International Journal of Multidisciplinary Engineering in Current ResearchISSN: 2456-4265, Volume 6, Issue 7, July 2021, http://ijmec.com/

# EXPLOITING AI AND SOURCE BLOCKCHAIN FRAMEWORK TO MITIGATE RISKS IN CLOUD MANUFACTURING IN INDUSTRY3.0

**Dr. Bhargav Gangadhara,**

[1]*Senior Technical lead/Director, Jack Henry and Associates, USA*

[1] *bhargavmtechmem@gmail.com*

**Abstract:** Cloud manufacturing is an evolving networked framework that enables multiple manufacturers to collaborate in providing a range of services, including design, development, production, and post-sales support. The framework operates on an integrated platform encompassing a range of Industry 4.0 technologies, such as Industrial Internet of Things (IIoT) devices, cloud computing, Internet communication, big data analytics, artificial intelligence, and blockchains. The connectivity of industrial equipment and robots to Internet opens the cloud manufacturing to the massive attack surface of cybersecurity and cybercrime threats caused by external and internal attackers. The impacts can be severe because physical infrastructure of industries is at stake. One potential method to deter such attacks involves utilizing blockchain andartificial intelligence to track the provenance of IIoT devices. This research explores a practical approach to achieve this goal by gathering provenance data associated with operational constraints defined in smart contracts and identifying deviations from these constraints through predictive auditing using artificial intelligence. A software architecture comprising IIoT communications to machine learning for comparing the latest data with the predictive auditing outcomes and logging appropriate risks was designed, developed, and tested. The state changes in the smart ledger of smart contracts were linked with the risks such that the blockchain peers can timely detect high deviations and take actions. The research defined the constraints related to physical boundaries and weight lifting limits allocated to three forklifts and showcased the mechanisms of detecting risks ofbreaking these constraints with the help of artificial intelligence. It also demonstrated state change rejections by blockchains at medium and high-risk levels. This research followed software development in Java 8 using JDK 8, CORDA blockchain framework, and Weka package for random forest machine learning. As a result of this, the model, along with its design and implementation, has the potential to enhance efficiency and productivity, foster greater trust and transparency in the manufacturing process, booster risk management, strengthen cybersecurity, and advance sustainability efforts.

**Keywords:** provenance; blockchain; smart contract; predictive auditing; cloud manufacturing risks; industrial internet of things.

## I. INTRODUCTION

Cloud manufacturing, as the name suggests, is a framework of operational planning, scheduling, monitoring, and control of manufacturing operations using hosted applications on the cloud computing [1–3]. Traditional manufacturing systems were controlled by programmable logic controllers (PLCs) operated by the local on-plant computers, which were capable of running manufacturing operations in limited physical spaces. The software systems used for materials planning, operations scheduling, monitoring, and control were also hosted within the data centers ofthe manufacturing plants. These systems were not connected to Internet as they were networked

using proprietary protocols and connections. Hence, the manufacturing operations were not exposed to cyber security threats in their traditional designs. For developing dynamic capabilities to respond to rapid demand and supply changes, manufacturers entered in strategic alliances to cover larger customer bases and meet the demands during normal times as well as during uncertainties and shocks [4–6]. In the collective operations, manufacturers needed to integrate their operations, which was possible by creating digital values using digitalization systems by Industry 4.0 technologies [7,8]. The digital value proposition could be achieved by making the PLCs operate with open communication protocols by transforming them into cyber-physical systems. The newly evolved Industrial Internet of Things (IIoTs) was used to transform the PLCs into cyber communication devices such that they could interface with the Internet and be controlled remotely. With this technological development and the already recognized problem of disconnected manufacturing silos and crunch of resources in computing, memory, and storage capacities of the data centers operated and maintained by manufacturing organizations, cloud manufacturing became a viable option for the future of manufacturing.

Interfacing PLCs with the Internet for controlling them through cloud manufacturing applications hosted on the cloud computing opened the gate for cyber threats to manufacturing organizations [9]. Cyber security threats are already prominent in manufacturing industry. About 75% of oil and gas industry have suffered at least one successful attack causing measurable business impacts by 2017. Power grids have suffered about 15% of the total number of cyber attacks in 2017. More recent statistics reported by Varonis and Forbes websites [10,11] reflect the ongoing trends of cyber attacks on manufacturing systems. Their reports stated that malicious power shell scripts targeted at cyber physical devices (detected and blocked) increased by 1000% to about 5200 monthly average attacks in 2021 and 2022 [10,11]. Normally, protection against remote code execution tactics is robust but rogue IIoT devices installed by insiders can cause a major loophole especially by using malicious and non-transparent algorithms [13,14]. The more worrying trend is about insiders creating deliberate loopholes in the cyber physical systems of manufacturing plants thus opening an attack surface for external exploits to penetrate and use the compromised cyber physical systems as launchpads [12,15–17]. The activity is reported to be about 30% of the overall number of attacks [10,11]. The extent to which unsolicited IIoT devices can be sneaked into manufacturing networks has not been estimated yet. However, 30% of the 5200 cyber attacks on IIoT devices in 2021 and 2022 were carried out through insider activity, which reports a significant trend that is expected to increase [10,11]. In order to address these challenges, cyber security threats need to be visualized with a different perspective as compared with those threats in self-hosted manufacturing and supply chain computerized control systems [17].

This research presents design, prototyping, and testing of controls employing Artificial Intelligence (AI) and Provenance Blockchain framework for protecting organizations using cloud manufacturing applications against the cyber security risks. As these organizations are having their PLCs transformed to cyber physical systems, they should be certified and accounted at the time of inception and during their operations. As reviewed in the literature review, provenance is the dynamic metadata of systems and devices that captures their "data about their manipulation history” (including change of ownership and assignment to various roles). As further reviewed in literature review, blockchains can be used to form trusted networks of partners operating their assets in supply chains and logistics for serving common demands and orders. It is hereby emphasized that if the provenance system can be deployed on such blockchains to capture the “real-time operational data about the manipulation history” of cyber physical systems used in logistics and supply chain operations for cloud manufacturers, it can help in mitigating cyber security risks to them by conducting AI-enabled predictive auditing. The research questions of this study are the following:

1. What are the risks associated with cloud manufacturing in Industry 4.0?
2. How can provenance blockchain be used to provide greater transparency and traceability in thecloud manufacturing process using AI-enabled predictive auditing?
3. How can this system help in mitigating cloud manufacturing risks in Industry 4.0?

In this research, the first research question is answered through literature review, the second research question is answered through designing, developing, and testing a software prototype, and the third research question is answered through a critical analysis of the software prototype keeping in context the findings in the answer to the first research question. The next section presents a review of literature.

## II. LITERATURE REVIEW

The modern era for the manufacturing sector is highly competitive, dynamic, and complex withuncertainties beyond the controls of individual companies [4,5]. To compete, survive, and flourish in this environment, manufacturing organizations need to develop “dynamic capabilities” to manage rapid changes as per the demand and competitive dynamics of their target markets [4–6]. Building dynamic capabilities require strategic alliances among multiple manufacturing organizations and use of modern technology to develop incremental improvements and rapid adjustments of manufacturing resources, processes, knowledge, and skills through management controls. The strategic alliances can be executed by creating a joint cloud manufacturing portal of applications that can monitor and control the manufacturing processes of the plants of the collaborating companies inthe strategic alliance [1–3]. The Industry 4.0 technologies and processes are viewed as the foundation for developing dynamic capabilities for cloud manufacturing [3,4,7,8]. Industry 4.0 technologies and processes have influence on digitalization, digital value creation, real time knowledge of markets and demands, quick production and marketing, ability to use and reuse materials and resources optimally, and sustainable development [4].

As introduced in the introduction section, cloud manufacturing comprises PLCs transformed ascyber physical systems running the manufacturing controls of several plants collaborating through the cloud-hosted applications for serving the demand dynamics. As the cyber physical systems are interfaced with the Internet, they are prone to cyber threats. Some of the known cyber security risks to cloud manufacturing systems are the following [12,15–19]:

(a) Eavesdropping attack: an attack mechanism in which, the communications from authorized devices to others like them are captured in between by eavesdropping devices (called listeners);

(b) Masquerading attack by capturing packets of unsecured IIoTs: an attack mechanism in which an unauthorized cyber physical system captures packets from unsecured

IIoTs and masqueradesas an authorized controller to the cloud hosted manufacturing applications;

(c) Distributed Denial of Service (DDoS): an attack mechanism in which, massive scale storms of packets are bombarded to the cyber physical systems through unprotected Internet connections compromised by the attackers thus overloading the computing systems, networking links, and cyber physical controllers;

(d) Side channel attacks: these are penetration attacks through the side channels into the manufacturing network, which are less monitored or ignored by the monitoring systems;

(e) Cross-side scripting attacks: these are Trojan scripts that can be mixed with the running scripts through SQL injection;

(f) Automated code-based attacks (such as Bots): these are penetration attacks caused by pre- programmed automatic codes;

(g) Exploit-based attacks: these are attacks orchestrated through open-source programs created by hacking experts;

(h) Identity thefts (of authorized IIoT devices): these are caused by eavesdropping attacks to capture authentication and authorization details of IIoT devices;

(i) Insider trading and proliferation: insiders engaged in malicious activities such as injecting Trojan codes in running programs or opening firewall ports for external attackers to launch theirexploits;

(j) Fake sensor data feeding and actuation attacks in control systems: these are well planned targeted attacks. For example, the attacker may send false signals of lowering valve pressure in a key pipeline thus causing the control system to gradually increase the pressure in the pipelineleading to an explosion;

The above list provides the answer to the first research question. The cloud manufacturing partners using blockchain technology for smart contracts to execute logistics and supply chain operations can integrate provenance data of IIoT devices with the blockchain [20–22]. Blockchains comprise of nodes integrated in the form of a chain such that contracts signed for logistics and supply chain operations can be stored in them in the form of encrypted blocks identified and integrated through hash functions. Provenance of computing devices and software systems is a separate database describing the characteristics, ownership, operational modes, and several such details about those computing devices [23]. Provenance on cloud computing can help in running forensic analysis of past events if recorded separately in addition to the data generated by the events [24]. This can ensure transparency, data fidelity and protection, privacy issues of the data collected, quality control, and intellectual property protection [25]. In cloud manufacturing, the IIoTs and the cyber physical systems enabled by them can be tracked closely using their provenance data [26–28]. The cyber physical systems can be compromise by hackers in several ways. Some known concerns are the following [15–19]:

(a) Validating the identity of cyber physical system enabled with IIoT communications;

(b) Tracking rapid deployments and Internet-enabling of millions of cyber physical systems;

(c) Traceability of cyber physical systems added, modified, and removed; especially installed on mobile assets;

(d) Validating fidelity of sensor data sent for influencing process events interpreted out of thesensory data and the decision-making algorithms running the actuation commands;

(e) Establishing accountability and liability of individuals owning the cyber physical systems;

(f) Inter-cloud assurances of cyber security;

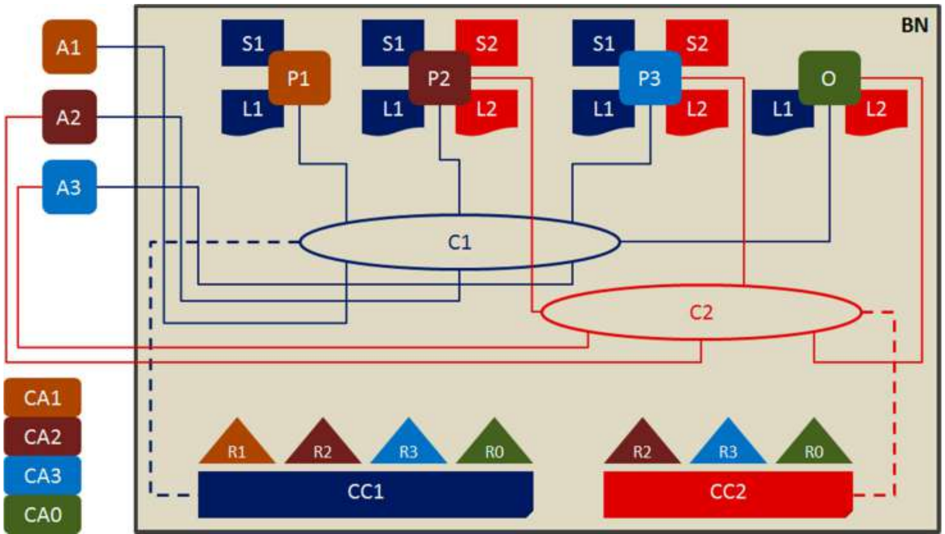
(g) Algorithmic transparency (accountability of performance and behaviours of algorithmsdeployed for controlling operations of cyber physical systems);

(h) Cyber physical systems indulging into erroneous or malicious processing thus affecting theexecution of smart contracts negatively;

The above six concerns form the problem for which, the prototype solution is designed and tested in this research. When manufacturers integrate their cyber physical systems driving their manufacturing processes for cloud manufacturing, they can also integrate their Enterprise Resource Planning (ERP) systems through blockchains [29]. Modern ERPs have interfacing to popular blockchain frameworks such as Hyperledger and Ethereum. In this manner, the IIoTs and their enabled cyber physical systems are also hooked to the blockchain [30,31]. The data collected from them can be stored on big data systems to monitor their events through continuous auditing. By using artificial intelligence, the predictions can be carried out such that the actual events versus predicted events can be compared [32]. To understand how this can work, a design scenario of a blockchain- controlled manufacturing network for creating, executing, and monitoring smart contracts using Hyperledger Fabric (based on explanation in Reference 33) is presented in Figure 1. This design can help in visualizing where the provenance data streams and artificial intelligence can be positioned inthis blockchain.

**Figure 1.** A design scenario of a blockchain-controlled manufacturing network for creating, executing, and monitoring smart contracts using Hyperledger Fabric (drawn based on the detailed explanationprovided in Reference 33).

In the scenario shown in Figure 1, four cloud manufacturing organizations R0, R1, R2, and R3 decide to setup a manufacturing blockchain network BN for signing, sharing, and monitoring smart logistics and supply chain contracts. R0 is the contracting authority and others are contracting vendors. They agree to setup to two network channels C1 and C2 governed by policies-based network configurations CC1 and CC2, respectively. C1 is shared by R0, R1, R2, and R3, and C2 is shared by R0, R2, and R3. Thus, R1 has no business relation with C2 and thus is denied access to it. The A1, A2, and A3 are cloud applications deployed by R1, R2, and R3 to interact with the network through the authorized channels. To interact with the network, R1, R2, and R3 need to authorize peers P1, P2, and



P3 representing them. R0 authorizes O for managing orders to R1, R2, and R3 through the peers P1, P2, and P3. The peers P1, P2, and P3 are authorized to access the block chain network BN using cryptographic keys issued by certification authorities CA1, CA2, and CA3, respectively. The ordering authority manages C1 and C2 network channels to interact with P1, P2, and P3. P1 has access to C1 only whereas P2 and P3 have access to both C1 and C2. When orders are placed, the smart contracts are signed digitally using signatures issued by CA1, CA2, and CA3 to P1, P2, and P3 for the contracting vendors, and the digital signature issued by CA0 to O for the contracting authority. Further to digitally signing the contracts, CA1, CA2, and CA3 issue X.509 certificates to the components identified as belonging to the organizations R1, R2, and R3, respectively. The certification authorities are also used to sign on transactions to affirm their approvals. On signing, the smart contracts are stored in the smart ledgers L1 and L2 belonging to the network segments C1 and C2, respectively. P1 stores a copy of L1 only (as it, and its company R1 do not have any business connection with C2), whereas P2 and P3 store a copy each of L1 and L2. The events related to the smart contracts L1 and L2 (such as approved logs of works completed as per the contractual terms) are stored in state databases S1 and S2. P1 maintains a copy of S1, and P2 and P3 maintain a copy of S2 and S3 (as per their access rights). All copies of state databases are synchronized. The ordering authority O need not maintain a copy of the state databases because organization R0 is not contributing to state changes. However, O can inspect S1 and S2 at will.

The above design scenario represents a vanilla blockchain network for creating, executing, and monitoring smart contracts and the events linked with their closure. This research added provenance capturing in real time and predictive analytics using artificial intelligence in which, random forest

algorithm was used. The blockchain framework selected for this research was CORDA [34,35], which is lightweight and can be installed, programmed, and executed in a personal laptop having i5 or i7 processor and 16GB of RAM. The blockchain contract rules were configured in the form of predictive auditing such that the updates are accepted only when the risks predicted by AI are within the prescribed limit. The methodology for conducting the primary research is presented in the next section.

## III. METHODOLOGY

This research is an original conceptualisation of solution to the concerns related to employment of cyber physical systems in cloud manufacturing networks, identified by the references [15] to [19].This research was designed to learn by experiencing a conceptualised design of provenance in CORDA blockchain framework available with open source codes. The knowledge was developed through experiencing the coding process and running tests by simulating scenarios of provenance data anomalies using simulated production data collection in logistics processes. The random forest machine learning algorithm was coded in such a way that it could predict numerical values of operating parameters and detect risk levels based on boundary parameters. The risk levels were appended to the event records confirming work-in-progress and completions as per the terms of the smart contracts. Thus, the operations manager monitoring the event records can view the risk levels and investigate the specific IIoT devices in question. As the events data may be collected from a group of IIoT devices, the entire group may require investigation till a rogue IIoT device is located by the investigators. As this is an original conceptualisation to solve the research problem, it required learning through several rounds of trials and experiences till the final results obtained are satisfactory. Keeping this approach in mind, the philosophy selected is pragmatic and the data collection and analysis shall be both qualitative and quantitative [36–38]. The learning was both inductive and deductive. The knowledge gained from literature sources and technical documentationof Hyperledger and CORDA frameworks were qualitative. The knowledge gained from the machinelearning actions when inbound data is manipulated was quantitative. The accuracy analysis of machine learning was quantitative, as well. Inductive research was required to learn the design and operations of blockchains from the technical documentation of Hyperledger and CORDA frameworks, and programming techniques from the CORDA manuals. Deductive research was required to test the design idea of the researcher by appending the provenance control based on anomaly detection by machine learning.

The state databases S1 and S2 are the main ERP-linked systems getting regular updates on events completed as per the smart contracts stored in smart ledgers L1 and L2. State changes in the S1 and S2 are facilitated by the ERP applications A1, A2, and A3 on behalf of organisations R1, R2, and R3 serving R0 through their respective smart contracts. Hence, the responsibility and accountability of accuracy and integrity of events data being fed to S1 and S2 state databases inside the manufacturingblockchains are with administrators of A1, A2, and A3, which are positioned as ERP data systems outside the blockchain. In this research, A1, A2, and A3 are the focal points for building strong provenance security controls. The provenance in this research is not merely static metadata or occasional changes to it, albeit comprises operational and allocation data and rules. In this research, three forklifts are assigned to different physical zones and are assigned to carry different ranges of weights. The location and weight data streams are considered as dynamic provenance feeds.

The proposed modification makes the network as blockchain network with provenance (BNP). The applications A1, A2, and A3 have Message Queuing Telemetry Transport (MQTT) interfaces on which, they receive provenance data from IIoT devices attached to the three forklifts. The data from the IIoT devices were used to change the states of state databases inside the blockchain by the blockchain peers P1, P2, and P3. The machine learning (ML) was implemented to make predictions of risk levels by comparing the latest data arrived with their predicted values. The provenance data was stored by the ML in a database called ProvDB. The algorithm planned for machine learning was random forest. The machine learning shall be trained using historical data in the ProvDB database. To begin a credible learning cycle about 500 records are planned to begin with, in the ProvDB

database. The initial records were considered as IIoT inputs from the three forklifts identified as Asset01, Asset02, and Asset03. The random forest was tasked to make asset-wise independent predictions. Hence, it was coded to first export the asset-wise data in separate files and then makes separate and independent predictions about their next state values. The risk levels were calculated by comparing the next state predicted values with the current state values received. The risks were defined as per the physical boundaries within which, the assets were allowed to operate and the loading limit on each asset. The physical boundaries of movements of Asset01 (A01) were: X = 1 to 200, Y = 1 to 200, and Z = 1 to 200 feet, and weight = 100 KG. The physical boundaries of movements of Asset02 (A02) were: X = 201 to 400 feet, Y = 201 to 400 feet, and Z = 201 to 400 feet, and weight = 125 KG. The physical boundaries of movements of Asset03 (A03) were: X = 401 to 600 feet, Y = 401 to600 feet, and Z = 401 to 600 feet, and weight = 150 KG. In real world, it can be a multistorey warehouse in which, reach truck forklift machines have been assigned to fixed boundaries. If they breach these boundaries, they will enter in the zones of other forklifts and cause accidents. There can also be issues of wrong forklifts assigned for jobs not suitable for them (such as under capacities of weight lifting). The extent of breach defines the level of risk. For example, if a forklift has breached only the boundaryof another forklift, the risk level will be logged as LOW but if a forklift breaches up to the centre of the zone of another forklift, the risk level will be logged as HIGH. Four risk levels were assigned: NONE, LOW, MEDIUM, and HIGH.

The state data was entered along with the predictive risks estimation made by the machine learning (ML) by the blockchain peers. Once authorised because of risk levels within acceptable limit, the IIoT devices were trusted to provide genuine events updates from the running processes, which can be used for changing the states in the state databases of the blockchain. However, if the risks arenot in the acceptable range, the blockchain state changes would not occur and the peers will be suggested to conduct investigations. The CORDA rules in the blockchain were programmed as the following:

For all assets:

X should be less than or equal to 600; Y should be less than or equal to 600; Z should be less than or equal to 600;

Weight should be less than or equal to 150; Risk level should be either NONE or LOW;

To simulate IIoT data transfers, a Message Queuing Telemetry Transport (MQTT) server called Apache ActiveMQ was used. The data in the ProvDB was sent from the Apache ActiveMQ in the form of topic publisher data sent by a publisher file coded as Publisher.Java, which will have records matching the database structure of the ProvDB database. The first column comprised of device keysused for registration and the remaining columns constituted the numerical data collected from the sensors in the running processes. The topic publisher data sent to a subscribed listener code called Listener.java represented the numerical data collected from the sensors. For the purpose of this research, the topic publisher data was constructed manually and sent through the Apache ActiveMQ MQTT Broker server. In real industrial applications, the Publisher.Java shall be embedded as a firmware deployed in physical IIoT devices such that the topic publisher data will be constructed automatically by the industrial sensors integrated in the IIoT devices. For this research, the values arechanged manually in the code itself to test different risk levels. The smart contract monitoring can be done with two quality objectives: right forklifts should be assigned to right zones and right weight loading capacities, and the forklifts should not breach their operating boundaries and enter zones ofother forklifts (unless reallocated operationally).

In this research, deploying real IIoT devices in a laboratory environment was avoided because the study is about detecting anomalies in the data collected from them and not about the engineering of the IIoT devices. The provenance data needs to be streamed to the big data systems through highly secured and encrypted channels with appropriate key exchanges, as stated in the studied by References [38] to [42]. It should be noted that streaming data from IIoTs may not be possible throughencrypted links from the devices. IIoT devices are low capacity low-end systems. Implementing

cryptography at the level of cyber physical systems may not be feasible. Hence, chances of breaches are possible. Provenance data validation is needed in Industry 4.0. This is the value additionproposed and tested in this research.

The topic publisher data constructed manually comprised of a set of data values tagged to a process at periodic intervals. At every transmission, the values were varied slightly as is expected in a stable industrial process (like, up to 10 percent). Intermittently, larger variations were also injected in the data. The machine learning was programmed to learn from the ongoing data streams and predict the next combination of data. A decision rule using Random Forest algorithm was programmed to compare the predicted versus actual arrival of the next combination of the data. The risks were logged in the form of alerts about four variables: X-axis movement, Y-axis movement, Z- axis movement, and weight lifted. There were boundaries assigned to the four variables. At no breach, the risks were marked as NONE, at one breach the risks were logged as LOW, at two breachesrisks were logged as MEDIUM, and at three and above breaches, risks were logged as HIGH. The risks were logged in a log file but not be passed on to the blockchain immediately at their occurrences. Their information was passed on to the state databases of the blockchain only when ten consecutive risk detection events of at least medium level have been logged by the rules engine. The log in the blockchain was not intended to be taking any automated action but to inform the peers P1, P2, and P3 to begin investigation about specific devices identified as changing their behaviours. The machinelearning predictive algorithm was coded using Weka package of JDK 8 and the rules engine were coded using core Java 8 coding.

The primary research environment and the tests conducted are reported in the next section. The primary research followed the design of Figure 2 and the scenario explained in this section.

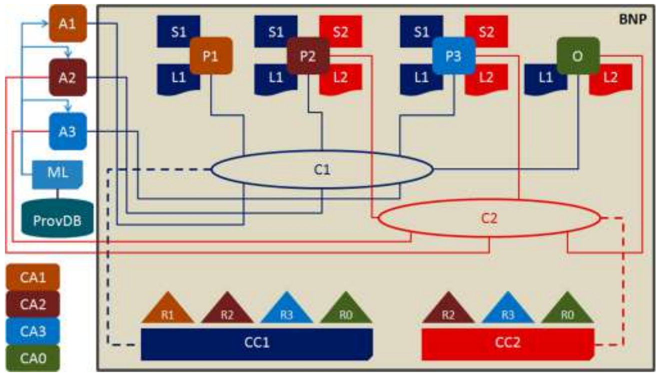


**Figure 2.** Addition of Provenance and Machine Learning to the architecture shown in Figure 1 (Author's own efforts).

## 2. Primary Research

The primary research was conducted by building the software architecture within a laptop environment running Ubuntu 22.04 operating system (a popular distribution of Linux). The software architecture and runtime flow used for the research project is shown in Figure 3.

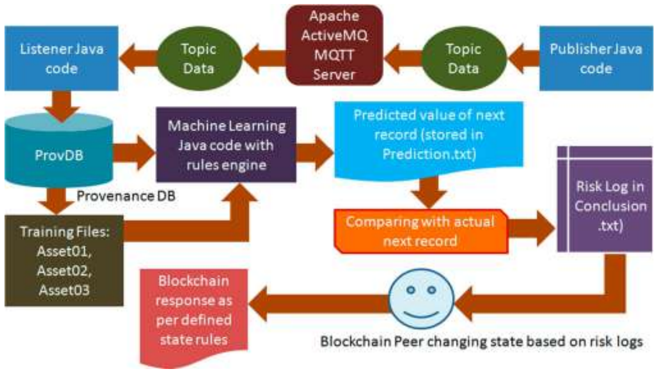

Listener Java code
Topic Data
Apache ActiveMQ MQTT Server
Topic Data
Publisher Java code
ProvDB
Provenance DB
Machine Learning Java code with rules engine
Predicted value of next record (stored in Prediction.txt)
Risk Log in Conclusion .txt)
Training Files: Asset01, Asset02, Asset03
Comparing with actual next record
Blockchain response as per defined state rules
Blockchain Peer changing state based on risk logs

**Figure 3.** Software architecture used for the primary research (Author's own efforts).

Six main components were created and interfaced in the software architecture:

(a) MQTT Broker Server using Apache ActiveMQ: The package ActiveMQ-5.15.0 was used to setup a MQTT broker server in Ubuntu 22.04. This package was selected because of its compatibility with Java 8 version.

(b) Publisher Java code: programmed using Java 8; Java 8 was selected because CORDA for open source development is available with packages compatible with up to Java 8 version only. Professional versions of CORDA are available on higher versions of Java. As stated in previoussection, this code should be embedded in the firmware of the IIoT devices. However, for the purpose of testing this research executed the code several times through Ubuntu 22.04 terminal by manipulating the input data at every event.

(c) Listener Java code: programmed using Java 8: this file is responsible to receive the data transmitted through the MQTT connections and save them into a ProvDB file to be read by the machine learning algorithm.

(d) Provenance database in the ARFF format: The file was created by the Listener java code in ProvDB.csv;

(e) Machine Learning code in Java with and in-built rules engine called WEKA: The package selected was weka-3.7.0 because it is compatible with Java 8; the rules or risk assessment and logging were defined as detailed in the previous section; the machine learning code picks up the latest data from the ProvDB and then compares it with the latest next state predictions using 80% records for training and 20% for testing. Based on the predictions the machine learning populates the respective state files of Assets A01, A02, and A03 along with the latest risk values. The verbose report to be analysed by the blockchain peers is entered in conclusion.txt file in append mode (new records added to the older records).

(f) CORDA blockchain framework with state rules defined in Java 8: To create smart contract state rules in CORDA, six Java files were configured: IOUState.java, IOUContract.java, IOUSchemaV1.java, ContractTests.java, ExampleFlow.java, and FlowTests.java. IOU is the name of smart contract tested in this experiment. While configuring these files, a database file named "iou.changelog-v1.xml" is configured automatically. This is the change log database comprisingstate changes of the smart contract named IOU. All the variables created in Schema and other files such as States and Flow, should have an existing record in the change log database.

The tests were run several times following the steps as stated below. These steps are the main steps comprising of several technical sub-steps for each of them.

- Step 1: Run the ActiveMQ console;
- Step 2: Run the Listener.java file; this opens the MQTT connections in the ActiveMQ console;
- Step 3: Run the Publisher.java and transmit data for location coordinates and weight carried outby a Forklift;
- Step 4: Run the machinelearning.java file;

The outputs will be generated in ProvDB (Provenance database with the latest record of assetplain.txt appended, asset01.arff, asset02.arff, asset03.arff, output.txt, prediction.txt, and finally, Conclusion.txt;

- Step 5: Open the CORDA console through IntelliJ Idea;
- Step 6: Try entering the latest values received in assetplain.txt, and analysing the risk log in conclusion.txt and entering the appropriate risk value (updates sent to the smart ledger of the smart contract);
- Step 7: Observe the responses from the CORDA smart ledger and write the full report by analysing it and all the files generated by the machine learning code;

The above steps were run for several combinations of input data values about the position and weight carried by three Forklifts identified as Asset01, Asset02, and Asset03 in the blockchain database. The results are discussed in the next section.

## IV. DISCUSSION

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

The results of all the tests conducted revealed two main states:

(a) the machine learning algorithm decides that risk is either at NONE or at LOW level such that the state change in the CORDA smart contract is allowed;

(b) the machine learning algorithm decides that risk is either at MEDIUM or at HIGH level such that the state change in the CORDA smart contract is prohibited instructing the blockchain peers to conduct investigations;

100 tests were conducted by varying the values following a structured approach. The programming of data was about a scenario in which three reach truck forklifts (a type of forklifts suitable for high rise warehousing for vertical storage) are allocated to three different operating zones in a warehouse. All the three zones have dimensions of 200 X 200 X 200 feet. They are touching each other but are not interconnected. They were called Zone1, Zone2, and Zone3 in the testing. Zone1 is on the ground, Zone2 is located on a landfill about 200 feet high and Zone3 is located on an adjacent landfill about 400 feet high. A simple schematic created in Blender 3D software is presented in Figure4:

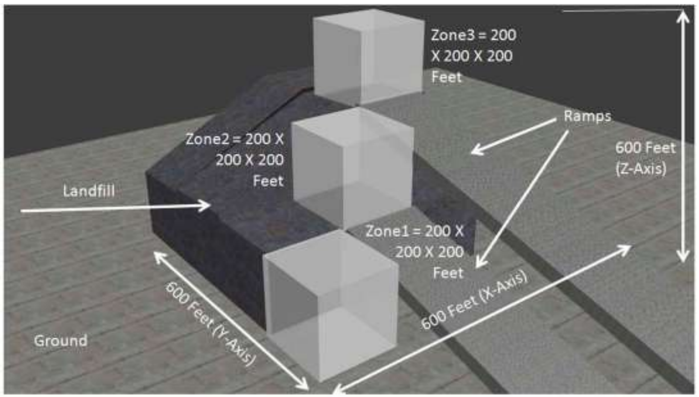


**Figure 4.** Physical layout of the warehouse and the three zones programmed for running the testing steps (author's own design).

As discussed in the Section 3, the three reach truck forklifts, named as Asset01, Asset02, and Asset03 were allocated to Zone1, Zone2, and Zone3, respectively. The physical boundaries ofmovements of the three assets were:

- Asset01 (A01): X = 1 to 200, Y = 1 to 200, and Z = 1 to 200 feet, and weight = 100 KG;
- Asset02 (A02): X = 201 to 400 feet, Y = 201 to 400 feet, and Z = 201 to 400 feet, and weight = 125KG;
- Asset03 (A02): X = 401 to 600 feet, Y = 401 to 600 feet, and Z = 401 to 600 feet, and weight = 150KG;

It may be observed that the forklifts can be shifted between zones only when they are taken out because Zone2 and Zone3 are accessible through ramps. Hence, breach of boundaries is possible only through planned allocations. The forklifts cannot breach their boundaries on their own. This is the reason this scenario was designed (that is, having no inadvertent breaches unless human actors are involved). The risk levels were defined based on how serious was the breach of these constraints. If a forklift is found to be breaching X or Y, the risk may be LOW because it might have been taken out from the warehouse in a parking place just to give it some rest to cool it down or for some maintenance and repairs. However, if there is a breach in Z-axis constraints (which will be along with breaches in X and Y axes because the forklifts have to be taken out and shifted through ramps), the risk level logged will be MEDIUM to HIGH depending upon how far they have being taken away. The breach in weight levels along with location breach shall generate HIGH risk logging only.

The 100 tests conducted were by entering location data with incremental changes, and later even the permissible weights were also breached. Initially, the forklifts were kept within their zones withno breach in weight as well. The risk logs were found as NONE. Thereafter, the forklifts were breached only at X and Y axes by imagining their locations outside their zones but not entering other zones. For example, the forklifts were positioned at various locations on the ground outside Zone1 and also on the landfill or the ramps outside Zone2 and Zone3. The risks were found to be logged as LOW. Thereafter, forklifts were entered into other zones and the risk levels were logged as MEDIUM to HIGH depending upon how far they were taken. Finally, when the weights were also breached, the risk levels of HIGH only were logged.

In this research, the machine learning code was written in such a way that at every data set received, it conducts prediction of next state values and then compares them with the data received to log the risks. However, in reality there would never be an abrupt jump to risk levels MEDIUM or HIGH. The transition will be gradual as the assets are moving along their respective paths. This is the reason the tests also were conducted by following paths defined in a test scenario. When enteringdata into blockchain, the blockchain peers should not jump to conclusions. They should follow the trends carefully to see if there are real risks. The use of predictive analytics by AI helps in reducing unnecessary false positives. For example, if the reallocations have happened quite a few times in thepast, they will reflect in the predicted values. Thus, the differences between the predicted and actual values will vary significantly only when sudden outliers are caused in the data streams. If a forklift has never been taken to another zone but has been taken out for cooling down or repairs several times in the past, the predictive values will detect breach only when the Z-value crosses its normal operating range.

The system was found to abide by the rules and produce either of the above two states (a) and

(b) without failing on even one of the tests. There were no false positive and false negative risks identified during the tests. However, the actual decision on true positives should be with the blockchain peers as they will compare the risk values with other data available, such as reallocations recorded in the ERP. Normally, the blockchain peer may come across false positives most of the time and hence enter NONE and LOW risks in the smart contract state updating. When they come acrossreal MEDIUM and HIGH risks, they anyways cannot enter them in the smart contract as it will refuseto change the states. In such instances, the blockchain peers will be left with no option but to investigate the risks.

At this stage, the third research question is answered by explaining how this system can solve the concerns raised by References [15] to [19] stated in the Literature Review. This research addresses the concerns to some extent justified as the following:

(a) *Validating the identity of cyber physical systems enabled with IIoT communications:* This concern is clearly addressed in this research as the identity of the cyber physical system is registered in theMQTT broker server, in the ProvDB of the machine learning, in the training and testing database of machine learning, and in the smart contract of the blockchain.

(b) *Tracking rapid deployments and Internet-enabling of millions of cyber physical systems*: With the system of smart contracts and smart ledgers in place, even millions of cyber physical systems shall be registered and prepared for tracking and tracing before assignment to smart contracts. However, the IT capacities of the cloud manufacturing and the network bandwidth should be sufficient to handle volumes of data generated by millions of cyber physical systems in real time.

(c) *Traceability of cyber physical systems added, modified, and removed; especially installed on mobile assets*: The machine learning shall continuously track the operational state changes of the cyberphysical systems and log risks accordingly for the blockchain peers monitoring the system. Major changes like addition, modification, and removal cannot go unnoticed by the machine learning. There may be false alarms because of communication interruptions (like data received in a topic stops temporarily) but the traceability and tracking will always be ON.

(d) *Validating fidelity of sensor data sent for influencing process events interpreted out of the sensory data and the decision-making algorithms running the actuation commands*: This is a tough challenge. The validity of fidelity of sensor data can only be made by decision-making algorithms comprising of engineering knowledge integrated. This system can, however, report inconsistencies in the state changes through its predictive capability and log risks. This may be of some help to the engineers operating the cyber physical systems.

(e) *Establishing accountability and liability of individuals owning the cyber physical systems*: This concern will be addressed by the system designed. At the time of registration of the assets, the ownership and accountability details will be recorded in the blockchain. The smart contracting parties will be fully liable for the assets registered and allocated to the contract. Further, blockchain peers from all contracting parties will monitor the operations thus ensuring timely detection of risks logged by the machine learning.

(a) *Inter-cloud assurances of cyber security*: Inter-cloud assurance is possible in this solution if the same MQTT broker and machine learning systems are implemented for all the contracting parties interfacing through multi-cloud blockchain. If multiple brokers and machine learning systems need to be implemented in a multi-site environment then replication of data between the Conclusion.txt files implemented in multiple clouds should be implemented.

(b) *Algorithmic transparency (accountability of performance and behaviours of algorithms deployed for controlling operations of cyber physical systems):* This is another tough challenge to be addressed. Performance and behaviours of algorithms require much deeper monitoring and control by sophisticated systems having full knowledge about the operating behaviours and performance metrics of the algorithms. This system can help by detecting changes in the already progressingpatterns and reporting them as risks at different levels depending upon the rules defined in the machine learning code and the blockchain state rules.

(c) *Cyber physical systems indulging into erroneous or malicious processing using exploits, scripting attacks, bots, device identity theft, and other means thus affecting the execution of smart contracts negatively*: Detection of exploits, scripting attacks, bots, etc. need to be enabled through intrusion detectionand prevention systems. This solution can detect operational anomaly caused by such malicious software attacks through machine learning but cannot detect presence of the software. Any anomaly causing negative execution of smart contract will be detected through machine learning and traced to the device using provenance data. Negative execution will breach the blockchain state transition rules and hence transactions will be rejected promoting for investigations. Attempts of device identity theft will also be difficult for the attackers because three levels of registration in MQTT broker, machine learning ProvDB database, and the blockchain smart contract will cause deterrence for the attack planners. There is high chance that the attackers willnot be able to plan a perfect breach of this entire system, although they should never be underestimated.

The challenges of provenance verification system to identify the IIoT devices accurately and build traceability of doubtful devices in the network, and the challenges of provenance detection of bindings, fault tolerance, integrity and confidentiality verifications through data, chain, and origin integrity verifications, access controls, and protection of keys during sharing may be solved to some extent following the solution of continuous operational risks monitoring in this research. Once devices are registered in the blockchain, they will be treated as trustworthy in the system tested in this research. However, this will not be a permanent perception built about the devices even if they follow all the routines and key exchanges for valid registration. Devices may be subject to investigation if their operational boundaries are breached and risks from medium onwards are logged because the blockchain state change will not be allowed. The next state prediction will alwaysbe based on the historical gradual state changes and hence any drastic variations will be detected promptly. Further, the predictability can also cover chances of devices breaching their operational boundaries in due course of their operations. As the blockchain peers are monitoring the records periodically and updating state changes in the blockchain, they will have opportunities to detect such probabilities well in advance and correct the course of operations of the devices to mitigate such risks proactively. There may be chances of some false positives as the devices may have been reallocated deliberately through mutual agreements among the blockchain contracting parties. However, reallocations should always be done through new smart contracts such that the MQTT broker server and the machine learning rules could be updated.

IIoT sensing streams used as provenance data validated by artificial intelligence for making state changes in smart contracts stored in blockchains can have several business benefits. The smart contract state rules can be defined to enforce any policies on the whole cloud manufacturing network.In future, this model and its design and coding may be useful not only for cyber security risk mitigation but also for increase in efficiency and productivity, increase in trust and transparency in the manufacturing process, and promoting sustainability. For example, if the contract demands emission levels from logistics and transportation equipment to be below defined thresholds, the IIoT data sensed and the machine learning driven risk levels thus logged can be useful in accepting or rejecting state changes in the smart contracts putting compliance pressure.

## V. CONCLUSIONS

This research was conceptualized with three research questions replicated as the following:

1. What are the risks associated with cloud manufacturing in Industry 4.0?
2. How can provenance blockchain be used to provide greater transparency and traceability in thecloud manufacturing process using AI-enabled predictive auditing?

3. How can this system help in mitigating cloud manufacturing risks in Industry 4.0?

In response to the first research question, the cyber security threats to cloud manufacturing were listed based on review of literature. Sophisticated threats like code injections, side channel attacks, covert channels, exploits, malware, and DDoS may be mitigated on cloud computing because of high end security controls. However, cloud manufacturing is dependent upon the data streams from IIoT devices attached with the process event sensors in the plant machinery, robots, and logistics equipment. They may not be protected from the sophisticated threats because of low computing and storage power requiring use of low end thin operating systems (such as Lubuntu, which is a light weight version of Ubuntu). If IIoT devices are compromised especially by insiders, they can be used as launch pads for attacking the manufacturing infrastructure. The concerns related to IIoT devices were identified separately, which were related to cyber security and beyond related to operational reliability, trust, and quality assurance. Their behavioral trends need to be monitored to find out if they are compromised making them rogue devices. A promising solution evolving in scientific research is using provenance blockchain employing predictive capabilities of AI. This concept was adopted in this research as the second research question. The second research question was answered by studying the provenance, blockchain, and AI solutions for cloud manufacturing as separate themes, which were combined in a design realized within a Ubuntu laptop environment. A scenario was imagined in which, three reach truck forklifts allocated to three separate zones in a warehouse having constraints in the form of physical operating boundaries and weights. AI was programmed to detect breaches to the constraints and logging the risks using random forest predictive analysis and an in-built rules engine within the coding. Thereafter, a smart contract system was programmedin CORDA framework including the risks in the state change rules for tracking the events conducted to fulfill the smart contract's requirements. The transparency and traceability were ensured by making the event logs, predictive AI results, and the risk logs transparent to all blockchain peers andthe customer.

With this system in place, the third research question was framed to explore how it can help in mitigating the cloud manufacturing risks. To answer this question, a number of tests were conductedto deeply experience the behavior of this system. The layout of the warehouse imagined for this research was drawn in Blender 3D software and presented for visualizing the risky scenarios. Thereafter, the possible risky scenarios were discussed. As this system operates in real time, risk logs happen at each data transmission event and data comparison between AI predictions and actual. Using AI predictions shall reduce the chances of false positives as the risk levels will increase gradually by following the paths of the vehicles on their way to breaching the boundaries. However, the blockchain peers need to correlate the risk values with all other data available in the ERP systems. If they detect deliberate human actions logged in the system officially, then they can safely assume the risks to be in control and update the events in the blockchain smart contracts. However, if the human actions are found to be not declared in ERP officially, then they can delay updating the state changes and first investigate the reasons. It was visualized that such real time monitoring of risks can help in risk mitigation to any IIoT related risks, such as quality risks, reliability risks, sustainability risks, efficiency risks, productivity risks, and any other area of concern of the engineers. The right kind of IIoTs and sensors need to be selected, and the Java rules defined for risk assessment needs to be customized as per the variables being monitoring. The random forest algorithm will simply predict new numbers based on its training and testing data, and the rules engine shall detect and publish the associated risk levels and their related actions.

## References

[1] Fu, J. A practical resource-searching method for manufacturing grid. Int. J. Adv. Manuf. Technol. 2014, 74, 335–340. [CrossRef]

[2] Tasgetiren, M.F.; Pan, Q.K. A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. Inf. Sci. 2011, 181, 3459–3475. [CrossRef]

[3] Li, B.H.; Zhang, L.; Wang, S.L.; Tao, F.; Cao, J.W.; Jiang, X.D.; Song, X.; Chai, X.D. Cloud manufacturing: A new service-oriented networked manufacturing model. Comput. Integr. Manuf. Syst. 2010, 16, 1–7. (In Chinese) [CrossRef]

[4] Tao, F.; Li, C.; Liao, T.W.; Laili, Y. BGM-BLA: A New Algorithm for Dynamic Migration of Virtual Machines in Cloud Computing. IEEE Trans. Serv. Comput. 2016, 9, 910–925. [CrossRef]

[5] Cao, Y.; Wu, Z.; Liu, T.; Gao, Z.; Yang, J. Multivariate process capability evaluation of cloud manufacturing resource based on intuitionistic fuzzy set. Int. J. Adv. Manuf. Technol. 2016, 84, 227–237. [CrossRef]

[6] Zhou, J.; Wang, M. Cloud Manufacturing Service Paradigm for Group Manufacturing Companies. Adv. Mech. Eng. 2014, 6, 740725. [CrossRef].

[7] Dr. Bhargav Gangadhara, "Optimize Utility in Computing-Based Manufacturing Systems Using Service Models and Development Models", Presented and published at International Journal of Computer Science and Information Security, IJCSIS Editorial Board, Vol -14, Page 5 , 2016.

[8] Zhang, L.; Luo, Y.L.; Tao, F.; Ren, L.; Guo, H. Key technologies for the construction of manufacturing cloud. Comput. Integr. Manuf. Syst. 2010, 16, 2510–2520. (In Chinese) [CrossRef]

[9] Hu, Y.J.; Chang, X.F.; Wang, Y.; Wang, Z.L.; Shi, C.; Wu, L.Z. Cloud manufacturing resources fuzzy classification based on genetic simulated annealing algorithm. Mater. Manuf. Process. 2017, 32, 1109–1115. [CrossRef]

[10] Dr. Bhargav Gangadhara, "Optimize utility in cloud-Based manufacturing systems using service models and development models", Presented and published at International Journal of Innovative Research in Advanced Engineering.IJIRAE Editorial Board, August 11 - 12, 2016.