VindSec-Llama — Fine-Tuned Meta's Llama-3 LLM, Federated Learning, Blockchain and PBOM-enabled Data Security Architecture for Wind Energy Data Platforms

Eranga Bandara*, Safdar H. Bouk*, Sachin Shetty*, Ross Gore*, Sastry Kompella§, Ravi Mukkamala*,
Abdul Rahman†, Peter Foytik*, Mohan Sunkara*, Xueping Liang‡,
* {cmedawer, sbouk, sshetty, mukka, pfoytik, rgore }@odu.edu
Old Dominion University, Norfolk, VA, USA
† abdulrahman@deloitte.com
Deloitte & Touche LLP
‡ xuliang@fiu.edu
Florida International University, USA
§ skompella@nexcepta.com
U.S. Naval Research Laboratory

***Abstract*—Current wind energy data platforms face significant challenges in securing and managing extensive data from both offshore and onshore wind farms. These challenges include vulnerabilities to cyber-attacks, data tampering, breaches, complex data-sharing issues due to privacy concerns and regulatory compliance, and a lack of scalability and flexibility in analytical tools for real-time data processing. This paper proposes a novel multi-layered data security architecture, termed "VindSec-Llama," to address these challenges. It integrates Generative AI, blockchain, federated learning, and Pipeline Bill of Materials (PBOM) to enhance data analytics, model development, and security across several layers, including Infrastructure, Data Lake, Federated Learning, MLOps, Data Provenance, and LLM. Each layer is designed to meet specific functional requirements, such as handling large datasets, facilitating secure federated learning, automating risk management, and ensuring data provenance and traceability. The platform, deployable in server environments (cloud or on-premises), complies with the Risk Management Framework (RMF) guidelines and security standards. It features a blockchain-enabled, coordinator-less federated learning system to enhance data privacy and security by enabling the development of privacy-preserving machine learning models with data from different wind farms. Automation plays a pivotal role throughout VindSec-Llama, with Meta's custom-trained Llama-3 LLM used for generating remediation scripts in the Infrastructure Layer and for producing PPBOM in the MLOps Layer. The Llama-3 LLM has been quantized and fine-tuned using Qlora to ensure optimal performance on consumer-grade hardware. The MLOps pipeline setup, a critical functionality of VindSec-Llama, ensures seamless integration and deployment of machine learning models, embodying best practices in continuous integration and delivery. This setup is geared towards maximizing security, compliance, and operational efficiency. End-to-end data provenance in the system is captured as ModelCards and NFT objects. A prototype of the platform has been implemented within a wind-energy testbed with the collaboration of Department of Energy US, illustrating its practical applications and benefits.**

***Index Terms*—DevSecOps, Generative-AI, LLM, Llama-3, Blockchain, NFT, PBOM, Wind-Energy**

I. Introduction

Wind energy, a cornerstone of sustainable power generation, increasingly relies on sophisticated data platforms to optimize performance and ensure reliability. These platforms collect, process, and analyze vast amounts of data from sensors, turbines, and environmental inputs to facilitate real-time decision-making, predictive maintenance, and energy management [1]. However, the security of these data platforms is a critical concern, as they are frequent targets for cyber-attacks that can lead to data breaches, system disruptions, and even physical damage to the infrastructure. The proprietary nature of the data and its importance to grid stability and energy forecasting further exacerbate these security vulnerabilities [2]. Traditional security measures often fall short due to the complexity and dynamic nature of the networked environments in which wind farms operate. Additionally, the integration of these platforms with public grids and the internet increases exposure to potential cyber threats, underscoring the need for advanced, resilient security architectures that can adapt to the evolving landscape of cyber risks while maintaining the integrity and availability of critical wind energy data [3].

To address these challenges, this paper introduces a novel data security architecture for wind energy data platforms, known as "VindSec-Llama." This platform incorporates Meta's custom-trained Llama-3 LLM [4], [5], blockchain [6], [7], federated learning [8], and PBOM [9], [10], all aimed at enhancing data analysis, model development, and security. VindSec-Llama is structured into several integrated layers: the Infrastructure Layer, Data Lake Layer, Federated Learning Layer, MLOps Layer, Data Provenance Layer, and the LLM Layer. The deployment of this platform in an infrastructure (e.g., cloud or on-premises) adheres strictly to RMF guidelines

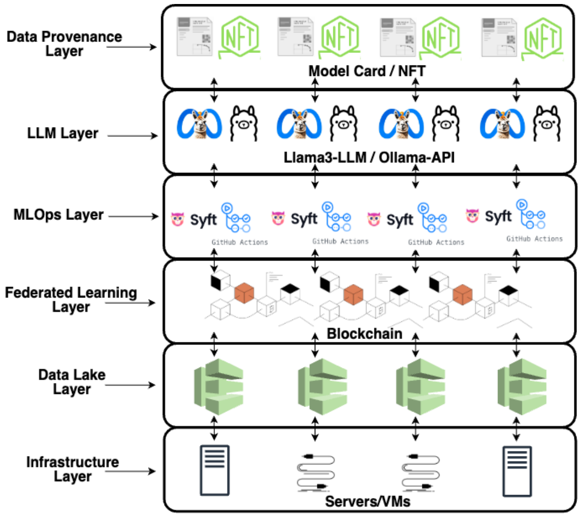


Fig. 1: Platform layered architecture.

and security standards [2]. All vulnerabilities in the infrastructure are scanned and identified against SCAP Security Guide (SSG) frameworks, such as STIG and PCI-DSS [11]. These identified vulnerabilities are fixed using remediation scripts generated by Meta's custom-trained Llama-3 LLM. The blockchain-enabled, coordinator-less federated learning system [8] allows the development of privacy-preserving machine learning models using data from different wind farms. The MLOps pipeline setup promotes the seamless integration and deployment of machine learning models, aligning with best practices in continuous integration and delivery. Moreover, the custom-trained Llama-3 LLM is used to produce PBOMs in the MLOps Layer, where each supply chain/pipeline information is tracked as PBOMs. The data provenance information in the system is captured as Model Cards [12] and NFTs [13], and stored in the underlying blockchain. A prototype of the platform has already been implemented within a wind-energy testbed, demonstrating its practical applications and significant benefits. This paper will detail our major contributions to this field.

1) Introduction of the Data Platform, which integrates Generative AI, blockchain, federated learning, and PBOM to address key security and operational challenges in wind energy data management.
2) Integration of a novel, blockchain-enabled, coordinator-less federated learning system within VindSec-Llama, enhancing data privacy and security across distributed wind energy data networks.
3) Integration of automation processes using fine-tuned Llama-3 LLM, including the generation of remediation scripts and PBOMs, facilitating rigorous security practices and efficient model deployment in server environments (e.g., cloud, on-premises).
4) Demonstration of VindSec-Llama's effectiveness through a prototype deployed within a wind-energy testbed at VMASC in Virginia, USA, in collaboration with the US Department of Energy.

II. System Architecture

Figure 1 describes the architecture of the platform. The proposed platform comprises six layers: 1) Infrastructure Layer, 2) Data Lake Layer, 3) Federated Learning Layer, 4) MLOps Layer, 5) LLM Layer, and 6) Data Provenance Layer. Below is a brief description of each layer:

A. *Infrastructure Layer*

The Infrastructure Layer is composed of an array of on-premises and cloud servers, including hardware servers and cloud virtual machines (VMs), each integral to the operational framework of the platform. In compliance with RMF guidelines, these servers are mandated to be managed securely, adhering to established standards such as NIST SP 800-53 [2]. This encompasses regular scanning for vulnerabilities and timely remediation of identified security issues. However, automating the RMF process for this infrastructure presents considerable challenges. These challenges stem from the inherent complexity of the system, the need to align with diverse standards like NIST SP 800-53, and the requirement to support continuous Authority to Operate (ATO) [14]. To address these challenges, the VindSec-Llama platform proposes an end-to-end RMF automation system. This system integrates custom-trained Meta's Llama-3 LLM with OpenSCAP, a tool for assessing and maintaining security compliance. The vulnerability scanning performance with OpenSCAP and the vulnerability fixing scripts (e.g., Ansible or Puppet [11]) generate through the fine-tuned Meta's Llama-3 LLM. These scripts are then executed to address the vulnerabilities within the server infrastructure. This approach culminates in a fully automated RMF system.

B. *Data Lake Layer*

The Data Lake Layer plays a pivotal role, particularly within the context of a blockchain-enabled, coordinator-less federated learning system [8]. In this system, various peers participate in the federated learning tasks, each acting as a node within the blockchain network. Notably, each peer maintains its own data lake, contributing to the decentralized nature of the system. Therefore, the Data Lake Layer is comprised of these individual data lakes, distributed across the various peers participating in the federated learning tasks. This structure allows the federated learning system to build privacy-preserving machine learning models using data from different wind farms.

C. *Federated Learning Layer*

The Federated Learning Layer within the proposed VindSec-Llama architecture addresses the limitations of conventional federated learning systems. Traditional federated learning frameworks typically rely on a centralized coordinator to aggregate local machine learning models, a structure that is inherently vulnerable to attacks and privacy breaches. Moreover, these systems often fall short in providing satisfactory transparency and provenance of the resulting machine learning models [8]. To overcome these challenges, the VindSec-Llama platform employs a blockchain-enabled, coordinator-less federated learning system. This approach ensures that each peer participating in the federated learning tasks is connected as a node within the blockchain network. Crucially, each peer maintains its own data lake, where real data is stored securely. These peers independently train their local machine-learning models using the data from their respective data lakes. Subsequently, these local models are aggregated to form a comprehensive global model. This allows for the building of privacy-preserving machine learning models with the data from geo-distributed wind farms [15].

D. MLOps Layer

The MLOps Layer in the proposed VindSec-Llama orchestrates the continuous integration and delivery of various services within the system. This layer leverages CI/CD pipelines, such as GitHub Actions, to continuously build and update federated learning models and the services that utilize these models [16]. An essential function of the MLOps Layer is to deploy these models and associated services in the server infrastructure using cloud-native container orchestration systems like Docker and Kubernetes [17]. One of the main challenges in cloud-native software deployment is mitigating supply-chain attacks and addressing vulnerabilities inherent in open-source tools. Additionally, tracking the entire development lifecycle and managing data provenance are complex tasks. The proposed VindSec-Llama platform addresses these challenges using SBOMs and PBOMs [10], [18]. Each code change (e.g., done as pull requests) in the ML models is scanned through SBOMs to identify vulnerabilities and dependencies. Furthermore, a custom-trained Meta's Llama-3 LLM generates PBOMs specific to each pull request. These PBOMs are created as JSON schemas, detailing critical aspects such as pull request information, test results, identified vulnerabilities in the built container, and the pull request's status and timestamps.

E. LLM Layer

The LLM Layer serves the generative AI functions across the aforementioned layers. Central to this layer is the deployment of the custom-trained Meta's Llama-3 LLM. This LLM is used to generate server-hardening scripts (based on the vulnerabilities identified in the Infrastructure Layer) and PBOMs in the MLOps Layer. To prepare the Llama-3 LLM, we've undertaken a meticulous training process, collaborating with Qlora to incorporate a 4-bit quantized pre-trained language model with Low-Rank Adapters (LoRA) [19], as shown in Figure2. During the training phase, we meticulously feed the model with a dataset encompassing pull-request information, test result data, software package vulnerability scan results, Ansible/Puppet remediation scripts, statuses, and the desired PBOM format represented as a JSON schema [10]. As a result, the Llama-3 LLM becomes proficient in responding to requests for the generation of new PBOMs for software updates, utilizing the input data provided (which typically includes pull-request details, SBOM information, statuses, and more) and Ansible/Puppet remediation script generation. The fine-tuned Llama-3 LLM runs on Ollam [20].

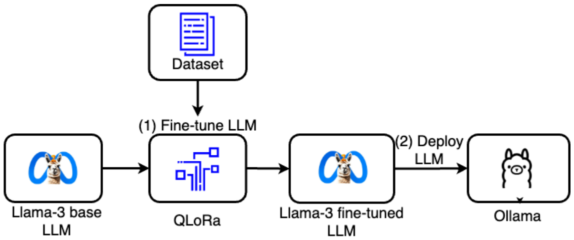


Fig. 2: Fine-tune Llama-3 LLM with Qlora.

F. Data Provenance Layer

The Data Provenance Layer is designed to manage the data provenance functions associated with federated learning, RMF automation of the server infrastructure, and CI/CD pipeline verification. In the Federated Learning Layer, the data provenance information of the ML models is meticulously recorded using Model Cards [12]. These cards capture the evolution of the model data from its origin, encompassing aspects such as ML model ownership and the storage locations of the ML models. For the RMF automation in the Infrastructure Layer, the system statuses, including vulnerabilities and their remediation, are also encapsulated within customized NFT tokens. In the MLOps Layer, the data provenance information pertaining to the end-to-end software and pipeline verification is stored and managed via Model Cards. These cards also provide detailed information about all activities involved in the scanning and pipeline verification processes.

III. Functionality

The platform encompasses eight main functionalities: 1) LLM fine-tuning, 2) Server infrastructure setup, 3) Server vulnerability scanning and fixing, 4) Federated model training, 5) MLOps pipeline setup, 6) SBOM/PBOM generation, 7) Data provenance handling, and 8) Attack mitigation. This section delves into the specifics of these functions.

A. LLM Fine-Tuning

VindSec-Llama utilizes a custom-trained LLama-3 LLM for PBOM generations. The LLama-3 LLM is precisely fine-tuned and trained specifically for PBOM generation with the software-supply chain information. To accomplish this, we undergo a rigorous training process, collaborating with QLoRA to convert a 4-bit quantized pre-trained language model into LoRA, thereby optimizing performance even on consumer-grade hardware. During the training phase, we carefully feed the model with a rich dataset encompassing the pull-request information, test result data, software package vulnerability scan results, SBOM data, statuses, and the desired PBOM format represented as a JSON schema 11. As a result, the LLama-3 LLM becomes proficient in responding to requests for the generation of new PBOMs for software updates, utilizing the input data provided, which typically includes pull-request details, SBOM information, statuses, and more.

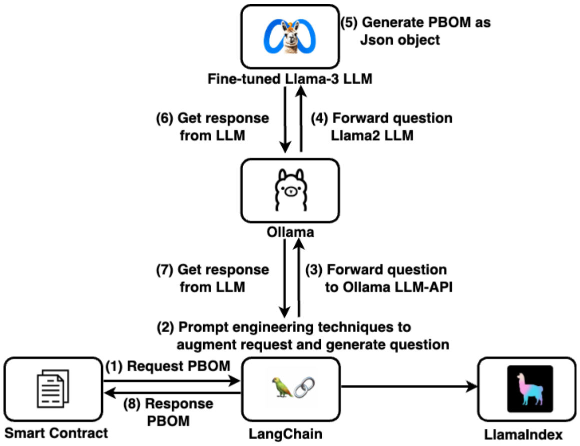

(5) Generate PBOM as Json object
Fine-tuned Llama-3 LLM
(6) Get response from LLM
(4) Forward question Llama2 LLM
Ollama
(7) Get response from LLM
(3) Forward question to Ollama LLM-API
(2) Prompt engineering techniques to augment request and generate question
(1) Request PBOM
(8) Response PBOM
Smart Contract
LangChain
LlamaIndex

Fig. 3: PBOM generation flow with Ollama LLM-API, Llama-3 LLM, LlamaIndex, LangChain and Smart Contracts.

By utilizing the privacy-preserving custom-trained LLM, we effectively mitigate the data privacy issues inherent in the use of commercial LLMs such as OpenAI GPT.

The fine-tuned LLama-3 LLM operates within Ollama [20], as depicted in Fig. 2. Interaction with Meta's LLama-3 LLM is facilitated through Ollama's LLM API, Llamaindex [21], and LangChain [22]. This robust interface enables streamlined and effective communication with the language model, as illustrated in Fig. 3.

B. Server Infrastructure Setup

The next functionality of the proposed platform is the setup of the server infrastructure, a foundational step critical for the successful deployment and operation of the platform. The infrastructure setup encompasses the deployment of a network of cloud/on-premises servers, which may include both hardware servers and VMs. Key to this setup is adherence to stringent security protocols and compliance standards, such as NIST SP 800-53, to ensure the infrastructure aligns with military-grade security requirements [2].

C. Server Vulnerability Scanning and Fixing

A key function of the server infrastructure is to handle vulnerability scanning of servers onboarded onto the platform. For instance, consider scanning vulnerabilities in Ubuntu 20.04 servers based on the STIG guidelines and fixing them according to the STIG standard. OpenScap contains SCAP documents related to Ubuntu 20.04 servers, such as "scap-security-guide-0.1.60/ssg-ubuntu2004-ds.xml" [11]. The SCAP document includes variable providers that relate to different compliance standards such as STIG and CIS. The blockchain smart contract interacts with the OpenScap API on each server and instructs it to perform the scan according to the STIG profile. The scan generates a scan/audit report which includes the server's STIG compliance score, based on the vulnerabilities found in the system, and detailed information about the vulnerabilities. Our system then offers automated server hardening capabilities based on the identified vulnerabilities. To achieve this, we have leveraged the custom-trained Llama-3 LLM to automatically produce an Ansible playbook or bash script, aligning with security compliance standards such as STIG. The Ansible playbook is dynamically generated by considering the vulnerabilities identified within the system and adhering to STIG compliance requirements, as shown in Figure13. Once the server hardening playbook is ready, the smart contract triggers its execution on the respective server, effectively implementing the necessary hardening measures. The identified vulnerabilities and system statuses are recorded as NFT objects and stored in the blockchain ledger for further verification and auditing functions.

D. Federated Model Training

In the federated learning environment of VindSec-Llama, numerous peers collaborate. Each peer maintains their own set of data, which is stored in an off-chain data lake. These data lakes could be geographically distributed across different wind energy power plants. In federated learning, initially, a leader peer is selected among the peers based on a consensus algorithm to handle coordination functions. The leader peer manages the creation of the federated pipeline, the initialization of model parameters, and the aggregation of the global model. The leader peer initializes the federated pipeline and model parameters and publishes them on a blockchain [8]. Subsequently, other peers retrieve the initial model and model parameters from the blockchain and train their local ML models (according to the model parameters) with their own datasets (stored in the off-chain storage). Once all peers have created their local models, the leader peer averages them into a global model using algorithms like stochastic gradient descent (SGD). Finally, the global model block is transmitted to all peers, as shown in Figure 4. The data provenance information of the ML models is captured as Model Cards.

E. MLOps Pipeline Setup

The MLOps pipeline setup is designed to streamline and optimize the machine learning (ML) lifecycle within the platform. This setup involves establishing a robust and automated pipeline for the continuous integration, delivery, and deployment of machine-learning models and associated services. The core objective of the MLOps pipeline is to enhance the efficiency, reproducibility, and reliability of ML model development and deployment processes, aligning them with best practices and operational requirements of military intelligence. Central to this setup is the utilization of CI/CD tools, such as GitHub Actions, which automate the building, testing, and deployment of federated learning models and other related services [16]. This automation ensures that any updates or changes to the ML models or their dependencies are seamlessly integrated and deployed without disrupting the platform's operations. The MLOps pipeline also leverages containerization technologies like Docker and orchestration tools such as Kubernetes, facilitating the deployment of ML models in a cloud-native environment [17]. This approach ensures scalability and aids in the efficient management of resources.

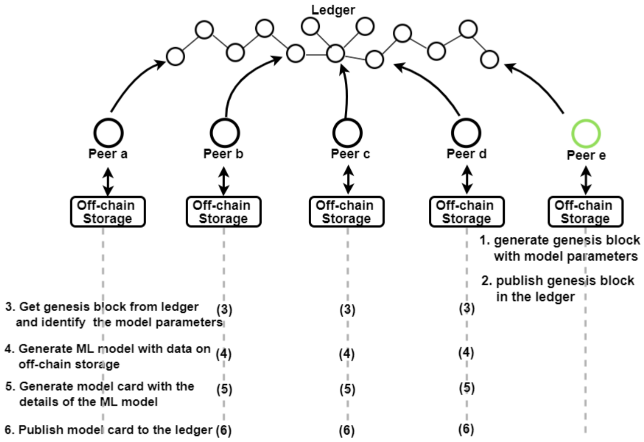

Ledger
Peer a
Peer b
Peer c
Peer d
Peer e
Off-chain Storage
Off-chain Storage
Off-chain Storage
Off-chain Storage
Off-chain Storage
1. generate genesis block with model parameters
2. publish genesis block in the ledger
3. Get genesis block from ledger and identify the model parameters
4. Generate ML model with data on off-chain storage
5. Generate model card with the details of the ML model
6. Publish model card to the ledger
(3)
(4)
(5)
(6)

Fig. 4: VindSec-Llama federated learning pipeline.

F. SBOM/PBOM Generation

For each code change (e.g., pull request [16]) of the federated learning models, the system performs a comprehensive scan of the containers, meticulously generates SBOMs (in SPDX-JSON or CycloneDX formats) [18], and identifies vulnerabilities within the system. Similarly, PBOMs are meticulously generated for each pull request, leveraging the capabilities of our custom-trained Meta's Llama-3 LLM. As soon as a pull-request is initiated in the ML Model, the fine-tuned Meta's Llama-3 LLM creates PBOMs based on data including pull-request details, unit test results, status, and case numbers. These PBOMs link with the respective SBOM analysis results of the pull request. The generated PBOMs are encoded as NFT tokens and stored in the underlying blockchain ledger. In this way, the system maintains a thorough and immutable record of all component details and their evolution, bolstering transparency and accountability throughout the software development lifecycle [23].

G. Data Provenance Handling

The platform also records data provenance information of federated learning model training as model card objects to provide transparency and traceability of the processes and ensure compliance with regulatory requirements. A model card is a standardized format used for documenting the performance and associated metadata of machine learning models. The information about the vulnerabilities identified in the servers, server hardening statuses, and PBOMs generated for each code change of the ML models are captured and recorded as NFT tokens. This information can be useful for auditors, regulators, and stakeholders to assess the effectiveness of the scanning and fixing processes and the overall security posture of the system. These model cards and NFTs are stored in the blockchain ledger, ensuring their immutability and tamper-proof nature [24].

H. Attack Mitigation

The platform's multifaceted security approach, incorporating blockchain, NFTs, and AI-driven PBOM generation, forms a formidable defense against a wide array of potential cyberattacks. The immutability of blockchain ledger entries ensures the integrity of data and resists attempts to manipulate identities. The decentralization of federated learning enhances resilience against attacks targeting centralized control points. Additionally, the model cards-based data provenance handling of ML models facilitates auditing and effectiveness verification. The PBOMs enhance security by proactively identifying vulnerabilities in the ML models and tracing the supply chain information. When deploying a model, the supply chain data can be verified, effectively countering pipeline attacks, including supply chain vulnerabilities. The end-to-end RMF automation of the servers and the system facilitates the requirement to support continuous ATO. The NFTs add an extra dimension of transparency and accountability, specifically with their representation of vulnerabilities, system statuses, and pipeline verification [23]. The integrated system stands as a comprehensive defense mechanism, bolstering system security and resilience against an array of potential threats.

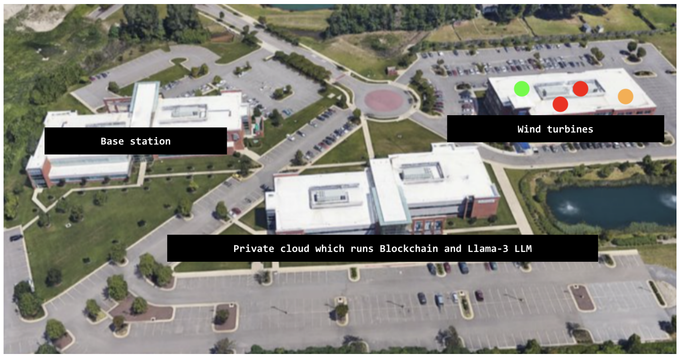


Fig. 5: Proposed large-scale testbed with wind turbines, on-prem Llama-3 LLM and blockchain in VMASC Virginia US.

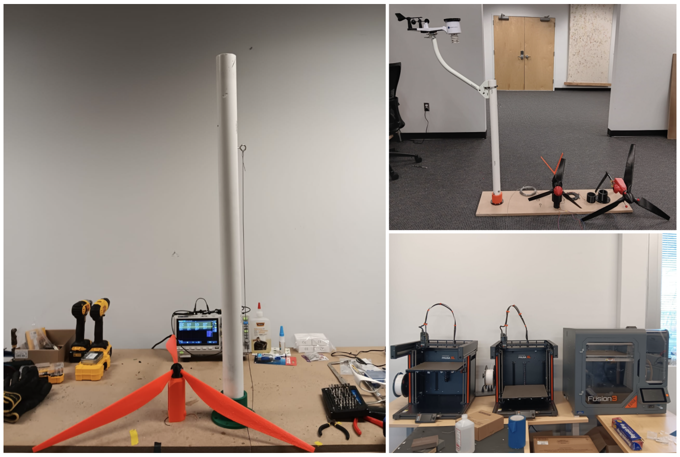


Fig. 6: 3D-printed wind turbines, wind speed and direction detectors, and the 3D printing setup used in the testbed.

IV. Implementation, Testbed Setup, and Evaluation

The proposed VindSec-Llama system testbed has been established at VMASC in Virginia, USA, in collaboration with the US Department of Energy, as shown in Figure 5. We have

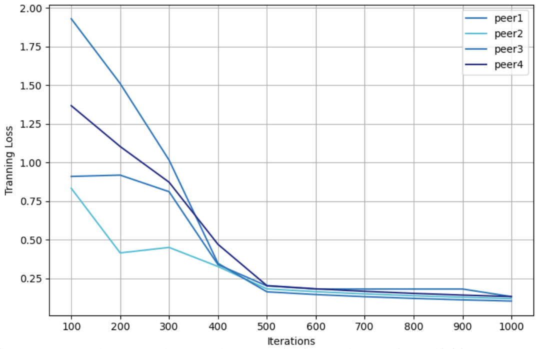


Fig. 7: Federated model training loss in different peers

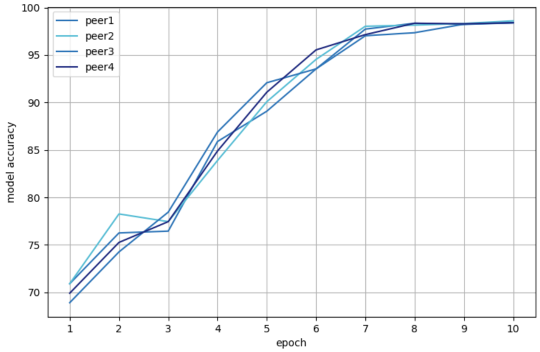


Fig. 8: Federated model accuracy in different peers

3D printed all elements of the wind turbines, including the tower, nacelle, rotor and hub, blades, wind speed/direction sensors and other components, as shown in Figure 6. The blockchain [6], [7] and other services are deployed on the infrastructure layer (e.g., cloud/on-prem servers). The Llama-3-8B model [5] has been used and trained for PBOM generation. The quantized Llama-3-8B LLM runs with Ollama on a consumer-grade server (without using a GPU). The LLM fine-tuning/training process was conducted through Qlora with a 4-bit quantized pre-trained language model using Low-Rank Adapters (LoRA) [19]. The platform's performance is evaluated in two key areas: LLM and blockchain-enabled federated learning.

In the evaluation of blockchain-enabled federated learning, we focused on assessing the accuracy and training loss of federated learning models, as well as the performance of the blockchain system. The federated learning process involved numerous iterations to refine the model's accuracy. In this evaluation, we trained the model over 1,000 iterations and plotted both the accuracy and the training loss. Figure 7 illustrates the variation in total training loss across different peers in each iteration. Figure 8 displays the accuracy of the federated machine learning model. Additionally, the block generation time, measured while increasing the number of peers up to seven, is shown in Figure 9.

In the LLM evaluation, we analyzed the responses from the LLM regarding its ability to generate server-hardening scripts and PBOMs. The Llama-3 LLM has been custom-trained to specialize in generating PBOMs and vulnerability-fixing Ansible/Puppet scripts. Prompts guide the custom-trained Llama-3 LLM in understanding the specific requirements and context of each service [25]. Figure 10 illustrates an example prompt used to instruct the LLM for generating a PBOM object. The prompt is designed to encapsulate the nuances of inputs (e.g., pull request information, SBOM information, vulnerability statuses, etc.), thereby enabling the LLM to generate targeted PBOMs. Figure 11 shows an example of a PBOM object generated by the LLM. These PBOMs are created as structured JSON schemas, encompassing critical details such as software update (e.g., pull request) information (creator, verifier, approver, timestamps), test results, identified vulnerabilities, and the status of the pull request. Figure 12 provides an example prompt used to instruct the LLM to generate server-hardening Ansible scripts based on the identified vulnerabilities. The Ansible playbook is then dynamically generated by considering the vulnerabilities identified within the system and adhering to the STIG compliance requirements, as shown in Figure 13.

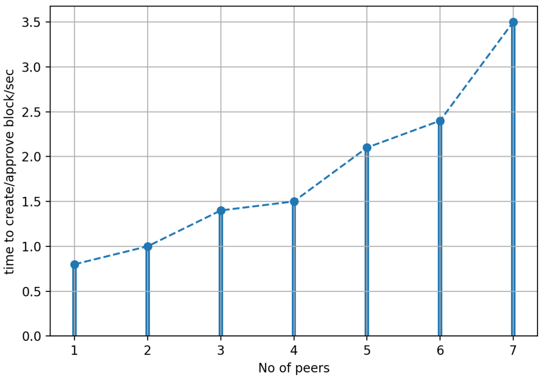


Fig. 9: Block creation time and # of blockchain peers

V. Related Work

Various researchers have attempted to enhance the security and privacy of cloud-native software services and data platforms. The key elements and architecture of these research initiatives are outlined in this section. Table I summarizes the contrast between these projects and the proposed platform.

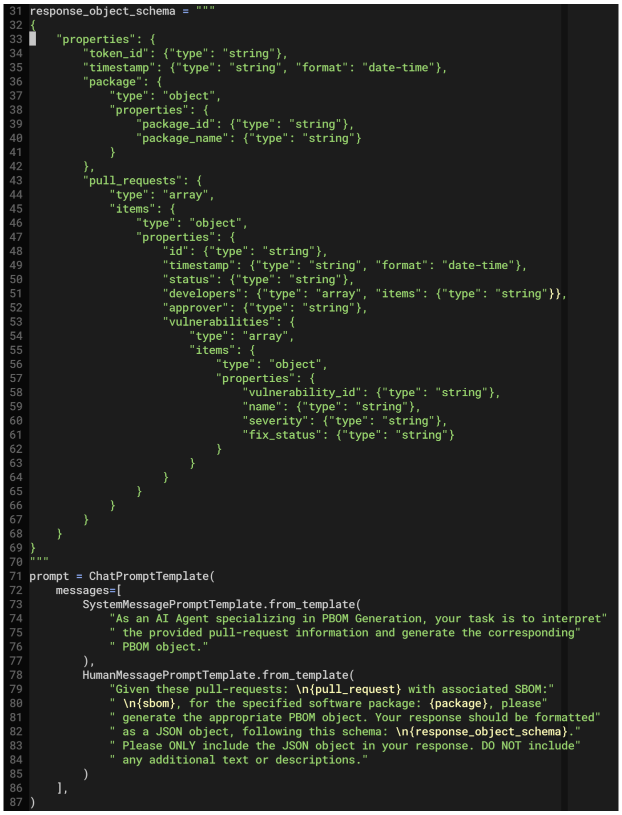
White et al. [14] introduce Continuous RMF, a novel approach utilizing blockchain technology to address delays in product release cycles and enhance security and functionality. Drljevic et al. [26], through Perspective RMF, explore the potential of blockchain technology to revolutionize business transactions by introducing a trust model based on algorithms. In the Let'strace system, Bandara et al. [24] present a blockchain-based cyber supply chain provenance platform that integrates TUF and In-ToTo frameworks, verifying software updates and enhancing supply chain security. In the Vind system, Bandara et al. [27], present a blockchain-based cyber supply chain provenance platform to address vulnerabilities in the Energy Delivery Systems supply chain. In SmartGrid-RMF [2], Aberibole et al. evaluate the impact of blockchain on decentralizing smart grids using the NIST conceptual model.

A. Conclusion and Future Works

This paper presents a data security architecture, "VindSec-Llama," incorporating Generative AI, blockchain, federated

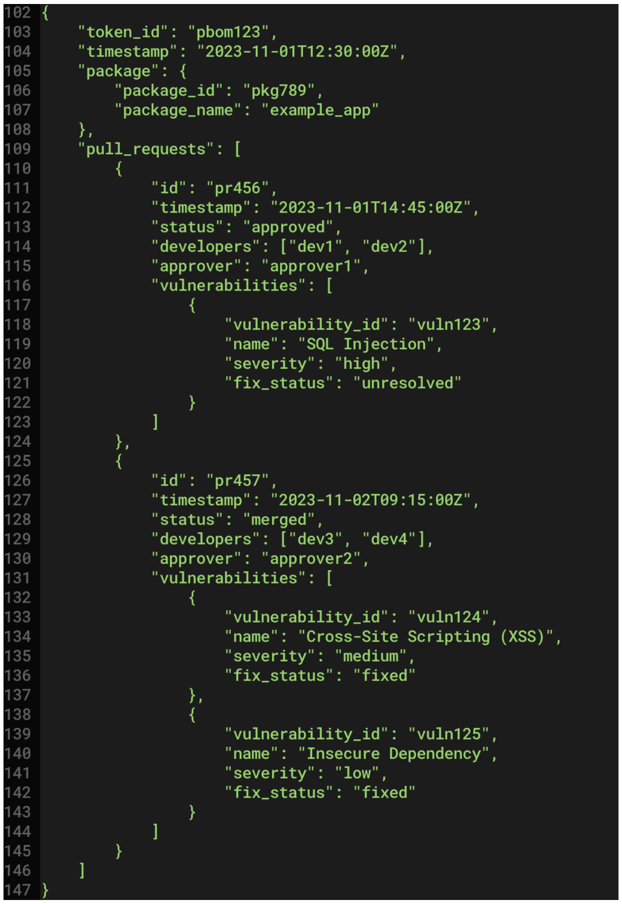
TABLE I: Cloud-native security framework comparison

Platform	Centralized/ Distributed	Blockchain Support	Supported RMF Frameworks	Data Provenance Support	NFT Support	Continuous ATO Support	AI Integration
VindSec-Llama	Distributed	✓	NIST, PCI-DSS	✓	✓	✓	✓
Contineous RMF [14]	Distributed	✓	NIST	✗	✗	✗	✗
Perspective RMF [26]	Distributed	✓	N/A	✗	✗	✗	✗
Letstrace [24]	Distributed	✓	N/A	✓	✗	✗	✗
Vind [27]	Distributed	✓	N/A	✓	✗	✗	✗
SmartGrid-RMF [2]	Distributed	✗	NIST	✓	✗	✓	✗



```
response_object_schema = """
{
    "properties": {
        "token_id": {"type": "string"},
        "timestamp": {"type": "string", "format": "date-time"},
        "package": {
            "type": "object",
            "properties": {
                "package_id": {"type": "string"},
                "package_name": {"type": "string"}
            }
        },
        "pull_requests": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "id": {"type": "string"},
                    "timestamp": {"type": "string", "format": "date-time"},
                    "status": {"type": "string"},
                    "developers": {"type": "array", "items": {"type": "string"}},
                    "approver": {"type": "string"},
                    "vulnerabilities": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "vulnerability_id": {"type": "string"},
                                "name": {"type": "string"},
                                "severity": {"type": "string"},
                                "fix_status": {"type": "string"}
                            }
                        }
                    }
                }
            }
        }
    }
}
"""
prompt = ChatPromptTemplate(
    messages=[
        SystemMessagePromptTemplate.from_template(
            "As an AI Agent specializing in PBOM Generation, your task is to interpret"
            " the provided pull-request information and generate the corresponding"
            " PBOM object."
        ),
        HumanMessagePromptTemplate.from_template(
            "Given these pull-requests: \n{pull_request} with associated SBOM:"
            " \n{sbom}, for the specified software package: {package}, please"
            " generate the appropriate PBOM object. Your response should be formatted"
            " as a JSON object, following this schema: \n{response_object_schema}."
            " Please ONLY include the JSON object in your response. DO NOT include"
            " any additional text or descriptions."
        )
    ],
)
```

Fig. 10: PBOM generation prompt.



```
{
    "token_id": "pbom123",
    "timestamp": "2023-11-01T12:30:00Z",
    "package": {
        "package_id": "pkg789",
        "package_name": "example_app"
    },
    "pull_requests": [
        {
            "id": "pr456",
            "timestamp": "2023-11-01T14:45:00Z",
            "status": "approved",
            "developers": ["dev1", "dev2"],
            "approver": "approver1",
            "vulnerabilities": [
                {
                    "vulnerability_id": "vuln123",
                    "name": "SQL Injection",
                    "severity": "high",
                    "fix_status": "unresolved"
                }
            ]
        },
        {
            "id": "pr457",
            "timestamp": "2023-11-02T09:15:00Z",
            "status": "merged",
            "developers": ["dev3", "dev4"],
            "approver": "approver2",
            "vulnerabilities": [
                {
                    "vulnerability_id": "vuln124",
                    "name": "Cross-Site Scripting (XSS)",
                    "severity": "medium",
                    "fix_status": "fixed"
                },
                {
                    "vulnerability_id": "vuln125",
                    "name": "Insecure Dependency",
                    "severity": "low",
                    "fix_status": "fixed"
                }
            ]
        }
    ]
}
```

Fig. 11: PBOM object generate by LLM.

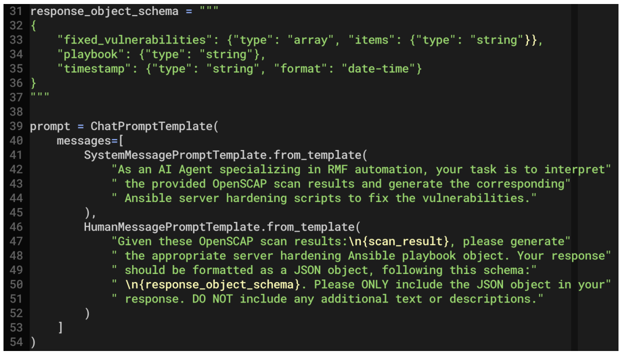
learning, and PBOM to address the prevailing security challenges in wind energy data management. The VindSec-Llama platform offers a robust solution that not only enhances data security and privacy but also streamlines operational efficiencies through automation and advanced data processing techniques. The introduction of a blockchain-enabled, coordinator-less federated learning system represents a significant leap forward in secure data sharing across distributed networks, crucial for the wind energy sector. Automation is a pivotal aspect of VindSec-Llama, with Meta's custom-trained Llama-3 LLM generating remediation scripts to address identified vulnerabilities in the server infrastructure. Data provenance information is captured as ModelCards and NFTs and stored in the underlying blockchain, ensuring transparency and traceability. The practical deployment of VindSec-Llama within a wind-energy testbed has demonstrated the platform's capacity to handle real-world data scales and complexities, affirming its effectiveness and adaptability. As the demand for renewable energy continues to grow, so does the need for secure and efficient data management platforms. VindSec-Llama stands as a testament to the potential of integrating advanced technological frameworks to address these needs, paving the way for more secure, efficient, and resilient energy infrastructures globally. For future work, we plan to incorporate the proposed data platform in a real wind-energy test environment in collaboration with Dominion Energy VA, USA.

Acknowledgements

This work is supported by the Department of Energy under Award Number DE-CR0000036.

References

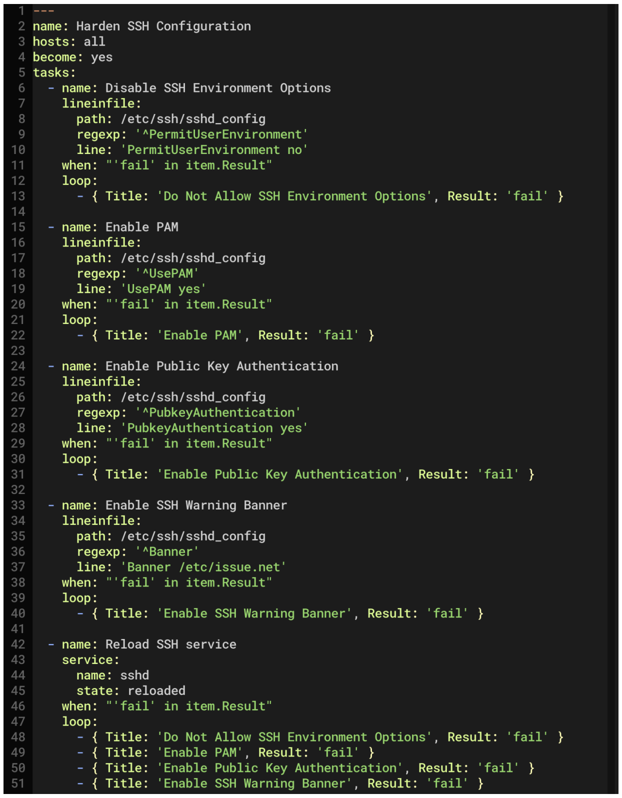
[1] J. Staggs, D. Ferlemann, and S. Shenoi, "Wind farm security: attack surface, targets, scenarios and mitigation," *International Journal of Critical Infrastructure Protection*, vol. 17, pp. 3–14, 2017.



```
response_object_schema = """
{
    "fixed_vulnerabilities": {"type": "array", "items": {"type": "string"}},
    "playbook": {"type": "string"},
    "timestamp": {"type": "string", "format": "date-time"}
}
"""

prompt = ChatPromptTemplate(
    messages=[
        SystemMessagePromptTemplate.from_template(
            "As an AI Agent specializing in RMF automation, your task is to interpret"
            " the provided OpenSCAP scan results and generate the corresponding"
            " Ansible server hardening scripts to fix the vulnerabilities."
        ),
        HumanMessagePromptTemplate.from_template(
            "Given these OpenSCAP scan results:\n{scan_result}, please generate"
            " the appropriate server hardening Ansible playbook object. Your response"
            " should be formatted as a JSON object, following this schema:"
            " \n{response_object_schema}. Please ONLY include the JSON object in your"
            " response. DO NOT include any additional text or descriptions."
        )
    ]
)
```

Fig. 12: Ansible playbook generation prompt.



```
---
name: Harden SSH Configuration
hosts: all
become: yes
tasks:
  - name: Disable SSH Environment Options
    lineinfile:
      path: /etc/ssh/sshd_config
      regexp: '^PermitUserEnvironment'
      line: 'PermitUserEnvironment no'
    when: "'fail' in item.Result"
    loop:
      - { Title: 'Do Not Allow SSH Environment Options', Result: 'fail' }

  - name: Enable PAM
    lineinfile:
      path: /etc/ssh/sshd_config
      regexp: '^UsePAM'
      line: 'UsePAM yes'
    when: "'fail' in item.Result"
    loop:
      - { Title: 'Enable PAM', Result: 'fail' }

  - name: Enable Public Key Authentication
    lineinfile:
      path: /etc/ssh/sshd_config
      regexp: '^PubkeyAuthentication'
      line: 'PubkeyAuthentication yes'
    when: "'fail' in item.Result"
    loop:
      - { Title: 'Enable Public Key Authentication', Result: 'fail' }

  - name: Enable SSH Warning Banner
    lineinfile:
      path: /etc/ssh/sshd_config
      regexp: '^Banner'
      line: 'Banner /etc/issue.net'
    when: "'fail' in item.Result"
    loop:
      - { Title: 'Enable SSH Warning Banner', Result: 'fail' }

  - name: Reload SSH service
    service:
      name: sshd
      state: reloaded
    when: "'fail' in item.Result"
    loop:
      - { Title: 'Do Not Allow SSH Environment Options', Result: 'fail' }
      - { Title: 'Enable PAM', Result: 'fail' }
      - { Title: 'Enable Public Key Authentication', Result: 'fail' }
      - { Title: 'Enable SSH Warning Banner', Result: 'fail' }
```

Fig. 13: Ansbile playbook generated by the LLM.

[2] A. Aderibole, A. Aljarwan, M. H. U. Rehman, H. H. Zeineldin, T. Mezher, K. Salah, E. Damiani, and D. Svetinovic, "Blockchain technology for smart grids: Decentralized nist conceptual model," *IEEE Access*, vol. 8, pp. 43 177–43 190, 2020.

[3] E. Sabev, R. Trifonov, G. Pavlova, and K. Rainova, "Cybersecurity analysis of wind farm scada systems," in *2021 International Conference on Information Technologies (InfoTech)*. IEEE, 2021, pp. 1–5.

[4] S. Arora, B. Yang, S. Eyuboglu, A. Narayan, A. Hojel, I. Trummer, and C. Ré, "Language models enable simple systems for generating structured views of heterogeneous data lakes," *arXiv preprint arXiv:2304.09433*, 2023.

[5] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[6] E. Bandara, X. Liang, P. Foytik, S. Shetty, N. Ranasinghe, and K. De Zoysa, "Rahasak-scalable blockchain architecture for enterprise applications," *Journal of Systems Architecture*, p. 102061, 2021.

[7] E. Bandara, D. Tosh, P. Foytik, S. Shetty, N. Ranasinghe, and K. De Zoysa, "Tikiri-towards a lightweight blockchain for iot," *Future Generation Computer Systems*, 2021.

[8] E. Bandara, X. Liang, S. Shetty, R. Mukkamala, A. Rahman, and N. W. Keong, "Skunk—a blockchain and zero trust security enabled federated learning platform for 5g/6g network slicing," in *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2022, pp. 109–117.

[9] V. Axelsson and F. Larsson, "Understanding the software bill of material for supply-chain management in open source projects," 2023.

[10] pbom.dev, "Pbom - pipeline bill of materials," https://pbom.dev/, 2024, accessed: 2024-04-21.

[11] J. A. Webb, M. W. Henderson, and M. L. Webb, "An open source approach to automating surveillance and compliance of automatic test systems," in *2019 IEEE AUTOTESTCON*. IEEE, 2019, pp. 1–8.

[12] A. Wadhwani and P. Jain, "Machine learning model cards transparency review: Using model card toolkit," in *2020 IEEE Pune Section International Conference (PuneCon)*. IEEE, 2020, pp. 133–137.

[13] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges," *arXiv preprint arXiv:2105.07447*, 2021.

[14] J. White and C. Daniels, "Continuous cybersecurity management through blockchain technology," in *2019 IEEE Technology & Engineering Management Conference (TEMSCON)*. IEEE, 2019, pp. 1–5.

[15] Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5171–5183, 2020.

[16] A. Decan, T. Mens, P. R. Mazrae, and M. Golzadeh, "On the use of github actions in software development repositories," in *2022 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2022, pp. 235–245.

[17] M. K. Abhishek, D. R. Rao, and K. Subrahmanyam, "Framework to deploy containers using kubernetes and ci/cd pipeline," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, 2022.

[18] V. Axelsson and F. Larsson, "Understanding the software bill of material for supply-chain management in open source projects," 2023.

[19] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[20] T. Reason, E. Benbow, J. Langham, A. Gimblett, S. L. Klijn, and B. Malcolm, "Artificial intelligence to automate network meta-analyses: Four case studies to evaluate the potential application of large language models," *PharmacoEconomics-Open*, pp. 1–16, 2024.

[21] D. Schumacher, "V3ctron— data retrieval & access system for flexible semantic search & retrieval of proprietary document collections using natural language queries." *Available at SSRN*, 2023.

[22] S. Ott, K. Hebenstreit, V. Liévin, C. E. Hother, M. Moradi, M. Mayrhauser, R. Praas, O. Winther, and M. Samwald, "Thoughtsource: A central hub for large language model reasoning data," *arXiv preprint arXiv:2301.11596*, 2023.

[23] E. Bandara, D. Tosh, S. Shetty, and B. Krishnappa, "Cyscpro-cyber supply chain provenance framework for risk management of energy delivery systems," in *2021 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2021, pp. 65–72.

[24] E. Bandara, S. Shetty, A. Rahman, and R. Mukkamala, "Let'strace—blockchain, federated learning and tuf/in-toto enabled cyber supply chain provenance platform," in *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*. IEEE, 2021, pp. 470–476.

[25] G. Marvin, N. Hellen, D. Jjingo, and J. Nakatumba-Nabende, "Prompt engineering in large language models," in *International Conference on Data Intelligence and Cognitive Informatics*. Springer, 2023, pp. 387–402.

[26] N. Drljevic, D. A. Aranda, and V. Stantchev, "Perspectives on risks and standards that affect the requirements engineering of blockchain technology," *Computer Standards & Interfaces*, vol. 69, p. 103409, 2020.

[27] E. Bandara, S. Shetty, D. Tosh, and X. Liang, "Vind: A blockchain-enabled supply chain provenance framework for energy delivery systems," *Frontiers in Blockchain*, p. 28, 2021.