ISSN: 2959-6386 (Online), Volume 2, Issue 3

Journal of Knowledge Learning and Science Technology

Journal homepage: https://jklst.org/index.php/home

Data Guardianship: Safeguarding Compliance in AI/ML Cloud Ecosystems

Samir Vinayak Bayani[1], Sanjeev Prakash[2], Lavanya Shanmugam[3]

[1]Broadcom Inc, USA

[2]RBC Capital Markets, USA

[3]Tata Consultancy Services, USA

Abstract

AI has found widespread application across various sectors, including security, healthcare, finance, and national defense. However, alongside its transformative advancements, there has been an unfortunate trend of malicious exploitation of AI capabilities. Concurrently, the rapid evolution of cloud computing technology has introduced cloud-based AI systems. Regrettably, vulnerabilities inherent in cloud computing infrastructure also pose risks to the security of AI services. We observe that the integrity of training data is pivotal, as any compromise therein directly impacts the efficacy of AI systems. Against this backdrop, we assert the paramount importance of preserving data integrity within AI systems. To address this imperative, we propose a data integrity architecture guided by the National Institute of Standards and Technology (NIST) cyber security framework. Leveraging block chain technology and smart contracts emerges as a fitting solution to tackle integrity challenges, owing to their characteristics of shared and decentralized ledgers. Smart contracts facilitate automated policy enforcement, enable continuous monitoring of data integrity, and mitigate the risk of data tampering.

Keywords: data integrity; AI systems; cloud computing; block chain

Article Information:

Article history: Received: 01/09/2023 Accepted:15/09/2023 Online: 30/09/2023 Published: 30/09/2023

DOI: https://doi.org/10.60087/jklst.vol2.n3.p456

[i] ***Correspondence author: Samir Vinayak Bayani***

Introduction

Artificial Intelligence (AI) stands as one of the most disruptive technological advancements of recent times. Originating in 1956, AI has experienced exponential growth, marked notably by milestones such as AlphaGo's victory over the world Go champion in 2016 and the launch of Waymo's self-driving taxi service by Google in 2018. Its applications extend across diverse sectors including national security, finance, healthcare, criminal justice, transportation, and smart cities. However, alongside its transformative potential, AI has also been used for nefarious purposes, exemplified by attacks such as adversarial examples in self-driving cars and coordinated assaults on AI-controlled robotic systems.

The dynamic landscape of AI development intersects with the rapid evolution of cloud computing technology, offering new avenues for AI researchers and developers. Cloud-based AI systems leverage the infrastructure provided by Cloud Service Providers (CSPs), enabling efficient utilization of resources and accessibility from any location with internet connectivity. While cloud computing presents numerous advantages, it also introduces challenges, particularly concerning data integrity and privacy. Users must entrust their data to potentially untrusted environments, raising concerns about breaches in data integrity and security.

The importance of data integrity within AI systems cannot be overstated. Compromised training data can lead to erroneous outcomes, posing significant risks, especially in critical domains. As such, safeguarding data integrity emerges as a critical imperative. Addressing these challenges requires a proactive approach, integrating security considerations into the design and implementation of AI systems.

In this context, we propose an architecture aimed at addressing data integrity issues within cloud-based AI systems. Drawing inspiration from the National Institute of Standards and Technology (NIST) cybersecurity framework, our architecture provides a structured approach to ensure continuous data integrity provisioning. Leveraging blockchain technology and smart contracts, we enhance the integrity of the machine learning pipeline, bolstering trust and security in cloud-based AI environments.

Our contributions encompass the proposal of a system architecture aligned with NIST cybersecurity principles, comprising modules that address key aspects of security and integrity. Additionally, we integrate blockchain and smart contracts into our architecture to automate policy enforcement and enhance trust between users and CSPs.

The remainder of this paper is structured as follows: Section 2 explores existing vulnerabilities and threats in AI and cloud environments, while Section 3 reviews related research. In Section 4, we detail our proposed architecture, followed by an evaluation and discussion in Section 5. Finally, we conclude our findings in Section 6.

Background

AI Environment

Within the AI data pipeline, there exist three primary avenues for potential adversarial interference [4]:

1. Attacks against the data utilized for training and decision-making.
2. Attacks against the classifier in the training environment.
3. Attacks against models in the deployment environment.

These adversarial scenarios are intrinsic to the machine learning pipeline and necessitate thorough examination of vulnerabilities and challenges present in both AI and cloud environments.

To elucidate these challenges, this section delves into the typical AI data pipeline and highlights integrity issues across its phases, as depicted in Figure 1. Furthermore, Table 1 provides an overview of possible attack methods associated with each phase.

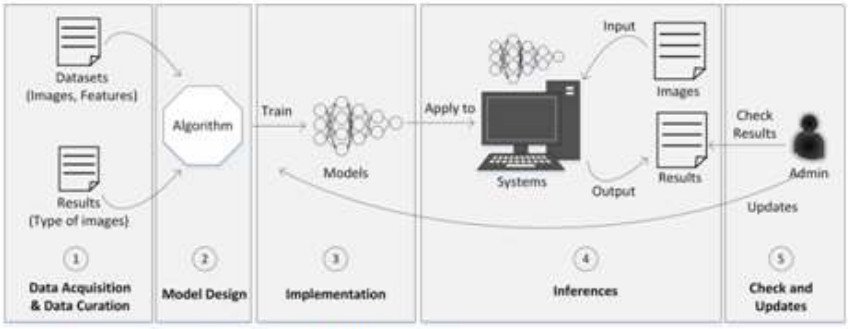


Figure 1. Common AI's data pipelines.

Table 1. ML pipeline's challenges.

Phase	Vulnerabilities and Attacks	Integrity Issues
A1. Data acquisition and curation	Data poisoning [16–19]	Adversary alter the datasets, so the results will be in a way that attacker desires.
A2. Model design	No specific issue related to ML	Generic issues related to choices of hardware/software deployment, or external services.
A3. Implementation	Data Poisoning [16–19], Back-door attack [20,21]	Such as phase A1, adversary attempts to corrupt the training datasets. Furthermore, this phase could be an entrance to back-door attack.
A4. Inferences	Adversarial examples [22–26]	Adversary try to manipulating the inputs that cause AI system to misclassify it and behave incorrectly.
A5. Check and updates	Backdoor attack [20,21]	This attack is being triggered if the adversary successfully manipu-lated the training datasets at training phase.

A1. Data Acquisition and Curation

Before an AI system can effectively learn a model tailored to a specific task, it requires access to extensive datasets. In the data acquisition phase, users compile training datasets that align with their desired outcomes. For instance, in

tasks like image classification or detection, datasets consist of collections of images. Data can be sourced from various outlets, including proprietary datasets or well-known repositories such as CIFAR10 or MNIST. During the data curation phase, users engage in activities like data formatting, noise removal, and labeling. This phase is pivotal as it directly influences the accuracy of the resultant model.

Vulnerabilities and Attacks—Data Poisoning: This threat occurs when attackers introduce erroneous or mislabeled data to train AI models. For instance, images of stop signs may be mislabeled to evade detection by the algorithm, posing risks on roads. Such risks underscore the importance of meticulous control over training datasets to safeguard AI decision-making processes. Additionally, data poisoning has been observed in recommender systems, where adversaries inject manipulated data to influence suggestions made by the system.

Issues—The integrity of training data is paramount in the machine learning (ML) pipeline as it dictates the accuracy of outcomes. Ensuring data integrity is crucial, particularly in supervised learning where correct labeling is essential for optimal results. Data poisoning represents a significant threat to data integrity, albeit accidental poisoning may also occur due to inconsistent or unfit data inputs.

A2. Model Design

In this phase, users determine the AI algorithms to be employed and set hyperparameters such as the number of nodes, layers, learning rate, biases, and activation functions.

Issues—While there are no specific ML system-related issues in this phase, general considerations like hardware/software deployment and choice of external services should be accounted for. Moreover, human error poses a risk of algorithmic inaccuracies.

A3. Implementation

The implementation phase comprises training and testing. Training is pivotal as it establishes the baseline behavior of the system by iteratively running algorithms with input datasets to predict outputs with minimal error. Testing, on the other hand, validates the model's performance using datasets not utilized in training to ensure correct parameter configurations.

Vulnerabilities and Attacks—Backdoor Attack: Primarily targeting the training phase, backdoor attacks rely on data poisoning to introduce examples with triggers into the training dataset. This manipulation associates the trigger with a target class, causing the model to misclassify inputs containing the trigger during inference.

Issues—Integrity challenges arise when adversaries attempt to influence, corrupt, or alter training data. This can be achieved by inserting adversarial inputs into existing data or directly altering the training dataset, potentially impacting the accuracy of results. Additionally, the training phase serves as an entry point for backdoor attacks triggered in subsequent phases.

A4. Inferences

Once the model is prepared, it is applied to applications or systems. In cases like image classification or detection, inputs are typically in image format, and the system classifies the image based on the trained model, providing the result.

Vulnerabilities and Attacks—Adversarial Examples: Among the most prevalent threats in the AI environment are adversarial examples. Attackers manipulate inputs to induce errors in AI systems. Even imperceptible perturbations in digital images can lead AI algorithms to completely misclassify them. Various types of adversarial examples exist, including:

• Fast Gradient Sign Method (FGSM): A method introduced by Goodfellow et al. for effective adversarial training. FGSM quickly generates adversarial examples by involving a single back-propagation step.

• DeepFool: Proposed by Moosavi-Dezfooli et al., DeepFool finds minimal L2 adversarial perturbations in an iterative manner, capable of generating perturbations smaller than FGSM.

• Jacobian-based Saliency Map Attack (JSMA): Introduced by Papernot et al., JSMA efficiently executes targeted attacks by calculating the Jacobian matrix of the score function, restricting small L0 perturbations.

• Basic Iterative Method (BIM): Proposed by Kurakin et al. to enhance FGSM's performance by employing an iterative optimizer for multiple iterations.

• Universal Perturbations: Unlike specific network-based attacks, universal perturbations can fool classifiers across various images with high probability, as demonstrated by Moosavi-Dezfooli et al. attacking 85.4% of test samples in the ILSVRC 2012 dataset under a ResNet-152 classifier.

Issues—Integrity challenges arise from the various adversarial examples mentioned earlier. The adversary's objective is to disrupt the model. In this phase, although the adversary cannot poison the training data or tamper with model parameters, they can still access deployed models.

A5. Check and Updates

Administrators routinely assess the accuracy and performance of systems. If necessary, systems undergo retraining to update models.

Vulnerabilities and Attacks—Backdoor Attack: As discussed in phase A3, if the adversary successfully implants a trigger, it can be activated in this phase when the system detects trigger images, yielding the targeted class instead of the correct label.

Issues—Integrity issues may manifest as backdoor attacks deployed during training data at the implementation phase are triggered during parameter or model updates.

How to Defend AI Systems

The aforementioned challenges underscore the susceptibility of AI systems to data alteration or modification. When inaccurately labeled data is utilized, it can lead to diminished accuracy and erroneous system behavior. Consequently, various strategies have been devised to protect AI systems from such threats, categorized into two defense mechanisms: AI-algorithm-based and architecture-based. The former focuses on fortifying AI systems against ML attacks, while the latter aims to design architectures that prevent adversaries from infiltrating the system, thereby reducing the risk of data manipulation.

AI-Algorithm-Based Defenses

Researchers continuously seek defense mechanisms against AI system attacks, though adversaries persistently seek methods to circumvent them. Below, we present several examples of AI-algorithm-based defense mechanisms along with the attacks they effectively counter. These mechanisms are classified into two categories: complete defense and detection-only.

1. Adversarial Training: This widely adopted defense mechanism targets adversarial examples encountered during the inference phase. By retraining the model with adversarial examples while retaining correct labels, the model learns to disregard adversarial inputs, thereby enhancing accuracy. However, a drawback is that the model becomes 'immune' only to attacks it has been trained against previously.

2. Data Compression: JPEG compression has been shown to effectively counter adversarial attacks like FGSM and DeepFool by removing high-frequency components in images. However, excessive compression may lead to loss of accuracy.

3. Randomization: Adding random resizing and padding operations during inference can mitigate adversarial effects by introducing variability into input images.

4. Gradient Regularizations and Adversarial Training: Combining gradient regularization with adversarial training can enhance robustness against attacks like FGSM and JSMA. However, this approach significantly increases the training complexity.

5. SafetyNet: This method utilizes a Radial Basis Function SVM classifier to detect adversarial examples based on differences in ReLU activation patterns, effectively identifying examples generated by various attacks.

6. Convolution Filter Statistics: By analyzing statistics on convolutional layer outputs, a cascade classifier can detect adversarial examples with high accuracy.

7. Perturbation Rectifying Network (PRN): PRN aims to defend against universal perturbations by adding additional 'pre-input' layers to the model. A separate detector trained on Discrete Cosine Transform features identifies perturbations, enabling accurate classification.

8. GAN-Based Defenses: Generative Adversarial Networks (GANs) have been utilized to improve robustness against adversarial perturbations. For instance, generator networks generate adversarial perturbations while the classifier learns to correctly classify both original and adversarial examples.

9. Denoising/Feature Squeezing: Feature input spaces are often excessively large, offering adversaries numerous options. Feature squeezing reduces input features, making it harder for adversaries to generate adversarial examples. Spatial smoothing and color bit depth reduction are effective squeezing methods.

These AI-algorithm-based defense mechanisms offer various strategies to mitigate the impact of adversarial attacks on AI systems, contributing to enhanced robustness and security.

Table 2. Examples of AI-Algorithm-based defense mechanisms.

Defense Mechanisms	Effective to	Category
Adversarial training	Adversarial examples	Complete-defense
Data compression	FGSM, DeepFool	Complete-defense
Randomization	Adversarial examples	Complete-defense
Gradient regularizations + adversarial training	FGSM, JSMA	Complete-defense
SafetyNet	FGSM, BIM, DeepFool	Detection-only
Convolution filter statistics	Adversarial examples	Detection-only
Perturbation Rectifying Network (PRN)	Universal perturbations	Complete-defense
GAN-based	Adversarial perturbations	Complete-defense
Denoising/Feature squeezing	Adversarial perturbation to an image	Detection-only

Each of these defense mechanisms possesses its own strengths, weaknesses, and trade-offs. However, it is imperative to highlight that researchers and developers continue to explore and refine these algorithm-based defense mechanisms to achieve superior results.

Architecture-Based Defense

Architecture-based defense operates by constructing an architecture aimed at preventing adversaries from infiltrating the system, thereby reducing the likelihood of malicious modification of ML datasets. Certain features can be integrated into the architecture to bolster its resilience against data integrity violations, including:

- Strengthening the authentication mechanism to prevent fake users from impersonating legitimate ones.
- Enhancing the authorization mechanism by restricting user permissions based on their designated roles, thereby preventing unauthorized users from engaging in arbitrary behavior and ensuring that only authorized users can interact with the services.
- Monitoring datasets' integrity throughout the ML lifecycle phases using hash algorithms. This enables comparison of dataset hashes to detect any alterations by attackers.
- Implementing logging and monitoring of data flow and user activities to promptly identify suspicious actions.

As depicted in Table 3, we summarize the merits and weaknesses of these two defense mechanisms by assigning a plus (+) sign to indicate the phases covered by each mechanism and a minus (-) sign to indicate the phases not covered. Given that AI-algorithm-based defense operates at the algorithm level, it covers the ML lifecycle phases from implementation (A3) to check and updates (A5). However, it may fall short in ensuring data integrity during the collection and curation of training datasets (A1) and algorithm configuration (A2). Conversely, architecture-based defense, with the implementation of the aforementioned examples, can span phases A1 to A5.

Nevertheless, in our view, optimal defense for cloud-based AI systems entails incorporating both mechanisms, as they each possess strengths and weaknesses that can complement one another. We will delve further into our proposed architecture-based method in Section 4.

Table 3. AI-Algorithm-based vs. Architecture-based in ML pipeline.

Phase	AI-Algorithm-Based	Architecture-Based
Data acquisition and curation	–	+
Model design	–	+
Implementation	+	+
Inferences	+	+
Check and updates	+	+

Cloud Environment

Cloud computing technology offers numerous advantages for its users, including simplicity and rapid deployment. Instead of investing in building infrastructure with their own resources, users can conveniently leverage the services provided by Cloud Service Providers (CSPs). However, entrusting a third party to manage their systems and data necessitates a high level of trust in the chosen entity [52]. Additionally, users must remain vigilant about vulnerabilities and threats inherent in cloud computing. When utilizing cloud services, users transfer their resources from their secure perimeter to a CSP whose security measures may not be fully known. Furthermore, there are risks associated with data transmission. In the context of cloud-based AI systems, users are required to migrate sensitive

data such as training data, models, parameters, and configurations to the CSP. Compromise of these data could result in unintended system behaviors.

Drawing from the list of risks outlined by the Open Web Application Security Project (OWASP), we identify potential vulnerabilities and threats within the cloud environment. OWASP enumerates ten potential risks in cloud computing [53], and we focus on those related to data integrity within the scope of our paper. We categorize the challenges in the cloud environment into two main areas: system access and cloud infrastructure.

In the first category, risks pertain to accountability, data ownership, service credibility, and data integration. Users face concerns regarding the credibility and security of their data when storing and transmitting it to the CSP. Entrusting data to a third party introduces an additional layer of risk [54].

In the second category, risks center around multi-tenancy and infrastructure security. A distinguishing feature of cloud computing is its shared infrastructure, where multiple tenants share cloud resources and services. Failures in the multi-tenancy system pose potential risks, such as inadvertent access by one user to another user's data on the same host. Further details are provided in Table 4.

Category	Risks [53]	Vulnerabilities	Issues
System Access	R1. Accountability and Data Ownership, R6. Service and Data Integration	C1. Account and service hijacking	Adversary could gain access to the cloud resources and services
		C2. Malicious insiders	Leaked important data to adversary
		C3. Lack of authentication and authorization mechanisms	Impersonate real user to compromise the data, resources, and services
Cloud Infrastructure	R7. Multi-tenancy, R9. Infrastructure Security	C4. Insecure API gateway	Exposed to unauthorized data access that could lead to a black-box attack
		C5. Security misconfiguration	Breach in API, account and service hijacking
		C6. Multi-tenancy failure	One tenant can access neighbor's data or resources. Adversary could use it to harm data integrity

System Access

C1. Account and Service Hijacking: This threat arises from various tactics such as phishing, fraud, exploiting software vulnerabilities, and credential reuse. Attackers can illicitly acquire user credentials, thus gaining unauthorized access to services [55–57].

C2. Malicious Insiders: This threat, familiar to most organizations, involves individuals with insider access exploiting their privileges. The severity of the repercussions depends on the level of access, as individuals with higher privileges can access sensitive data and services. Malicious insiders pose significant risks, including theft of confidential data, reputational damage, financial losses, and productivity disruptions [55–57].

C3. Lack of Authentication and Authorization Mechanisms: Authentication and authorization serve as the primary defenses against unauthorized access to the cloud environment. Authentication verifies the identity of users, distinguishing between legitimate users and adversaries, while authorization controls data access by defining the access levels for each authenticated user. Absence of these mechanisms can lead to various compromises, such as unauthorized access to personal information and cloud services, loss of data privacy, and data leakage [55–57].

Cloud Infrastructure

C4. Insecure API Gateway: API gateways facilitate client interactions with cloud services. Inadequate security measures in API gateways can expose organizations to numerous threats, including anonymous access, credential reuse, and non-encrypted data transmission. Additionally, insecure API gateways can lead to account and service hijacking, data loss, and leakage. Vulnerabilities in API gateways can be exploited in black-box attacks in cloud-based AI systems, enabling adversaries to misuse insecure APIs to query ML models [55–57].

C5. Security Misconfiguration: Misconfigurations may occur at various levels, including frameworks, web servers, application stacks, or browsers. For instance, using a browser with weak security settings can lead to security misconfiguration. Such misconfigurations may result in interface breaches, API vulnerabilities, or account and service hijacking. It is imperative to regularly audit security configurations and utilize browsers or frameworks that enforce robust security policies [55].

C6. Multi-Tenancy Failure: Multi-tenancy is fundamental in cloud computing, allowing cloud vendors to share resources among multiple users. However, failures in multi-tenancy can compromise system integrity and expose users' data. For example, one user may inadvertently access another user's data, posing risks of data tampering by adversaries. Ensuring robust multi-tenancy mechanisms is crucial to maintaining data integrity in cloud environments [55].

Proposed Architecture

Architecture Requirements

After analyzing the vulnerabilities and challenges outlined in Section 2, we have identified several requirements necessary for constructing a robust data integrity architecture for cloud-based AI systems, as summarized in Table 6. Below, we present our proposed solutions that address these requirements:

Identity and Access Control Management: Proper identification and authorization of users before accessing cloud services are crucial for preventing data integrity compromises. Lack of authentication and authorization mechanisms can leave the system vulnerable to adversaries. Our solution involves using digital signatures to verify users' identities each time they access the system and its services.

Consistency and Completeness: Maintaining consistency and completeness of ML datasets is vital to ensure accurate results. Any tampering or imbalance in the training data can lead to biased decisions by the AI system. We address this requirement by employing hash functions to monitor and verify data integrity. Additionally, we record this information in a blockchain, leveraging its decentralized nature to detect and signal any data alterations.

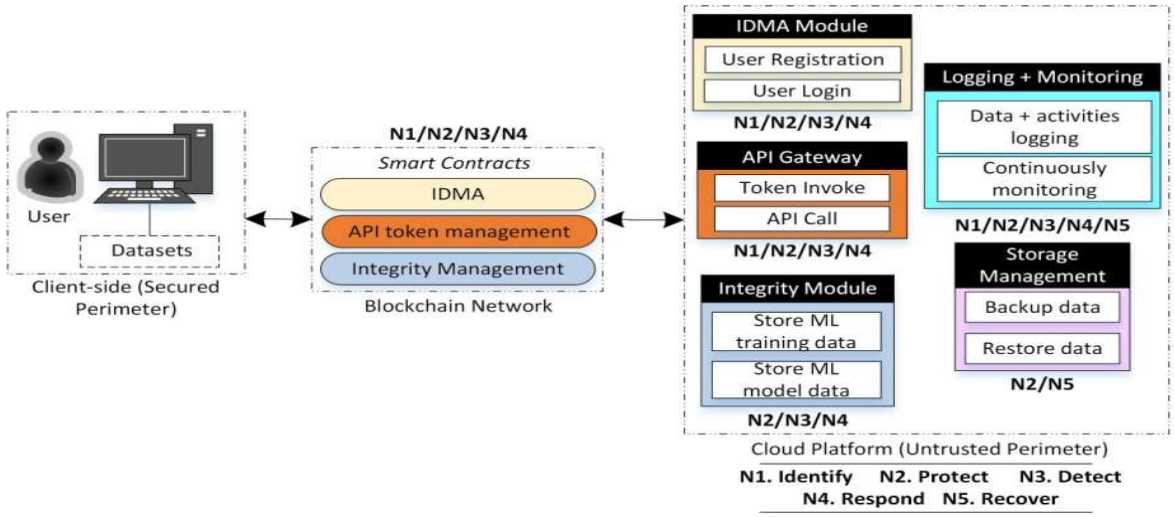
Non-Repudiation: Malicious insiders pose a significant threat to system integrity, as they can exploit their access privileges to leak or modify data undetected. To address this, users are required to sign data changes to verify their identity. Furthermore, we implement logging and monitoring mechanisms to track and flag any unusual user activities within the system.

Trusted Service Level Agreement (SLA): SLAs serve as contractual agreements between customers and CSPs, establishing trust between both parties. However, the possibility of breaches or trust issues remains. Our solution involves using smart contracts to automate and enforce SLAs, minimizing vulnerabilities exposed to unauthorized parties. This approach enhances policy enforcement across AI pipeline phases and mitigates cloud vulnerabilities.

Each of these proposed solutions aims to strengthen the overall security posture of cloud-based AI systems, addressing specific vulnerabilities and ensuring robust data integrity protection.

Table 6. Requirements for data integrity architecture for Cloud-based AI Systems.

Requirement	Covered AI Pipeline	Covered Cloud Vulnerabilities	Description	Our Proposed Solutions
Identity and Access Control Management	A2	C1, C3, C4, C5	Prevent adversary to impersonate the real user to login to cloud environment and gain control over the data.	We use a digital signature to verify the user's identity every time they enter the system and use services.
Consistency and Completeness	A1, A5		Prevent adversary to alter the training data, mislabelled the training data to another value, and disrupt the consistency in datasets.	We use a hash function to keep track to ensure no violation of data integrity and record it in the blockchain.
Non-repudiation	A2	C2	Prevent malicious insiders that has direct access to services and data to leak the information to the adversary or even modify it undetected.	Whenever users changes the data, they need to sign it to verify their identity.
Trusted SLA	A1, A2, A5	C1–C6	Trust issues between user and CSP.	We use smart contracts to bind the trust between users and CSP that can automate the SLA process.



Architecture Details

We present our proposed architecture that fulfills the four requirements outlined above in Figure 2.

In this architecture, there are three main components: the user on the client-side, smart contracts in the blockchain network, and the cloud platform/CSP (assumed to be untrustworthy). Users interact with the cloud platform by performing actions such as uploading ML datasets, training ML models, and using AI services via API calls. To ensure data integrity throughout the ML lifecycle in the cloud environment, we propose an architecture based on the NIST cybersecurity framework. The mapping of our modules to the NIST framework guidance is depicted in Figure 2, and the analysis of the mapping is discussed in Section 5. The architecture consists of five modules: Identity Management and Access Control (IDMA), API Gateway, Integrity Module (IM), Logging Monitoring, and Storage Management. Additionally, we design six protocols for the first three modules: IDMA, API gateway, and IM.

Identity Management and Access Control Module (IDMA): This module handles user authentication and access control by assigning roles to users and ensuring proper authentication before accessing cloud services. Users must prove their authenticity before accessing the cloud system to prevent impersonation by adversaries. New users are assigned roles and permissions upon registration. Role-Based Access Control (RBAC) is utilized as a policy enforcement mechanism, with three roles defined: General User, Log Admin, and System Admin.

1. General User: Authorized to access personal data and cloud services through API calls with an additional token.
2. Log Admin: Authorized to access logging data for system analysis.
3. System Admin: Authorized to manage user roles and access user information.

Conclusion:

In the rapidly evolving landscape of AI and machine learning (ML) technologies, the integration of cloud computing has brought forth unprecedented opportunities for innovation and efficiency. However, with these advancements

come significant challenges related to data integrity, security, and compliance within cloud-based AI/ML ecosystems. As organizations increasingly rely on these technologies to drive critical decision-making processes, safeguarding data integrity and ensuring compliance with regulatory frameworks have become paramount concerns.

Throughout this paper, we have delved into the intricate interplay between AI/ML technologies and cloud computing environments, identifying vulnerabilities, threats, and potential solutions to mitigate risks and enhance data guardianship. From the data acquisition and curation phase to the deployment of AI models and ongoing monitoring, each stage of the AI/ML lifecycle presents unique challenges that must be addressed to maintain data integrity and regulatory compliance.

We have outlined a comprehensive architecture that incorporates both algorithm-based defenses and architecture-based mechanisms to protect against threats such as data poisoning, adversarial attacks, and unauthorized access. By leveraging cryptographic techniques, blockchain technology, and role-based access control, organizations can establish robust defenses to safeguard data integrity, authenticate users, and enforce access controls within cloud-based AI/ML ecosystems.

Furthermore, we have underscored the importance of compliance with regulatory frameworks such as GDPR, HIPAA, and CCPA, which impose stringent requirements for the protection of sensitive data and the rights of individuals. By integrating compliance measures into the design and implementation of AI/ML systems, organizations can ensure transparency, accountability, and trustworthiness in their data practices.

In conclusion, safeguarding compliance and data guardianship in AI/ML cloud ecosystems requires a multi-faceted approach that encompasses technological innovations, regulatory adherence, and organizational best practices. By adopting a proactive stance towards data security, privacy, and regulatory compliance, organizations can harness the full potential of AI/ML technologies while minimizing risks and building trust among stakeholders. As we continue to navigate the evolving landscape of AI/ML and cloud computing, the pursuit of data guardianship will remain a foundational imperative for organizations seeking to thrive in the digital age.

References :

[1]. Anyoha, R. Exploring the Evolution of Artificial Intelligence. Retrieved from: https://bit.ly/3x2jid7 (accessed on March 3, 2021).

[2]. Marr, B. Unveiling the Remarkable Milestones of Artificial Intelligence. Retrieved from: https://bit.ly/3oLq2s3 (accessed on December 1, 2020).

[3]. West, D.M.; Allen, J.R. Unraveling the Impact of Artificial Intelligence on Our World. Retrieved from: https://brook.gs/3CyGQrp (accessed on December 1, 2020).

[4]. Herpig, D.S. Enhancing the Security of Artificial Intelligence. 2019; p. 48. Retrieved from: https://bit.ly/3nMt2F9 (accessed on September 15, 2020).

[5]. Brundage, M.; Avin, S.; Clark, J.; Toner, H.; Eckersley, P.; Garfinkel, B.; Dafoe, A.; Scharre, P.; Zeitzoff, T.; Filar, B.; et al. Anticipating, Preventing, and Mitigating the Malevolent Use of Artificial Intelligence. arXiv 2018, arXiv:1802.07228.

[6]. Foremski, T. Investigating Trust Issues with Tech Giants. Retrieved from: https://zd.net/3mCATVe (accessed on May 18, 2021).

[7]. Wang, Y.; Wen, J.; Zhou, W.; Luo, F. Introducing a Novel Dynamic Model for Evaluating Cloud Service Trust in Cloud Computing. In Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, August 1–3, 2018; pp. 10–15.

[8]. Pillai, A. S. (2023). Advancements in Natural Language Processing for Automotive Virtual Assistants Enhancing User Experience and Safety. *Journal of Computational Intelligence and Robotics*, *3*(1), 27-36.
https://thesciencebrigade.com/jcir/article/view/161

[9]. Sarker, M. (2022). Towards Precision Medicine for Cancer Patient Stratification by Classifying Cancer By Using Machine Learning. *Journal of Science & Technology*, *3*(3), 1-30.
DOI: https://doi.org/10.55662/JST.2022.3301

[10]. Manoharan, A., & Sarker, M. REVOLUTIONIZING CYBERSECURITY: UNLEASHING THE POWER OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR NEXT-GENERATION THREAT DETECTION. *DOI: https://www. doi. org/10.56726/IRJMETS32644*.

DOI : https://www.doi.org/10.56726/IRJMETS32644

[11]. Bappy, M. A., & Ahmed, M. (2023). ASSESSMENT OF DATA COLLECTION TECHNIQUES IN MANUFACTURING AND MECHANICAL ENGINEERING THROUGH MACHINE LEARNING MODELS. *Global Mainstream Journal of Business, Economics, Development & Project Management*, *2*(04), 15-26.

DOI: https://doi.org/10.62304/jbedpm.v2i04.67

[12]. Hossain, M. I., Bappy, M. A., & Sathi, M. A. (2023). WATER QUALITY MODELLING AND ASSESSMENT OF THE BURIGANGA RIVER USING QUAL2K. *Global Mainstream Journal of Innovation, Engineering & Emerging Technology*, *2*(03), 01-11.
DOI: https://doi.org/10.62304/jieet.v2i03.64

[13]. Sharma, Y. K., & Harish, P. (2018). Critical study of software models used cloud application development. *International Journal of Engineering & Technology, E-ISSN*, 514-518.

https://scholar.google.com/citations?view_op=view_citation&hl=en&user=Fxv3elcAAAAJ&citation_for_view=Fxv3elcAAAAJ:d1gkVwhDpl0C

[14]. Padmanaban, H. (2023). Navigating the intricacies of regulations: Leveraging AI/ML for Accurate Reporting. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, *2*(3), 401-412.

DOI: https://doi.org/10.60087/jklst.vol2.n3.p412

[15]. Padmanaban, P. H., & Sharma, Y. K. (2019). Implication of Artificial Intelligence in Software Development Life Cycle: A state of the art review. *vol*, *6*, 93-98.

https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Implication+of+Artificial+Intelligence+in+Software+Development+Life+Cycle%3A+A+state+of+the+art+review&btnG=

[16]. PC, H. P., Mohammed, A., & RAHIM, N. A. (2023). *U.S. Patent No. 11,762,755*. Washington, DC: U.S. Patent and Trademark Office.

https://patents.google.com/patent/US11762755B2/en

[17]. Miah, S., Rahaman, M. H., Saha, S., Khan, M. A. T., Islam, M. A., Islam, M. N., ... & Ahsan, M. H. (2013). Study of the internal structure of electronic components RAM DDR-2 and motherboard of nokia-3120 by using neutron radiography technique. *International Journal of Modern Engineering Research (IJMER)*, *3*(60), 3429-3432

https://shorturl.at/nCJOQ

[18]. Rahaman, M. H., Faruque, S. B., Khan, M. A. T., Miah, S., & Islam, M. A. (2013). Comparison of General Relativity and Brans-Dicke Theory using Gravitomagnetic clock effect. *International Journal of Modern Engineering Research*, *3*, 3517-3520.
https://shorturl.at/hjm37

[19]. Miah, M. H., & Miah, S. (2015). The Investigation of the Effects of Blackberry Dye as a Sensitizer in TiO2 Nano Particle Based Dye Sensitized Solar Cell. *Asian Journal of Applied Sciences*, *3*(4).
https://shorturl.at/iyJQV

[20]. Miah, S., Miah, M. H., Hossain, M. S., & Ahsan, M. H. (2018). Study of the Homogeneity of Glass Fiber Reinforced Polymer Composite by using Neutron Radiography. *Am. J. Constr. Build. Mater*, *2*, 22-28.
https://shorturl.at/joDKZ

[21]. Miah, S., Islam, G. J., Das, S. K., Islam, S., Islam, M., & Islam, K. K. (2019). Internet of Things (IoT) based automatic electrical energy meter billing system. *IOSR Journal of Electronics and Communication Engineering*, *14*(4 (I)), 39-50.

[22]. Nadia, A., Hossain, M. S., Hasan, M. M., Islam, K. Z., & Miah, S. (2021). Quantifying TRM by modified DCQ load flow method. *European Journal of Electrical Engineering*, *23*(2), 157-163.

https://shorturl.at/csuO3

[23]. Miah, S., Raihan, S. R., Sagor, M. M. H., Hasan, M. M., Talukdar, D., Sajib, S., ... & Suaiba, U. (2022). Rooftop Garden and Lighting Automation by the Internet of Things (IoT). *European Journal of Engineering and Technology Research*, *7*(1), 37-43.
DOI: https://doi.org/10.24018/ejeng.2022.7.1.2700

[24]. Prasad, A. B., Singh, S., Miah, S., Singh, A., & Gonzales-Yanac, T. A Comparative Study on Effects of Work Culture on employee satisfaction in Public & Private Sector Bank with special reference to SBI and ICICI Bank.

[25]. Ravichandra, T. (2022). A Study On Women Empowerment Of Self-Help Group With Reference To Indian Context.

https://www.webology.org/data-cms/articles/20220203075142pmwebology%2019%20(1)%20-%2053.pdf

[26]. Kumar, H., Aoudni, Y., Ortiz, G. G. R., Jindal, L., Miah, S., & Tripathi, R. (2022). Light weighted CNN model to detect DDoS attack over distributed scenario. *Security and Communication Networks*, *2022*.
https://doi.org/10.1155/2022/7585457

[27]. Ma, R., Kareem, S. W., Kalra, A., Doewes, R. I., Kumar, P., & Miah, S. (2022). Optimization of electric automation control model based on artificial intelligence algorithm. *Wireless Communications and Mobile Computing*, *2022*.
https://doi.org/10.1155/2022/7762493

[28]. Devi, O. R., Webber, J., Mehbodniya, A., Chaitanya, M., Jawarkar, P. S., Soni, M., & Miah, S. (2022). The Future Development Direction of Cloud-Associated Edge-Computing Security in the Era of 5G as Edge Intelligence. *Scientific Programming*, *2022*.
https://doi.org/10.1155/2022/1473901

[29]. Al Noman, M. A., Zhai, L., Almukhtar, F. H., Rahaman, M. F., Omarov, B., Ray, S., ... & Wang, C. (2023). A computer vision-based lane detection technique using gradient threshold and hue-lightness-saturation value for an autonomous vehicle. *International Journal of Electrical and Computer Engineering*, *13*(1), 347.
https://shorturl.at/ceoyJ

[30]. Patidar, M., Shrivastava, A., Miah, S., Kumar, Y., & Sivaraman, A. K. (2022). An energy efficient high-speed quantum-dot based full adder design and parity gate for nano application. *Materials Today: Proceedings*, *62*, 4880-4890.

https://doi.org/10.1016/j.matpr.2022.03.532