Trustworthy Machine Learning using Secure Distributed Matrix Computation

Draft Document (for Review)

***Abstract*—Recent advancements in artificial intelligence (AI) and large language models (LLMs) have shown that AI models can exhibit human-level capabilities in performing some tasks. With such abilities, they have convinced researchers that further research on artificial intelligence should be performed more responsibly and by considering different dimensions of AI technologies. There are several aspects of artificial intelligence and machine learning that we should pay more attention to. Among others, trustworthiness and alignment are two important open problems in the fields of machine learning and artificial intelligence that seek effective and more reliable solutions.**

In this article we discuss how cryptographic constructions and techniques from secure computation can be adapted to enhance the trustworthiness of AI and ML models. Particularly we focus on constructions based on secure distributed computation and coded computing based on polynomials. Polynomials have shown to be very effective mathematical tools for designing solutions to different problems in various areas of engineering and technology. They are universal approximators and powerful encoding techniques, that can also behave as abstract or mathematical bridges connecting different engineering domains (e.g., secure computation and trustworthy machine learning).

***Index Terms*—Trustworthy Artificial Intelligence, Trustworthy Machine Learning, Secure Computation, Secure Distributed Matrix Computation, SDMC, Large Language Models, LLMs, Trustworthy AI, Trustworthy ML, AI Safety, ML Safety, Value Alignment, Alignment, AI, ML.**

I. Introduction

Recent advancements in artificial intelligence (AI) and particularly large language models (LLMs) have surprised most people by showing sort of human-level capabilities in doing different tasks (such as question-and-answering as well as text generation and summarization). With such capabilities, researchers, particularly the pioneers of AI, have started to speak out about the potential future risks of advanced AI tools/technologies. More importantly, they have been recommending that further research on AI should be carried out more responsibly by considering different dimensions of AI technologies and tools.

There are at least two main aspects of artificial intelligence that should be carefully researched. On one hand, AI tools and technologies can be used for a whole variety of good purposes. They can be used as search engines or digital assistants allowing people to benefit from their capabilities (e.g., for researchers and software developers to boost their productivity and to extract knowledge from huge text corpus from the Internet). They can potentially be used in education, in finance and e-commerce [1], and many other application domains [2]. Overall, AI technologies can potentially help the global economy boom significantly[1].

Besides the potentially beneficial applications mentioned above, there are contrary views that highlight the other side of AI technologies and how they might adversely affect humans' lives. Some believe that AI systems can eliminate many jobs[2] or can gradually replace humans in different sectors. Therefore, humans might go out of the loop in different businesses [3]. More importantly, there are various studies that highlight AI and ML tools come with different shortcomings and challenges such as algorithmic bias, fairness, ethical and data privacy concerns, hallucination, inaccuracy and unreliability in certain scenarios/environments [4].

The aforementioned issues related to AI and ML have mainly been formulated as the concepts of *trustworthiness* (or robustness) and *alignment* (or value alignment) in artificial intelligence and machine learning [5]. Trustworthiness and robustness have been the center of researchers' attention in the last couple of decades. There are various studies that discuss the problems related to trust in ML and AI systems; and there have been ongoing research on how trust-related issues in AI can be mitigated [6]. Alignment is another research problem in machine learning that has been more or less pointed out in the previous research [7]. With the hypes around recently introduced AI tools, e.g., OepnAI's ChatGPT and other ChatBots, alignment has been highlighted as an important problem more than before.

Motivated by the importance of these problems and that they might become more crucial in the near future, in this article we briefly review the important open problems in AI and ML. We also highlight some high-profile principles of AI. Having the AI principles in mind, we then discuss how cryptographic and mathematical tools can be used for alleviating trust-related issues in AI and ML and tackling relevant challenges. We particularly focus on polynomial coded computation and techniques from secure multiparty computation (MPC). Both secure computation and polynomial coded computing are well-known techniques and have attracted significant attention in the last decades.

[1] https://www.mckinsey.com/fi/news/generative-ai-holds-huge-economic-potential-for-the-global-economy

[2] https://www.businessinsider.com/ai-radically-reshape-job-market-global-economy-employee-labor-innovation-2023-8

Polynomials, in particular, have already proven that they can be very helpful and effective tools for solving problems in various areas of engineering and technology. Some prominent polynomial-based solutions with engineering applications include error-correcting codes (e.g., Reed-Solomon codes [8]), secret sharing (e.g., Shamir secret sharing [9]), verifiable secret sharing (VSS) [10], fully homomorphic encryption (FHE) [11], zero-knowledge proofs (ZKP) [12], arithmetization in zero-knowledge proof systems (e.g., R1CS [13] and QAP [14]), Lagrange coded computing (LCC) [15], private polynomial computing (PPC) [16], verifiable polynomial delegation (VPD) [17], [18], polynomial commitment schemes (e.g., the FRI protocol [19]), straggler mitigation in distributed computing [15], secure matrix multiplication [15], etc. Polynomials have also been studied in artificial neural networks, see e.g., polynomial neural network [20], deep polynomial neural networks [21], Lagrangian neural networks [22], and polynomial classifiers [23], etc.

In this article we highlight some of the well-known applications of polynomials, particularly those relevant to secure computation and trustworthy machine learning. We provide insights on how various mathematical properties of polynomials make them powerful tools for solving various problems in engineering and technology. Furthermore, we present the concept of secure distributed matrix computation (SDMC), which is a coded computing technique based on polynomials that extends the idea of secret sharing from sharing data to sharing both data and computation in a secure way.

The proposed secure distributed matrix computation (SDMC) scheme enables applying the principles of AI in real applications and also mitigating the drawbacks of AI tools to some extent. In particular, SDMC allows humans to be in the loop in multiparty computation and distributed AI (DAI) systems wherein humans and AI systems are collaboratively working toward completing a task. Even more, SDMC has built-in capability for guaranteeing the security and privacy of the shared data in such DAI systems.

A. Article Organization

This article is organized as follows. We continue by section II in which we highlight some of the important open problems and challenges of AI and ML. We also discuss some high-profile principles of AI (e.g., the Asilomar AI principles). Since in this article we leverage mathematical and cryptographic techniques as potential solutions for AI and ML related problems, in section III we highlight some of the interesting applications of polynomials in various areas of engineering and technology. The provided overview will be helpful for understanding the potentials of polynomials as powerful encoding techniques for bridging the gap between secure computation and trustworthy machine learning; and thus designing effective solutions for open problems in AI and ML. In section IV we present the notion of secure distributed matrix computation (SDMC), which in fact, is a polynomial-based solution for alleviating trust-related issues (or trustworthiness) in multiparty computation AI and ML applications. In section V we discuss some potential future research directions. The article is concluded in section VI.

II. Open Problems in AI and AI Principles

In this section we review some of the important open problems and challenges in artificial intelligence and machine learning safety. We also highlight some of the high-profile AI principles.

A. Open Problems in AI and ML Safety

There are various open problems in artificial intelligence and machine learning safety [5], which are expected to become more important in the near future. In this section we provide a brief overview of them.

The amount of research on AI and ML safety seems to be marginal; although, big AI companies have recently decided to invest more on trustworthiness and safety aspects of artificial intelligence and machine learning research and technologies. Nevertheless, the manuscript [5] is one of the works that categorizes open problems in machine learning safety into four broad categories, namely robustness, monitoring, alignment, and systemic safety [5]. Other important problems in AI and ML safety include, data security and privacy, algorithmic bias and fairness, verification and validation. The main theme behind most of these problems is how to build machine learning and artificial intelligence models that respect human values and we humans can truly trust them in various situations.

Table I summarizes some important open problems in AI and ML safety. This article aims at proposing mechanisms mainly for secure and trustworthy machine learning (SaTML). Hence, in this section we discuss the problems that are more relevant to the topic of this article, i.e., trustworthy ML and AI using cryptographic constructions and based on techniques from privacy-preserving and secure multiparty computation.

The notion of trust in general has been studied from various perspectives and quite extensively. Some early and fundamental works include the trust model attributed to Mayer, Davis and Schoorman [24] and a computational model for trust [25]. The Mayer's trust model considers three pillars or key factors contributing to trust, namely ability, benevolence, and integrity [24]. Marsh's computational model [25], on the other hand, has focused on the computational aspect of trust and has provided a precise formalism of the concept of trust that can be embedded in AI systems, particularly in distributed AI systems (DAI).

Following these works [24], [25], there have been numerous works for modeling and measuring trust. For example, in [26] an information theoretic approach for measuring trust has been provided that relates the notion of trust to the concept of uncertainty in information theory (which in turn can be related to uncertainty quantification in machine learning models). A brief overview of different trust evaluation models can be found in [27], in which the authors have also used cryptographic techniques for secure trust evaluation (STE).

Another important open problem in artificial intelligence and machine learning research is the alignment problem

Open Problems in AI and ML Safety [5]	
Robustness	How to create AI and ML models that are robust and reliable against adversarial attacks and in unusual situations.
Monitoring	How to detect abnormal usage and unexpected functionalities of AI and ML models.
Alignment	How to design AI and ML models that respect true and universally-accepted human values.
Systemic Safety	How to address or handle risks and attacks to AI and ML systems.

TABLE I
OPEN PROBLEMS IN AI AND ML SAFETY [5]

(sometimes referred to as value alignment). With recent advancements in AI, this problem is expected to become more important, thus requiring further research for designing effective solutions. As discussed in [7], the alignment problem in AI can be a challenging one; and it might be too soon to judge what approaches can be better solutions for this problem. Nevertheless, some earlier relevant researches have argued the potential of solutions based on (hybrid) inverse reinforcement learning (IRL) [28] and cooperative inverse reinforcement learning (CIRL) [29] (see also [30]). These approaches along with alignment verification methods (e.g., [31]) might be helpful tools for aligning the intents of AI & ML models with true human values and steering AI systems properly based on the AI principles [31].

The third research challenge which, during the last few years, has become more important than before is data security and privacy [32]. With the increased use of digital devices and online/cloud services in our daily lives, data security and privacy are expected to become more crucial than ever; thus seeking more effective solutions. While data security, in general, has been studied quite extensively in the last few decades, addressing security and privacy issues in machine learning applications is still challenging because of the very large dimensions of machine learning models and the big data phenomenon. Therefore, further research on security and privacy-related aspects of machine learning and artificial intelligence models is required; so that more effective solutions can be developed.

B. AI Principles

Artificial intelligence safety has been an important concern since early days of AI research [32], [33]. Over the past decades several sets of AI principles have been developed. Their main goals have been to provide general guidelines for how AI research and development should be conducted and how to develop beneficial AI systems. The Asilomar AI principles [34] is an example of AI principles that provide some guidelines for beneficial AI development. Another framework of principles for AI in society has been provided in [32] (also in [35]). This framework [32] proposes five principles based on an analysis of several high-profile AI principles (including the Asilomar, Montreal, IEEE, EGE, AIUK, Partnership AI principles) and other guidelines (e.g., AI4People) [32]. In this article, we refer to this set of five principles [32] as 5Principles4AI.

The Asilomar AI principles covers a broad list of AI-related topics which are categorized into three categories, namely research, ethics and values, and long-term issues. Each category highlights several sub-categories that are outlined in Table II. The framework of [32], on the other hand, develops its principles based on existing bioethics principles and by adding an extra aspect which is specific to AI (i.e., explicability). There are various other sets of rules or principles for AI. Asimov's Three Laws of Robotics [36] is another example, that can perhaps provide some fundamental rules for developing standard and carefully-designed guidelines for ethical artificial intelligence and machine learning systems [37]. The aforementioned AI principles are summarized and highlighted in Table II and Table III.

The above two sets of AI principles provide general guidelines about how AI research and development should be conducted. Furthermore, they highlight the ultimate goals of AI tools and technologies (i.e., to promote common good and well-being of sentient creatures as well as eliminating all sort of discrimination). In particular, the unified framework of five AI principles in society [32] integrates the common principles of several other high-profile AI principles. With these principles in mind, in the next sections we discuss how mathematical and cryptographic techniques can be used for addressing open problems in artificial intelligence and machine learning safety.

III. CRYPTOGRAPHIC AND MATHEMATICAL TECHNIQUES FOR TRUSTWORTHY AI AND ML SAFETY

In this section we highlight cryptographic and mathematical techniques that can be useful for alleviating the issues of trust, data privacy as well as alignment in machine learning and artificial intelligence models. We start by reviewing some of the important properties and applications of polynomials, that have proven their effectiveness as powerful and universal problem solvers in various areas of engineering and technology. Some well-known examples include error-correcting codes, secret sharing, Lagrange coded computing, as well as post-quantum cryptographic schemes such as multivariate cryptography and FHE schemes based on the ring of polynomials, etc.

A. Polynomials as Powerful Encoding Techniques

Polynomials (as some fundamental mathematical structures) have been around for a long time. In different eras in history they have had different applications; and there have been various polynomial-based problems that have interested mathematicians. Among others, Diophantine equations are polynomial equations with integer coefficients; and the well-known Fermat's last theorem is an example of a Diophantine equation [38], [39]. Fermat's last theorem is closely related to elliptic curves, which are the foundation of an important branch of

The Asilomar AI Principles [34]			
Research	Ethics and Values		Long-Term Issues
- Research Goals - Research Funding - Science-Policy Link - Research Culture - Race Avoidance	- Safety - Value Alignment - Human Values - Responsibility - Failure Transparency - Judicial Transparency - AI Arms Race	- Personal Privacy - Shared Benefit - Shared Prosperity - Liberty and Privacy - Human Control - Non-subversion	- Capability Caution - Importance - Risks - Common Good - Recursive Self-Improvement

TABLE II
THE ASILOMAR AI PRINCIPLES [34]

5Principles4AI: The Unified Framework of Five Principles for AI in Society [32]	
Beneficence	AI technologies should promote the common good, well-being of sentient creatures, human dignity, and help sustain the planet.
Non-maleficence	AI researchers and developers should develop AI technologies responsibly, so that personal privacy is guaranteed and the AI technologies are not misused.
Autonomy	AI developments should promote the autonomy of all human beings; and there should be a balance between human-led and machine-led decision making.
Justice	AI technologies should promote justice, fairness, as well as shared benefit and prosperity. Furthermore, AI developments should eliminate all sorts of discrimination.
Explicability	AI decision making processes should be transparent, understandable, and interpretable. Explicability specifies intelligibility (how AI processes work) and accountability (who is responsible for the work logic of AI technologies).

TABLE III
THE UNIFIED FRAMEWORK OF FIVE PRINCIPLES FOR AI IN SOCIETY [32]

applied cryptography, called elliptic curve cryptography (ECC) [38]. Other remarkable efforts on polynomials were done by Joseph-Louis Lagrange, whose works such as the celebrated Lagrange interpolation have been used in engineering domains quite extensively (e.g., in secret sharing [9], error-correcting codes [8], Lagrange coded computing [15], and Lagrangian neural networks [22]).

In this article, we review some of the well-known applications of polynomials, by focusing mainly on those that are more relevant to data security and privacy as well as to secure computation and trustworthy machine learning. We highlight the intuition behind how polynomials and their properties have been utilized as mathematical tools for providing solutions to various engineering and technology problems. The discussed properties of polynomials as a rich mathematical structure provide insights on how to utilize polynomial-based techniques for tackling important problems in machine learning and artificial intelligence areas.

B. Secret Sharing

Secret sharing [9] is one of the appealing applications of polynomials. The main idea behind secret sharing [9] is how to divide a secret (i.e., a piece of data D) into different pieces in such a way that the data can be reconstructed only if a certain number of the pieces are available. Secret sharing relies on a very fundamental property of polynomials, that is by having a sufficient number of points on a polynomial, the polynomial can be reconstructed (thanks to Lagrange interpolation) [9]. However, any number of points less than a certain threshold cannot give any information about the secret.

In order to share a piece of data or secret, the data is encoded in a polynomial; and the shares of the data are generated by simply evaluating the polynomial on different points in a predefined domain (e.g., $0, 1, 2, ..., n \in Z$). The data recovery (or secret reconstruction) is guaranteed by a nice property of polynomials; that is, by having t points on a polynomial of degree $t-1$, the polynomial (thus the secret data) can be reconstructed. However, by any number of points less than t points, the polynomial cannot be reconstructed. The parameter t is called the threshold which indicates the minimum number of points required for polynomial reconstruction.

In this particular application of polynomials [9] the fact that a certain number of points on a polynomial is needed for its proper reconstruction has been used for addressing a very important problem in data security and engineering. It is interesting to note that this property of polynomials (yet with a different interpretation) has been used for addressing other important problems in engineering, e.g., for error-correcting codes such as Reed-Solomon error-correcting codes [8], and for verifiable computation in ZKP systems (see e.g., R1CS and QAP [40], [41]).

The idea of secret sharing [9] has been the foundation for many other important subsequent works with applications to cryptography (interested readers might refer to these survey [42], [43] for further discussion on secret sharing and its applications in other domains). With *data* garnering more values as the digital gold or digital oil, secret sharing techniques can provide effective tools for sharing data while preserving its security and privacy. It is worth noting that another well-known secret sharing scheme is attributed to Blakley [44].

C. Lagrange Coded Computing and Private Polynomial Computing

Two other interesting applications of polynomials in private computing are: Lagrange coded computing (LCC) [15] and private polynomial computing (PPC) [16]. Lagrange coded

computing (LCC) [15] is a novel technique for addressing engineering problems in distributed computation settings, in which different workers are collaboratively performing a computational task. LCC, that leverages Lagrange polynomials, provides solutions to three important problems in distributed computing, i.e., resiliency for alleviating the issue of stragglers, security for dealing with malicious or Byzantine parties (or workers), privacy guarantees for the distributed datasets [15]. LCC achieves this by encoding the dataset and sharing the encoded dataset among the workers; and letting the workers to perform the computation on the encoded data.

Private polynomial computing (PPC) [16] generalizes the idea of private information retrieval (PIR) to private computation. PPC incorporates and borrows ideas from well-known techniques based on polynomials, such as Reed-Solomon error-correcting codes [8], Shamir secret sharing [9], Lagrange coded computation [15]; and provides schemes for evaluating polynomials on Lagrange encoded data. Similar to LCC, PPC provides solutions to the challenges of stragglers in distributed computation as well as privacy preservation of data using data hiding techniques such as secret sharing [9]. It is worth noting that Reed-Solomon (RS) codes [8] are maximum distance separable (MDS) codes and are closely related to Shamir secret sharing scheme [45]. These properties make RS codes powerful encoding techniques for various applications in which data privacy matters (e.g., private information retrieval from MDS coded databases [46]).

D. Polynomials as Universal Privacy-Enhancing Techniques

The useful properties of polynomials and their widespread usage in cryptography and secure computation make polynomials a sort of universal privacy-enhancing techniques. Other than secret sharing [9], Lagrange coded computing [15], and private polynomial computing [16], there are various other interesting and important applications of polynomials. Some prominent applications include fully homomorphic encryption (FHE), which is empowered by the RLWE problem [47] based on the ring of cyclotomic polynomials, quadratic arithmetic programming (QAP) [14] and rank-1 constraint systems (R1CS) [13], verifiable polynomial delegation (VPD) [17], [18], polynomial commitment schemes [17], and much more. On the other hand, polynomials are universal approximators and have been used in various other applications, e.g., in polynomial neural networks [20], deep polynomial neural networks [21], and polynomial classifiers [23].

The robustness and usability of polynomial-based techniques can be seen quite well with their applications in zero-knowledge proof (ZKP) systems, particularly for the arithemtization of computation and verifiable computation [48], [49]. In ZKP applications, typically there are two parties, namely a prover and a verifier. The interactions between the two parties are simulated using polynomials in the form of computational protocols, e.g., IOP protocols and FRI protocol (fast Reed-Solomon interactive oracle proof) [19]. In these applications the parties use polynomial-based techniques (e.g., R1CS and QAP) for encoding their data and arithmetization of their computations. These techniques enable tying and entangling the data and computation with rigorous mathematical relations. Applying such techniques on the data and computations enforces the parties to follow the protocol's rules. In other words, if a party does not follow the rules of the computational protocol, the party can be detected by the protocol.

The extensive utilization of polynomials in engineering applications and cryptographic constructions gives polynomials the capability to behave as sort of abstract bridges between secure computation and various other engineering applications. This property, in turn, can be very handy for designing privacy-preserving and privacy-enhancing solutions, e.g., for trustworthy machine learning and artificial intelligence safety. In particular, polynomials can be useful tools for verification of artificial neural networks [50], [51] and their robustness [52]–[54].

E. Polynomials as Robustness Verification Techniques for ANN

As mentioned in section II (also in Table I), robustness of machine learning and artificial intelligence models is an important open problem in these fields. There have been ongoing researches for developing robust ML models as well as techniques for verifying the robustness of ML models [52]–[54].

Among other approaches, polynomial-based techniques and concepts have been interesting ways for addressing the issue of robustness in some machine learning models. Some polynomial-based techniques, that can be useful for enhancing the safety and robustness of ML models, include polynomial optimization for robustness verification of artificial neural networks [54], polytopes and polynomial zonotopes for robustness of reinforcement learning [55], robustness and verification of neural networks [50], [51], [56], safety verification [57] of neural networks, reachability analysis of neural networks [58], and support vector machines [59]. Polytopes and zonotopes can also be helpful for designing provable defense mechanisms against adversarial attacks on (deep) neural networks [60], [61] as well as for the verification of Lagrangian neural networks [62] and Lipschitz neural networks [50].

As machine learning and artificial intelligence tools become more pervasive, it is important to design robust machine learning models and to develop defense mechanisms against adversarial attacks on ML and AI models. Cryptographic constructions, particularly techniques based on polynomials, are rigorous mathematical tools that can be used both for addressing data security and privacy issues as well as for designing more robust ML and AI models.

IV. From Secret Sharing to Sharing Computation Securely: Secure Distributed Matrix Computation (SDMC)

In this section we present the idea of secure distributed matrix computation (SDMC). This idea is built on top of several other important foundational works, including Lagrange coded

computation, secure matrix multiplication, and secret sharing techniques [15], [63], [64], and [9].

As discussed in the previous sections, polynomials have been utilized in various areas of engineering and technology for solving different important problems. Earlier applications of polynomials seem to be mainly for encoding the data, e.g., Reed-Solomon error-correcting codes [8], secret sharing schemes [39], the NTRU cryptosystem [65], etc. However, in recent years with the advancements in computation technologies and the big data phenomenon, the applications of polynomials have been extended to encoding computations as well. Some examples that we discussed in the previous section include Lagrange coded computing (LCC) [15], private polynomial computing (PPC) [16], fully homomorphic encryption (FHE) schemes [11], as well as applications in ZKP systems such as FRI protocol [19], R1CS and QAP [40], [41].

In this section we leverage the existing techniques even further for applications to trustworthy and privacy-preserving machine learning. Our idea is to utilize well-known techniques based on polynomials, e.g., coded computing, secure matrix multiplication, and secret sharing, for performing matrix computation in a distributed and secure fashion.

We would like to emphasize that in previous works, e.g., in Lagrange coded computing [15] and private polynomial computing [16], the goal of the polynomial-based constructions was to deal with the issue of stragglers, malicious parties, and data privacy. More specifically, in those previous works, the parties could be untrusted parties (where a party is a worker performing some computational tasks). For the setting of the problem in this article, the participating parties are assumed to be honest and trustworthy parties, typically human entities that want to be in the loop in multiparty-computation scenarios that deal with distributed AI systems (DAI). The main purpose is to design a mechanism in which human entities can participate in DAI systems appropriately. Most importantly, the proposed mechanism can be helpful for ensuring data privacy and balancing between human-led and machine-led decision processes, which are important principles in the Unified Framework of Five Principles for AI in Society (see Table III in Section II).

A. Assumptions and Problem Settings

The setting for the research problem that we want to address in this article is as follows. We assume there are some AI system(s) that are supposed to be programmed for performing some tasks using AI and ML models. One approach is to program the AI systems and let them perform their jobs. Instead of this classic approach, a different paradigm is to deploy secure distributed computation (with human entities in the loop) and to perform the computations securely. Therefore, in addition to AI system(s) we assume there are some trusted human entities that want to be in the loop, while the AI systems are doing their jobs. Then, to perform some computational tasks (i.e., to evaluate a function), the parties distribute the computation and evaluate the function in a distributed manner. To achieve this, they use the secure distributed matrix computation techniques that we discuss in following subsections.

B. Secure Matrix Multiplication

Matrix multiplication is a commonly-used operation with significant applications in machine learning applications that deal with big data sets. Two novel secure matrix multiplication techniques have been proposed in [63] and [64]. In what follows we provide an overview of the proposed methods.

1) Secure Matrix Multiplication based on [63]: The proposed technique for secure matrix multiplication in [63] relies on $(\alpha, \beta)-$polynomial codes and works as follows [63].

Given two matrices $A \in \mathbb{F}_q^{k\times l}$ and $B \in \mathbb{F}_q^{l\times m}$, the goal is to compute $A \times B$ in a secure distributed manner. Here, $\mathbb{F}_q$ is a large finite field; and k, l, and m are positive integers representing the dimensions of the matrices. To perform the matrix multiplication securely, first the matrices need to be encoded using polynomial encoding techniques, e.g., $(\alpha, \beta)-$polynomial codes [63]. In order to encode the matrices, each of the matrices is partitioned horizontally and vertically (along its rows and columns), as follows [63]:

$$A = \big(A_1, A_2, \ldots, A_m\big) \quad \text{and} \quad B = \big(B_1, B_2, \ldots, B_n\big) \tag{1}$$

The A and B matrices can then be encoded using the following equations:

$$\widetilde{A_i} = \sum_{j=1}^{m} A_j x_i^{\alpha j} \tag{2}$$

$$\widetilde{B_i} = \sum_{j=1}^{n} B_j x_i^{\beta j} \tag{3}$$

where x_i for $i = 1, ..., n$ denotes a unique number assigned to each party (or computational worker). Assuming each party can store and process $\frac{1}{m}$ fractions of matrix A, then the parameters α and β are defined as: $\alpha = 1$ and $\beta = m$, as suggested in [63].

To compute the multiplication of A and B, each party (say party i) needs to perform specific computations on the received shares (or encoded version) of the matrices as follows:

$$\widetilde{C_i} = \widetilde{A_i}^T \times \widetilde{B_i} = \sum_{j=1}^{m}\sum_{k=1}^{n} A_j^T B_k x_i^{km+j} \tag{4}$$

Once the parties are done with their computations, they will pass their results to a specified party so that the remaining parts of the computation can be performed. The computation result is finalized and determined after all the parties provide the shares of their results. In other words, without the results of a certain number of the parties, the final computation result cannot be reconstructed. This will be useful for ensuring that the computation is carried out in a distributed manner.

2) Secure Matrix Multiplication based on [64]: Some other novel approaches for secure multix multiplication were proposed in [64]. In particular, the fully secure distributed matrix multiplication of [64] works as follows.

Similar to the approach of [63], two matrices $A \in \mathbb{F}_q^{k\times l}$ and $B \in \mathbb{F}_q^{l\times m}$ are given and the goal is to compute $A \times B$ in a secure distributed manner. To achieve this, the two matrices first need to be partitioned; and then to be encoded using polynomial encoding techniques. The scheme of [64] suggests the following partitioning and encoding approaches:

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_r \end{pmatrix} \qquad \text{and} \qquad B = \big(B_1, B_2, \ldots, B_r\big) \tag{5}$$

The A and B matrices can then be encoded using the following equations [64]:

$$\widetilde{A_i} = \sum_{j=1}^{r} A_j x_i^{j-1} + \sum_{k=1}^{l} R_{A_k} x_i^{r+k-1} \tag{6}$$

$$\widetilde{B_i} = \sum_{j=1}^{r} B_j x_i^{(j-1)(l+r)} + \sum_{k=1}^{l} R_{B_k} x_i^{(r+k-1)(l+r)} \tag{7}$$

where $i = 1, ..., n$ denotes a unique number assigned to each party. l is the number of computing parties that may collude. $\widetilde{A_i}$ and $\widetilde{B_i}$ are the encoded matrices for party with $ID = i$. Furthermore, R_{A_k} is a random matrix with uniformly selected elements corresponding to submatirx A_k.

Once the A and B matrices are encoded and the encoded matrices are distributed among the parties, the parties can perform operations on their corresponding shares. Particularly, for performing a multiplication operation on A and B, the parties need to do the following computation [64]:

$$\widetilde{A_i} \times \widetilde{B_i} = \sum_{j=1}^{r}\sum_{k=1}^{r} A_j B_k x^{T_1} + \sum_{j=1}^{r}\sum_{k=1}^{l} A_j R_{B_k} x^{T_2} \tag{8}$$

$$+ \sum_{j=1}^{l}\sum_{k=1}^{r} R_{A_j} B_k x^{T_3} + \sum_{j=1}^{l}\sum_{k=1}^{l} R_{A_j} R_{B_k} x^{T_4} \tag{9}$$

where the variables j and k vary between the range specified in the lower and upper bounds of the summations in the equations. Furthermore, $T_1 = j + (k-1)(l+r) - 1$; $T_2 = j+(k+r-1)(l+r)-1$; $T_3 = j+r+(k-1)(l+r)-1$; and $T_4 = j+r+(k+r-1)(l+r)-1$, as suggested in [64].

It should be noted that since the matrices were encoded, each party by itself cannot get the final result of the computation. To be able to compute the final result of the computation, the participating parties need to provide their computation results. This property of the coded computation will be useful for having human parties to be in the loop while the computation are performed in a distributed AI application scenario or in a distributed AI (DAI) system.

C. Secure Matrix Addition

The encoding techniques discussed in the previous subsections can be used for designing secure distributed matrix addition schemes as well. With addition and multiplication gates on matrices, it would be possible to perform arbitrary computation on matrices defined over finite fields.

Mathematically speaking, assuming two matrices $A \in \mathbb{F}_q^{k\times l}$ and $B \in \mathbb{F}_q^{k\times l}$ are given and the goal is to compute $A + B$ in a secure distributed manner. We use the (α, β)-polynomial codes [63] to encode the two matrices (by setting $\beta = m$):

$$\widetilde{A_i} = \sum_{j=1}^{n} A_j x_i^{mj} \tag{10}$$

$$\widetilde{B_i} = \sum_{j=1}^{n} B_j x_i^{mj} \tag{11}$$

where x_i for $i = 1, ..., n$ is a unique number assigned to each party. Recall that m is the ratio (or portion) of the matrices that each party (or computational node) can store and process.

Given $\widetilde{A_i}$ and $\widetilde{B_i}$ as the encoded version of the matrices A and B, the addition of the two matrices can be computed distributedly as follows:

$$\widetilde{C_i} = \widetilde{A_i} + \widetilde{B_i} = \sum_{j=1}^{n} A_j x_i^{mj} + \sum_{j=1}^{n} B_j x_i^{mj} \tag{12}$$

This approach is an extension of the well-known Shamir secret sharing scheme [9]. In secret sharing it is possible to compute the addition of two field elements, whereas with secure distributed addition of matrices it is possible to compute the addition of matrices over a finite field, e.g., $A+B \in \mathbb{F}_q^{k\times l}$.

D. Secure Distributed Matrix Computation

The secure distributed matrix addition and multiplication schemes that we discussed above enable us to perform arbitrary secure operations on matrices. In particular, since polynomials are universal approximators, this property of polynomials can be used for representing arbitrary functions as polynomials. With a function represented as a polynomial, it would be straightforward to evaluate it using the arithmetic gates (i.e., addition and multiplication operations) over matrices on a certain finite field, i.e., F_q.

It is worth mentioning that polynomial representation of functions for their secure evaluation has been utilized in other applications, e.g., using Taylor series or Chebychev polynomials for the evaluation of secure comparison problem or the activation functions of neural networks. For example, in [66]–[68] Chebyshev polynomials have been used as approximations for the ReLU activation functions in neural networks.

The idea of secure distributed matrix computation is as follows. Given a function f and some matrices A and B over a finite field, the goal is to evaluate the function over the matrices in a secure distributed fashion. To achieve this, two main steps need to be done. First the matrices need to be encoded

with appropriate techniques. This can be done using the encoding techniques, e.g., with (α, β)-polynomial codes [63], that were briefly discussed in subsection IV-B. The second step is to arithemtize the functions using arithmetic gates and transform it into polynomial representation. Once the data and function are properly encoded using polynomial codes and in polynomial representation, the parties can then evaluate the function on the encoded data in a secure distributed manner.

After all the parties compute their results, they need to send the obtained results to a main party (which is a party who wants to get the function evaluation). The main party can then utilize some interpolation technique, e.g., Lagrange interpolation, to obtain the final result of function evaluation. It should be noted that the computation result can be obtained only if a certain number the parties provide their results. The minimum number of data points that are required for reconstruction of results is called the threshold. We have formalized the required steps for secure distributed matrix computation (SDMC) in Protocol 1.

Protocol 1: Secure Distributed Matrix Computation (SDMC)

Input: Matrices $A \in \mathbb{F}_q^{k\times l}$ and $B \in \mathbb{F}_q^{k\times l}$.
Requirements: There are n parties, who want to perform a distributed computation task.
Objective: Compute $f(A, B)$ in a secure distributed manner.
Output: $y = \widetilde{f}(\widetilde{A}, \widetilde{B})$, which is the encoded version of $y = f(A, B)$.

1 Encode the matrices A and B using the appropriate encoding techniques (e.g., using (α, β)-polynomial codes as discussed in section IV-B):

$$\widetilde{A_i} = \sum_{j=1}^{n} A_j x_i^{mj} \qquad \widetilde{B_i} = \sum_{j=1}^{n} B_j x_i^{mj}$$

2 Distribute the encoded matrices among the parties.

3 Represent the function f as a polynomial using arithmetic gates:

$$f(A, B) = \sum g(\widetilde{A_i}, \widetilde{B_i})$$

where g is a multiplicative function of $\widetilde{A_i}$ and $\widetilde{B_i}$

4 Evaluate each arithmetic gate using the appropriate distributed addition or multiplication techniques:

$$\widetilde{C_i} = \widetilde{A_i} + \widetilde{B_i} = \sum_{j=1}^{n} A_j x_i^{mj} + \sum_{j=1}^{n} B_j x_i^{mj}$$

$$\widetilde{C_i} = \widetilde{A_i}^T \times \widetilde{B_i} = \sum_{j=1}^{m}\sum_{k=1}^{n} A_j^T B_k x_i^{km+j}$$

5 The parties send their result to the main computational node.

6 The main computational node uses interpolation to obtain the final value of the function.

7 Return result:

$$y = \widetilde{f}(\widetilde{A}, \widetilde{B})$$

V. Discussion and Future Research Directions

In the last couple of decades there have been significant amounts of progress in different domains of digital technology. With the rise of advanced AI technologies, big data phenomenon, the emergence of disruptive technologies such as Blockchain and cryptocurrencies, as well as quantum computation, the world is going to gradually move into a new era. While these advances bring many opportunities, they have also the impetus for adversly affecting our lives and the society. As recommended by the AI principles, which were discussed in Section II, researchers should also study and pay attention to different aspects of these technologies, in particular the safety and security of AI tools and systems (as summarized in Table I in Section II). In what follows we highlight some future research directions that can be helpful with this regard.

First and foremost, it is important to design effective and comprehensive AI guidelines and to increase the awareness of researchers and developers in the AI community. This allows researchers and engineers to apply the guidelines to their works and conduct their research and development more responsibly. The Asilomar AI principles and the unified framework of five principles for AI in society (5Principles4AI) can be the foundations for more comprehensive AI principles. Each section or set of guidelines in the aforementioned AI principles highlight some dimensions or aspects of AI technologies that require further attention and research. For example, balancing human-led and machine-led decision making processes is an important aspect in the 5Principles4AI (under the category of autonomy in Table III in Section II-B). The secure distributed matrix computation scheme that we presented in this article can be a useful tool for keeping human in the loop (HIL) so that they can ensure the balance between human and machines in distributed AI systems (DAI). It also provides data privacy measures thanks to its underlying data encoding and coded computing techniques.

Trustworthiness and robustness of machine learning models, value alignment, and data privacy are some major aspects of artificial intelligence and machine learning safety. Particularly, robustness of ML models has been the center of researchers attention in the last couple of decades [6], [52], [54] (which is also among the open problems summarized in Table I in Section II). As the AI and ML systems become more persuasive, it is important to design effective and robust models that can be relied on in different environments and unusual (unseen) situations. In addition, with the rise of large language models (LLMs), data privacy is expected to become more important, and therefore seeks effective solutions.

Another interesting area of research is the study of trust mechanisms in distributed AI (DAI) systems. In particular, the techniques in decentralized computation and Blockchain might be helpful for designing effective distributed trustworthy AI systems empowered by computational trust models and cryptographic constructions. A related disruptive technology is the concept of decentralized autonomous organizations (DAO), in which DAI systems can be used. Trust and reputation mechanisms can be very useful mechanisms for improving decentralized autonomous organizations (DAO) systems [69], [70] as well.

Last but not least, some mathematical techniques have been very effective tools for addressing various problems in different areas of engineering and technology. Polynomial-based techniques are among such tools that can be useful for addressing open problems or tacking challenges of AI systems. For example, polynomial zonotopes and polytopes [56] have attracted attention for designing robust machine learning models as well as for stability analysis and verification of neural networks [50], [57].

VI. Conclusion

In this article we reviewed some of the challenges and open problems in artificial intelligence and machine learning safety, namely trustworthiness, data privacy, and alignment. We also highlighted a couple of high-profile principles of artificial intelligence, including the Asilomar principles for AI and a unified framework of five principles for AI in society (5Principles4AI). We then discussed how mathematical and cryptographic techniques can be used for secure and trustworthy machine learning and AI safety.

In particular, secure distributed matrix computation (SDMC) can be a useful mathematical tool for enforcing AI principles in distributed AI systems (DAI). SMDC is based on rigorous mathematical foundations such as coded computation and secret sharing. It provides better data privacy guarantees and allows humans to be in the loop in multiparty computation scenarios in which AI systems are collaborating. It also enables balancing human-led and machine-led decision processes, which is an important AI principle as stated in the framework of five principles for AI in society. Mathematical and cryptographic constructions can be a useful tool for tackling open problems in AI. In the future research we would like to study the applicability of other mathematical techniques for improving the safety and security of AI and ML systems.

References

[1] M. Dowling and B. Lucey, "Chatgpt for (finance) research: The bananarama conjecture," *Finance Research Letters*, vol. 53, p. 103662, 2023.

[2] P. P. Ray, "Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, 2023.

[3] M. Song, X. Xing, Y. Duan, J. Cohen, and J. Mou, "Will artificial intelligence replace human customer service? the impact of communication quality and privacy risks on adoption intention," *Journal of Retailing and Consumer Services*, vol. 66, p. 102900, 2022.

[4] A. Abd-Alrazaq, R. AlSaad, D. Alhuwail, A. Ahmed, P. M. Healy, S. Latifi, S. Aziz, R. Damseh, S. A. Alrazak, J. Sheikh, *et al.*, "Large language models in medical education: Opportunities, challenges, and future directions," *JMIR Medical Education*, vol. 9, no. 1, p. e48291, 2023.

[5] D. Hendrycks, N. Carlini, J. Schulman, and J. Steinhardt, "Unsolved problems in ml safety," *arXiv preprint arXiv:2109.13916*, 2021.

[6] P. Xiong, S. Buffett, S. Iqbal, P. Lamontagne, M. Mamun, and H. Molyneaux, "Towards a robust and trustworthy machine learning system development: An engineering perspective," *Journal of Information Security and Applications*, vol. 65, p. 103121, 2022.

[7] I. Gabriel, "Artificial intelligence, values, and alignment," *Minds and machines*, vol. 30, no. 3, pp. 411–437, 2020.

[8] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[9] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[10] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pp. 383–395, IEEE, 1985.

[11] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, *et al.*, "Homomorphic encryption standard," *Protecting privacy through homomorphic encryption*, pp. 31–62, 2021.

[12] S. GOLDWASSER, S. MICALI, and C. RACKOFF, "The knowledge complexity of interactive proof systems," *SIAM journal on computing*, vol. 18, no. 1, pp. 186–208, 1989.

[13] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for r1cs," in *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*, pp. 103–128, Springer, 2019.

[14] R. Gennaro, C. Gentry, B. Parno, and M. Raykova, "Quadratic span programs and succinct nizks without pcps," in *Advances in Cryptology–EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings 32*, pp. 626–645, Springer, 2013.

[15] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. A. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1215–1225, PMLR, 2019.

[16] N. Raviv and D. A. Karpuk, "Private polynomial computation from lagrange encoding," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 553–563, 2019.

[17] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *Advances in Cryptology-ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings 16*, pp. 177–194, Springer, 2010.

[18] J. Zhang, T. Xie, Y. Zhang, and D. Song, "Transparent polynomial delegation and its applications to zero knowledge proof," in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 859–876, IEEE, 2020.

[19] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Fast reed-solomon interactive oracle proofs of proximity," in *45th international colloquium on automata, languages, and programming (icalp 2018)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[20] S.-K. Oh, W. Pedrycz, and B.-J. Park, "Polynomial neural networks architecture: analysis and design," *Computers & Electrical Engineering*, vol. 29, no. 6, pp. 703–725, 2003.

[21] G. G. Chrysos, S. Moschoglou, G. Bouritsas, J. Deng, Y. Panagakis, and S. Zafeiriou, "Deep polynomial neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 8, pp. 4021–4034, 2021.

[22] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," *arXiv preprint arXiv:2003.04630*, 2020.

[23] W. M. Campbell, K. T. Assaleh, and C. C. Broun, "Speaker recognition with polynomial classifiers," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 4, pp. 205–212, 2002.

[24] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An integrative model of organizational trust," *Academy of management review*, vol. 20, no. 3, pp. 709–734, 1995.

[25] S. P. Marsh, "Formalising trust as a computational concept," 1994.

[26] Y. L. Sun, W. Yu, Z. Han, and K. R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 305–317, 2006.

[27] M. G. Raeini and M. Nojoumian, "Secure trust evaluation using multipath and referral chain methods," in *Security and Trust Management: 15th International Workshop, STM 2019, Luxembourg City, Luxembourg, September 26–27, 2019, Proceedings 15*, pp. 124–139, Springer, 2019.

[28] T. Arnold and D. Kasenberg, "Value alignment or misalignment - what will keep systems accountable?," in *AAAI Workshop on AI, Ethics, and Society*, 2017.

[29] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[30] J. F. Fisac, M. A. Gates, J. B. Hamrick, C. Liu, D. Hadfield-Menell, M. Palaniappan, D. Malik, S. S. Sastry, T. L. Griffiths, and A. D. Dragan, "Pragmatic-pedagogic value alignment," in *Robotics Research: The 18th International Symposium ISRR*, pp. 49–57, Springer, 2020.

[31] D. S. Brown, J. Schneider, A. Dragan, and S. Niekum, "Value alignment verification," in *International Conference on Machine Learning*, pp. 1105–1115, PMLR, 2021.

[32] L. Floridi and J. Cowls, "A unified framework of five principles for ai in society," *Machine learning and the city: Applications in architecture and urban design*, pp. 535–545, 2022.

[33] N. Wiener, "Some moral and technical consequences of automation: As machines learn they may develop unforeseen strategies at rates that baffle their programmers.," *Science*, vol. 131, no. 3410, pp. 1355–1358, 1960.

[34] "Asilomar AI Principles." https://futureoflife.org/ai-principles.

[35] L. Floridi, J. Cowls, M. Beltrametti, R. Chatila, P. Chazerand, V. Dignum, C. Luetge, R. Madelin, U. Pagallo, F. Rossi, *et al.*, "An ethical framework for a good ai society: Opportunities, risks, principles, and recommendations," *Ethics, governance, and policies in artificial intelligence*, pp. 19–39, 2021.

[36] I. Asimov, "Three laws of robotics," *Asimov, I. Runaround*, vol. 2, 1941.

[37] S. L. Anderson, "Asimov's "three laws of robotics" and machine metaethics," *Ai & Society*, vol. 22, pp. 477–493, 2008.

[38] L. D Antonio, "Teaching elliptic curves using original sources," *MAA NOTES*, vol. 68, p. 25, 2005.

[39] A. Shell-Gellasch and D. Jardine, *From calculus to computers: using the last 200 years of mathematics history in the classroom*, vol. 68. Cambridge University Press, 2005.

[40] J. Eberhardt and S. Tai, "Zokrates-scalable privacy-preserving off-chain computations," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1084–1091, IEEE, 2018.

[41] S. Liu, "Privacy protection revolution: Zero-knowledge proof," in *2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI)*, pp. 394–397, IEEE, 2022.

[42] A. Chandramouli, A. Choudhury, and A. Patra, "A survey on perfectly secure verifiable secret-sharing," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–36, 2022.

[43] V. Attasena, J. Darmont, and N. Harbi, "Secret sharing for cloud data security: a survey," *The VLDB Journal*, vol. 26, no. 5, pp. 657–681, 2017.

[44] G. R. Blakley, "Safeguarding cryptographic keys," in *Managing Requirements Knowledge, International Workshop on*, pp. 313–313, IEEE Computer Society, 1979.

[45] R. J. McEliece and D. V. Sarwate, "On sharing secrets and reed-solomon codes," *Communications of the ACM*, vol. 24, no. 9, pp. 583–584, 1981.

[46] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from mds coded data in distributed storage systems," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7081–7093, 2018.

[47] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 1–35, 2013.

[48] D. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 501–512, 2012.

[49] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in *Annual Cryptology Conference*, pp. 111–131, Springer, 2011.

[50] C. Schilling, M. Forets, and S. Guadalupe, "Verification of neural-network control systems by integrating taylor models and zonotopes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8169–8177, 2022.

[51] N. Kochdumper, C. Schilling, M. Althoff, and S. Bak, "Open-and closed-loop neural network verification using polynomial zonotopes," in *NASA Formal Methods Symposium*, pp. 16–36, Springer, 2023.

[52] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "Ai2: Safety and robustness certification of neural networks with abstract interpretation," in *2018 IEEE symposium on security and privacy (SP)*, pp. 3–18, IEEE, 2018.

[53] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020.

[54] T. Chen, *Robustness verification of neural networks using polynomial optimization*. PhD thesis, Université Paul Sabatier-Toulouse III, 2022.

[55] N. Kochdumper, H. Krasowski, X. Wang, S. Bak, and M. Althoff, "Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes," *IEEE Open Journal of Control Systems*, vol. 2, pp. 79–92, 2023.

[56] N. Kochdumper and M. Althoff, "Sparse polynomial zonotopes: A novel set representation for reachability analysis," *IEEE Transactions on Automatic Control*, vol. 66, no. 9, pp. 4043–4058, 2020.

[57] Y. Zhang and X. Xu, "Safety verification of neural feedback systems based on constrained zonotopes," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 2737–2744, IEEE, 2022.

[58] H.-D. Tran, D. Manzanas Lopez, P. Musau, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson, "Star-based reachability analysis of deep neural networks," in *Formal Methods–The Next 30 Years: Third World Congress, FM 2019, Porto, Portugal, October 7–11, 2019, Proceedings 3*, pp. 670–686, Springer, 2019.

[59] F. Ranzato and M. Zanella, "Robustness verification of support vector machines," in *Static Analysis: 26th International Symposium, SAS 2019, Porto, Portugal, October 8–11, 2019, Proceedings 26*, pp. 271–295, Springer, 2019.

[60] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International conference on machine learning*, pp. 5286–5295, PMLR, 2018.

[61] A. Mustafa, S. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao, "Adversarial defense by restricting the hidden space of deep neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3385–3394, 2019.

[62] M. Jordan, J. Hayase, A. Dimakis, and S. Oh, "Zonotope domains for lagrangian neural network verification," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8400–8413, 2022.

[63] Q. Yu, M. Maddah-Ali, and S. Avestimehr, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[64] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2018.

[65] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *International algorithmic number theory symposium*, pp. 267–288, Springer, 1998.

[66] S. Obla, X. Gong, A. Aloufi, P. Hu, and D. Takabi, "Effective activation functions for homomorphic evaluation of deep neural networks," *IEEE Access*, vol. 8, pp. 153098–153112, 2020.

[67] E. Hesamifard, H. Takabi, and M. Ghasemi, "Cryptodl: towards deep learning over encrypted data," in *Annual Computer Security Applications Conference (ACSAC 2016), Los Angeles, California, USA*, vol. 11, 2016.

[68] R. Podschwadt and D. Takabi, "Classification of encrypted word embeddings using recurrent neural networks.," in *PrivateNLP@ WSDM*, pp. 27–31, 2020.

[69] Y.-Y. Hsieh, J.-P. Vergne, P. Anderson, K. Lakhani, and M. Reitzig, "Bitcoin and the rise of decentralized autonomous organizations," *Journal of Organization Design*, vol. 7, no. 1, pp. 1–16, 2018.

[70] S. Wang, W. Ding, J. Li, Y. Yuan, L. Ouyang, and F.-Y. Wang, "Decentralized autonomous organizations: Concept, model, and applications," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 870–878, 2019.