PERSPECTIVE

Can Sidechains and AI Save the Edge? A Perspective on Scalability and Security in IoT-Fog Blockchains

Saeed Javanmardi[1] **and Marco Scarpa**[1,*]

[1] Department of Engineering, University of Messina, 98166 Messina, Italy

Abstract

IoT-Fog networks create big challenges in scalability, latency, and security at the edge. This paper gives a simple perspective on how *sidechains* and *artificial intelligence* (*AI*) can help. Sidechains process local transactions near devices, increase throughput, and reduce delay. AI methods, such as Random Forest, detect abnormal traffic, predict workload peaks, and adjust network settings (for example, block time and number of validators). We present clear, small formulas for throughput, latency, checkpoint cost, and security, and we show how these formulas guide design choices (block time T_b, confirmations k, checkpoint period τ). The goal is an easy model that links design choices to performance and security, helping readers build faster, safer, and more adaptive IoT-Fog Blockchain networks.

Keywords: blockchain, sidechain, IoT-Fog networks, scalability and security, artificial intelligence.

1 Introduction

IoT-Fog architectures are becoming more important as the number of connected devices grows rapidly. In many smart environments, such as smart cities, health systems, and industrial networks, a huge amount of data is generated close to the users. This data often needs to be processed and validated quickly, without relying only on distant cloud servers. The edge layer, also called the fog layer, plays a key role in supporting local computation, storage, and communication [1].

Blockchain can help verify transactions between IoT devices without needing a central authority. However, most public Blockchain platforms suffer from low throughput and high confirmation time. When thousands or millions of devices send transactions simultaneously, the network becomes congested and performance drops sharply. Standard Blockchains were not built for the speed and scale of fog–IoT networks. This gap between IoT needs and Blockchain limitations motivates new designs that can scale better at the edge [2].

Sidechains can help solve these problems. A sidechain is a separate Blockchain that runs next to the mainchain and can process transactions on its own. It handles local work and sometimes anchors its state to the mainchain. This lowers the load on the main network, raises throughput, and cuts delay for local actions. In fog–IoT networks, sidechains let local gateways or fog nodes check and store transactions near the devices [3]. They still keep global consistency by sending checkpoints to the mainchain from time to time. This

Submitted: 22 December 2025
Accepted: 17 March 2026
Published: 23 March 2026

Vol. 1, **No.** 1, 2026.
10.62762/JSS.2025.237121

***Corresponding author:**
✉ Marco Scarpa
mscarpa@unime.it

Citation
Javanmardi, S., & Scarpa, M. (2026). Can Sidechains and AI Save the Edge? A Perspective on Scalability and Security in IoT-Fog Blockchains. *Journal of Systems Scalability*, 1(1), 23–28.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

mixed design gives local speed and global security together.

Artificial intelligence (AI) techniques add another important dimension to this discussion. AI can examine network behavior, find intrusions, predict traffic changes, and optimise resources in fast-changing systems. For example, a Random Forest model can learn normal patterns and then detect abnormal transaction flows or strange communication data. By using AI with sidechains, fog nodes become smarter and more flexible. They can find threats faster and keep the network stable in real time. They can also change settings when needed, share the work better, and use their resources well [4].

In this paper, we use the term AI in a practical way. We mean data-driven decision methods that help the sidechain at the edge. In particular, we focus on two functions: (1) anomaly detection for security, and (2) adaptive resource and parameter tuning (for example, block time and number of validators). These functions can be implemented with machine learning (such as Random Forest), but they can also be implemented with simpler rule-based methods. We use the word AI as an umbrella term for these mechanisms, and we do not claim general or human-like intelligence. Our architecture is similar to MEC because it splits work between a local layer and a stronger layer. In our case, the sidechain handles IoT transactions near gateways, and the mainchain provides global anchoring through checkpoints. However, unlike classic MEC, we also focus on Blockchain trust, consensus, and security anchoring. We also use AI for parameter adaptation and anomaly detection at the edge.

There is a clear need for new designs that can combine local processing speed with strong security guarantees. Sidechains can improve performance, and AI methods can improve adaptability and security. However, their combined potential at the edge has not been clearly analyzed or presented in a simple analytical way. The main contributions of this paper are as follows:

- We give a clear and simple explanation of how sidechains can improve throughput and reduce latency in IoT-Fog Blockchain networks.
- We present how AI, and in particular Random Forest models, can support security and dynamic adaptation at the edge.
- We develop simple analytical models for sidechains, AI, and their combination, including basic formulas and numerical examples that help understand the impact on performance and security.

2 Sidechains: Analytical Modeling

One big benefit of sidechains is higher throughput (more transactions per second). Because a sidechain works on its own, it can use a lighter consensus mechanism and shorter block times. So it can process more transactions in less time than the main chain. At the same time, latency is reduced because the validation does not depend on the global network. Devices can receive confirmations from their local sidechain in seconds or milliseconds, instead of waiting for blocks to be finalized on the main chain.

We can describe this using simple formulas. Let:

$$\text{TPS} = \frac{B}{T_b}$$

where B is the number of transactions in each block, and T_b is the block time. If the sidechain uses a smaller T_b, then TPS becomes higher. We define $T_{\text{network}}^{\text{local}}$ as the network communication delay inside the local domain (e.g., IoT device to gateway or sidechain nodes). We define $T_{\text{network}}^{\text{global}}$ as the network communication delay over the wide-area path (e.g., sidechain to mainchain or cloud).

For latency, we can write:

$$L_{\text{side}} = T_{\text{network}}^{\text{local}} + T_b$$

$$L_{\text{main}} = T_{\text{network}}^{\text{global}} + T_b^{\text{main}}$$

where L_{side} is the latency of the sidechain, L_{main} is the latency of the main chain, and T_b^{main} is the block time of the main chain. Because local network time is usually much shorter than global time, sidechain latency is much lower.

Confirmations (k) and checkpoint period (τ) are two key parameters in our model. k is the number of blocks that must be added after a transaction before we treat it as confirmed. A larger k gives more safety, but it increases the confirmation time, that is, the time until a transaction is treated as confirmed. In a simple form, the sidechain confirmation time can be approximated as $k \times T_b$. τ is the time between two checkpoints that the sidechain sends to the mainchain. A smaller τ gives stronger anchoring, but it adds more overhead on the mainchain.

Table 1. Parameters used in the analytical model for the combination of Sidechains and AI.

Symbol	Description
TPS_{side}	Throughput of the sidechain
TPS_{AI}	Extra throughput gained by AI adaptation
$\text{TPS}_{\text{total}}$	Total throughput after combining sidechain and AI
P_d	Attack detection probability of the AI model
P_{chain}	Baseline security probability of the sidechain
P_{secure}	Total security probability after combining both methods

3 Artificial Intelligence: Analytical Modeling

One important use of AI is in intrusion detection. In this section, AI refers to machine-learning based mechanisms used at the fog/edge layer. We use AI to support intrusion/anomaly detection and dynamic adaptation of sidechain settings. Random Forest is one example, but the same goals can be achieved with other learning methods or lightweight heuristics. Fog nodes can collect network and transaction data and use machine learning models to detect abnormal behaviors. For example, a Random Forest model can be trained to recognise attack patterns, such as flooding, unauthorized access, or unusual transaction flows. In this paper, **RF** means **Random Forest**. When a suspicious pattern is found, the networks can respond quickly by blocking traffic, isolating nodes, or adjusting the Blockchain consensus rules. This improves the security of both the sidechain and the mainchain.

We can write this idea with a simple formula. Let:

$$P_d = \frac{TP}{TP + FN}$$

where P_d is the detection probability, TP is the number of true positives, and FN is the number of false negatives. A higher P_d means the Random Forest can detect more attacks correctly.

Another benefit of AI is that it can support the dynamic configuration of sidechain parameters. Instead of using fixed rules, AI can watch the network and change settings automatically. For example, if the delay gets longer, the network can make block times shorter or add extra validators. If security risks increase, it can make the consensus rules stronger or send checkpoints to the main chain more often. This behavior can also be shown with simple rules:

$$T_b^{new} = \begin{cases} T_b - \Delta t, & \text{if latency is high} \\ T_b, & \text{otherwise} \end{cases}$$

$$V^{new} = \begin{cases} V + 1, & \text{if risk is high} \\ V, & \text{otherwise} \end{cases}$$

where T_b is the block time, Δt is the adjustment step, and V is the number of validators.

This kind of intelligent control makes the Blockchain architecture more flexible and efficient, because it can change itself based on network conditions and security risks.

4 Combining Sidechains and AI: A Perspective

Sidechains and AI can work together to solve many of the problems that exist in IoT-Fog Blockchain networks. Sidechains focus on improving scalability and reducing delay by processing local transactions close to where they happen. AI focuses on learning from data, detecting problems, and making better decisions automatically. When these two technologies are combined, they can support each other and make the edge layer stronger and smarter. Table 1 summarizes the key parameters used in our analytical model for this combination.

A simple example can help illustrate this idea. Imagine a fog network where each local gateway runs its own sidechain. All local IoT devices send their transactions to this sidechain, which validates and records them quickly. At the same time, each gateway also runs a Random Forest model to monitor traffic and security events. The AI model watches for abnormal patterns, such as sudden increases in transaction volume or suspicious activity. If it detects a possible attack or overload, it can trigger actions immediately, such as blocking malicious traffic, increasing the number of validators, or shortening block intervals. This makes the network both faster and more secure.

This combination also allows better use of resources. By predicting future traffic, AI can help the sidechain prepare for busy periods in advance. For example, if

the AI model expects a high number of transactions in the next few minutes, the network can adjust block size or validation speed early. This avoids sudden congestion and keeps latency low even when many devices are active. It also reduces the pressure on the main chain because the sidechain can handle local peaks more efficiently.

We can describe this interaction without ad hoc numbers. Let TPS_{side} be the base throughput of the sidechain. Let $h \in [0, 1]$ be the prediction hit rate of the AI model (measured on validation data), and let γ be the average throughput gain (tx/s) when the adaptation is correctly triggered. Here, the *prediction* means predicting a near-future traffic peak (overload) in a short time window (e.g., the next few minutes). We model this as a binary classification task: *peak vs. no-peak* (or *adaptation needed vs. not needed*). A *hit* means the model correctly predicts the class and triggers the right action. For this reason, using a hit/miss rate is suitable here. If a regression model is used to predict the exact traffic value, it can still be converted to this binary decision by using a threshold (peak if predicted traffic is above a limit). Then

$$\text{TPS}_{\text{total}} = \text{TPS}_{\text{side}} + h\,\gamma.$$

If false alarms add overhead o (tx/s), we define $p_{FP} \in [0, 1]$ as the false-positive occurrence rate over all evaluated samples, with

$$p_{FP} = \frac{FP}{N},$$

where N is the total number of evaluated samples. Then the net effect is

$$\text{TPS}_{\text{total}} = \text{TPS}_{\text{side}} + h\,\gamma - p_{FP}\,o,$$

with bounds

$$\text{TPS}_{\text{side}} \le \text{TPS}_{\text{total}} \le \text{TPS}_{\text{side}} + \gamma.$$

Here $h = \frac{TP}{TP+FN}$ is taken from the AI confusion matrix on a peak-detection task, while p_{FP} shows how often false alarms occur in the full evaluated stream. This form is more suitable for system overhead, because the cost appears each time a false positive is triggered, independently of the number of true negatives. We define

$$\text{TPS}_{\text{AI}} = h\gamma - p_{FP}o$$

as the extra throughput gained by AI adaptation. Then we can also write

$$\text{TPS}_{\text{total}} = \text{TPS}_{\text{side}} + \text{TPS}_{\text{AI}}.$$

For security, let P_d be the detection rate of the AI model measured on a labeled validation set (true positives over all attacks), and let P_{chain} be the baseline stop rate of the sidechain measured in your testbed (stopped attacks over all attacks without AI).

If AI and chain act independently, a simple combination is:

$$P_{\text{secure}} = 1 - (1 - P_d)(1 - P_{\text{chain}}).$$

Under the independence assumption, we have

$$P_{secure} = P_d + P_{chain} - P_d P_{chain}.$$

This value is also a tighter upper bound than $\min(1,\, P_d + P_{chain})$, especially when P_d and P_{chain} are high. Therefore, under independence, we use $P_d + P_{chain} - P_d P_{chain}$ as the upper bound.

If independence is not safe to assume, use the inclusion–exclusion form with overlap P_{both} measured from experiments:

$$P_{\text{secure}} = P_d + P_{\text{chain}} - P_{\text{both}},$$

with the bounds

$$\max(P_d,\, P_{\text{chain}}) \;\le\; P_{\text{secure}} \;\le\; \min\big(1,\, P_d + P_{\text{chain}}\big).$$

How to compute in practice (brief): 1) train RF, evaluate P_d on validation traces; 2) run baseline sidechain, record P_{chain}; 3) run both together, measure P_{both}; 4) report P_{secure} with CIs.

Using sidechains also has drawbacks compared to using one centralized Blockchain (or one single main chain). First, the system becomes more complex because we must manage multiple ledgers, checkpointing, and cross-chain links (peg or bridge). This added complexity can create new attack points, especially around checkpoints and cross-chain transfer logic. Second, sidechains reduce globality, because data and state are split across chains. A node on one sidechain may not see the full global state in real time, so global consistency can be weaker and more delayed. Third, security on a sidechain can be lower than on a single strong chain, because the validator set can be smaller and easier to attack. For these reasons, sidechains need careful design of validators, checkpoint rules, and monitoring to keep security and global consistency acceptable.

Sidechains also have some important challenges. If there is an error in consensus between sidechains,

or if cross-chain synchronization fails, the system can produce inconsistent states for a period of time. This may cause delayed confirmations, problems in cross-chain transfers, extra recovery cost, and in some cases a higher risk of double spending. Compared with a centralized approach, sidechains offer better scalability and decentralization, but they are more complex to manage. A centralized system is simpler and can keep global consistency more easily because decisions are made in one place. However, it also creates a single point of failure and reduces fault tolerance. For this reason, sidechain-based systems need careful coordination, secure checkpoint rules, and strong monitoring.

There are still some challenges to consider. Running AI models at the edge requires computational power and good-quality data. If the data is noisy or incomplete, the AI model may produce wrong predictions or false alarms. In addition, the coordination between sidechains and the main chain must remain secure. Checkpoints, validator selection, and peg mechanisms need to be well designed so that attackers cannot exploit the interaction between components. Balancing performance, security, and resource usage is a key research question. Despite these challenges, combining sidechains with AI provides a powerful approach to saving the edge. It combines the power of Blockchain and smart data methods. This shows that future IoT-Fog networks can become faster, safer, and more flexible by using these two technologies together.

5 Research Summary

This paper presents analytical models that quantify how sidechains and AI affect throughput, latency, and security in IoT–Fog Blockchain environments. The results show that sidechain throughput can be expressed as $TPS = \frac{B}{T_b}$, where reducing the block time T_b or increasing block size B directly raises transaction capacity, while latency decreases due to local communication paths compared to the main chain. The latency gap between sidechain and main chain is modeled as $L_{\text{side}} = T_{\text{local}}^{\text{network}} + T_b$ versus $L_{\text{main}} = T_{\text{global}}^{\text{network}} + T_b^{\text{main}}$, demonstrating faster confirmation under fog conditions.

For AI, the Random Forest intrusion detector contributes to security and dynamic resource tuning, with detection probability $P_d = \frac{TP}{TP+FN}$ and false-positive occurrence rate $p_{FP} = \frac{FP}{N}$, where N is the total number of evaluated samples. These are integrated into a throughput adaptation model $TPS_{\text{total}} = TPS_{\text{side}} + h\gamma - p_{FP}o$, where h is prediction accuracy, γ represents the throughput gain during correct scaling, and o is the overhead caused each time a false alarm is triggered.

The combined security of sidechains and AI is represented as $P_{\text{secure}} = 1 - (1 - P_d)(1 - P_{\text{chain}})$ under independence assumptions, and bounded by $\max(P_d, P_{\text{chain}}) \leq P_{\text{secure}} \leq \min(1, P_d + P_{\text{chain}})$. Overall, the findings show that properly tuned sidechains can improve throughput and latency at the edge, and that AI can increase intrusion detection and maintain performance, but both methods require careful parameter selection for checkpoint period τ, block size B, block time T_b, and confirmations k to avoid high overhead and preserve consistency with the main chain.

6 Conclusion

This paper gave a simple view of how sidechains and AI can help IoT-Fog Blockchains. Sidechains increase throughput by local processing and reduce delay near devices, while still keeping trust with periodic checkpoints to the main chain. AI methods (e.g., Random Forest) improve security and adaptability by detecting abnormal traffic, predicting peaks, and tuning settings such as block time and confirmations. We also presented small analytical models that link design choices to performance and security. The key knobs are block time T_b, confirmations k, and checkpoint period τ. These models help engineers choose T_b and B for speed, pick k for safety delay, and size τ to control anchor overhead. For security, combining the chain's baseline protection with AI detection increases the overall stop rate of attacks.

Data Availability Statement

Not applicable.

Funding

This work was partially funded by European Union - Next generation EU - PNRR - Missione 4, Componente 2, Investimento 1.1 - Bando PRIN 2022 PNRR - Decreto Direttoriale n. 1409 del 14-09-2022 - Progetto PaB-PIF, CUP J53D2301493, Project Id. P20227W8ZC.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Javanmardi, S., Nascita, A., Pescapè, A., Merlino, G., & Scarpa, M. (2025). An integration perspective of security, privacy, and resource efficiency in IoT-Fog networks: A comprehensive survey. *Computer Networks, 270*, 111470. [CrossRef]

[2] Reshi, I. A., & Sholla, S. (2025). IBF network: enhancing network privacy with IoT, blockchain, and fog computing on different consensus mechanisms. *Cluster Computing, 28*(3), 208. [CrossRef]

[3] Yang, L., Jiang, R., Pu, X., Wang, C., Yang, Y., Wang, M., ... & Tian, F. (2024). An access control model based on blockchain master-sidechain collaboration. *Cluster Computing, 27*(1), 477-497. [CrossRef]

[4] Paramesha, M., Rane, N., & Rane, J. (2024). Big data analytics, artificial intelligence, machine learning, internet of things, and blockchain for enhanced business intelligence. *Artificial Intelligence, Machine Learning, Internet of Things, and Blockchain for Enhanced Business Intelligence (June 6, 2024)*. [CrossRef]

Saeed Javanmardi is a postdoctoral researcher at the University of Messina in Italy. He received his PhD in Information Technology from the University of Naples Federico II. His research focuses on IoT-Fog networks, Blockchain security, and resource management. He has worked on intrusion detection systems and task scheduling, especially in drone and fog environments. During his academic career, he has published several journal and conference papers and collaborated with research groups and industry partners in Italy. His goal is to design secure and efficient network solutions for modern IoT applications. (Email: saeed.javanmardi@unime.it)

Marco Scarpa is a full professor at the Department of Engineering, University of Messina, Italy. He was born on 4 March 1969 in Italy and holds a PhD in Computer Science. His research interests include distributed computing, computer reliability, and wireless sensor networks. He teaches courses in computer systems and basic computer science, and he has contributed to many research projects and scientific publications in computing and engineering. (Email: marco.scarpa@unime.it)