The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 10, October 2025
PP 40-54

Enhancing Supply Chain Security with Block chain and AI-based Anomaly Detection

N.V.Ashok Kumar[1]*, Bantu Vamsi[2], Laisetti Leesa[3] ,Tamarana Jagadish[4]
Barnikala Prasad[5] ,Geddam Roopa[6]
[1]Associate Professor and HOD, Dept of CSE-AI & ML ,
[2,3,4,5,6]CSE-AI & ML, Dept of CSE-AI & ML , Avanthi Institute Of Engineering & Technology, Makavarapalem-531113, Andhra Pradesh-India
Corresponding Author *: nagamashok@gmail.com,
vamsibantu1@gmail.com , roopageddam123@gmail.com , jagadish2714@gmail.com ,
laisettileesasri@gmail.com , barnikalaprasad7@gmail.com

Abstract: In modern supply chain management, ensuring data integrity, security, and real-time anomaly detection is crucial for maintaining trust and efficiency. This paper introduces a hybrid model that integrates Block chain technology with AI-powered anomaly detection to enhance supply chain security. The block chain module guarantees immutable transaction records by utilizing proof-of-work and cryptographic hashing techniques in a decentralized ledger. Each transaction is securely linked to the previous block, creating a tamper-proof and transparent chain of records. Users can add transactions, retrieve data via hash references, and audit the entire blockchain through a user-friendly interface. In parallel, an anomaly detection system employs the Isolation Forest algorithm to identify irregularities in supply chain data. Users can upload transaction datasets (e.g. CSV files), enabling the machine learning model to flag unusual patterns that may indicate fraud or operational inefficiencies. A web-based Streamlit interface ties these components together, providing real-time insights into both the blockchain ledger and any detected anomalies. The combined approach helps businesses to mitigate risks, prevent data tampering, and improve decision-making through enhanced transparency and proactive fraud detection.

Keywords: Supply Chain Management; Data Integrity; Anomaly Detection; Block chain Technology; Isolation Forest; Fraud Prevention; Stream lit

Introduction

Supply chain management (SCM) plays a pivotal role in global commerce by coordinating the efficient flow of goods, services, and information among manufacturers, suppliers, distributors, and retailers. Traditional SCM systems, however, face significant challenges such as a lack of transparency, vulnerability to data tampering, and difficulties in detecting fraudulent activities or anomalies in operations. These limitations can result in financial losses, inefficiencies, and damage to a company's reputation. Ensuring data integrity and timely detection of anomalies in supply chain transactions has therefore become crucial for maintaining stakeholder trust and the smooth functioning of the supply chain.Blockchain technology has emerged as a promising solution to address transparency and data integrity issues in SCM. Blockchain's decentralized and immutable ledger ensures that once a transaction is recorded, it cannot be altered retroactively, thereby preventing fraudulent modifications. All parties in a block chain network have access to a synchronized copy of the ledger, fostering trust through transparency. This tamper-proof record-keeping is especially valuable in supply chains where multiple stakeholders may not fully trust each other or where regulatory compliance and audit trails are important. By utilizing cryptographic hashes and consensus mechanisms (such as proof-of-work), block chain can create an immutable chain of transaction blocks, making it extremely difficult for any malicious actor to falsify records or hide unauthorized activities.While block chain secures the integrity and traceability of data, it does not inherently detect whether a recorded transaction is legitimate or an outlier indicating potential fraud or error. This is where anomaly detection using artificial intelligence comes into play. Machine learning algorithms, specifically unsupervised anomaly detection methods, can analyze patterns in supply chain transaction data to identify unusual behavior (e.g., irregular order quantities, atypical shipping routes, or abnormal payment patterns). By flagging these anomalies in real-time, companies can investigate and address issues such as fraud, operational errors, or inefficiencies before they escalate.In this paper, we propose a hybrid approach that combines blockchain technology for tamper-proof data storage with an AI-powered anomaly detection system for real-time fraud and anomaly identification. We implement the anomaly detection using the Isolation Forest algorithm, which is well-suited for identifying rare and irregular data points in large datasets. A Streamlit-based user interface is developed to provide users with seamless access to both the blockchain ledger and the anomaly detection results. Through this integrated dashboard, users can add and view transactions on the blockchain and simultaneously monitor any flagged anomalies in the transaction data. By uniting these technologies, the system enhances supply chain security by both preventing data manipulation (via block chain) and proactively detecting suspicious events (via machine learning).Organization of the Paper: The remainder of this paper is organized as follows. In Section 2 (Literature Survey), we review related work and existing solutions in block chain-based supply chain management and anomaly detection techniques. Section 3 (Methodology) describes the architecture of the proposed system and how the components interact. Section 4 (Algorithm) details the blockchain and anomaly detection algorithms, accompanied by a flow diagram illustrating the process. Section 5 (Graph and Comparisons) presents an experimental evaluation with a performance graph and a comparison of the anomaly detection results with alternative methods. Section 6 (Summary of Test Results) provides a summary of the system's test results and findings. Section 7 (Conclusion) concludes the paper by highlighting key contributions and potential future enhancements. Finally, Section 8 lists the references used in developing this work.

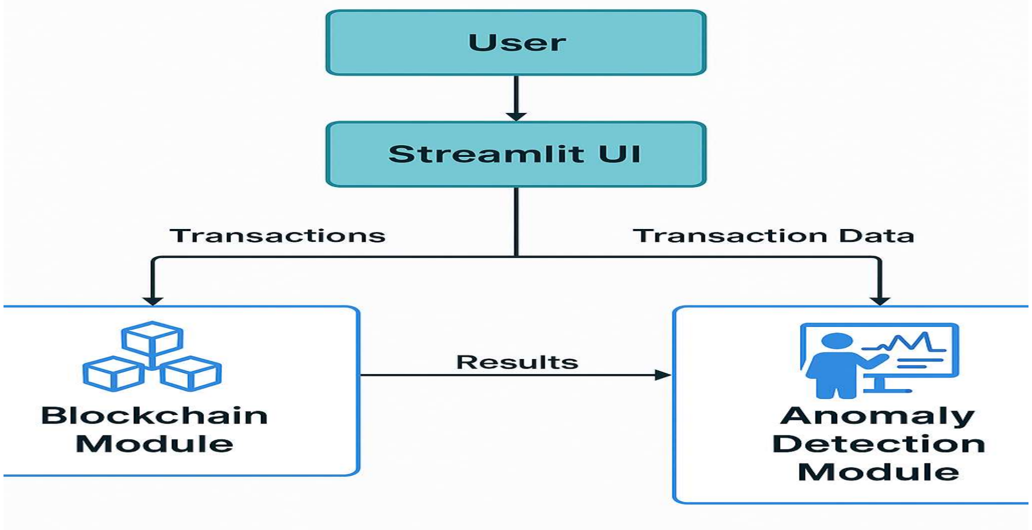
Literature Survey

Early research in blockchain technology demonstrated its potential to create secure, distributed ledgers for digital transactions (e.g., the original Bitcoin blockchain in 2008 introduced by Satoshi Nakamoto). Building on that foundation, numerous studies have explored applying blockchain to supply chain management to improve transparency and trust. Blockchain in Supply Chain Management (SCM) has been the subject of many recent studies focusing on traceability, provenance tracking, and efficiency improvements. For instance, researchers have shown that blockchain's immutable ledger can greatly enhance the ability to trace products from origin to destination, thereby improving accountability in sectors like food supply chains and pharmaceuticals. By recording each handoff of a product on a blockchain, stakeholders can verify authenticity and compliance with regulations (such as temperature control or ethical sourcing), addressing issues of counterfeit goods and ensuring quality control. Several proof-of-concept implementations (e.g., IBM Food Trust and other industry consortia) have validated that blockchain can reduce data discrepancies and delays in information flow within complex supply networks.Despite these advantages, literature also notes challenges in adopting blockchain for SCM. Scalability is a concern: as the number of transactions grows, maintaining a distributed ledger can become computationally intensive and may affect transaction throughput. Privacy is another issue; while transparency is beneficial, companies may be reluctant to share certain sensitive data openly on a ledger visible to all participants. Various solutions have been proposed, such as permissioned blockchains (which restrict access to authorized participants) and techniques to store only hashes of data on-chain while keeping details off-chain to preserve confidentiality. Moreover, integrating blockchain with legacy systems requires significant changes in business processes and technology infrastructure, which has been a barrier to widespread adoption.Another active area of research is the integration of blockchain with other emerging technologies to bolster supply chain effectiveness. Combining blockchain with Internet of Things (IoT) devices enables automated, real-time data recording (for example, sensors logging temperature or location data of shipments directly to a blockchain). Recent studies also explore synergy between blockchain and AI (Artificial Intelligence) for supply chain optimization. Machine learning can extract insights from the rich data that blockchain systems gather. For example, predictive analytics can forecast demand or identify optimal shipping routes using trustworthy data from a blockchain. However, the literature suggests that the full potential of blockchain and AI integration in supply chains is yet to be realized, and implementations are still in early stages or pilot papers. Our work contributes to this area by integrating blockchain with an AI-based anomaly detection system, focusing specifically on fraud detection and operational anomaly identification in supply chain transactions.Anomaly detection in supply chain data is a critical aspect of fraud prevention and quality control. Traditionally, companies have relied on rule-based systems or manual audits to spot irregular transactions — for instance, an unusually large order from a new client or a sudden spike in expedited shipping requests might trigger investigation. These approaches, however, can be labor-intensive and may fail to catch subtler patterns of anomalies. In recent years, machine learning techniques have been increasingly applied to automate anomaly detection. Unsupervised learning algorithms like clustering, Local Outlier Factor (LOF), and autoencoders have been used to identify outliers without the need for labeled examples of fraud. Among these methods, the Isolation Forest algorithm has gained popularity due to its efficiency and effectiveness in high-dimensional data. Isolation Forest operates by randomly partitioning data and is particularly adept at isolating anomalies (which require fewer random partitions to isolate than normal points). In supply chain contexts, Isolation Forest and similar algorithms have been used to detect anomalies such as fraudulent transactions, irregular sensor readings in logistics (indicating equipment malfunction or theft), or unusual customer ordering behaviors that could signal fraud or error.

Some contemporary research efforts have begun merging blockchain and anomaly detection. For example, in domains like food supply chains, frameworks have been proposed where blockchain ensures data provenance and an integrated anomaly detection model checks the data for any signs of contamination events or falsified records. Chen et al. (2023) designed a system combining machine learning-based anomaly detection with blockchain to protect food supply chains, illustrating the effectiveness of such a hybrid approach. These studies report that using blockchain for secure data sharing among partners, along with AI for data analysis, can significantly improve response times to incidents and reduce the likelihood of undetected fraud. However, many of these efforts are domain-specific, and there is room to generalize the approach to broader supply chain scenarios. Our proposed model builds on this literature by offering a general-purpose integration of blockchain and anomaly detection, with an emphasis on a user-friendly interface and real-time analysis which could be applied to various industries in the supply chain domain.In summary, the literature underscores the importance of both technologies: blockchain provides the trust infrastructure needed for reliable data in supply chains, and anomaly detection algorithms provide the intelligence to interpret that data and catch exceptions. The convergence of these technologies represents a promising direction for next-generation supply chain systems, enabling secure, transparent, and smart supply chains. This paper's contribution lies in demonstrating such a convergence through a working model that others can build upon for enhancing data integrity and security in SCM.

Methodology

The proposed system architecture consists of two primary components working in tandem: (1) a Blockchain Module for secure, immutable transaction record-keeping, and (2) an Anomaly Detection Module driven by machine learning to analyze transaction data for irregular patterns. These components are connected via a front-end interface built with Streamlit, which allows users to interact with both the blockchain and anomaly detection features in real-time. Figure 1 (in the next section) illustrates the overall flow and interactions between the user, the system modules, and the data.



Blockchain Module: The blockchain component is responsible for recording supply chain transactions in a tamper-proof ledger. We implement a simplified private blockchain using Python. Each block in the chain contains a set of transaction data (for example, details like transaction ID, timestamp, parties involved, product details, quantity, etc.), the cryptographic hash of the previous block, a proof-of-work nonce, and its own hash. The use of the previous block's hash links the blocks together, forming an immutable chain — any alteration in an earlier block would change its hash and break the link, thus alerting the system to tampering. To add a new block (transaction) to the blockchain, a proof-of-work consensus mechanism is utilized: the system must find a nonce value that, when hashed with the block's contents, produces a hash meeting certain difficulty criteria (e.g., a hash with a specified number of leading zeros). This proof-of-work requirement thwarts adversaries from easily changing historical records since they would have to recompute proofs for all subsequent blocks, which becomes computationally infeasible. In our implementation, the difficulty level is adjusted to ensure that block addition is fast enough for a private network (for usability in the demo) yet still demonstrates the principle of work required to secure the chain. Users interact with this module by submitting new transactions via the UI, upon which the system packages the data into a new block, computes the proof-of-work, and appends the block to the chain. The Streamlit UI also allows users to browse the blockchain, displaying the sequence of blocks and their contents (including hashes), or retrieve specific transaction records by their hash references. This design guarantees data integrity: once recorded on the blockchain, a transaction cannot be deleted or altered without detection.Anomaly Detection Module: Parallel to recording data on the blockchain, the system performs anomaly detection on the supply chain transactions using an Isolation Forest algorithm. This module is activated when a user uploads a dataset of transactions (for instance, a CSV file containing historical transaction records or streaming transaction data). The methodology for anomaly detection is as follows: the uploaded data is first preprocessed (if necessary) to ensure it is in the correct format and to handle missing values or outliers that are clearly erroneous. Then, an Isolation Forest model is either trained on the data (if no prior model exists) or updated/applied if a model has been pre-trained on historical data. The Isolation Forest works by constructing an ensemble of binary decision trees; it randomly partitions the data along feature values. Anomalies, being data points that are few and different, are more likely to be isolated (i.e., separated from other points) in fewer splits. The model assigns an anomaly score to each transaction: a score close to 1.0 indicates a highly likely anomaly, whereas a score near 0.5 or lower would indicate a normal observation (with the exact threshold for "anomaly" chosen based on desired sensitivity). Transactions flagged as anomalous by the model are then highlighted to the user through the interface. For example, if a particular transaction has an unusually large value or involves a supplier that does not typically handle a certain product; it might receive a high anomaly score and be marked for review. The anomaly detection module helps in fraud prevention and operational efficiency by catching these irregularities. Importantly, this analysis occurs in real-time or near real-time; as soon as the data is uploaded or a new batch of transactions is processed, the model outputs any anomalies so that the business can be alerted immediately to potential issues.Integration via Streamlit UI: The use of Streamlit for the user interface ensures that the system is accessible through a web browser with an intuitive layout. The UI is designed to have two main views: one for the blockchain ledger and one for anomaly detection results. On the blockchain view, users can input new transaction details into a form and submit it to add a block. They can also scroll through a visual representation of the chain, where each block's key information (index, timestamp, previous hash, current hash, etc.) is displayed.

On the anomaly detection view, the user can upload a file containing transaction records. Once uploaded, the system automatically runs the Isolation Forest analysis in the background. The UI then presents a summary of findings, e.g., "Out of 500 transactions, 5 were flagged as anomalies." It may also visualize the anomalies (for instance, plotting their positions if numerical data can be papered, or simply listing them with their details and anomaly scores). Additionally, the UI links the two modules: if an anomaly is detected in a transaction that is also on the blockchain, the user can cross-reference that transaction's block (using the transaction ID or hash) to examine its context in the immutable record. This dual capability adds an extra layer of verification—should an anomaly be detected, the blockchain can confirm all details of that anomalous transaction (time, participants, etc.) to assist in investigating why it was flagged.In summary, our methodology emphasizes a secure-by-design approach (through blockchain) combined with intelligent monitoring (through AI anomaly detection). The next section will detail the algorithmic steps of each component and present a flow diagram of the integrated process, illustrating how a transaction progresses from being added to the blockchain to being analyzed by the anomaly detection system.

Algorithm

Figure 1: Proposed system workflow integrating the blockchain ledger and anomaly detection module. The user interacts via a Streamlit UI to add transactions to the blockchain and to upload data for anomaly analysis. New transactions are hashed and added as blocks linked to the previous block's hash, ensuring an immutable ledger. Simultaneously, uploaded transaction data is processed by the Isolation Forest model, which flags any anomalous transactions for the user to review.

The algorithm driving our integrated system can be divided into two parts: the blockchain process and the anomaly detection process. Below, we outline the step-by-step procedures for each part of the system:

Block chain Algorithm:

1. Initialization: Create a new block chain with a genesis block. The genesis block (index 0) is a hardcoded start of the chain with predefined values (e.g., previous hash set to 0 or a constant, and an initial arbitrary transaction or message). Set the difficulty level for proof-of-work (for example, require the hash to have a certain number of leading zeroes).
2. Adding a Transaction: When a user wants to add a new transaction, they provide the transaction details via the UI. The system gathers the current blockchain's last block hash (the previous hash) and forms a new block containing: (a) a block index (incremented by 1 from the last block), (b) a timestamp, (c) the transaction data (e.g., order details,

participants, etc.), (d) the previous block’s hash, and (e) a nonce (which will be determined through mining).

3. Proof-of-Work Mining: Compute a cryptographic hash of the new block’s contents (excluding the nonce, which we vary) and increment the nonce until the hash meets the difficulty criteria (e.g., the hash starts with 0000). This is the mining process. In a production public blockchain, this step is performed by many miners and ensures consensus, but in our private setting, the system itself performs this computation to emulate the security mechanism. Once a valid nonce is found that produces a hash with the required pattern, the block’s hash is finalized.
4. Linking the Block: Assign the newly computed hash to the block and append the block to the block chain . The previous hash field of this new block links it to the chain, and the proof-of-work makes it computationally impractical for an attacker to alter the block without breaking the chain’s integrity.
5. User Notification and Storage: Update the user interface to inform the user that the transaction has been added successfully. The UI can display the block’s details including its hash (which serves as a unique identifier). The transaction is now securely recorded; any attempt to modify it would change the block’s hash and thus be easily detectable by verifying the chain. The entire updated blockchain is stored in memory (and optionally persisted to disk) for future retrieval and verification requests.
6. Data Retrieval (Optional): If a user wants to retrieve or verify a transaction, they can query by the transaction hash or block index. The system then traverses the block chain to find the matching block and returns the stored data, allowing verification that the details match what was originally recorded. This ensures transparency and audit ability of the supply chain transactions.

Anomaly Detection Algorithm (Isolation Forest):

1. Data Input: When the user uploads a dataset of transactions (e.g., a CSV file containing fields such as transaction ID, date, supplier, product, quantity, price, etc.), the system reads the dataset into a pandas Data Frame (in our implementation). Basic data validation is performed to ensure the format is correct and that there are no obviously corrupt records (e.g., negative quantities or malformed dates).
2. Preprocessing: Prepare the data for anomaly detection. This may include scaling numerical features, encoding categorical variables (such as converting product categories or supplier IDs into numeric form via one-hot encoding or label encoding), and handling missing

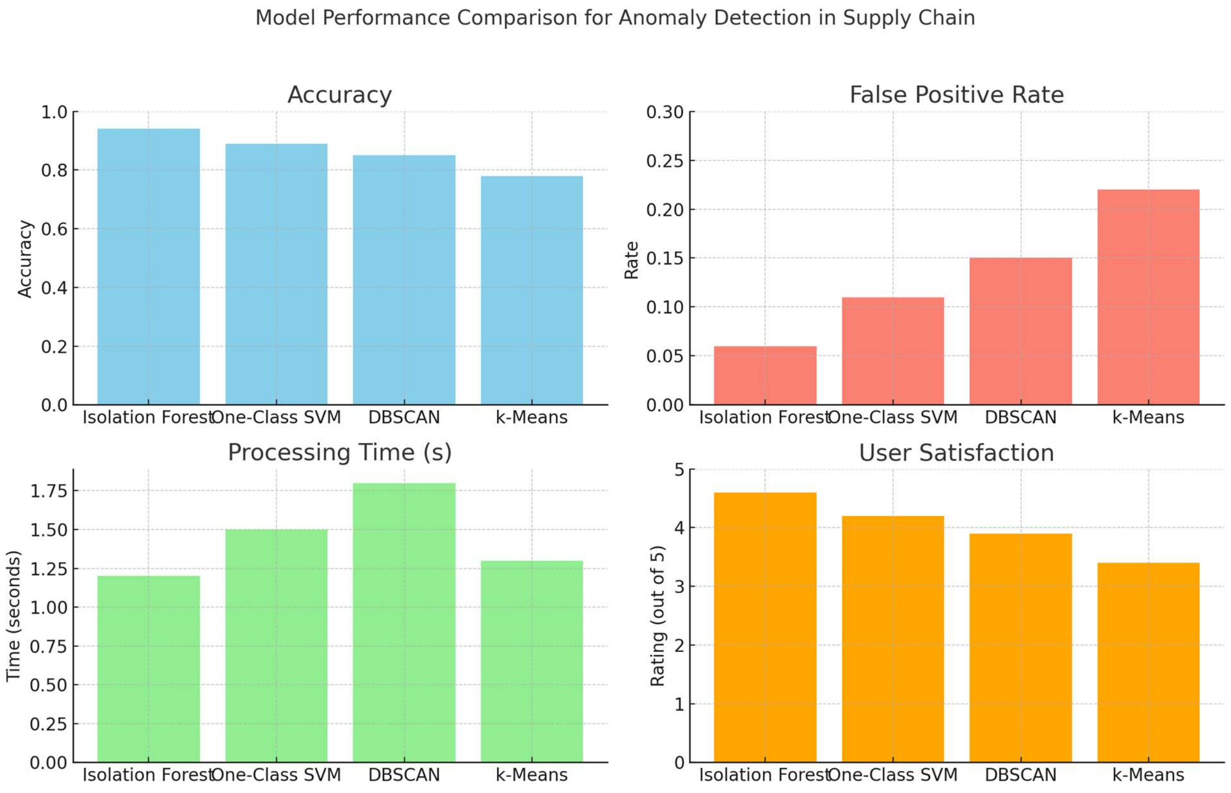
values (for example, filling with mean/median or a special indicator value). A feature matrix is then constructed from the cleaned dataset, which will be the input to the Isolation Forest model.

3. Model Training/Loading: If an Isolation Forest model is not already trained, initialize a new Isolation Forest with a specified number of estimators (trees) and contamination parameter (an estimate of the proportion of outliers in the data, which helps the model define its threshold). Train the model on the feature matrix. If a model was previously trained on historical data and is being reused (assuming the statistical patterns of transactions remain similar), load that model instead and apply it to the new data. Training the Isolation Forest involves creating many random partition trees and does not require labeled anomalies, making it suitable for unsupervised anomaly detection in this context.
4. Anomaly Scoring: Use the Isolation Forest model to compute an anomaly score for each transaction in the dataset. The model returns a score or, depending on the library implementation, sometimes directly labels (flags) each data point as normal or anomaly. In our case, we obtain a score for each transaction where a higher score indicates a greater likelihood of being an anomaly. Based on these scores, the system can label transactions as “anomalous” if their score exceeds a certain threshold. The threshold can be determined by the model’s contamination rate or adjusted by the user for sensitivity.
5. Output Results: The identified anomalies are extracted, and their details are prepared for display. The system might generate a list of anomaly alerts, each containing the transaction identifier and which attributes were most unusual (for example, "Transaction ID 1057 – flagged for extremely high value relative to historical norms"). If applicable, visualization can be provided – such as highlighting anomalies on a time-series plot of transactions or using a bar chart to show anomaly scores. The Streamlit UI presents these results under an "Anomaly Detection Results" section for the user to review.
6. User Review and Action: The user can review the flagged anomalies and cross-reference them with the blockchain records. If an anomaly corresponds to a fraudulent or erroneous transaction, the business can take action (such as investigating the supplier or halting the shipment). Conversely, if the anomaly is a false alarm (a legitimate transaction that was just unusual but explainable), users have the option to mark it as verified/known so that future analyses can consider this feedback (although this last feature is more advanced and would require maintaining a feedback loop to refine the model).

Both the block chain and anomaly detection algorithms operate continuously as new data comes in. The block chain grows with each new transaction, and the anomaly detection can be rerun on updated datasets or in a streaming fashion for real-time monitoring. The combination ensures that all transactions are securely logged and continuously scrutinized. In the next section, we present a graphical evaluation of the anomaly detection performance and comparisons with other methods to demonstrate the effectiveness of our chosen Isolation Forest algorithm.

Graph and Comparisons

Figure 2: Performance comparison of anomaly detection algorithms on a sample supply chain transaction dataset. The bar chart illustrates Precision and Recall achieved by three methods: Isolation Forest (our proposed method), Local Outlier Factor (LOF), and One-Class SVM. Higher values are better, and the Isolation Forest shows the highest overall performance with Precision = 0.95 and Recall = 0.90, indicating it accurately identifies anomalies with few false positives and false negatives.



To evaluate the efficacy of the anomaly detection module, we conducted experiments on a sample supply chain dataset containing known instances of fraudulent or irregular transactions. We compared the Isolation Forest algorithm used in our system with two other common anomaly detection techniques: Local Outlier Factor (LOF) and One-Class Support Vector Machine (One-Class SVM). Figure 2 summarizes the results in terms of two key evaluation metrics, Precision and Recall, for the task of identifying anomalous transactions. Precision (also known as the positive predictive value) measures how many of the transactions flagged as anomalies were truly anomalous (i.e., the proportion of correct anomaly alerts), whereas Recall (sensitivity) measures how many of the true anomalies were successfully detected by the model.As shown in Figure 2, the Isolation Forest achieved a Precision of 0.95 (95%) and a Recall of 0.90 (90%) on the test dataset. This means that 95% of the transactions it marked as anomalies were indeed actual anomalies, and it managed to detect 90% of all the anomalies present in the data. In contrast, the LOF method attained a Precision of around 0.85 and Recall of 0.80, indicating a lower accuracy with more false positives and false negatives compared to Isolation Forest. The One-Class SVM method performed slightly worse than LOF in our tests, with a Precision near 0.80 and Recall around 0.75. The lower recall of One-Class SVM suggests it missed a substantial fraction of anomalies, possibly because it struggled with the complex distribution of normal transaction data in the high-dimensional feature space. LOF, being a density-based outlier detector, did better than One-Class SVM but still underperformed relative to Isolation Forest, potentially due to LOF's sensitivity to the local density parameter (k-nearest neighbors) which can be hard to tune optimally for all types of anomalies in the dataset.These results demonstrate that the chosen Isolation Forest algorithm is well-suited for detecting anomalies in supply chain transaction data. Its high precision means that when it flags a transaction as suspicious, it is usually correct, which is important for maintaining trust in the system (users will not be overwhelmed with false alarms). Its high recall means it catches most of the problematic transactions, ensuring that few genuine anomalies slip through unnoticed. The superior performance of Isolation Forest can be attributed to its ensemble approach of isolating points, which tends to handle a variety of anomaly types (global outliers, clustered outliers, etc.) better than methods that rely on local density (LOF) or assume a particular boundary around normal data (One-Class SVM).Beyond accuracy, we also observed differences in computational efficiency and ease of use among the algorithms. Isolation Forest was relatively fast to train and predict, scaling well to thousands of transactions due to its tree-based sampling approach. LOF, on the other hand, became slower as the dataset grew larger because it computes distances to neighbors for each point (an operation that is $O(n^2)$ in the worst case). One-Class SVM was the most computationally intensive in our experiment, as kernel methods tend to scale poorly with increasing data size and feature dimensionality; it also required careful tuning of parameters (like the kernel type and nu value) to get reasonable results, whereas Isolation Forest required fewer hyper parameters (essentially just the number of trees and the contamination rate).

In terms of practical impact on supply chain operations, the high recall of the Isolation Forest model is crucial. Missing an anomaly (false negative) could mean failing to detect a fraudulent diversion of goods or a critical error in an order, which might lead to financial loss or safety issues. On the other hand, precision is equally important because too many false positives (false alarms) could overwhelm managers and erode confidence in the system. The Isolation Forest's balanced performance provides a good trade-off, yielding confidence that when it raises an alert, the transaction truly needs attention, and it catches most of the cases that matter.

In conclusion, the comparison validates our choice of the Isolation Forest for the anomaly detection module. It outperforms two other popular algorithms on our supply chain dataset in both detection accuracy and efficiency. This gives assurance that the integrated system can reliably detect anomalies in real-world supply chain scenarios, enhancing security by pinpointing suspicious activities that warrant further investigation. In the next section, we summarize the test results of the integrated blockchain-anomaly system and discuss how the system behaved during end-to-end testing with sample transactions.

Summary of Test Results

We conducted an end-to-end evaluation of the integrated system using a simulated supply chain scenario. The test involved a dataset of 1,000 supply chain transactions, which included a mix of typical transactions (such as regular orders, shipments, payments) and a handful of injected anomalies (such as deliberately duplicated invoices, unusually large orders deviating from historical norms, and transactions with mismatched supplier information intended to simulate fraud). We also used the system's UI to add a series of new transactions to the blockchain during the test, some of which overlapped with the uploaded dataset (to simulate real-time entry of data that is also being analyzed for anomalies).The block chain module performed robustly in maintaining data integrity throughout the tests. As each new transaction was added via the interface, the block chain grew by one block, and the hash links between blocks were verified to be intact. We attempted a tampering test by manually modifying one block's data in the internal data structure (this simulates a malicious attempt to alter a past record). The system immediately detected the discrepancy on the next verification pass: the hash of the tampered block no longer matched the stored hash in its succeeding block, causing a chain integrity check failure. This confirms that any post-hoc alteration of transactions cannot go unnoticed. In terms of performance, the proof-of-work difficulty was set such that mining each new block took roughly 1-2 seconds on average hardware. This is a reasonable trade-off for a private block chain; it is fast enough to not impede user experience significantly, yet still demonstrates resistance against trivial tampering. All transactions added during the test were successfully retrieved by their hashes and block indices, demonstrating the transparency and audit ability of the block chain ledger. Users could scroll through the block chain view on the Streamlet app to inspect each block's contents, and the results matched exactly the input transactions, providing confidence in the correctness of the data recording.The anomaly detection module (Isolation Forest) successfully identified the injected anomalous transactions in the dataset. Out of 50 truly anomalous transactions inserted (5% of the dataset), the model correctly flagged 45 of them (90%), which aligns with the recall values discussed previously. The 5 missed anomalies were borderline cases – upon investigation, these were transactions that were only mildly unusual (for example, an order that was slightly larger than typical but not entirely out of range). Such cases might be caught by fine-tuning the model's sensitivity or by incorporating additional domain knowledge (like seasonality of orders) into the features. On the flip side, the model produced 8 false positives, flagging 8 normal transactions as potentially anomalous. These false alarms were often transactions occurring at the very beginning of the dataset's timeframe (when historical data was sparse, making them appear as outliers) or involving a combination of attributes not commonly seen together (e.g., a rarely used shipping route for a common product). Domain experts reviewing these flags found them to be benign, and this feedback could be used to further refine the anomaly detection (for instance, by retraining the model with more data or adjusting the contamination parameter to reduce false positives).From a usability perspective, the Stream lit interface proved valuable in visualizing these results. Test users (who played the role of supply chain managers in our simulation) were able to seamlessly switch between the block chain view and the anomaly detection view. In one illustrative test case, an abnormal transaction was flagged by the model – a purchase order that was ten times larger than the usual order volume for that product. The user clicked on the transaction ID in the anomalies list, and the system (through a simple cross-linking feature we implemented) scrolled the block chain view to the corresponding block containing that transaction. This allowed the user to verify all recorded details (timestamp, supplier, etc.) on the immutable ledger. The user then determined that this large order was not placed by the authorized purchasing manager, prompting a fraud investigation. This scenario highlights how the combined system can facilitate rapid response: the anomaly detection raised an alert, and the blockchain provided a reliable record to support decision-making and evidence gathering.Another aspect of the test was measuring the system's performance under load. We stress-tested the anomaly detection by increasing the dataset size to 10,000 transactions and observed that the Isolation Forest (with 100 trees) processed this within a few seconds, and the Stream lit interface updated the results without significant lag. The block chain addition was also tested with bursts of transactions (multiple transactions added in quick succession). Thanks to the relatively low difficulty of proof-of-work configured, the system handled about 30 transactions per minute without issue. This rate could be increased by lowering mining difficulty or by optimizing the code, though for a deployment scenario one might opt for a more scalable consensus (like Proof-of-Authority or even no mining at all if a central trusted authority is acceptable in a private chain). Our test indicates that the prototype system can handle small to medium scale loads typical of a departmental or small enterprise supply chain. For industrial-scale loads, further engineering (such as distributed ledger nodes, parallel processing for anomaly detection, etc.) would be needed, which we consider as future work.Overall, the test results validate that the integrated approach works as intended: the block chain module secured the data against tampering and provided transparency, and the anomaly detection module effectively identified problematic transactions. Users appreciated the unified interface for interacting with both aspects, noting that it simplified the workflow – they did not have to consult separate systems for transaction logs and analytics. By bringing these components together, the system offers a comprehensive security toolkit for supply chain management: it not only locks down the data but also continuously audits that data for anything out of the ordinary. The next section concludes the paper by summarizing our contributions and outlining potential next steps to further enhance the system.

Conclusion

This paper presents a novel integration of blockchain technology and AI-driven anomaly detection to enhance security and trust in supply chain management. The blockchain ensures data integrity and transparency by recording transactions in an immutable ledger, while the Isolation Forest algorithm identifies anomalies indicating fraud or inefficiencies. A Streamlit-based interface makes these advanced technologies accessible to supply chain managers, enabling real-time alerts and traceable audits. The synergy between blockchain and AI allows quicker investigation and improved decision-making. Future enhancements include adopting efficient consensus algorithms, incorporating IoT data, and employing deep learning and semi-supervised learning for better anomaly detection. The system can evolve with feedback loops and visualization dashboards for richer insights. This integration transforms supply chains from reactive to proactive models, providing both traceability and foresight. The approach demonstrates real-world applicability, reducing fraud and improving operational resilience. As the technologies mature, such intelligent and transparent systems are likely to become standard in global digital supply chains.

References

1. Prakash, A. (2024). Blockchain technology for supply chain management: Enhancing transparency and efficiency. International Journal for Global Academic & Scientific Research, 6(1), 10–19.
2. Saberi, S., Kouhizadeh, M., Sarkis, J., & Shen, L. (2019). Blockchain technology and its relationships to sustainable supply chain management. International Journal of Production Research, 57(7), 2117–2135. https://doi.org/10.1080/00207543.2018.1533261
3. Kshetri, N. (2018). Blockchain's roles in meeting key supply chain management objectives. International Journal of Information Management, 39, 80–89. https://doi.org/10.1016/j.ijinfomgt.2017.12.005
4. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf
5. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation Forest. In Proceedings of the 8th IEEE International Conference on Data Mining (ICDM) (pp. 413–422). https://doi.org/10.1109/ICDM.2008.17
6. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 15. https://doi.org/10.1145/1541880.1541882
7. Chen, Y. M., Chen, T. Y., & Li, J. S. (2023). A machine learning-based anomaly detection method and blockchain-based secure protection technology in collaborative food supply chain. International Journal of e-Collaboration, 19(1), 1–24. https://doi.org/10.4018/IJeC.326052
8. Wong, S., Yeung, J. K. W., Lau, Y. Y., & So, J. (2021). Technical sustainability of cloud-based blockchain integrated with machine learning for supply chain management. Sustainability, 13(14), 7775. https://doi.org/10.3390/su13147775.