Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.071124

ARTICLE

Blockchain and Smart Contracts with Barzilai-Borwein Intelligence for Industrial Cyber-Physical System

Gowrishankar Jayaraman[1], Ashok Kumar Munnangi[2], Ramesh Sekaran[3], Arunkumar Gopu[3] and Manikandan Ramachandran[4,*]

[1]Department of Computer Science and Engineering, JAIN (Deemed-to-be University), Banglore, 562112, Karnataka, India
[2]Department of Information Technology, Siddhartha Academy of Higher Education-Deemed to be University, Vijayawada, 520007, India
[3]Department of Computer Science and Engineering, Dayananda Sagar University, Bengaluru, 562112, Karnataka, India
[4]School of Computing, SASTRA Deemed University, Thanjavur, 613401, Tamil Nadu, India

*Corresponding Author: Manikandan Ramachandran. Email: srmanimt75@gmail.com

Received: 31 July 2025; Accepted: 24 November 2025; Published: 12 January 2026

ABSTRACT: Industrial Cyber-Physical Systems (ICPSs) play a vital role in modern industries by providing an intellectual foundation for automated operations. With the increasing integration of information-driven processes, ensuring the security of Industrial Control Production Systems (ICPSs) has become a critical challenge. These systems are highly vulnerable to attacks such as denial-of-service (DoS), eclipse, and Sybil attacks, which can significantly disrupt industrial operations. This work proposes an effective protection strategy using an Artificial Intelligence (AI)-enabled Smart Contract (SC) framework combined with the Heterogeneous Barzilai–Borwein Support Vector (HBBSV) method for industrial-based CPS environments. The approach reduces run time and minimizes the probability of attacks. Initially, secured ICPSs are achieved through a comprehensive exchange of views on production plant strategies for condition monitoring using SC and blockchain (BC) integrated within a BC network. The SC executes the HBBSV strategy to verify the security consensus. The Barzilai–Borwein Support Vectorized algorithm computes abnormal attack occurrence probabilities to ensure that components operate within acceptable production line conditions. When a component remains within these conditions, no security breach occurs. Conversely, if a component does not satisfy the condition boundaries, a security lapse is detected, and those components are isolated. The HBBSV method thus strengthens protection against DoS, eclipse, and Sybil attacks. Experimental results demonstrate that the proposed HBBSV approach significantly improves security by enhancing authentication accuracy while reducing run time and authentication time compared to existing techniques.

KEYWORDS: Industrial CPS; security; artificial intelligence; blockchain; smart contract; heterogeneous

1 Background

A blockchain is an immutable distributed ledger that securely records transactions and tracks assets across industrial networks. Traditional industrial security processes are slow and labor-intensive, particularly with increased data exchange, monitoring, and automation. Industrial Cyber-Physical Systems (ICPSs) integrate cyber and physical worlds, enabling data-driven, intelligent, and collaborative industrial operations. However, ICPSs are highly vulnerable to attacks, such as DoS, eclipse, and Sybil attacks, making security a major concern.

Blockchain and Smart Contracts offer strong potential for securing [1] Industrial IoT (IIoT) data by ensuring transparency, privacy, and tamper-resistant communication. Existing blockchain-based IIoT approaches improve security and reduce delays; however, they still suffer from limitations such as unaddressed block validation time, inadequate legislative compliance, and inability to minimize runtime or attack success probability [2].

To overcome these issues, this study proposes an Barzilai (HBBSV) method. It aims to reduce runtime, enhance authentication accuracy, and minimize attack success by classifying the component features as secure or insecure. Smart Contracts verify production plant data, whereas the Barzilai–Borwein Support Vectorized algorithm strengthens detection against common attacks.

Main Contributions of HBBSV (Crisp Points)

1. Reduced runtime with lower attack success probability.
2. Smart Contract and blockchain-based verification to speed up authentication and ensure secure condition monitoring in ICPSs.
3. Barzilai–Borwein supports vectorized classification to filter secure components and eliminate insecure components to improve security.

Motivation

This study uses the HBBSV method to classify component features within a production line, thereby improving security, authentication accuracy, and authentication time. As blockchain-based smart contracts face security challenges such as DDoS attacks, this study integrates blockchain and smart contracts to ensure secure and timely access to industrial CPSs.

The remainder of this paper is structured as follows: Section 2 reviews related work, Section 3 presents the HBBSV method, Section 4 discusses experiments and results, and Section 5 concludes the study.

2 Related Works

A review of industrial blockchain applications in [3] highlighted security and integrity issues, but many prior studies lacked adequate security and privacy considerations. Two privacy-preserving schemes, DeepPAR and DeepDPA, were introduced in [4] for ICPSs; DeepPAR protects input privacy and update secrecy, while DeepDPA ensures backward secrecy but does not reduce runtime. CPS standards and components have been summarized in [5], although attack detection was missing.

A C2P cyber-physical risk model using Bayesian networks and SHS was proposed in [6]; however, other steady-state methods were not explored. The CPS studies in [7,8] did not address latency. In [9], privacy-preserving ICPSs and blockchain applications (e-government, e-health, cryptocurrencies, smart cities, and cooperative ITS), but did not minimize execution time. IoT–blockchain challenges were reviewed in [10], and storage limitations were overlooked in.

The CPS prototype in [11] faced safety issues owing to its scale and complexity. The safety control tasks in [12] did not reduce subsystem preservation time. Blockchain-enabled Safe-aaS in [13] enhanced security but lacked a hybrid blockchain. IoT dataset sharing for ICPSs in [14] involves third-party risk. AI and Blockchain offer promising solutions to enhance cybersecurity in smart cities were proposed in [15], whereas a blockchain-IIoT framework in [16] omitted data authorization and storage rules. Digital tokens for manufacturing traceability in [17] did not reduce the runtime. A novel pattern of malware validation scheme based on blockchain technology was introduced in [18]. The POMS in [19] ignored anonymity and transparency.

The hybrid machine learning-blockchain approaches performed in [20] to ensures data integrity, secures communication. The PDI security model in [21] identified issues at the process, data, and infrastructure levels, but was not applied to trustworthy industrial AI. The Machine Learning and Blockchain Synergy in [22] for focus on smart contract (SC).

The digital twin studies in [23,24] faced limitations in terms of blockchain suitability and large-scale data handling. A several security challenges in IoT-enabled SG applications to support sustainability [25]. A blockchain-based framework [26] to leverages decentralized security, smart contracts, and edge computing, and cybersecurity in ADN [27].

A blockchain technology with privacy-preserving framework [28] to achieve the required level of security while improving system efficiency. A cyber-security trust model [29] to provides multi-risk protection for secure data transmission in cloud computing. A deepCLG hybrid learning model [30] to improve network intrusion detection systems (NIDSs). A distributed intrusion detection framework [31] based on fog computing.

The weighted and extended isolation forest algorithms [32] for the real-time detection of cyber-attack transactions. An integrated approach using Deep Neural Networks and Blockchain technology (DNNs-BCT) [33] to improve the detection and prevention in IoT environments. An investigative report based on cyber vulnerability detection [34] using AI, ML, and DL. A novel framework [35] to integrates artificial intelligence (AI), blockchain, and smart contracts. A fully decentralized system based on ethereum smart contracts and Interplanetary File System (IPFS) [36] for IIoT. A Hyperledger Fabric-based blockchain network [37] for EVCSs to mitigate these risks.

The symbols of $\emptyset$, W^T and C_i described in below notation Table 1.

Table 1: Notation table

Ø	**Scaling function**
$MAX\left(W^T C_i + b\right) > 0$	The components are present in the positive quadrant
$MAX\left(W^T C_i + b\right) < 0$	The components are present in the negative quadrant
$SIGN\left[MAX\left(W^T C_i + b\right)\right] = 1$	Label is Negative
$SIGN\left[MAX\left(W^T C_i + b\right)\right] = -1$	Label is Positive

3 Methodology

Smart contracts are programs deployed on a blockchain that automatically execute agreement terms when predetermined conditions are met. They remove the need for intermediaries and ensure immediate, trustworthy outcomes. A Cyber-Physical System (CPS) is a computer-controlled system integrating computation with physical processes. CPSs are core components of IIoT and Industry 4.0, enabling real-time intelligent applications through interconnected sensors, aggregators, and actuators. They monitor and manipulate physical objects to create efficient, reliable, and secure smart environments. CPS applications include smart cities, healthcare, manufacturing, transportation and grids. However, connecting cyber and physical layers introduces major security risks. To enhance security, this study proposes a blockchain–smart contract framework that enables secure, tamper-resistant transactions without third parties. The HBBSV method addresses the common attacks.

Sybil attacks: multiple fake identities

Eclipse attacks: isolating a victim node

DoS attacks: overwhelming nodes with traffic

The blockchain serves as a distributed transaction ledger that ensures data integrity. Smart contracts verify the contract execution and support secure on-chain transactions. In this study, on-chain transactions secure industrial CPS data, and the HBBSV smart contract model reduces runtime and lowers attack success probability.

Fig. 1 illustrates HBBSV: manufacturer components (MC) send timestamps and eight predicted features to the blockchain, whereas actual component data (training data) are stored in component C for validation.

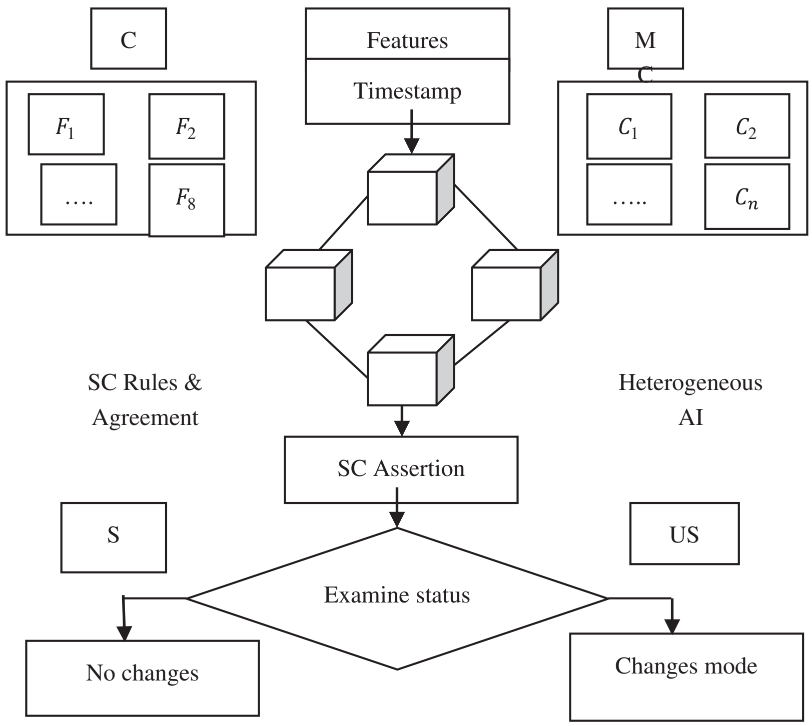

Figure 1: Block diagram of HBBSV

3.1 Dataset Discription

The proposed method was applied to two datasets. The first is the production plant data for the condition-monitoring dataset. This dataset was obtained from https://www.kaggle.com/datasets/inIT-OWL/production-plant-data-for-condition-monitoring (accessed on 23 November 2025). It includes 8 features. The production plant data for the condition monitoring dataset included the conditions of an important component within the production lines, which is usually not available directly through a sensor and must be derived from a multitude of available signals.

On the other hand, the Versatile Production System dataset obtained from https://www.kaggle.com/datasets/inIT-OWL/versatileproductionsystem?resource=download (accessed on 23 November 2025) comprises six features as well as a large number of instances. The number of CSV files obtained in different ways: delivery model, dosing model, Filling_ALL.module.csv, Filling_CapGrabber.module, Filling_CapScrewer.module, Filling_CornPortioning.module, Filling_Pump.module, Production.csv, Storagemodule. The dataset was mainly utilized to produce csv files in 7729 instances and ranged from 100 to 1000 components.

3.2 System Model

Let us consider the input production plant data for condition monitoring of the respective plants, concentrating on the prediction of the component condition within production lines. These components are mathematically represented as follows.

$$C = C_1, C_2, \ldots, C_n \tag{1}$$

where, 'C' represents the components of interest for condition monitoring; the state of a component is vital for a plant to successfully function and achieve high-quality products. From the input dataset, 8 features '$F_1, F_2, \ldots, F_8$' corresponding to the components is mathematically formulated via the component matrix 'CM', given by:

$$CM = \begin{vmatrix} C_1F_1 & C_1F_2 & \ldots & C_1F_8 \\ C_2F_1 & C_2F_2 & \ldots & C_2F_8 \\ \ldots & \ldots & \ldots & \ldots \\ C_nF_1 & C_nF_2 & \ldots & C_nF_8 \end{vmatrix} \tag{2}$$

where 'n' number of components are considered. Each component includes 8 features (i.e., $F_1, F_2, \ldots, F_8$). with the aid of the component matrix, consisting of 8 features for each component, 'MC' requests the smart contract 'SC' on the blockchain 'BC' to establish its viability. Each request in the 'BC' network represents a transaction 'T_i' and is mathematically formulated as follows:

$$Req \rightarrow (T_1, T_2, \ldots, T_n) \tag{3}$$

On the other hand, production plant data for conditional monitoring are confirmed by admin if $Training_{data} = Predicted_{data}$.

3.3 BC and SC for ICPSs

In this ICPS context, BC and SC refer to blockchain and smart contracts, and not social categories. The smart contract (SC) performs both the administrative and conditional checks. It contains Component Functions (CF) and Component Events (CE) at specific blockchain addresses. CF represents the code through which component Ci executes transactions Ti, while CE notifies all components of the system events.

All SC actions are governed by smart contract codes that are visible across the blockchain network, ensuring rule-based trust and transparency. In the proposed method, a single smart contract SC manages component information (SC_C). Each manufacturer interacts with SC_C to enroll the components and update theirfeatures. This unified management reduces inconsistencies and minimizes the probability of attack success.

Validating individual blocks with Smart Contracts before adding them to a blockchain is complex and requires an AI-based analysis. Therefore, the Heterogeneous Barzilai–Borwein AI (HBBAI) model was introduced. Fig. 2 shows the block structure used for ICPS condition monitoring, containing data, timestamp (TS), components C_i, and features F_i. Using SVM-based ICPSs with blockchain and smart contracts, the HBBAI model determines whether each component's condition is within the production line. If so, no security breach is detected; otherwise, the component is flagged as insecure.

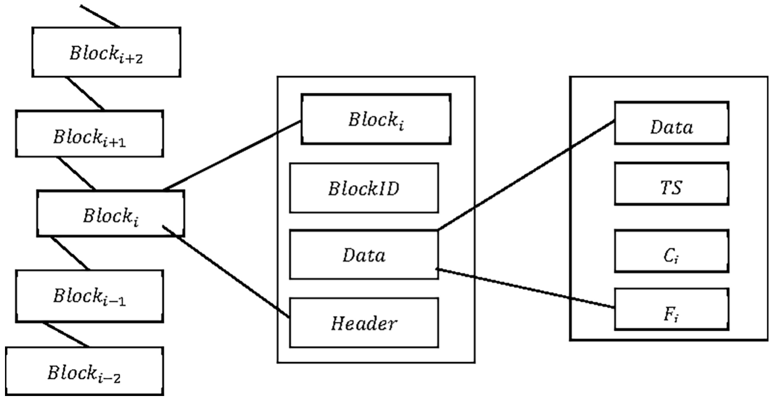

Figure 2: Sample block structure for condition monitoring in industrial-based CPS

For this process, a binary-labeled training set of components is defined as

$$D = \{(p_1, q_1), (p_2, q_2), \ldots, (p_n, q_n)\} \tag{4}$$

where '$p_i \in C_i$' and '$q_i \in \{+1, -1\}$'. Then, the optimization using the conventional AI model is represented as

$$MIN \frac{1}{2} W^T W + C\alpha_i, such\ that\ (q_i \emptyset (p_i) + b) \geq 1 - \alpha_i \tag{5}$$

where '$\emptyset$' represents the scaling function that is used to scale training data into a higher dimensional feature space. 'W' denotes the normal vector to the hyperplane, where the scaling function is applied to identify the separated hyperplane with a higher margin.

$$q_i = Q_i MAX\left(W^T C_i + b\right) \tag{6}$$

$$Q_i' = SIGN\left[MAX\left(W^T C_i + b\right)\right] \tag{7}$$

where the component with maximum '$W^T C_i + b$' represents an illustrative component of the production plant data. In the positive component of the production plant data, '$MAX\left(W^T C_i + b\right) > 0$' indicates that the components are present in the positive quadrant. On the other hand, '$MAX\left(W^T C_i + b\right) < 0$' means that the component is in the negative quadrant. Eq. (7) corresponds to the component timestamp, where the label is negative if '$SIGN\left[MAX\left(W^T C_i + b\right)\right] = 1$' and is positive if '$SIGN\left[MAX\left(W^T C_i + b\right)\right] = -1$'. In the presence of Heterogeneous Occurrences, the heterogeneous function is defined by Barzilai–Borwein and mathematically written as:

$$\gamma_n = \frac{(C_n - C_{n-1})^T [\nabla F(C_n) - \nabla F(C_{n-1})]}{[\nabla F(C_n) - \nabla F(C_{n-1})]^2} \tag{8}$$

where '$F(C_0) \geq F(C_1) \geq F(C_2) \geq \ldots \geq F(C_n)$' converges to a local minimum using iterative step-size updates. A hyperplane was constructed to classify the components as secure (within the production line) or compromised. Although blockchain improves ICPS security, it introduces vulnerabilities, whereas the Barzilai–Borwein method remains a low-cost and efficient optimization approach.

Algorithm 1 uses the Barzilai–Borwein Support Vectorized ICPS to classify components using Smart Contract rules. An iterative hyperplane separates components within and outside the production line. Discarding unsafe components reduces runtime and attack success probability.

Algorithm 1: Barzilai–Borwein support vectorized ICPSs

Input: Components: $C = \{C_1, C_2, \ldots, C_n\}$, Features: $F = \{F_1, F_2, \ldots, F_8\}$, Transactions: $T = \{T_1, T_2, \ldots, T_n\}$
Output: Computationally efficient secured CPS
Steps:
1. **Begin**
2. For each component C with its corresponding features F and transactions T:
3. Form the matrix $CM = \begin{vmatrix} C_1F_1 & C_1F_2 & \ldots & C_1F_8 \\ C_2F_1 & C_2F_2 & \ldots & C_2F_8 \\ \ldots & \ldots & \ldots & \ldots \\ C_nF_1 & C_nF_2 & \ldots & C_nF_8 \end{vmatrix}$
4. Obtain requests as specified in $Req \rightarrow (T_1, T_2, \ldots, T_n)$
5. Perform the binary classification task $D = \{(p_1, q_1), (p_2, q_2), \ldots, (p_n, q_n)\}$
6. Each feature must be related to its component.
7. Components must belong to valid production lines.
8. The procedure is repeated for all 8 features only.
9. *If* $\text{SIGN}[\max(W^T C_i + b)] = 1$, assign a negative label.
10. *If* $\text{SIGN}[\max(W^T C_i + b)] = -1$, assign a positive label.
11. All 8 features are selected upon receipt.
12. CF and CE are acknowledged when components belong to a production line.
13. The procedure is repeated as conventional AI model
14. Evaluate the optimal global solution $q_i = Q_i MAX\left(W^T C_i + b\right)$
15. Record the corresponding component timestamp $Q_i' = SIGN\left[MAX\left(W^T C_i + b\right)\right]$
16. Evaluate the HO function $\gamma_n = \frac{(C_n - C_{n-1})^T [\nabla F(C_n) - \nabla F(C_{n-1})]}{[\nabla F(C_n) - \nabla F(C_{n-1})]^2}$
17. **End For**
18. **End**

4 Experimental Setup

The proposed HBBSV method was experimentally evaluated against four existing approaches: blockchain IIoT [1], blockchain for IIoT [2], an energy-efficient framework [20], and IoT-enabled active distribution networks [27]. All methods were implemented in Java and tested on a Windows 10 system with an Intel Core i3-4130 (3.40 GHz) processor, 4 GB of RAM, and 1 TB of storage.

Experiments used two datasets:

- Condition-Monitoring dataset (50–500 components)
- Versatile Production System dataset with six features and 7729 instances (100–1000 components)

The performance was assessed using run time, probability of attack success, authentication accuracy, authentication time, and security.

The results comparing HBBSV with the four baseline methods are presented in tables and graphs, showing the performance across all evaluation metrics.

4.1 Run Time

The runtime is the time required to identify whether the components are within the production lines and is computed as follows:

$$RT = \sum_{i=1}^{n} C_i * \text{Time}[\gamma_n] \tag{9}$$

where the run time 'RT' is measured in milliseconds (ms). As seen, 'RT' is computed based on the components involved in assessing 'C_i' and the time consumed in deriving the heterogeneous function '$\text{Time}[\gamma_n]$', in which components are aligned for production lines.

Table 2 and Fig. 3 show that the runtime increases as the number of components (50–500) increases. The proposed HBBSV method achieves a lower runtime than blockchain IIoT [1], blockchain for IIoT [2], energy-efficient framework [20], and IoT-enabled ADNs [27]. For 450 components, HBBSV records 160.55 ms, outperforming the baseline (175.35, 193.15, 200.25, and 152.63 ms). By verifying the component features through BC and SC before processing, HBBSV reduces unnecessary computations. Overall, HBBSV lowered the runtime by 12%, 22%, 17%, and 7% compared to the four existing methods.

Table 2: Comparison of runtime (ms) vs components using condition monitoring dataset

Components	**Run time (ms)**				
	Proposed HBBSV	**Blockchain IIoT approach [1]**	**Blockchain for IIoT [2]**	**Energy-efficient framework [20]**	**IoT-enabled active distribution networks [28]**
50	51.25	53.25	55.25	56.25	52.15
100	64.15	83.35	95.15	96.25	75.42
150	73.25	90.15	110.15	115.15	84.55
200	85.95	105.15	125.55	130.25	97.35
250	100.25	115.25	140.35	142.25	108.43
300	113.25	120.15	145.55	150.55	117.62
350	125.85	145.55	170.15	170.15	135.22
400	145.25	160.25	175.15	180.15	170.43
450	160.55	175.35	193.15	200.25	152.63
500	180.35	195.15	205.15	210.55	189.72

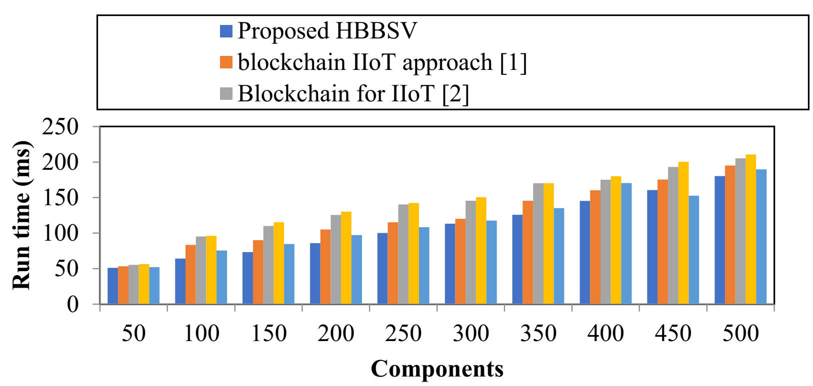

Figure 3: Run Time Measurements for Condition Monitoring Dataset [1,2]

Table 3 and Fig. 4 show that the runtime increases with 100–1000 components, and HBBSV consistently outperforms blockchain IIoT [1], blockchain for IIoT [2], energy-efficient framework [20], and IoT-enabled ADNs [27]. Using BC and SC with six component features, HBBSV achieved significantly lower runtime. Overall, it reduces runtime by 14%, 20%, 26%, and 9% compared to the four existing methods.

Table 3: Versatile production system dataset using comparison of runtime (ms) vs. components

Components	**Run time (ms)**				
	Proposed HBBSV	**Blockchain IIoT approach [1]**	**Blockchain for IIoT [2]**	**Energy-efficient framework [20]**	**IoT-enabled active distribution networks [28]**
100	50.42	52.25	54.15	57.35	51.55
200	61.35	75.34	85.15	76.38	69.42
300	70.63	80.25	90.35	85.55	78.35
400	79.38	91.45	105.75	110.25	82.35
500	85.25	103.25	110.65	126.85	90.33
600	93.75	110.75	125.35	140.45	100.72
700	100.65	125.35	130.45	160.75	120.43
800	115.45	140.52	142.85	170.15	130.84
900	130.82	150.35	153.15	190.45	142.33
1000	150.35	165.15	175.35	200.73	179.12

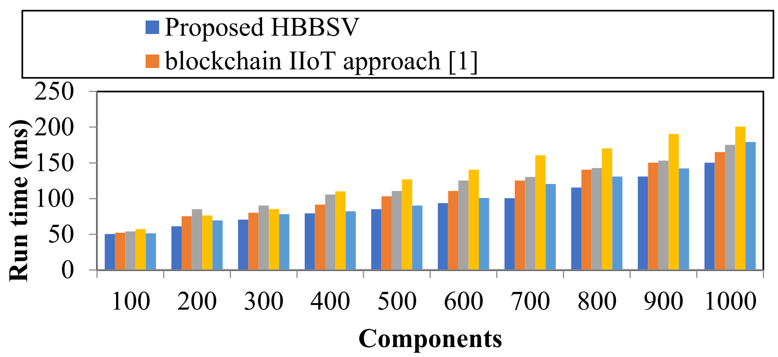

Figure 4: Run time measurements for versatile production system dataset [1]

4.2 Probability of Attack Success

An attack is any malicious attempt to harm or access a system, such as through data theft or denial-of-service. The probability of attack success is the percentage of components whose features are compromised among the total components.

$$PAS = \sum_{i=1}^{n} \frac{Prob_{CC}}{C_i} * 100 \tag{10}$$

where, the probability of attack success 'PAS' is measured in percentage based on the number of components considered for simulations 'C_i' and the probability of components being compromised '$Prob_{CC}$'.

Table 4 and Fig. 5 show the probability of attack success (PAS) for 50–500 components across all methods. With 50 components, HBBSV achieved 6% PAS, which is lower than blockchain IIoT [1] (8%), blockchain for IIoT [2] (12%), energy-efficient framework [20] (13%), and IoT-enabled ADNs [27] (7%). PAS increased as the component count increased, but HBBSV consistently maintained its lowest values.

Table 4: Probability of attack success (%) vs. components using condition monitoring dataset

Components	Probability of attack success (%)				
	Proposed HBBSV	Blockchain IIoT approach [1]	Blockchain for IIoT [2]	Energy-efficient framework [20]	IoT-enabled active distribution networks [28]
50	6	8	12	13	7
100	8	10	13	14	9
150	10	12	14	15	11
200	11	15	16	17	13
250	13	17	20	22	15
300	14	19	23	24	17
350	15	21	24	26	19
400	16	22	25	27	21
450	18	24	28	30	20
500	20	25	30	32	22

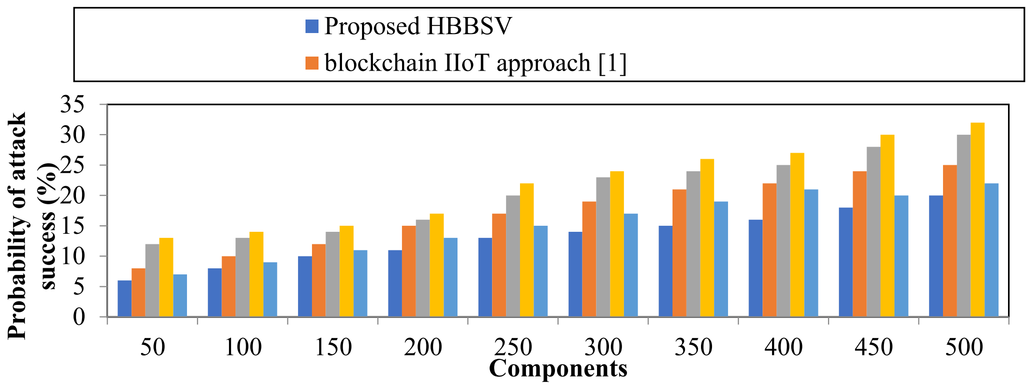

Figure 5: Probability of Attack Success Measurement for Condition Monitoring Dataset [1]

HBBSV performs better because the Barzilai–Borwein Support Vector algorithm separates compromised and uncompromised features through optimal hyperplane classification, thereby eliminating abnormal components before processing. As a result, HBBSV improves PAS by 24%, 37%, 41%, and 14% compared to methods [1,2,20,27], respectively.

Table 5 and Fig. 6 illustrate the measured probability of attack success using Versatile Production System datasets of the four methods. The components are plotted on the horizontal axis and the probability of attack success is plotted on the vertical axis. As shown in the graphical chart, the blue, brown, green, and violet lines indicate the probability of attack success of the proposed HBBSV, existing [1,2,20,27], respectively. AS a result, the proposed HBBSV method improves the probability of attack success by 21%, 34%, 39% and 13% compared to the existing blockchain IoT approach [1], blockchain for IIoT [2], energy-efficient framework [20] and compared to [27], respectively.

Table 5: Versatile production system dataset using probability of attack success (%) vs. components

Components	Probability of attack success (%)				
	Proposed HBBSV	Blockchain IIoT approach [1]	Blockchain for IIoT [2]	Energy-efficient framework [20]	IoT-enabled active distribution networks [28]
100	10	14	17	19	12
200	12	17	21	22	15
300	15	19	25	26	17
400	18	22	27	29	19
500	19	25	30	32	23
600	21	29	32	35	26
700	24	30	35	39	29
800	26	31	36	40	30
900	29	33	39	43	31
1000	32	35	42	45	33

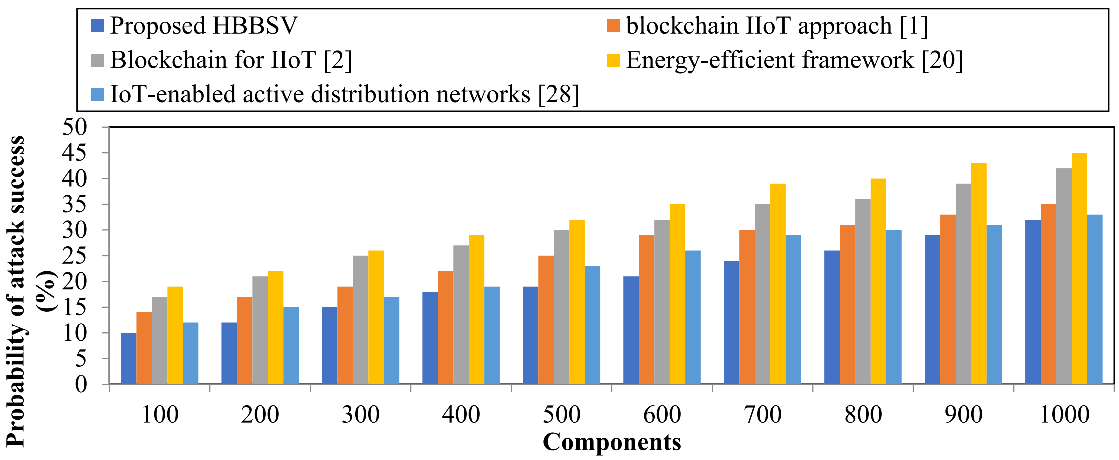

Figure 6: Probability of attack success measurement for versatile production system dataset [1,2,20,28]

4.3 Authentication Accuracy

Authentication accuracy 'AA' is measured as the ratio of the number of component features correctly authorized to the total number of components. AA is expressed as follows:

$$AA = \frac{No.of.\text{component features that are correctly authorized}}{\text{total number of components}} * 100 \tag{11}$$

From below Table 6, represents the comparison of authentication accuracy vs. components.

Table 6 and Fig. 7 show that HBBSV consistently achieves the highest authentication accuracy across 50–500 components, outperforming blockchain IIoT [1], blockchain for IIoT [2], energy-efficient frameworks [20], and IoT-enabled ADNs [27]. For 50 components, these methods achieved 76%, 68%, 70%, and 85% accuracy, respectively, while HBBSV reached 92%.

Table 6: Probability of authentication accuracy (%) vs. components using condition monitoring dataset

Components	Authentication accuracy (%)				
	Proposed HBBSV	Blockchain IIoT approach [1]	Blockchain for IIoT [2]	Energy-efficient framework [20]	IoT-enabled active distribution networks [28]
50	92	76	68	70	85
100	88	74	66	71	79
150	91	81	76	78	86
200	92	84	79	80	89
250	91	83	78	82	90
300	90	81	77	80	91
350	93	87	81	84	89
400	89	85	81	85	85
450	92	88	84	86	89
500	94	90	86	88	91

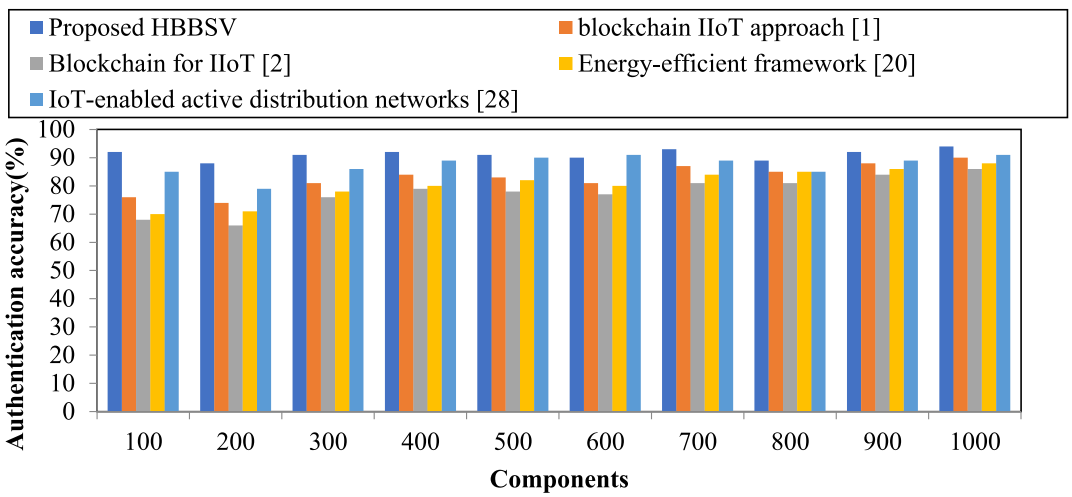

Figure 7: Authentication accuracy measurement for condition monitoring dataset [1,2,20,28]

The Barzilai–Borwein Support Vector algorithm enhances the HBBSV by optimally separating component features through hyperplane classification, retaining valid components, and removing invalid ones. This leads to improved authentication accuracy, with HBBSV outperforming the other four methods by 10%, 18%, 14%, and 4%, respectively.

From Table 7 and Fig. 8, a comparison graph for authentication accuracy with different components was obtained. The authentication accuracy results of the proposed HBBSVmethod were compared with those of the existing blockchain IoT approach [1], blockchain for IIoT [2], energy-efficient framework [20], and IoT-enabled active distribution network [27]. Among the four methods, the proposed HBBSV method showed the greatest ability to increase authentication accuracy. To conduct the experiments, 10 iterations were measured for 100–1000 components. In the first iteration with 100 components, the authentication accuracies of [1,2,20,27] improved by 80%, 78%, 75%, and 90%, respectively. In a greater comparison, the HBBSV method further improved authentication accuracy by 94%. Fig. 7 also reveals that as the number of components increases, the proposed HBBSV method continuously produces better authentication accuracy than the other methods. Consequently, the proposed HBBSV method achieved greater authentication accuracy by 9%, 10%, 2%, and 7% compared with the existing blockchain IoT approach [1], blockchain for IIoT [2], energy-efficient framework [20], and IoT-enabled active distribution network [27], respectively.

Table 7: Versatile production system dataset using probability of authentication accuracy (%) vs. components

Components	**Authentication accuracy (%)**				
	Proposed HBBSV	**Blockchain IIoT approach [1]**	**Blockchain for IIoT [2]**	**Energy-efficient framework [20]**	**IoT-enabled active distribution networks [28]**
100	94	80	78	75	90
200	93	84	73	76	91
300	89	82	80	80	88
400	91	86	82	84	90
500	93	84	86	88	92
600	92	82	88	90	91
700	94	88	90	92	93
800	90	86	89	91	88
900	93	90	88	91	92
1000	95	92	94	93	94

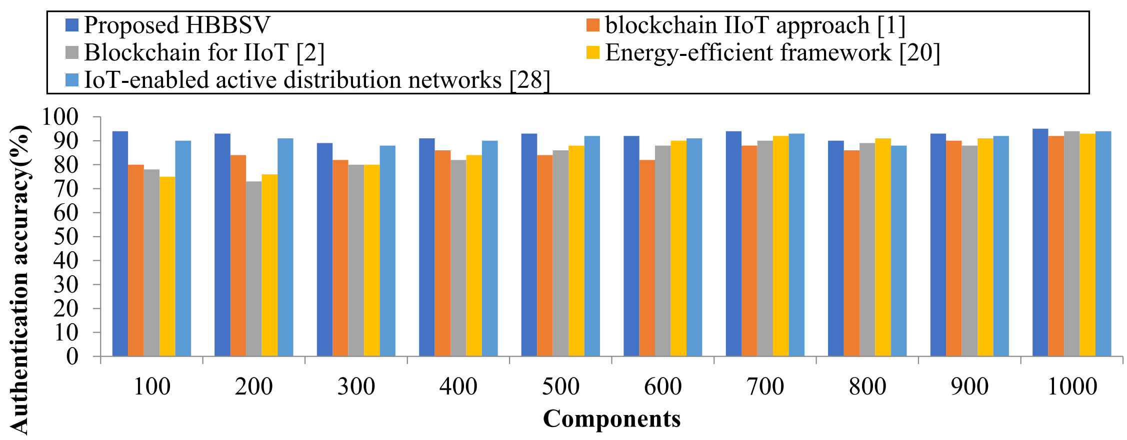

Figure 8: Authentication accuracy measurement for versatile production system dataset [1,2,20,28]

4.4 Authentication Time

Authentication time is defined as the amount of time required to identify the compromised component features. AT is mathematically estimated as follows:

$$AT = C_i * T((identifying\, the\, single\, component)) \tag{12}$$

where AT denotes the authentication time computed in milliseconds (ms); C_i denotes the total number of components, and T is the time taken to identify a single component. Table 8 presents a comparison between the authentication time and components.

Table 8 and Fig. 9 show that with 100 components, HBBSV achieves an authentication time of 20 ms, which is faster than blockchain IIoT [1] (24 ms), blockchain for IIoT [2] (25 ms), the energy-efficient framework [20] (28 ms), and IoT-enabled ADNs [27] (22 ms). Across all runs, HBBSV remained consistently faster because BC and SC validated components directly through CF and CE, ensuring feature consistency and reducing delays. Overall, HBBSV reduces the authentication time by 16%, 26%, 32%, and 8% compared with the four existing methods.

Table 8: Probability of authentication time (ms) vs. components using condition monitoring dataset

Components	Authentication time (ms)				
	Proposed HBBSV	**Blockchain IIoT approach [1]**	**Blockchain for IIoT [2]**	**Energy-efficient framework [20]**	**IoT-enabled active distribution networks [28]**
50	12	22	28	30	18
100	20	24	25	28	22
150	23	26	32	32	24
200	25	30	33	35	28
250	26	30	35	37	29
300	28	32	37	40	29
350	30	35	39	42	33
400	32	36	42	45	34
450	34	37	43	46	35
500	37	39	46	50	36

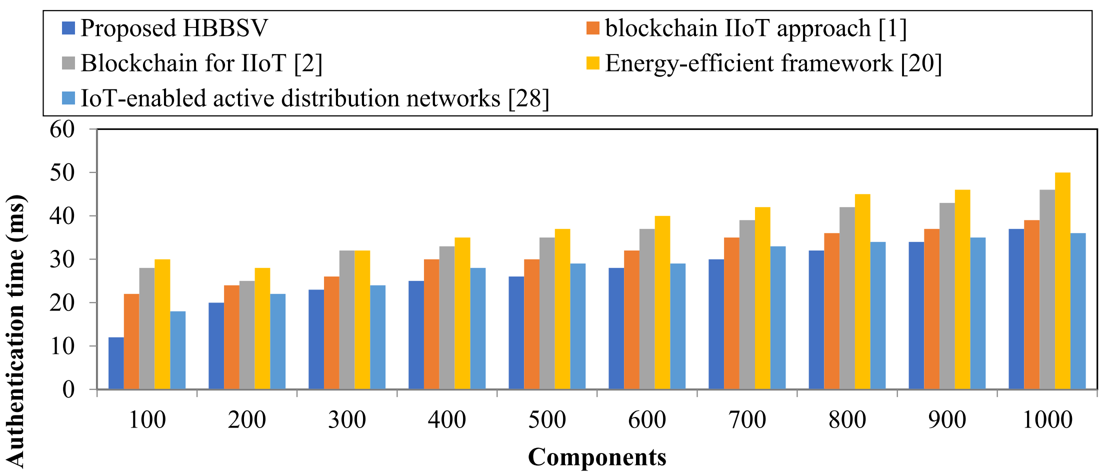

Figure 9: Authentication time measurement for condition monitoring dataset [1,2,20,28]

Table 9 and Fig. 10 present the performance results of authentication time for 100–1000 component features. The components are given on the x-axis and the time taken to identify the component features is represented on the y-axis. Ten different results were obtained for the four techniques, which confirms that the HBBSV method utilizes a smaller AT than the other conventional methods. In the first iteration with 100 components, the time required to detect the component features was observed to be 20, 25, 29 and 15 ms using the existing methods [1,2,20,27], respectively. Comparatively, the proposed HBBSV method required only 10 ms during the first iteration. Finally, the overall authentication time using the proposed HBBSV method was reduced by 21%, 11%, 8%, and 37% compared with that of the existing blockchain IIoT approach [1], blockchain for IIoT [2], and energy-efficient frameworks [20,27], respectively.

Table 9: Versatile production system dataset using probability of authentication time (ms) vs. components

Components	Authentication time (ms)				
	Proposed HBBSV	Blockchain IIoT approach [1]	Blockchain for IIoT [2]	Energy-efficient framework [20]	IoT-enabled active distribution networks [28]
100	10	20	25	29	15
200	15	22	27	27	19
300	20	24	29	31	22
400	23	27	30	34	25
500	25	32	33	36	29
600	27	33	35	39	31
700	29	35	38	41	32
800	31	35	40	44	34
900	33	38	42	45	36
1000	35	40	44	48	39

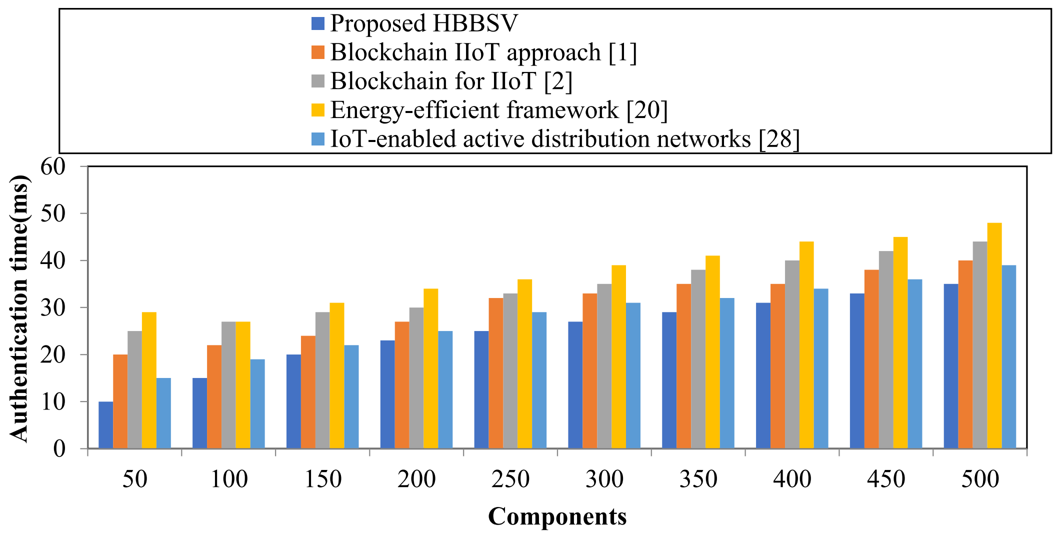

Figure 10: Authentication time (ms) for Versatile Production System dataset [1,2,20,28]

4.5 Security

Security is defined as the ratio of the number of component features compromised by authentic users without any modification to the total number of components and is mathematically expressed as

$$Security = \frac{\begin{array}{c}\text{No.component features being compromised by authentic users}\\ \text{without any modification}\end{array}}{\text{total number of components}} * 100 \quad (13)$$

Table 10 and Fig. 11 indicate that HBBSV provides higher security than the four existing methods across 50–500 components. With 50 components, methods [1,2,20] achieved 50%, 55%, and 53% security, respectively, whereas HBBSV reached 75%. For 100 components, HBBSV again led to 73% security. The Barzilai–Borwein SVM improves security by separating the secured and unsecured component features and eliminating the latter. Overall, HBBSV achieves higher security by 28%, 19%, 24%, and 5% compared with methods [1,2,20,27], respectively.

Table 10: Security (ms) vs. components using condition monitoring dataset

Components	Security (%)				
	Proposed HBBSV	**Blockchain IIoT approach [1]**	**Blockchain for IIoT [2]**	**Energy-efficient framework [20]**	**IoT-enabled active distribution networks [28]**
50	75	50	55	53	65
100	73	48	53	51	69
150	78	57	60	58	72
200	83	64	69	65	79
250	85	66	71	69	80
300	86	75	75	72	83
350	89	79	83	80	85
400	88	76	81	79	87
450	86	73	77	75	84
500	89	77	82	80	88

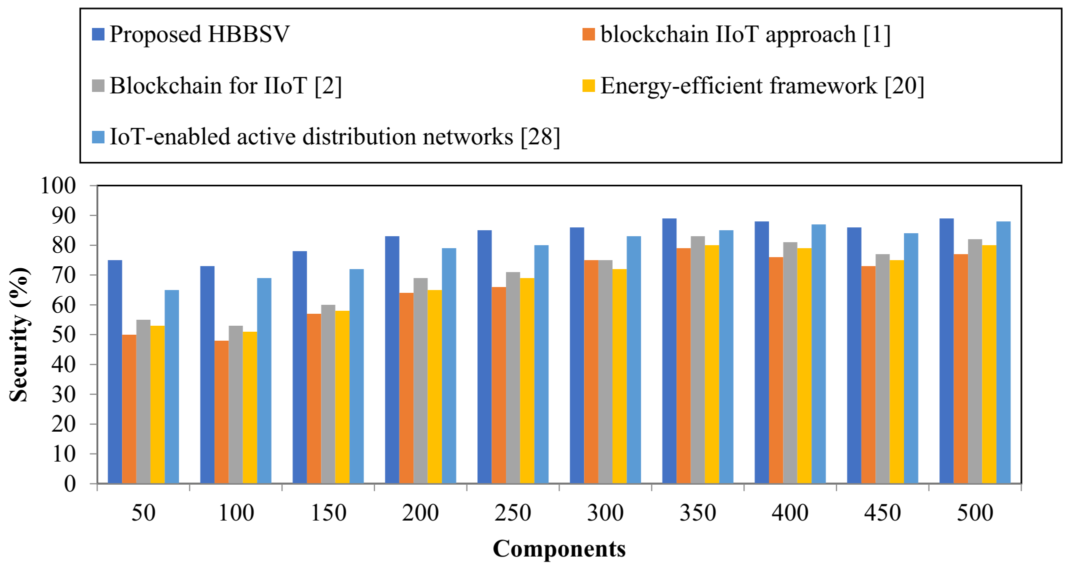

Figure 11: Security measurements for condition monitoring dataset [1,2,20,28]

Table 11 and Fig. 12 show that HBBSV consistently provides higher security than all the four existing methods for 100–1000 components. For example, with 100 components, the baselines achieved 49%–60% security, whereas HBBSV reached 70%, and similar gains were observed across all iterations. By removing unsecured features and resisting DoS, Eclipse, and Sybil attacks, or significance tests.

Table 11: Versatile production system dataset using security (ms) vs. components

Components	Security (%)				
	Proposed HBBSV	Blockchain IIoT approach [1]	Blockchain for IIoT [2]	Energy-efficient framework [20]	IoT-enabled active distribution networks [28]
100	70	49	50	52	60
200	71	47	51	50	65
300	75	54	58	55	70
400	80	60	64	63	76
500	83	63	69	66	79
600	86	69	73	70	82
700	87	75	78	75	84
800	87	76	82	78	86
900	85	73	75	74	85
1000	88	75	80	79	87

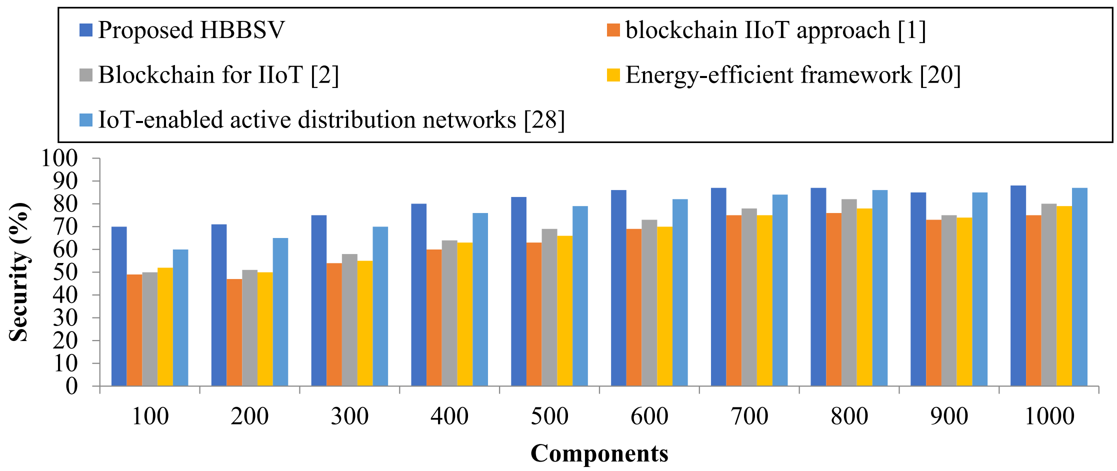

Figure 12: Security measurements for versatile production system dataset [1,2,20,28]

4.6 *Confidence Intervals*

Confidence intervals strengthen the results by showing a range of plausible values for an estimate, providing more insight than a simple yes/no significance test. They quantify precision and uncertainty, indicating the reliability of the findings and whether the effect is practically meaningful. While significance tests show that results are unlikely due to chance, confidence intervals reveal the range in which the true value likely lies.

5 Conclusion

This work proposes an AI-enabled smart-contract model, the Heterogeneous Barzilai–Borwein Support Vector (HBBSV), to secure ICPSs by reducing runtime and attack success probability. HBBSV uses blockchain and smart contracts to validate component conditions, and applies the Barzilai–Borwein SVM algorithm to detect abnormal features and enforce security rules. Its performance—evaluated through runtime, probability of attack success, authentication accuracy, authentication time, and security—shows significant improvements: with the condition-monitoring dataset, HBBSV achieved higher authentication accuracy (+14%), stronger security (+24%), lower attack success (−11%), reduced runtime (−17%), and shorter authentication time (−27%); with the second dataset, gains included +7% accuracy, +13% security, −27% attack success, −17% runtime, and −19% authentication time. Limitations include communication-based vulnerabilities, lack of unified safety–security analysis, high system complexity, smart-contract immutability, reliance on oracles, and limited legal recognition. Future work will integrate advanced AI-based smart contracts with IoT and blockchain to further enhance security and authentication performance.

Acknowledgement: Not applicable.

Funding Statement: Not applicable.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Gowrishankar Jayaraman, Ashok Kumar Munnangi; data collection: Manikandan Ramachandran; analysis and interpretation of results: Ramesh Sekaran, Arunkumar Gopu; draft manuscript preparation: Gowrishankar Jayaraman, Ashok Kumar Munnangi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Rathee G, Balasaraswathi M, Chandran KP, Gupta SD, Boopathi CS. A secure IoT sensors communication in industry 4.0 using blockchain technology. J Ambient Intell Humaniz Comput. 2021;12(1):533–45. doi:10.1007/s12652-020-02017-8.
2. Bai L, Hu M, Liu M, Wang J. BPIIoT: a light-weighted blockchain-based platform for industrial IoT. IEEE Access. 2019;7:58381–93. doi:10.1109/ACCESS.2019.2914223.
3. Alladi T, Chamola V, Parizi RM, Choo KR. Blockchain applications for industry 4.0 and industrial IoT: a review. IEEE Access. 2019;7:176935–51. doi:10.1109/ACCESS.2019.2956748.
4. Zhang X, Chen X, Liu JK, Xiang Y. DeepPAR and DeepDPA: privacy preserving and asynchronous deep learning for industrial IoT. IEEE Trans Ind Inform. 2020;16(3):2081–90. doi:10.1109/TII.2019.2941244.
5. Colombo AW, Karnouskos S, Kaynak O, Shi Y, Yin S. Industrial cyberphysical systems: a backbone of the fourth industrial revolution. IEEE Ind Electron Mag. 2017;11(1):6–16. doi:10.1109/mie.2017.2648857.
6. Huang K, Zhou C, Tian YC, Yang S, Qin Y. Assessing the physical impact of cyberattacks on industrial cyber-physical systems. IEEE Trans Ind Electron. 2018;65(10):8153–62. doi:10.1109/TIE.2018.2798605.
7. Casino F, Dasaklis TK, Patsakis C. A systematic literature review of blockchain-based applications: current status, classification and open issues. Telemat Inform. 2019;36:55–81. doi:10.1016/j.tele.2018.11.006.
8. Bernal Bernabe J, Canovas JL, Hernandez-Ramos JL, Torres Moreno R, Skarmeta A. Privacy-preserving solutions for blockchain: review and challenges. IEEE Access. 2019;7:164908–40. doi:10.1109/ACCESS.2019.2950872.
9. Reyna A, Martín C, Chen J, Soler E, Díaz M. On blockchain and its integration with IoT. Challenges and opportunities. Future Gener Comput Syst. 2018;88(3):173–90. doi:10.1016/j.future.2018.05.046.
10. Fernández-Caramés TM, Fraga-Lamas P. A review on the use of blockchain for the Internet of Things. IEEE Access. 2018;6:32979–3001. doi:10.1109/ACCESS.2018.2842685.
11. Leitão P, Colombo AW, Karnouskos S. Industrial automation based on cyber-physical systems technologies: prototype implementations and challenges. Comput Ind. 2016;81:11–25. doi:10.1016/j.compind.2015.08.004.
12. Jiang Y, Yin S, Kaynak O. Data-driven monitoring and safety control of industrial cyber-physical systems: basics and beyond. IEEE Access. 2018;6:47374–84. doi:10.1109/ACCESS.2018.2866403.
13. Roy C, Misra S, Pal S. Blockchain-enabled safety-as-a-service for industrial IoT applications. IEEE Internet Things Mag. 2020;3(2):19–23. doi:10.1109/IOTM.0001.1900080.

14. Banerjee M, Lee J, Choo KR. A blockchain future for Internet of Things security: a position paper. Digit Commun Netw. 2018;4(3):149–60. doi:10.1016/j.dcan.2017.10.006.
15. Denis A, Thomas A, Robert W, Samuel A, Kabiito SP, Morish Z, et al. A survey on artificial intelligence and blockchain applications in cybersecurity for smart cities. SHIFRA. 2025;2025:1–45. doi:10.70470/shifra/2025/001.
16. Zhao S, Li S, Yao Y. Blockchain enabled industrial Internet of Things technology. IEEE Trans Comput Soc Syst. 2019;6(6):1442–53. doi:10.1109/TCSS.2019.2924054.
17. Westerkamp M, Victor F, Küpper A. Tracing manufacturing processes using blockchain-based token compositions. Digit Commun Netw. 2020;6(2):167–76. doi:10.1016/j.dcan.2019.01.007.
18. Abed Mohammed M, Lakhan A, Zebari DA, Ghani MKA, Marhoon HA, Abdulkareem KH, et al. Securing healthcare data in industrial cyber-physical systems using combining deep learning and blockchain technology. Eng Appl Artif Intell. 2024;129:107612. doi:10.1016/j.engappai.2023.107612.
19. Production plant data for condition monitoring. [cited 2018 Sep 19]. Available from: https://www.kaggle.com/datasets/inIT-OWL/production-plant-data-for-condition-monitoring.
20. Song W, Zhu X, Ren S, Tan W, Peng Y. A hybrid blockchain and machine learning approach for intrusion detection system in industrial Internet of Things. Alex Eng J. 2025;127(1):619–27. doi:10.1016/j.aej.2025.05.030.
21. Leng J, Zhou M, Zhao JL, Huang Y, Bian Y. Blockchain security: a survey of techniques and research directions. IEEE Trans Serv Comput. 2022;15(4):2490–510. doi:10.1109/TSC.2020.3038641.
22. Sghaier R, El Hog C, Ben Djemaa R, Sliman L. Machine learning and blockchain synergy: opportunities and challenges for ML models and smart contracts. Blockchain Res Appl. 2025;2025:100411. doi:10.1016/j.bcra.2025.100411.
23. Leng J, Ruan G, Jiang P, Xu K, Liu Q, Zhou X, et al. Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey. Renew Sustain Energy Rev. 2020;132:110112. doi:10.1016/j.rser.2020.110112.
24. Leng J, Zhang H, Yan D, Liu Q, Chen X, Zhang D. Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop. J Ambient Intell Humaniz Comput. 2019;10(3):1155–66. doi:10.1007/s12652-018-0881-5.
25. Mohanta BK, Awad AI, Elsaka T, Kheddar H, Baraka E. Smart-contract-based blockchain-enabled decentralized scheme for improving smart-grid security. Internet Things. 2025;34:101811. doi:10.1016/j.iot.2025.101811.
26. Essaid M, Ju H. Blockchain solutions for enhancing security and privacy in industrial IoT. Appl Sci. 2025;15(12):6835. doi:10.3390/app15126835.
27. Li Z, Shahidehpour M, Liu X. Cyber-secure decentralized energy management for IoT-enabled active distribution networks. J Mod Power Syst Clean Energy. 2018;6(5):900–17. doi:10.1007/s40565-018-0425-1.
28. Patil SM, Dakhare BS, Satre SM, Pawar SD. Blockchain-based privacy preservation framework for preventing cyberattacks in smart healthcare big data management systems. Multimed Tools Appl. 2025;84(22):25547–66. doi:10.1007/s11042-024-20109-x.
29. Akbar M, Waseem MM, Mehanoor SH, Barmavatu P. Blockchain-based cyber-security trust model with multi-risk protection scheme for secure data transmission in cloud computing. Clust Comput. 2024;27(7):9091–105. doi:10.1007/s10586-024-04481-9.
30. Gulzar Q, Mustafa K. Enhancing network security in industrial IoT environments: a DeepCLG hybrid learning model for cyberattack detection. Int J Mach Learn Cybern. 2025;16(7):4797–815. doi:10.1007/s13042-025-02544-w.
31. Kumar P, Kumar R, Gupta GP, Tripathi R. A distributed framework for detecting DDoS attacks in smart contract-based blockchain-IoT systems by leveraging fog computing. Trans Emerg Telecommun Technol. 2021;32(6):e4112. doi:10.1002/ett.4112.
32. Feng Z, Li Y, Ma X. Blockchain-oriented approach for detecting cyber-attack transactions. Financ Innov. 2023;9(1):81. doi:10.1186/s40854-023-00490-6.
33. Sathyabama AR, Katiravan J. Enhancing anomaly detection and prevention in Internet of Things (IoT) using deep neural networks and blockchain based cyber security. Sci Rep. 2025;15(1):22369. doi:10.1038/s41598-025-04164-4.

34. Ali Laghari A, Khan AA, Ksibi A, Hajjej F, Kryvinska N, Almadhor A, et al. A novel and secure artificial intelligence enabled zero trust intrusion detection in industrial Internet of Things architecture. Sci Rep. 2025;15(1):26843. doi:10.1038/s41598-025-11738-9.
35. Alevizos L. Automated cybersecurity compliance and threat response using AI, blockchain and smart contracts. Int J Inf Technol. 2025;17(2):767–81. doi:10.1007/s41870-024-02324-9.
36. Dwivedi SK, Amin R, Vollala S. Smart contract and IPFS-based trustworthy secure data storage and device authentication scheme in fog computing environment. Peer-Peer Netw Appl. 2023;16(1):1–21. doi:10.1007/s12083-022-01376-7.
37. Chowdhury A, Shafin SS, Masum S, Kamruzzaman J, Dong S. Secure electric vehicle charging infrastructure in smart cities: a blockchain-based smart contract approach. Smart Cities. 2025;8(1):33. doi:10.3390/smartcities8010033.