internet of Things Summit (GIoTS),* Geneva, 2017, pp. 1–6.
68. OpenFog, OPC Foundation. https://opcfoundation.org/markets-collaboration/openfog/.
69. Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I., Improving language understanding by generative pre-training, 2018. https://www.mikecaptain.com/resources/pdf/GPT-1.pdf.
70. Beaulieu-Jones, B.K. et al., Privacy-preserving generative deep neural networks support clinical data sharing, *Circ: Cardiovasc. Qual. Outcomes,* 2019, vol. 12, no. 7, p. e005122. https://doi.org/10.1161/CIRCOUTCOMES.118.005122
71. Shokri, R. and Shmatikov, V., Privacy-preserving deep learning, *Proc. 22nd ACM SIGSAC Conf. on Computer and Communications Security,* Denver CO, Oct. 2015, pp. 1310–1321. https://doi.org/10.1145/2810103.2813687
72. Shamir, A., How to share a secret, *Commun. ACM,* 1979, vol. 22, no. 11, pp. 612–613.
73. Duan, J., Zhou, J., and Li, Y., Privacy-preserving distributed deep learning based on secret sharing, *Inf. Sci.,* 2020, vol. 527, pp. 108–127.
74. Akushsky, I.A. and Yuditsky, D.I., *Mashinnaya arifmetika v ostatochnykh klassakh* (Modular Arithmetic in Residue Classes), Moscow: Sovetskoe radio, 1968.
75. Bloom, J., A modular approach to key safeguarding, *IEEE Trans. Inf. Theory,* 1983, vol. 29, no. 2, pp. 208–210.
76. Mignotte, M., How to share a secret, *Proc. Workshop on Cryptography,* Springer, 1982, pp. 371–375.
77. Tian, T., Wang, S., Xiong, J., Bi, R., Zhou, Z., and Bhuiyan, M.Z.A., Robust and privacy-preserving decentralized deep federated learning training: focusing on digital healthcare applications, *IEEE/ACM Trans. Comput. Biol. Bioinformatics,* 2023. https://ieeexplore.ieee.org/abstract/document/10058838/.
78. Barzu, M., Ţiplea, F.L., and Drăgan, C.C., Compact sequences of co-primes and their applications to the security of CRT-based threshold schemes, *Inf. Sci.,* 2013, vol. 240, pp. 161–172.
79. Ge, Z., Zhou, Z., Guo, D., and Li, Q., Practical two-party privacy-preserving neural network based on secret sharing. http://arxiv.org/abs/2104.04709.
80. Paillier, P., Public-key cryptosystems based on composite degree residuosity classes, in *Proc. Conf. Advances in Cryptology – EUROCRYPT'99,* Stern, J., Ed., Berlin, Heidelberg: Springer, 1999, vol. 1592, pp. 223–238. https://doi.org/10.1007/3-540-48910-X_16
81. Benaloh, J., Dense probabilistic encryption, *Proc. Workshop on Selected Areas of Cryptography,* Kingston 1994, pp. 120–128. https://sacworkshop.org/proc/SAC_94_006.pdf.
82. Rivest, R.L., Shamir, A., and Adleman, L., A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM,* 1978, vol. 21, no. 2, pp. 120–126. https://doi.org/10.1145/359340.359342

83. ElGamal, T., A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Inf. Theory,* 1985, vol. 31, no. 4, pp. 469–472.
84. Chen, T. and Zhong, S., Privacy-preserving backpropagation neural network learning, *IEEE Trans. Neural Networks,* 2009, vol. 20, no. 10, pp. 1554–1564.
85. Gentry, C., *A Fully Homomorphic Encryption Scheme,* Stanford Univ., 2009.
86. Gentry, C., Computing arbitrary functions of encrypted data, *Commun. ACM,* 2010, vol. 53, no. 3, pp. 97–105.
87. Gentry, C. and Halevi, S., Implementing Gentry's fully-homomorphic encryption scheme, in *Proc. 30th Annu. Int. Conf. on the Theory and Applications of Cryptographic Techniques Advances in Cryptology-EUROCRYPT 2011,* Tallin, May 15–19, 2011, Springer, 2011, pp. 129–148.
88. Gentry, C., Halevi, S., Peikert, C., and Smart, N.P., Ring switching in BGV-style homomorphic encryption, in *Security and Cryptography for Networks,* Visconti, I. and de Prisco, R., Eds., Berlin, Heidelberg: Springer, 2012, vol. 7485, pp. 19–37. https://doi.org/10.1007/978-3-642-32928-9_2
89. Gentry, C., Sahai, A., and Waters, B., Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based, in *Proc. Annu. Conf. on Cryptology,* Springer, 2013, pp. 75–92.
90. van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V.V., Fully homomorphic encryption over the integers, in *Proc. Annu. Int. Conf. on the Theory and Applications of Cryptographic Techniques,* Springer, 2010, pp. 24–43.
91. van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V., Fully homomorphic encryption over the integers, in *Proc. Conf. Advances in Cryptology – EUROCRYPT 2010,* Gilbert, H., Ed., Berlin, Heidelberg: Springer, 2010. vol. 6110, pp. 24–43. https://doi.org/10.1007/978-3-642-13190-5_2
92. Cheon, J.H., Kim, A., Kim, M., and Song, Y., Homomorphic encryption for arithmetic of approximate numbers, in *Proc. Int. Conf. on the Theory and Application of Cryptology and Information Security,* Springer, 2017, pp. 409–437.
93. Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J., Cryptonets: applying neural networks to encrypted data with high throughput and accuracy, *Proc. Int. Conf. on Machine Learning,* New York, 2016, pp. 201–210. https://proceedings.mlr.press/v48/gilad-bachrach16.html.
94. van Elsloo, T., Patrini, G., and Ivey-Law, H., SEALion: a framework for neural network inference on encrypted data. http://arxiv.org/abs/1904.12840.
95. TensorFlow. https://www.tensorflow.org/?hl=ru.
96. Microsoft SEAL. https://github.com/microsoft/SEAL.
97. Benaissa, A., Retiat, B., Cebere, B., and Belfedhal, A.E., TenSEAL: a library for encrypted tensor operations using homomorphic encryption. http://arxiv.org/abs/2104.03152.
98. Chabanne, H., De Wargny, A., Milgram, J., Morel, C., and Prouff, E., Privacy-preserving classification on deep neural network, *Cryptol. ePrint Arch.,* 2017. https://eprint.iacr.org/2017/035.
99. Brakerski, Z., Gentry, C., and Vaikuntanathan, V., (Leveled) fully homomorphic encryption without bootstrapping, *ACM Trans. Comput. Theory (TOCT),* 2014, vol. 6, no. 3, pp. 1–36.
100. Lee, J.-W. et al., Privacy-preserving machine learning with fully homomorphic encryption for deep neural network, *IEEE Access,* 2022, vol. 10, pp. 30039–30054.
101. Ryffel, T., Tholoniat, P., Pointcheval, D., and Bach, F., ARIANN: low-interaction privacy-preserving deep learning via function secret sharing, Oct. 28, 2021. http://arxiv.org/abs/2006.04593.