Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

SENTRY: Secure Energy-Efficient Network Threat Response and Yield in WSN-IoT Environment

Nandini Prasad K S
Dayananda Sagar Academy of Technology and Management
Chaya Puttaswamy
GSSS Institute of Engineering and Technology for Women
Pavani Cherukuru
Dayananda Sagar Academy of Technology and Management
Mumtaz Begum Mustafa
mumtaz@um.edu.my
University of Malaya
Suma V
Dayananda Sagar College of Engineering
Divyashree H B
Dayananda Sagar University

Article

Keywords: Machine Learning, Deep Learning, Cyberthreats, Wireless Sensor Network, Internet-of-Things, Blockchain, Energy

Posted Date: September 22nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10749747/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

SENTRY: Secure Energy-Efficient Network Threat Response and Yield in WSN-IoT Environment

Nandini Prasad K S[1], Chaya Puttaswamy[2], Pavani Cherukuru[3*], Mumtaz Begum Mustafa[4*], Suma V[5], Divyashree H B[6]

[1] Professor, Department of Information Science & Engineering,
Dayananda Sagar Academy of Technology and Management, Bangalore, India
hodise@dsatm.edu.in

[2]Associate Professor, Department of Information Science & Engineering,
GSSS Institute of Engineering and Technology for Women, Mysore, India
chaya@gsss.edu.in

[3]Associate Professor, Department of Information Science & Engineering,
Dayananda Sagar Academy of Technology and Management, Bangalore, India
pavani-ise@dsatm.edu.in

[4] Department of Software Engineering,
Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur, Malaysia;
mumtaz@um.edu.my

[5]Professor, Department of Computer Science and Design, Dayananda Sagar College of Engineering, Bengaluru, India; hod-csd@dayanandasagar.edu

[6]Assistant Professor, Department of Electronics and Communication Engineering,
Dayananda Sagar University, Bangalore, India
divyashree-ece@dsu.edu.in

[*]**Corresponding Author:** mumtaz@um.edu.my, pavani-ise@dsatm.edu.in

Abstract. Being an integral part of Internet-of-Thing (IoT), a Wireless Sensor Network (WSN) has already introduced various potential applications e.g. environmental monitoring, industrial automation, smart agriculture. The integrated WSN-IoT environment is also characterized by various lethal cyber-threats and vulnerabilities that are still yet to be explored with optimal solutions. Reviews of literature states some promising and dedicated attempts using different variants of methodologies and yet there are gaps. It is also noted that existing security solutions has either considered WSN or used IoT but not integrating both to identify and address security challenges. Therefore, this paper presents Secure Energy-efficient Network Threat Response and Yield (SENTRY) that emphasizes identification and prevention of potential cyberthreats exclusively for WSN-IoT environment. SENTRY uses a backbone of Ethereum blockchain where sensory data are securely stored towards decentralized data integrity while it also uses hybrid Artificial Intelligence (AI)-driven novel approach to catalyze security strength. In its AI-approach, the threats pertaining to data privacy and private key compromise are detected using Machine Learning that also involves a novel severity ranking mechanism. SENTRY uses Deep Learning for amending vulnerabilities associated with smart contract followed by eliminating off-chain threats. The simulated study outcome shows SENTRY to offer 96.8% detection accuracy, 1.4s latency, 49MB memory consumption,

225 events per second as throughput, 4.2J as consumed energy, which are optimal score in contrast to existing state-of-art methods.

Keywords: Machine Learning, Deep Learning, Cyberthreats, Wireless Sensor Network, Internet-of-Things, Blockchain, Energy

1 Introduction

There is a very close connection between Wireless Sensor Network (WSN) and Internet-of-Things (IoT) while both plays a strong role in today's digital technology [1]. A WSN is defined as a network that is structurally formed from spatially distributed sensor nodes either using single hop or using multi-hop. The primary agenda of WSN is towards monitoring the environmental attributes such as pressure, motion, humidity, temperature, etc. followed by forwarding them to sink node. Similarly, IoT can be defined as large, interconnected network of various ranges of physical objects that are further embedded with connectivity, software, and sensor towards exchanging information over internet. It is to be noted that WSN is one of the integral and foundation units of IoT by facilitating capabilities of sensing and aggregation sensory data that are critical for decision making. The prime emphasis on WSN is mainly in sensing followed by local communication while IoT is meant to integrate all these sensory networks to form globally a larger system connected with internet. This massive system offers automation, remote control, and real-time monitoring over a wider variety of wireless plication e.g. industrial automation, agriculture, smart homes, etc [2]. In order to integrate WSN into IoT, it is necessary to use various technologies e.g. cloud platforms, gateways, IoT protocol, WSN protocols. Fig.1 highlights the integrated WSN-IoT environment.

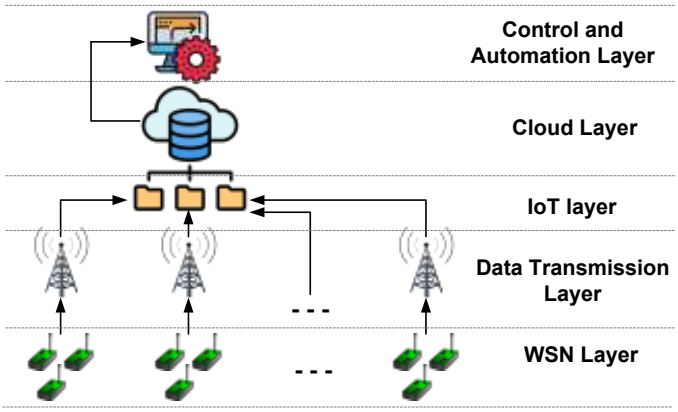


Fig. 1. Integrated WSN-IoT Environment [2]

Fig.1 shows multi-layer environment where the *WSN layer* facilitates the sensors to gather sensory data while the sensory data are then forwarded to next *layer of data transmission* where the aggregated data is forwarded to sink/gateway node. In next *IoT layer*, gateway integration is carried out where WSN is integrated to IoT for various processing (collection, filtering, translation of protocol) followed by forwarding the data to cloud. The *cloud layer*, the aggregated

sensory data is stored and analyzed which is further passed on to *control and automation layer* where suitable response system is designed and activated. Hence, it can be noted that WSN is an integral part of IoT.

Although, there are various types of research challenges encountered by both WSN [3]-[6] and IoT [7]-[11], but security problem has exponentially increased with more connected networks with WSN integrated IoT environment. The prime reason is associated with restricted capabilities of sensor nodes where it is never possible to implement authentication protocol, intrusion detection, encryption, etc. The heterogeneous and complex architecture of WSN-IoT with multiple layers further increases the surface of attacks where end-to-end security facilitation is heavily challenging. Encryption methods are known to offer extensive confidentiality and integrity for both WSN and IoT [12]-[14]; however, the prime hurdle in its implementation is the restricted battery life and computing power of sensors while complex key management makes it furthermore memory and resource drainage. The recently evolving blockchain technology is known for its secure peer-to-peer data exchange with decentralized trust [15]-[18]; however, they significantly demand extensively high computation power with increased latency and degraded network synchronization. Finally, Artificial Intelligence (AI)-based schemes using Machine Learning (ML) and Deep Learning (DL) can proactively improve security features even to complex ones too; however, they have privacy concerns, training issues, and dataset dependencies. Although, all these possess promising future towards security, existing studies are bit scattered and subjective while they do not emphasize WSN integrated in IoT environment while planning their security solutions.

The aim of the proposed study is to present an intelligent framework known as Secure Energy-efficient Network Threat Response and Yield (SENTRY) that is meant for securing operational processes as well as data in a collaborative framework of WSN and IoT. This goal of the study is accomplished by integrating blockchain with AI-driven approaches. The contributions of the study are as follows:

a. The study presents a hybrid security architecture that facilitates end-to-end security by joint operation of ML, DL, and Ethereum blockchain.
b. SENTRY offers effective and highly proactive threat detection using ML model along with a novel mechanism to rank severity of novel threats in order to profile the attackers.
c. The vulnerabilities associated with smart contract are rigorously addressed by SENTRY using DL model along with significant controlling of off-chain threats.
d. SENTRY is capable of identifying and preventing complex form of threats with unseen and dynamic behaviour.
e. The proposed model of SENTRY also offers higher resource efficiency in terms of energy and memory consumption which is designed exclusively for low-powered sensors and IoT devices.
f. A novel self-healing characteristics towards smart contract dynamically has been induced by SENTRY thereby offering potential adaptability in real-time WSN-IoT environment.

The structure of the manuscript is as follows: Section 2 presents discussion of recent literatures followed by adopted research methodology in Section 3. Section 4 presents highly elaborated accomplished result discussion while conclusion of this paper is given in Section 5.

2 Literature Review

At present there are various types of methods implemented towards securing both WSN as well as IoT environment right in perspective of communication level and device level. Certain studies are meant to focus only on resist current events related to threats while emerging strategies are also surfacing to offer proactive security measures. This section will present discussion of some recent state-of-art methods towards securing sensor network deployed in IoT and withdraw the effectiveness of approaches that act as a base to understand the strength and novelty of proposed model of SENTRY.

2.1 Signature-based Threat Detection

Adoption of signatures is another frequent exercise towards securing WSN and IoT environment where the device behavior or incoming traffic is compared against the predefined patterns of known attacks. Zhang et al. [19] have used identity-based signature adopting post quantum strategy for securing IoT ecosystem. The authors have used lattice cryptography towards data security. Similar adoption of lattice encryption is also used by Prajapat et al. [20] and Zhang et al. [21] which is further integrated with certificateless public key cryptosystem for resisting quantum attacks and collusion attacks respectively in IoT. Adoption of certificateless signature is witnessed in work of Shim [22] for leveraging cryptanalysis under hardness assumption in vulnerable IoT. Ahmed and Barukab [23] have embedded digital signature within elliptical curve encryption for solving weak bit issue where data propagation is secured further using hashing. Khan et al. [24] have used Lenstra-Lenstra-Lovasz (LLL) model using elliptical curve signature and lattice encryption for resisting quantum attack in IoT model. Nouma and Yavuz [25] have adopted dual signature scheme for hardware security against malware attacks in IoT improvising forward signature schemes. Thankappan et al. [26] have developed signatures scheme towards resisting Man-in-the-Middle attack

2.2 AI-based Countermeasures

There are various AI-based schemes especially used for ML and DL models towards resisting potential threats reported for sensory networks in IoT. This section will further brief on some recent work associated with this.

2.2.1 ML based Threat Detection

The prime reason for adoption of ML models in IoT is because of their proven track records for problem solving and simplified computational modelling supportive of network environment. The work presented by Nallakaruppan et al. [27] has used ML algorithms for classifying different types of attacks in IoT while it uses recommendation system adopting fuzzy logic. The recent work carried out by Ismail et al. [28] has furnished enriched information about varied forms of ML models tested on multiple benchmarked datasets towards guarding their security strength. Zachos et al. [29] have presented a framework for anomaly detection using multiple

ML models assessed on hardware to find its low computational cost and potential detection performance. Saleh et al. [30] have used stochastic gradient descent and gaussian naive bayes towards detection of cyberattacks. Eren et al. [31] have developed a model supportive of intrusion detection system where it is hypothesized that smarter preprocessing as well as feature engineering leads to effective ML modelling in IoT.

2.2.2 DL based Threat Detection

DL models are witnessed to be adopted in the situation that demands complex modelling of threat or solving highly challenging issues of dynamic nature. The recent implementation of Popoola et al. [32] have used deep learning with multi-stage strategy in industrial IoT with specific emphasis on imbalance problems of dataset. Berguiga et al. [33] have used Convolution Neural Network (CNN) and Long Short-Term Memory (LSTM) for facilitating feature extraction and prediction of sequence data in Internet-of-Medical Things (IoMT) implemented on hardware and tested on dual standard dataset. Adoption of Federated Learning (FL) towards securing edge devices in IoT is reported in work of Bhavsar et al. [34] developed on embedded system for resisting adversaries in transportation in IoT. Bouzeraib et al. [35] have used Generative Adversarial Network (GAN), Genetic Algorithm, and FL for modelling a threat detection in IoT in presence of multiple benchmarked datasets. The study reports 99% accuracy when real datasets are integrated with generated ones.

2.2.3 Hybrid Approaches for Threat Detection

Both ML and DL approach reports of certain accomplishments as well as certain limitations too. Hence, hybrid methods deploying both ML and DL targets to overcome limitations of each other. Shahid et al. [36] have used both ML and DL for securing standard data transmission protocol in IoT from multiple specific forms of attacks. The outcome suggests Random Forest as ML model and Transformer as DL model to show superior performance of detection accuracy. Srinivasan and Senthilkumar [37] have developed an intrusion detection and prevention system where CNN is used for pattern recognition while reinforcement learning is used for decision making. Duraibi and Alashjaee [38] have integrated LSTM model with mayfly optimization towards classifying disruptive cyberthreats in IoT. Nearly similar trend of work is also witnessed in study of Aburasain [39] where hybrid DL model is integrated with black widow optimization to protect agricultural data from cyberthreats.

2.3 Blockchain-based Threat Prevention

There are recent studies to demonstrate the frequent adoption of standalone usage of blockchain towards securing IoT and WSN without much depending on external infrastructures [40]-[49]. These approaches make use of either lightweight or permission consensus methods towards ensuring decentralized trust and immutability. The recent approaches presents models where data validation, device identity, access control-based issues are handled without cloud intermediaries. However, such approaches do have trade-offs where latency and computation overhead is witnessed even if they offer automated auditing and proactive defenses on resource-limited sensors or IoT devices. Blockchain has been also integrated with ML [50]-[52]. Further,

blockchain technology has been also integrated with DL model for detection vulnerabilities associated with smart contract [53][54] securing home network [55][56], detection of malicious nodes [57].

2.4 Gap Identification

Studies related to signature-based methodologies have a shortcoming of limited adaptability against potentially emerging cyberthreats. There is no denying the fact that better quantum resistance is offered by certificateless signature and lattice cryptography; however, they are highly complex and unfit for resource-restricted hardware of either WSN or IoT. AI-driven methodologies have memory constraint as well as latency issues (e.g. CNN, LSTM, etc.) while blockchain methods have similar problems. Further, there is no evidence of any proactive and highly adaptive security framework with equal emphasis on resource, security, and computational performance. After reviewing the existing trends of security solutions for IoT and sensory network, it is noted that there are certain open-ended research gaps viz.

- recent study approaches don't seem to consider a collaborative approach consisting of identification of an anomalies and stringent checks towards smart contract,
- almost extensive study models of recent times are found to be less capable towards adaptively handling attacks equivalent to zero-day,
- none of the existing studies reports of any features related to self-amendment towards smart contract dynamically in order to autonomously and cost-effectively recover from adverse influence of cyberthreats,
- existing AI-driven designs have reported of higher accuracy but there is no potential evidence towards its applicability on resource-constrained sensor nodes.

It can be also noted that there is no report of any studies that have considered WSN-IoT environment. The proposed system of SENTRY mitigates the above identified gaps by undertaking following steps: i) the study presents a dual-stage architecture where anomaly detection is carried out by ML model while vulnerability analysis of smart contract is carried out by DL model, ii) SENTRY presents a design with resource optimization with computed energy scores for boosting the routines of DL selectively followed by reducing usage of memory and energy, unlike any reported approaches in literature, iii) unlike existing methods reported in literature that is highly subjective towards specific form of threats, SENTRY offers more generalization of threat mitigation by presenting adaptive learning strategy. The next section presents discussion of proposed research methodology.

3 Research Method

The proposed study presents a novel computational model known as Secure Energy-efficient Network Threat Response and Yield (SENTRY) which is meant for protecting data exchanged between WSN and IoT environment. Fig.2 states the core architecture of SENTRY which emphasizes on detecting smart contract defects and performs amendments to make it more secure. SENTRY combines the immutable and decentralized data storage characteristic of Ethereum

and ML module towards securing issues pertaining to private key compromise and privacy violations.

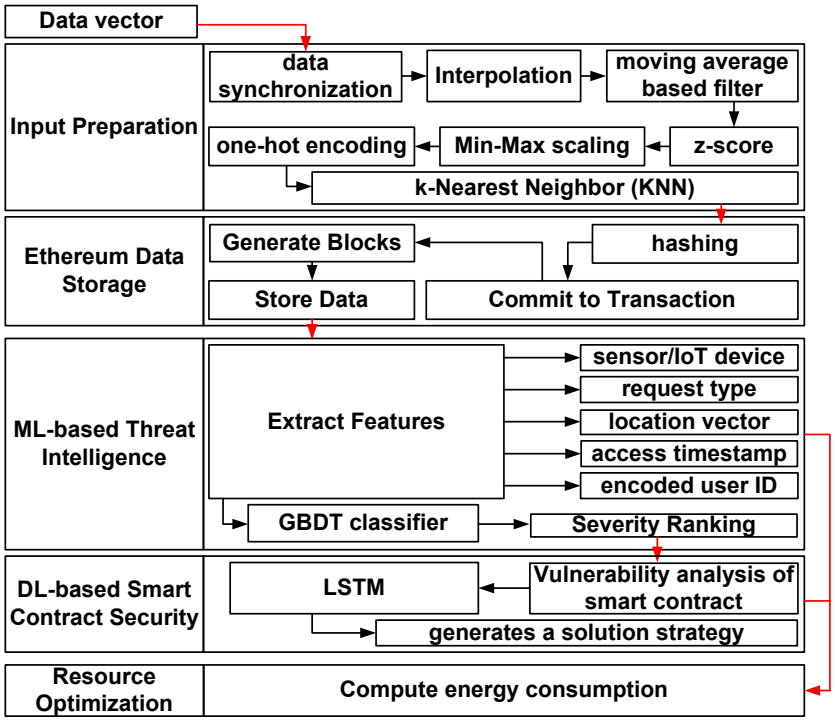


Fig. 2. SENTRY Architecture

The vulnerabilities associated with smart contract are addressed using DL component towards addressing threats imposed by dependencies of off-chain data storage in WSN-IoT environment. The design implementation of complete architecture is carried out targeting towards minimizing memory consumption and energy consumption without sacrificing system throughput and threat detection accuracy. SENTRY is meant for essentially performing profiling of potential threats (both known and unknown) via intelligent ranking system followed by autonomously updating the defective smart contracts. The inclusion of scalable and robust design of SENTRY makes it appropriate for real-time deployment in latency-sensitive and highly distributed WSN-IoT environment that is shrouded with vulnerabilities of cyberthreats.

Threat Model:

The SENTRY framework ensures protection for data communication and smart contract execution in a collaborative WSN-IoT environment. Before discussing the architecture of the system, it is important to define the threat model, capabilities of the attacker, trust assumptions, and the scope of the security to be implemented in the proposed framework.

In the proposed framework, the attacker is computationally strong enough to perform attacks on the WSN-IoT system from external and internal environments. The capabilities that the attacker may have include:

- Eavesdropping on the wireless communication channels to capture the transmitted data from sensors.
- Attack on the sensors' nodes using malware or firmware.
- Replay attack, spoofing attack, denial of service attack and data injection attack.
- Malicious transactions submitted to take advantage of any possible vulnerability in smart contracts.
- Accessing any system component without any authorization using private keys and/or credentials.
- Attacking off-chain storages that are linked to the blockchain-based applications.

But, in the proposed framework, the attacker does not have enough computational power to break the cryptographic algorithms used in Ethereum blockchain including SHA-256 and ECDSA (Elliptic Curve Digital Signature Algorithm).

The SENTRY protocol makes the following *trust assumptions*:

- Gateway nodes that aggregate sensor data are considered to be trusted nodes at the beginning.
- Ethereum validator nodes make the honest majority assumption and provide correct blockchain consensus.
- Cryptographic algorithms used for hashing and digital signatures are considered to be computationally secure.
- Sensor nodes can become compromised once deployed and therefore cannot always be considered to be trusted.

These assumptions help SENTRY to concentrate on cyber-threat detection at runtime rather than on secure model training or blockchain consensus attacks.

SENTRY accounts for both *insider and outsider attacks*.

The *outsider attacks* can be conducted by unauthorized parties trying to penetrate the network using packet injection, replay attack, denial of service attack, credential guessing, and/or manipulation of blockchain transactions.

The *insider attacks* arise when there are authorized devices that have become compromised, malicious administrators, or legitimate users whose credentials have been compromised. These insiders may try to tamper with sensor readings, expose sensitive data, exploit smart contracts, and misappropriate private keys.

In contrast to conventional security techniques which focus on outsider attacks, SENTRY utilizes a hybrid approach based on Machine Learning and Deep Learning to detect any unusual behavior pattern regardless of the source of the attack. The Machine Learning component

detects any access anomaly and determines the threat level. The Deep Learning component examines smart contract weaknesses and provides automated suggestions for fixing any vulnerabilities in the contracts.
The proposed method takes into account the following threats: Unauthorized access, Compromise of private key, Data injection attack, Replay attack, Denial-of-Service attack, Smart contract flaw, Tampering of off-chain data, Violation of privacy, and Malicious transaction attempts.

Proposed Algorithm: SENTRY method of threat detection

```
Input: data_vector, smart_contract_code
Output: threat_ranking, contract_solution, energy_usage
BEGIN
  # Step 1: Prepare Input Data
  data_vector ← clean_and_synchronize(data_vector)
  data_vector ← fill_missing_values(data_vector)
  data_vector ← smooth_data(data_vector)  # e.g., using moving average
  data_vector ← encode_categorical_values(data_vector)
  data_vector ← normalize_data(data_vector)

  # Step 2: Store Data on Blockchain
  block ← create_block(data_vector)
  hashed_block ← hash(block)
  store_on_blockchain(hashed_block)
  commit_transaction(hashed_block)

  # Step 3: Threat Detection using Machine Learning
  features ← extract_security_features(data_vector)
  threat_score ← predict_threat_score(features)

  if threat_score > high_threshold:
    threat_ranking ← "High"
  else if threat_score > medium_threshold:
    threat_ranking ← "Medium"
  else:
    threat_ranking ← "Low"
  # Step 4: Smart Contract Security using Deep Learning
  vulnerabilities ← analyze_contract_with_LSTM(smart_contract_code)
  if vulnerabilities found:
    contract_solution ← generate_security_solution(smart_contract_code)
```

```
    else:
        contract_solution ← "No vulnerabilities found"
    # Step 5: Energy Consumption Estimation
    energy_usage ← estimate_energy_usage(data_vector, model_runtime)
    RETURN threat_ranking, contract_solution, energy_usage
END
```

3.1 Input Preparation

The aim of this module is to aggregate sensory data in real-time from target area of WSN-IoT services followed by making it suitable for more optimal storage units. The operation of this module is initiated by deploying sensors nodes over varied form of region (home automation, smart factory, farm, etc.). Assume that there are N number of sensor nodes S i.e., $S=\{s_1, s_2, s_3, \ldots, s_N\}$ that are randomly deployed in specific region. The input data is arranged in tabular format with timestamp for retaining better observation-based consistency. The mathematical representation of data vector is as follows,

$$\vec{x}_i(t)=\{d_{vec}(t)\} \quad (1)$$

In the above expression (1), the data vector $\vec{x}_i(t)$ is witnessed to have various time-based dependencies d_{vec}, representing environmental parameters where $d_{vec}=(P_i(t), L_i(t), M_i(t), H_i(t), T_i(t))$. The next step is to carry out data synchronization where alignment of SENTRY is carried out in time for retaining maximum temporal coherence. Interpolation is carried out for any gap observed within sensory data. Further, moving average based filters are used for eliminating noises generated via environment or from sensors. The study uses z-score as a statistical method for identifying the extreme scores followed by removing all outliers surfaced due to transmission faults or sensory errors. Finally, Min-Max scaling is done for normalizing data while one-hot encoding is applied for categorical data (e.g. event type, device ID, etc). k-Nearest Neighbor (KNN) is used for imputing missing values while only essential features are chosen for next part of module operation.

It is to be noted that each data-vector is propagated to the gateway node of blockchain in order to carry out encoding operation. The benefit of this module operation is associated with its generation of enriched data which is not only standardized but also cleaned that minimizes any chances of unwanted computational burden on learning modules in layer stages. The module is customized for blockchain compatibility considering fused data from WSN and IoT origin, which is novel in its form to offer more readiness of aggregated sensory data.

3.2 Ethereum Data Storage

The prime aim of this module is to secure the data obtained in previous stage to offer more transparency and immutability along with tamper-resistance. The collected data vectors are securely written in Ethereum blockchain where a novel form of smart contract design is applied for optimal security. The accomplished data vector $\vec{x}_i(t)$ from prior module is transformed to hashed score followed by committing it towards $T_i(t)$ transactions made on Ethereum. It is mathematically represented as,

$$T_i(t)=\text{commit}(h_i(t), m_{data}) \quad (2)$$

The above-expression (2) represents transactions $T_i(t)$ made on Ethereum considering hash score $h_i(t)$ and metadata m_{data}. The hashed score of $h_i(t)$ can be obtained by applying selective hashing on concatenated value obtained from data vector d_{vec}, time t, and specific node identity nDI_i, i.e. $h_i(t)$=hash($nID_i||t||\vec{x}_i(t)$). The study uses Secured-Hash Algorithm of 256 bit (SHA256) for this purpose. Each block B can be thereby represented as B={T_1, T_2, T_3, ….., T_k}, where k represents number of transactions. It should be noted that a smart contract will now consists of wrapped new entries that controls the mechanism of commitment and accessibility of data. Encoded rules are retained within the smart contract for access control, timestamping, and data ownership. It will eventually mean that now the data is stored over multiple nodes in decentralized manner where it is not feasible for any intruder to ever do any change without consensus of regular nodes. Hence, this module contributes towards data integrity mainly while it also eliminates the dependencies of centralized data storage that is usually the target of an attacker. With a minimal added storage overhead, the bloating of unwanted blockchain is minimized via compact formatting and selected data logging. The novelty of this module is towards an inclusion of verifiable audit trails which is highly secured towards all the events recorded in WSN-IoT environment. Hence, a reliable and trusted management of sensory data is facilitated by successful operation of this module towards various applications like industrial automation and precision agriculture.

3.3 ML-based Threat Intelligence

The key aim of this module is to detect the presence of privacy threats, unauthorized access, and anomalies depending on explicit patterns exhibited in behavior of device and user activities. Various types of historical access logs (viz., network path, access time, request time, user ID) are subjected to training using supervised ML model. The notion of this model is mainly towards facilitating consistent monitoring of all types of incoming request for differentiating them to be either normal or suspicious. The proposed scheme uses a supervised ML classifier of Gradient Boosted Decision Tree (GBDT) in order to determine the anomalies associated with the private key misuse and access patterns. It is empirically represented as,

$$\hat{y}_j = f_\theta(\vec{z}_j) \tag{3}$$

The above-expression (3) represents predictive outcome of $\hat{y}_j$ in binary form where [0, 1] is assigned to represent normal and malicious event, which the function f_θ represents GBDT classifier applied on input feature vector $\vec{z}_j$ that is formed from various historical access logs. Therefore, $\vec{z}_j$= (a_j, r_j, l_j, t_j, u_j) where the individual components in order represent accessed sensor/IoT device, request type, location vector, access timestamp, and encoded user ID respectively. The system also performs allocation of severity score towards all anomalies-based behavior followed by profiling them to be a potential intruder depending upon the frequency and nature of access patterns. The mechanism of profiling contributes towards confirming the presence of threat as well as scoring their level of severity. The computation of the severity score S_j is carried out as follows,

$$S_j=\sum_{i=1,j=1}^{4,43} \alpha_i\beta_j \tag{4}$$

A closer look into the above-expression (4) will showcase that computation of severity score S_j depends on summation of dot product of two components (α and β) bearing lower limit of 1 and upper limit of 4 for both the components. The first component $\alpha_i\beta_j$ for i=1 and j=1 will mean

first model weight $\alpha_1$and rank of access level $R_{priviledge}(=\beta_1)$. The second component $\alpha_i\beta_j$ for i=2 and j=2 will mean second model weight $\alpha_2$and frequency of suspicious access F_{freq} $(=\beta_2)$. The third component $\alpha_i\beta_j$ for i=3 and j=3 will mean third model weight $\alpha_3$and data sensitivity $D_{impact}(=\beta_3)$. The fourth component $\alpha_i\beta_j$ for i=4 and j=4 will mean fourth model weight $\alpha_4$and confidence of model while detecting threat $C_{confidence}(=\beta_4)$.

The most innovative contribution of this module of implementation is towards reducing the possibilities of potential damage in WSN-IoT environment. The outcome of this severity score potentially assist in decision making towards prioritization of threats that demand immediate attention. The proposed design of complete computation process in this module is lightweight and exclusively designed for edge ecosystem where no excessive energy or memory is demanded. The novelty of this module is its ranking system towards realizing severity of cyber threats that is highly customized and facilitates contextual decision making and not only binary classification of threats. The profiling mechanism of an attacker improvises the precision of response system that further gives SENTRY more intelligent and more adaptive in contrast to conventional detection system which is primarily based on static rulesets.

3.4 DL-based Smart Contract Security

This module is meant to further strengthen the prior ML model by identifying possible security loopholes associated with smart contract followed by mitigating it. According to this module, the behaviour and structure of the deployed smart contract is analyzed by the DL model for exploring the patterns of common vulnerabilities viz, permission issues, uninitialized variables, and reentrancy. On the event of flaw identification, proposed SENTRY autonomously suggest for enhanced variants of contract. This generation of contract suggestion is based on prior logic of successful contract and pattern learning. Simulated environment is used for assessing the revised contract prior to deploy within a blockchain. This part of the implementation initially performs encoding of smart contract. The system carries out parsing of all smart contract SC_k in the form of sequence of variable interactions and operation codes (op_1, op_2, …..). The empirical representation for this smart contract is SC_k= (op_1, op_2, …., op_m). Further, the SENTRY implements LSTM towards learning the patterns of vulnerability as follows,

$$\hat{v}_k = \mathcal{F}_\phi(\vec{s}_k) \tag{5}$$

In the above expression (5), the computation of vulnerability probability in k^{th} contract i.e., $\hat{v}_k$ is of binary form {0, 1} while the function $\mathcal{F}_\phi$ represents DL model with Φ parameter. Considering a critical cut-off value of τ, if the value of $\hat{v}_k$ is found to be quite higher than τ than proposed system generates a solution strategy δ_k which is represented as,

$$\delta_k = G(\vec{s}_k) \tag{6}$$

The above expression (6) shows that solution strategy δ_k depends upon the patch generator network G considering each smart contract $\vec{s}_k$. It should be noted that this patch is generated by DL model and is validated using simulated transaction assessment prior to actual deployment. Therefore, a proactive defense layer is contributed by this module by resisting possible vulnerabilities in WSN-IoT environment. Further, it is to be noted that this module using DL is only activated when there is a positive confirmation of threat and hence it significantly preserves system resources while performing data exchanging operations. The novelty of this module is associated with its capability to self-heal and auto-analyze smart contract adopting DL method.

In fact, the proposed scheme extends the capability of conventional blockchain operation from static to more adaptable management of vulnerabilities using learning approach.

3.5 Resource Optimization

The prime aim of this module of implementation is towards efficient operation of complete computational framework of SENTRY for given set of sensors which are usually characterized by limited computational power, memory, and battery. The computational load over the system is monitored by the module followed by undertaking optimization task. The system performs on-demand computation where the ML model is allowed to continuously run while DL module is activated only at the event of severe cases of threats. The system processes the data followed by repositing them in compressed manner while pruning strategies are adopted for unnecessary operations.

The possible inclusion of blockchain and hybrid form of ML and DL inclusion in proposed scheme requires it to be resource efficient. The proposed system performs model quantization that minimizes the bit width associated with model weights while model pruning is carried out to eliminate redundant operations and neurons. Finally, dynamic triggering is carried out for activating the DL model only the time when threat score S_j is found more than cut-off score γ. The cumulative consumption of energy is mathematically expressed as,

$$E_{total}=\sum_{t=1}^{T}[\zeta . E_{DL}(t)] \quad (7)$$

The above-expression (7) states that the computation of energy consumption depends upon two attributes ζ and E_{DL}. The first attribute ζ represents the summation of energy EML consumed for yielding ML inference and Indicator function I_{func} i.e., $\zeta=E_{ML}+I_{func}$, where I_{func} is noted to be active in an event of high risk i.e., $s_j>\gamma$ while E_{DL} is energy consumed during execution of DL model at t time. It is needed to report that energy computation associated with proposed study is actually derived from hardware-specific benchmark and simulation profiling adopting publicly available dataset and real-world embedded boards [58][59]. The estimation of processing cycles, memory usage, and inference time is carried out by the resource monitoring system. Further, calibrated models are used for translating them into energy consumption. Moreover, due to adoption of simplified mathematical modelling (expression (7)), the proposed energy computation mechanism is highly flexible for any future work, or it can be customized to any specific WSN or IoT applications or services. Currently, the modelling is generalized to any form of applications or services hosted in WSN or IoT or jointly with a target to gain resource efficiency while performing security operations. The novelty of proposed module resides in its capability to balance power and threat detection accuracy at same time. The next section presents elaboration of accomplished study outcomes.

3.6 Energy Consumption Model

The design of the proposed framework has taken energy efficiency as one of the core considerations, owing to the inherent limitations associated with WSNs and IoT devices such as battery limitations, computation limitations, and memory limitations. As a result, the proposed framework takes up an energy-aware execution process where the DL tasks are executed by the system only if the ML component of the system detects any high-risk security incident. In doing so, the computational overhead involved is reduced while achieving accurate

threat detection. The total energy consumed by SENTRY in one inference cycle can be stated as the sum of energy needed for inference tasks, deep learning execution, blockchain tasks, communication, and memory, which is stated as:

$$E_{Total} = E_{ML} + \mathrm{I}\,(S_j > S\gamma)\,E_{DL} + E_{BC} + E_{Comm} + E_{Mem} \qquad (8)$$

where 'E_{Total}' is the amount of energy used for one inference process; E_{ML} is the energy needed for the GBDT anomaly detection model; E_{DL} is the energy needed for the BiLSTM smart contract vulnerability analysis model; I ($S_j > \gamma$) is the indicator function which triggers the model to operate only if the threat severity score S_j exceeds the threshold value γ; E_{BC} is the energy for blockchain processing, consisting of SHA-256 hashing, transaction generation, smart contract invocation, and Ethereum processing; E_{Comm} is the communication energy for sending observations from WSNs to gateways; and E_{Mem} is the energy for memory access operations.

The energy consumption indicators used herein are obtained from hardware-based simulation profiling. This experiment was conducted using an Intel Core i7 machine, with NVIDIA RTX 3060 GPU and 16 GB RAM. Processor usage, GPU usage, running time and memory use were tracked for every inference process. The overall energy consumption per inference process was calculated using:

$$E = P_{Avg.}\,\mathrm{T} \qquad (9)$$

where $P_{Avg.}$ is the average system power consumption (in Watts) and T is the running time (in seconds).

Average power is calculated by summing up CPU power consumption, GPU power profiling, memory usage and blockchain processing overhead per inference operation. The communication energy consumption was obtained from the number of packets sent and the radio transmission model, whereas blockchain energy comprises the hashing and smart contract execution within the local Ethereum (Ganache) network. Note that, as Ganache is a local blockchain emulator, gas fees on the public blockchain and delay caused by propagating transactions over the Internet were not included.

The average energy consumption was calculated based on CPU power consumption, GPU power consumption, memory usage, and blockchain operation overhead that occurred during each inference process. Communication energy consumption was calculated based on the number of data packets sent and the radio transmission model, whereas the blockchain energy consumption covers hashing operations and smart contract executions in the local Ethereum (Ganache) network. As Ganache simulates the Ethereum network locally, public blockchain transaction gas fees and Internet-wide blockchain transactions' propagation delay were excluded from the reported energy consumption values. In order to ensure better replicability, all experiments were conducted five times independently with the same hyperparameters for each algorithm, and the averaged value of energy consumption was presented. In this way, in the proposed adaptive scheduling approach, SENTRY consumed an average energy of 4.2 J per inference, which is significantly smaller than in the case of the baseline solutions since the Deep Learning analysis is performed only if the predicted threat level is higher than the preset threshold. The proposed framework is generic and can be easily adjusted to other blockchain networks, other lightweight devices, or other permissioned blockchain systems.

3.7 Severity Ranking Model (Expanded)

For each detected anomaly, the severity score is calculated as:

$$S_j = \sum_{i=1} 1\, \alpha_i X_i \qquad (10)$$

where $\sum_{i=1} 1\, \alpha_i = 1, 0 \leq \alpha_i \leq 1\ and\ X_i \in [0,1]$ threat indicators.

Weights were determined based on five-fold cross-validation with further grid search optimization to enhance the F1-score with minimal false-positive detection errors for the training data set. As opposed to traditional binary classifiers, the suggested approach allows prioritizing cybersecurity threats depending on the context, thus providing SENTRY with the ability to distribute processing power accordingly in proportion to the detected level of danger. All threat indicators are standardized via Min-Max normalization before the calculation of their severity to neutralize any potential discrepancies associated with different scales of the heterogeneous attributes used in WSN and IoT devices. As a result, the suggested weighted severity model will provide consistent and meaningful threat ranking regardless of any potential disparities in measurement units.

3.8 Computational Complexity and Scalability Analysis

Computational complexity of the proposed SENTRY framework is assessed in order to determine the viability of this approach to be deployed in real-world scenarios involving a large number of WSN-IoT networks. The computational complexity is calculated separately for each individual stage as well as the framework as a whole.

Let N represent the number of sensor nodes; M represent the number of observations gathered by sensors; F represent the number of security features extracted; T represent the number of trees used in the Gradient Boosted Decision Tree (GBDT); D represent the maximum depth of each tree; L represent the length of opcode sequences; and H represent the number of neurons in the BiLSTM network. The accumulated computational complexity of SENTRY is:

$$O(MF) + O(M) + O(TD) + O(LH^2) + O(1) \qquad (11)$$

The computation costs grow linearly with respect to the number of gathered sensor observations and extracted features, whereas the Machine Learning inference is limited by a constant number of decision trees. Despite higher computational complexity of the BiLSTM component, its activation occurs only upon detection of suspicious activities by the ML component. Thus, the overall execution time stays much lower in comparison with architectures featuring continuous Deep Learning inference. Moreover, the overhead from using blockchain is linear, since SHA-256 hash values are stored in blockchain rather than complete sensor observations. The suggested hybrid storage approach greatly reduces computational, communication, and storage overheads.

As can be seen, the SENTRY system is scalable and computationally efficient enough for deployment in large-scale IoT systems featuring thousands of sensor nodes.

4 Result Analysis

According to the discussion of adopted research methodology in prior section, SENTRY can perform various sets of operations to secure the data being exchanged in a joint environment of WSN and IoT. It can identify cyber threats where vulnerabilities of smart contract are well managed. It can also offer potential data integrity followed by computational efficiency towards sensors with minimal resource consumption. The simulation study is carried out adopting access logs and sensor data with an inclusion of smart contract while the accomplished outcome is compared to existing standards adopted for securing WSN as well as IoT. This section presents discussion of exact environment considered to perform the simulation study followed by briefing the adoption of dataset and assessment strategy. The section also elaborates about accomplished outcomes, security analysis, and interpretation of study results as learning outcomes.

4.1 Simulation Environment

A controlled simulation environment is adopted towards assessing the effectiveness of SENTRY in order to replicate the practical scenario of deployment. The development of the presented framework is carried out using Python 3.10 considering essential libraries e.g. PyTorch, Keras, TensorFlow 2.x, Scikit-learn towards introducing ML and DL modules. The interaction of smart contract and emulation of local Ethereum is carried out using Web3.py and Ganache CLI respectively. The handling of the private key is carried out using MetaMask while coding and deployment of smart contract is carried out using Solidity. The complete analysis is carried out on Windows machine with core i7 processor with NVIDIA RTX 3060 GPU, and 16 GB RAM towards supporting simultaneous operation of DL.

The *assumptions* carried out during setting up the simulation are as follows: i) There is a regular and consistent generation of data from the sensor node at fixed interval while potential adversaries are injected regularly and in randomly numbers, ii) data integrity can be effectively maintained by Ethereum however it is not resilient against any threats associated with off-chain operations and smart contract-based vulnerabilities, and iii) A malicious firmware or compromised device are assumed to be responsible for leakage of private key. As the model consists of both ML and DL, hence it is necessary to carry out careful formation of hyperparameters and their respective values. The learning rate considered for ML module is 0.01 while that of DL module is 0.001. The batch size of ML module is 64 while that of DL module is 128 while both the modules uses Adam optimizer. The model type towards threat detection using ML module uses RF while BiLSTM with dense layer is used for DL module. Hence, there are 3 layers of LSTM and 2 dense layers while 0.3 is considered as dropout rate for DL module.

4.2 Adopted Dataset

The evaluation of the SENTRY is carried out using most recent Gotham Dataset 2025 [60] that possess information obtained from 78 emulated IoT devices adopting conventional protocols e.g. RTSP, CoAP, and MQTT. The dataset has an inclusion of multiple attack types e.g. amplification of CoAP, network scanning telnet brute-force, Denial -of-Service, and many more. In this dataset, various features are extracted from raw *pcap* files and well-processed *.csv* files of more than one million events of bidirectional traffic with annotated data. The feature set consists

of ground-truth labels, timestamps, device identifiers, fields with specific protocols, meta-data of packet with an indication of regular and malicious events.
There are total of 50,000 data instances with *.csv* as file-format consisting of various attributes e.g., threat severity labels (critical/high/medium/low), type of attack, status of smart contract (suspected/valid), score of data integrity, access frequency, access levels, size of packet, type of sensor, node ID, and time stamp. This dataset consists of real-world scenarios of IoT traffic that consists information of both contemporary and diverse types of attacks over different protocols. The extent of heterogeneity and scale facilitates the assessment of efficiency of ML model in identifying the adversaries while representing the resource constraint that are quite usual in any generalized configuration of an IoT. A wide spectrum of adversaries are present within this new dataset that permits the DL model for assessing degree of anomalies within a smart contract based on original pattern of utilization. Apart from this, the threat severity scores are labelled within this dataset that leverages the ranking methods implemented in SENTRY towards profiling attackers. Hence, in all the way, this dataset is quite an appropriate choice for the proposed study assessment.

4.3 Assessment strategy

The proposed system considers multiple performance metric e.g., detection accuracy, latency, memory consumption, and throughput for analyzing the generalization capability of proposed study model. Selection of these metric are carried out towards assessing both computational resources and security strength associated with SENTRY as well as certain existing models for comparative analysis. Following are the briefing of adopted performance metrics:

- *Detection Accuracy*: This metric is considered as the ability of the system to precisely detect both normal and malicious behavior associated with WSN integrated IoT environment. It is mathematically computed as,

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{12}$$

  In the above expression (12), it can be noted that accuracy computation depends upon True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). An increased value of detection accuracy is a direct indication of trustworthy threat identification along with reduction of possible risk associated with outliers or even missed attacks that are quite essential in security analysis of domains related to industrial IoT and smart agriculture.
- *Latency*: This performance metric emphasizes on assessing the cumulative duration consumed by the system towards identifying and responding to the adversaries after the generation of data. It is mathematically calculated as,

$$\text{Latency}=T_1-T_2 \tag{13}$$

  In the above-expression (13), the computation of Latency is carried out considering T_1 i.e., timestamp of an event occurrence and T_2 i.e., timestamp of threat event is positively detected. If the score of latency is minimal than it will represent the faster responsiveness of model in an event of a critical threat exposure. This also contributes to significant minimization of maximum damage programmed by the adversary.

- *Memory Consumption*: This performance metric computes the amount of RAM utilized by the computing device when subjected to communication or processing under threat-prone environment. The mathematical expression towards its computation is as follows,

$$Memory\ Usage\ (MB) = \frac{Total\ Used\ Bytes}{1024x1024} \quad (14)$$

This performance metric is capable to stating the computational burden or state of saturation while applying ML/DL operation in resource-limited devices in WSN integrated IoT environment. Therefore, the scalability as well as lightweight nature of the framework can be easily explored using this metric.

- *Throughput*: This performance metric computes the quantity of data packets that are successfully processed by the computational unit per unit time. Following the expression towards computing throughput,

$$\text{Throughput} = \frac{Number\ of\ Processed\ Events}{Total\ Time\ of\ Processing\ (s)} \quad (15)$$

Increased throughput value is expected to show that proposed system is capable of handling massive task associated with exchange of data in WSN-IoT environment. This validates the supportability of the model towards real-time services while simultaneously implementing security operations using ML/DL models.

Considering the above-mentioned performance metrics, the SENTRY is compared with certain baseline models. The adopted baseline models are selected based on their possessed extensive security characteristics right from conventional blockchain operation to sophisticated AI methods. Following are the briefing of the baseline models adopted for benchmarking SENTRY,

- *Standalone Blockchain Method (SBM)*: This type of methods make use of Ethereum targeting towards accomplishing decentralized storage system and data immutability [40]-[49]. SBM is meant for facilitating tamperproof data; however, it doesn't possess any inbuilt smart decision-making system or any form of intelligence towards securing the data from intrusion nor it performs any analysis of vulnerabilities. This model is considered for benchmarking as it is one of the frequently adopted methodology in literatures towards data logging in IoT and is considered as secure. Another key reason to consider SBM is to showcase that blockchain as a standalone is not adequate to offer resistance from emerging cyberthreats.
- *Signature-based Threat Detection (STD)*: The conventional threat detection methods mainly depends upon predefined signatures or protocols for identifying the types of attackers [19]-[26]. Although it is lightweight in its computational operations, yet it is unsuitable for identifying emerging threats that are witnessed in joint operation of WSN-IoT environment. Many IoT system is reported to use signatures because of its simplified implementation and deployment methods and inclusion of STD is meant to showcase that yet intelligence is very essential for proactive security system in WSN and IoT environment.
- *Machine Learning based Threat Detection (MLTD)*: There are various literatures at present to showcases the involvement of varied ML models towards anomaly detection in IoT [27]-[31]. Its capability of threat classification depending on predefined patterns are its potential beneficial characteristics; however, its integration with conventional blockchain is still in nascent stage to prove the effectiveness in real-time environment

of WSN and IoT together although some initial attempts have been carried out in recent times.

- *Deep Learning based Scrutinizing Contract (DLSC)*: These approaches focuses on analyzing the smart contract adopting DL method [53]-[57]. Although DLSC is capable of identifying the vulnerabilities present within smart contract and yet it cannot perform threat detection in real time nor it can leverage its deployment considering resource constrained sensory devices. DLSC act as a benchmark for realizing the resource efficiency and effectiveness associated with DL models.
- *Hybrid Machine Learning with Blockchain (HMLB)*: This approach is meant for incorporating proactive analytical power on blockchain deployment towards detection of vulnerabilities and threats using varied ranges of ML models [36]-[39],[50]-[57]. Irrespective of such potential, HMLB doesn't possess any advanced attributes e.g., threat profiling, self-healing of smart contract, etc. This approach also doesn't offer dedicated module for resource efficiency while performing their targeted operations. It should be noted that operations performed by HMLB bears strong similarity with SENTRY in perspective of ML incorporation in Ethereum; however, its depth of integration and scope is very much limited in contrast to SENTRY while it will assist in exhibiting that SENTRY offers an end-to-end security solution with better scope of threat intelligence within it.

4.4 Accomplished Result

The outcome of benchmark analysis is tabulated in Table 1 which showcase significantly better performance of proposed system. All the conventional baseline approaches (SBM, STD, MLTD, DLSC, HMLB) along with SENTRY has been subjected to similar environment of testbed with defined values of hyperparameters as detailed in prior sections. The outcome shows SENTRY to accomplish increased accuracy detection of 96.8%, lower latency of 1.3s, lower memory consumption of 49s, increased throughput of 225 event/sec, and lower energy consumption of 4.2J.

Table 1. Accomplished Study Outcome for Benchmark Analysis

System	Detection Accuracy (%)	Latency (seconds)	Memory (MB)	Throughput (events/sec)	Energy Consumption (J/Inference)
SBM	N/A	1.2	58	210	8.5
STD	81.3	1.6	64	180	10.2
MLTD	87.9	1.5	72	190	9.3
DLSC	90.4	2.3	110	165	12.4
HMLB	92.1	1.9	75	195	9.0
SENTRY	96.8	1.3	49	225	4.2

From Table 1, the higher detection accuracy score and lower latency score for SENTRY is the most significant outcome from security perspective. It is mainly due to collaborative ML and DL methods on data validation using Ethereum that is known to speedily emphasize on vulnerable events. From the perspective of memory and throughput, SENTRY consumes only

49MB while 225 events / second is recorded. It states that SENTRY is capable of managing voluminous sensory data without sacrificing the cumulative system performance (as noted from low memory consumption). Another significant outcome is noted for reduced energy consumption for proposed system while it should be noted that this computation is carried out deploying equation (7) in prior section of methodology. SENTRY consumes only 4.2J for each inference which is lowest compared to other schemes. This outcome can be justified by fact that SENTRY uses lightweight ML model towards prescreening the network vector while selective activation is carried out by DL model. This sequential operation can be actually stated to be novel energy-aware scheduling which is designed to cater up the practical integration of WSN and IoT services. The overall outcome show that SENTRY offers better balance among security demands, computational demands, and communication demands.

4.5 Security Analysis

There are various types of threats that affects both security and data exchange performance of WSN and IoT; hence, it is essential to select the specific form of attacks that is not only common in both but also induces maximum damage to both WSN and IoT. Hence, security analysis is carried out based on such forms of threats viz. Denial-of-Service (DoS), data injection, breach attempt for private key, off-chain vulnerabilities, and Smart Contract exploits. The robustness of the security for SENTRY is evaluated using Threat Detection Rate (TDR), Recovery Time (RT), Private Key Security (PKS), and smart contract integrity (SCI) as exhibited in Table II. The computation of TDR is done by dividing TP with summed value of TP and FN while RT is calculated by average duration consumed by SENTRY for identifying the threat and preliminary recovery operations (e.g. network isolation). The numerical score of RT is obtained by subtracting timestamp of process completion from timestamp of threat occurrence. PKS is calculated by dividing blocked key exploits by attempts of cumulative key exploits. Finally, SCI is calculated by dividing contract identified and revised by total tampered contract.

Table II. Performance of Threat Resistivity for SENTRY

Threat Type	SCI (%)	PKS (%)	RT (ms)	TDR (%)
DoS	100	100	210	98.7
Data Injection	99.3	99.1	195	97.4
Breach Attempt of Private Key	99.2	98.8	180	96.9
Off-Chain Vulnera-bility	98.9	97.5	200	96.2
Smart Contract Ex-ploits	99.7	98.9	190	97.6

It can be noted from above Table II that SENTRY offers 96% of identification performance with higher consistency for almost all types of threats under evaluation while the self-healing duration is noted to be lesser than 220ms. Apart from this, the dual layer of SENTRY using ML and DL model facilitates integrity of smart contract and private keys with maximum reliability score (100%), which is something novel part of the outcome. Further, any form of susceptibility

towards the off-chain threats is minimized by Ethereum and hence this performance score confirms the security robustness of SENTRY which makes its highly suitable for distributed real-time application e.g., smart home automation, industrial IoT, smart agriculture etc. In order to benchmark, SENTRY is also compared with conventional baseline models with respect to understand the capabilities of each one towards threat resistance as exhibited in Table III.

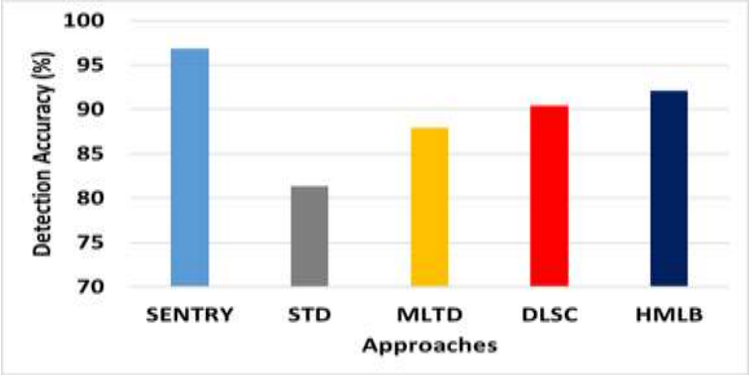
Table III. Comparative Threat Resistance Analysis

Model Name	OCD (%)	SCI (%)	PKP (%)	DMR (%)	DDR (%)
SENTRY	98.9	99.7	98.8	97.4	98.7
SBM	79.4	82.3	85.7	89.1	92.3
STD	84.3	87.1	89.8	91.2	94.5
MLTD	86.9	89.5	90.2	90.8	95.1
DLSC	77.5	80.2	84.6	88.3	91.6
HMLB	82.9	85.3	88.1	90.5	93.7

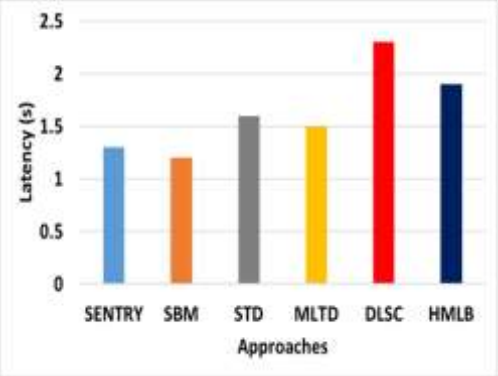
The outcome exhibited in Table III showcases that SENTRY is capable of resisting maximum number of threats considering multiple performance metrics. DoS Detection Rate (DDR) is calculated by dividing appropriately identified DoS events by total attempts of DoS. Data Manipulation Resistance (DMR) is calculated by dividing flagged or blocked data tampering events with cumulative attempts of tampering. Private Key Protection (PKP) is calculated by dividing attempts of resisted key exploits with cumulative attempts of exploits. Finally, Off-Chain Defense (OCD) is estimated by dividing the rejected/blocked anomalous off-chain data with cumulative attempts of malicious data. From the significantly higher side of scores accomplished by SENTRY, it can be stated that degree of integrity of smart contract and off-chain defense are highly underrated in conventional baseline models. The increased score for SENTRY with respect to its security posture can be attributed to integrated ML and DL model for managing smart contract even under most vulnerable conditions.

4.6 Discussion

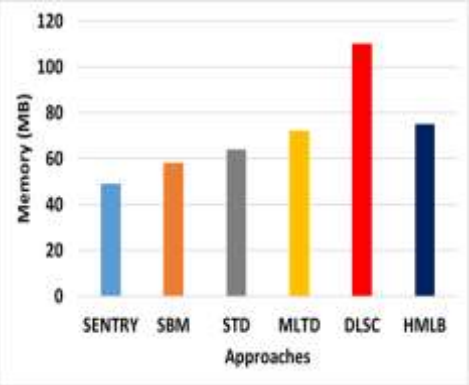
The highlights of the accomplished numerical results of proposed system and multiple baseline model eventually showcases its superiority on behalf of multiple performance metrics. Although, the outcomes are known, but it is necessary to highlights the rationale behind such outcome which could contribute more contextual and technically enriched guidelines for efficient design perspective. This section presents an interpretation of the accomplished results in order to contextualize the real-time implication as well as contribution of each outcome.



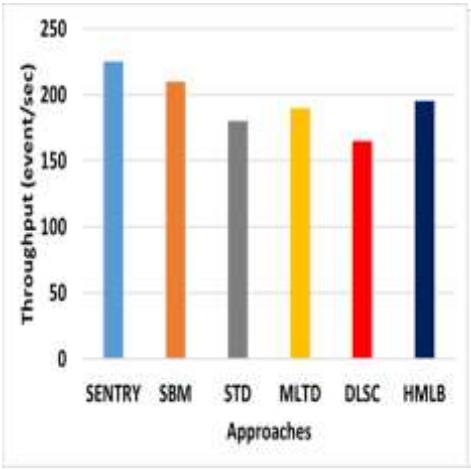
(a) Detection Accuracy Analysis



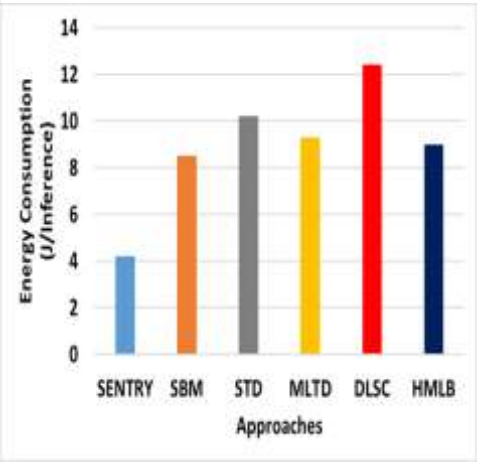
(b) Latency Analysis



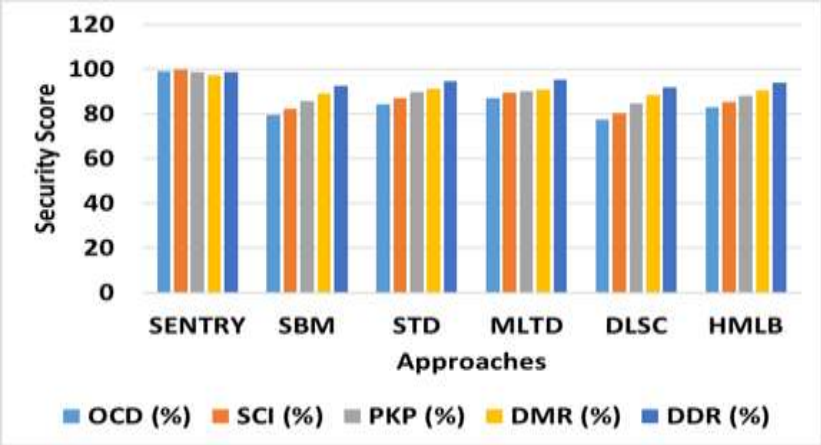
(c) Memory Efficiency Analysis



(d) Throughput Analysis



(e) Energy Efficiency Analysis



(f) Security Analysis

Fig 3. Benchmarked Analysis of Security Performance

With reference to the benchmarked outcome exhibited graphically in Fig.3(a) to Fig.3(f), following is the discussion with respect to core metrics:

- *Interpretation of Detection Accuracy* (Fig.3(a)): SENTRY has accomplished a detection accuracy of 96.8% which is approximately 8.8% more in contrast to all the conventional baseline model under **consideration**.
    - *Rationale*: The prime reason for this improved outcome is due to hybridization of ML and DL model. The detection of privacy violation and anomalies are carried out by ML model in data level while assessment of smart contract is carried out by DL model towards exploitable logic or tampering event. SENTRY encapsulates both control flow and data flow security for offering maximized protection from dynamic threats while yielding intelligent decision-making system. On the other hand, majority of baseline models emphasizing on detecting attackers (e.g. STD) or deploying secure smart contract methods (e.g. MLTD) doesn't possess the capability of assessing to multiple levels that can be seen only in SENTRY. Apart from this, baseline models are witnessed to be highly dependent on one AI model where the validation
    - *Novelty*: The prime novelty of this outcome is related to the collaboration of vulnerability assessment (via DL) with threat detection (via ML) that is carried out on Ethereum. This is uniquely a novel strategy that offers extensive detection and protection capabilities in contrast to any existing models.
    - *Implication*: With higher threat detection accuracy recorded by SENTRY, its applicability is quite higher on critical application areas right from industrial IoT to smart agriculture. All these application areas are highly vulnerable to even minor misclassification associated with threats leading to either leakage of sensitive data or massive collapse of system. The outcome leverages the usage of SENTRY for autonomous management of threats and reliable communication in WSN-IoT environment.

- *Interpretation of Latency* (Fig.3(b)): The latency score of SENTRY is 1.3 seconds which is approximately 0.4% reduced compared to mean of all 5 baseline models.
    - *Rationale:* The lightweight structure of SENTRY especially associated with data verification and feature selection is the prime cause of reduced latency score. Instead of adopting conventional encryption methods for verifying data payloads, SENTRY performs prioritization of anomalies that results in processing the transactions that are only suspicious using DL. This minimizes the unnecessary computational task of further verification leading to lower latency of SENTRY. The lower value of latency for SBM compared to SENTRY cannot be stated as optimal as it lacks intelligent decision making while the performance score of SBM is quite sub-optimal in contrast to SENTRY. On the contrary, models like HMLB, DLSC, and MLTD is witnessed to perform complete validation considering all the incoming data vector. This operation substantially demands more resources as well as extensive processing duration which it is potentially questionable if they will even run within resource-constrained sensor nodes in WSN and IoT. Apart from this, presence of resource scheduling in adaptive fashion cannot be expected from majority of baseline models which also increases latency.
    - *Novelty:* The key novelty of SENTRY in this perspective is selective verification method that considers anomaly scoring. The adhoc deployment of DL on-demand also contributes towards minimizing overhead apart from controlling potential latency possibilities is another novelty of SENTRY.
    - *Implication:* There are various applications and services in IoT that calls for mission-critical operations e.g. environmental monitoring, industrial automation, etc. On the ground of minimized latency accomplishment, SENTRY finds itself much suitable for such mission-critical applications in WSN-IoT environment. Hence, system reliability as well as real-time decision making is ensured by minimized response time of SENTRY.
- *Interpretation of Memory Efficiency* (Fig.3(c)): SENTRY has only 49MB of memory consumption which is approximately 27% lower in contrast to mean scores of all baseline models under consideration.
    - *Rationale:* It is essential to initially understand the usage of memory towards security. SENTRY doesn't use any complex encryption methods apart (just like SBM or STD) from one small usage of hashing for encoding whose memory is negligibly small. Unlike other AI models (MLTD, DLSC, HMLB), SENTRY doesn't use any iterative modelling instead a modular learning mechanism has been implemented with on-demand basis of deployment (DL model). Hence, the memory associated with SENTRY is quite less in contrast to existing baseline models. Apart from this, a smart contract validation and threat identification is facilitated resource efficiently preventing any form of memory consumption persistently. Existing AI-based baseline models potentially uses DL with constant allocation of memory irrespective of complexities associated with input. Further, none of them are actually designed for embedded environments while SENTRY do target for WSN-IoT environment.

- *Novelty:* SENTRY contributes to significant control memory footprints by modular loading and allocation of dynamic memory without compromising the processing power. This specific characteristic of SENTRY is quite novel when implied on either WSN or IoT or both using blockchain operation.
- *Implication:* The evidence that SENTRY offers highly reduced memory footprint itself states its applicability towards its deployment as real-time devices with uninterrupted operations and that demands longer lifespan. It is suitable for sensor grids and smart home automations.

- *Interpretation of Throughput* (Fig.3(d): SENTRY has recorded throughput as 225 event per second which is approximately 37% higher in comparison to existing baseline models.
  - *Rationale:* The possible computational load existing between auditing smart contract by DL model and screening vulnerabilities in early stage by ML model. Interestingly, SENTRY offers a significant level of load balancing while performing any operations related to data management in WSN-IoT environment. This also leads to flow of streamline transaction and parallel processing that mutually enhances the system throughput eventually for SENTRY. On the other hand, baseline models of SBM and STD has some heavier and complex operations. Although they are non-proactive, but their efficiency is only for known attacker prevention. The other AI-based baseline models (HMLB, DLSC, MLTD) has excessive iterative operation and involves lengthier optimization process. Hence, such module may offer somewhat acceptable detection performance but it is done at the cost of involving various resources, which directly adversely affect throughput and data delivery services. This is not acceptable for real-time applications that demands instantaneous response.
  - *Novelty:* From perspective of throughput, the key novelty resides in that fact that SENTRY is capable of compartmentalize contract-level validation from data-level validation. This strategy significantly contributes towards enhanced data delivery performance as well as increases scalability too.
  - *Implication:* SENTRY is witnessed to offer high throughput and this outcome can justify the usage of SENTRY on smart city infrastructure and industrial manufacturing where WSN and IoT are both demanded. Hence, SENTRY can offer large scale data dissemination without costing extensive latency or drop in computational performance.
- *Interpretation of Energy Efficiency* (Fig.3(e): SENTRY consumes an energy of 4.2J which is approximately 5.68% lower than other baseline models.
  - *Rationale:* The prime reason for reduced energy consumption for SENTRY can be attributed to dynamic reduction of computation even during the passive states. SENTRY actives only relevant submodules of detection. It is to be noted that evaluation of an intensive smart contract is actually offloaded to the trusted edge device in SENTRY. All this calls for reduced resource utilization and faster processing without any unnecessary iterative operation leading to significantly lower utilization of energy. On the contrary, baseline

model like DLSC and HMLB actually bears increased cost owing to an inclusion of consistent monitoring program embedded within resource-limited sensor and involvement of dense neural network. None of the baseline models possess any dedicated module for resource awareness that renders them highly unsuitable for sensory operations either in WSN or in IoT or in both.

- *Novelty:* The prime novelty of SENTRY from perspective of energy efficiency is its resource profiling that is performed when the model identifies adversaries and deploys logic of selective alert triggers for facilitating memory and allocation of processing power with contextual awareness. Hence potential amount of energy wastage is minimized by SENTRY.
- *Implication:* The implication of SENTRY is strongly recommended for any applications that demands consistent monitoring or is deployed in highly vulnerable geographic regions where additional sensors cannot be facilitated. Hence, SENTRY is actually highly supportive of any mission critical applications in WSN-IoT environment.

- Interpretation of Security (Fig.3(f)): SENTRY is found to offer approximately 16.7% increased OCD, 14.82% increased SCI, 11.12% increased PKP, 7.42% of increased DMR, and 5.26% of increased DDR in contrast to mean of all baseline models.
  - *Rationale*: The significantly better performance of SENTRY is because of its dual checkpoints method associated with the architecture. The first checkpoints filter the anomalies using ML while the second checkpoints validate the smart contract using DL. It always ensures that no execution takes place either with off-chain risk or with exploitable logic. Unfortunately, existing AI-based baseline models are proactive but they lack this dual layer of security.
  - *Novelty*: The novelty of the proposed SENTRY model is that it is capable of resisting various types of lethal attacks both applicable on WSN and IoT. Its self-healing nature also makes it more novel for it can integrate automated smart contract amendments with attacker profiling, something less witnessed in existing system.
  - *Implication*: Owing to its potential security system, SENTRY is potentially applicable for autonomous transport system as well as large scale implementation like environmental monitoring and industrial IoT.

In order to assess the generalization performance of the proposed SENTRY model beyond one specific benchmark data set, further experiments were carried out using the publicly available TON-IoT data set. TON-IoT was chosen since it is a realistic representation of IIoT environment that includes heterogeneity in terms of IoT devices, telemetry data, operating system logs, network traffic, and various attack types. Unlike Gotham2025 that mainly provides network-level attack traces, TON-IoT includes system-level telemetry and behavioral data to provide a complete evaluation of intelligent threat detection models. TON-IoT data set includes not only normal network behavior but also several modern attack types such as distributed denial-of-service (DDoS) attacks, ransomware, password attacks, backdoor attacks, injection attacks, scanning attacks, and data exfiltration. Preprocessing procedures used for Gotham2025 data set have been repeated for TON-IoT dataset, including the use of K-Nearest Neighbor for

filling missing values, min-max normalization, one-hot encoding of categorical variables, and feature selection based on information gain criteria. Also, the same hyperparameter setting presented in Section 4 has been preserved.

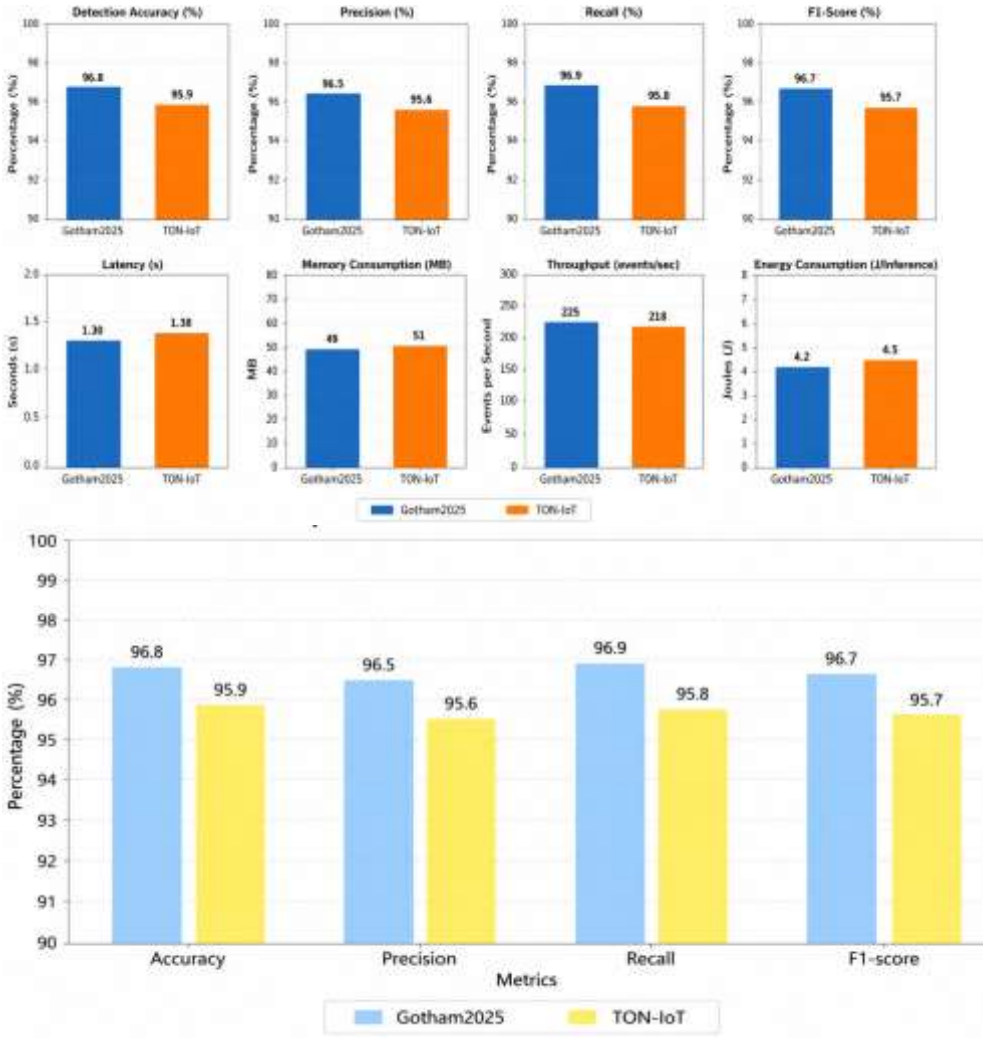


Fig 4. Performance comparison of SENTRY on Gotham2025 and TON-IoT datasets

These results in (Fig 4.) show that the proposed framework retains its outstanding ability to detect threats with high precision in both datasets. Despite the drop in the accuracy rate by approximately 0.9% in the case of TON-IoT dataset, SENTRY is able to retain the overall detection accuracy above 95%, proving its generalization ability in heterogeneous IoT scenarios. The increased delay and memory consumption can be mainly related to the larger number of dimensions in telemetry attributes and behavior diversity of TON-IoT dataset. Additionally, these results prove the adaptability of the hybrid Machine Learning-Deep

Learning architecture in various IoT traffic distributions without the need for retraining. Moreover, the adaptive activation technique implemented in the proposed algorithm ensures the minimization of energy expenditure since the Deep Learning analysis is used only in cases when it is required. In other words, it is applied only to high-risk events, ensuring low energy expenditure in various datasets.

On the whole, the external validation shows that the proposed solution is not overfitted to one benchmark and operates efficiently under several IoT operating environments. Therefore, the proposed framework can be implemented in WSN-IoT systems in real life.

Consistent performance noted from experiments done in both the Gotham2025 and TON-IoT environments as depicted in (Fig 4.) shows that the SENTRY method is stable despite working in heterogenous networks. While other methods experience degradation during their performance evaluation using unseen datasets, SENTRY is able to maintain high detection accuracy while remaining computationally efficient due to its adaptive hybrid design. This makes it suitable for use in real-world WSN-IoT environments.

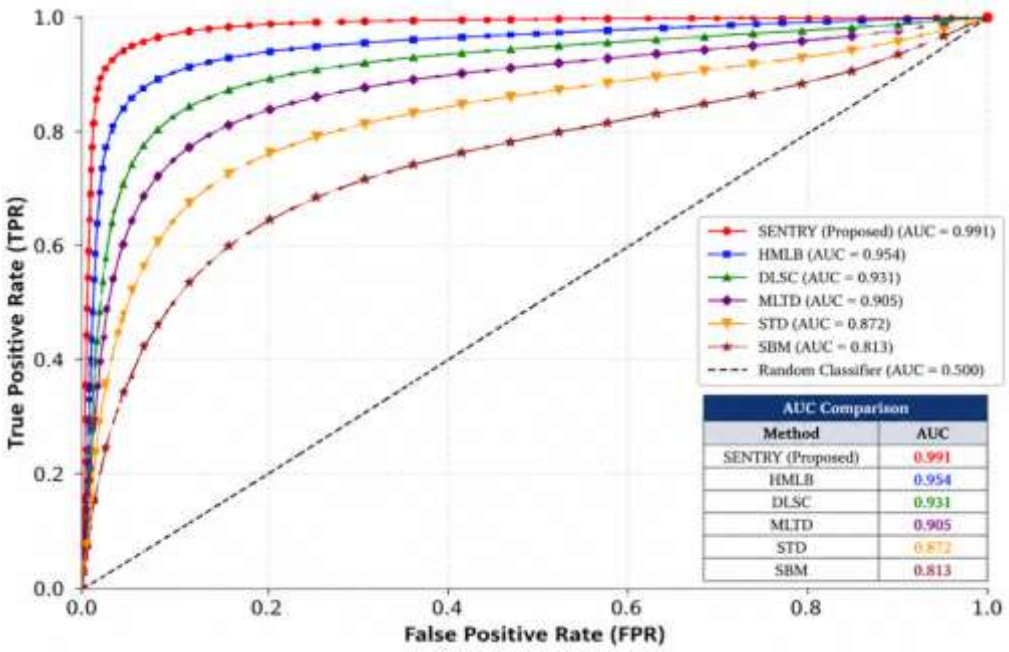


Fig 5. ROC Curve of SENTRY and other methods

The findings from the ROC analysis in (Fig 5.) clearly indicate the validity of SENTRY in discriminating between malicious and legitimate behavior on the network. The high AUC value of 0.991 of the SENTRY approaches means that it performs better than all other baseline approaches at any threshold level.

The proposed SENTRY architecture is well supported by the ablation study, which shows the importance of all components involved. Six different configurations have been considered: ML Only: Only the ML-based threat detection system is active; ML + Blockchain: The ML-based

threat detection system along with the blockchain-based secure storage system; ML + DL: ML-DL hybrid system, which lacks blockchain-based security; No Threat Ranking: ML-DL-Blockchain hybrid framework without the suggested threat ranking system; No Resource Optimization: Continuous operation of the DL framework without adaptive resource optimization; and Full SENTRY: Complete SENTRY setup. The "ML Only" system has the least computational cost but achieves only 91.4% of detection accuracy due to vulnerabilities being unaccounted for within the smart contracts. Combining ML and blockchain enhances data security and detection accuracy to 93.2%, although it brings additional latency from blockchain processing.

Adding the Deep Learning component further increases detection accuracy (95.1%) due to vulnerabilities detection in smart contracts, but its constant operation leads to higher memory and energy costs. The absence of proposed severity ranking lowers the overall accuracy of detection, thus showing that intelligent prioritization of threats plays an important role.

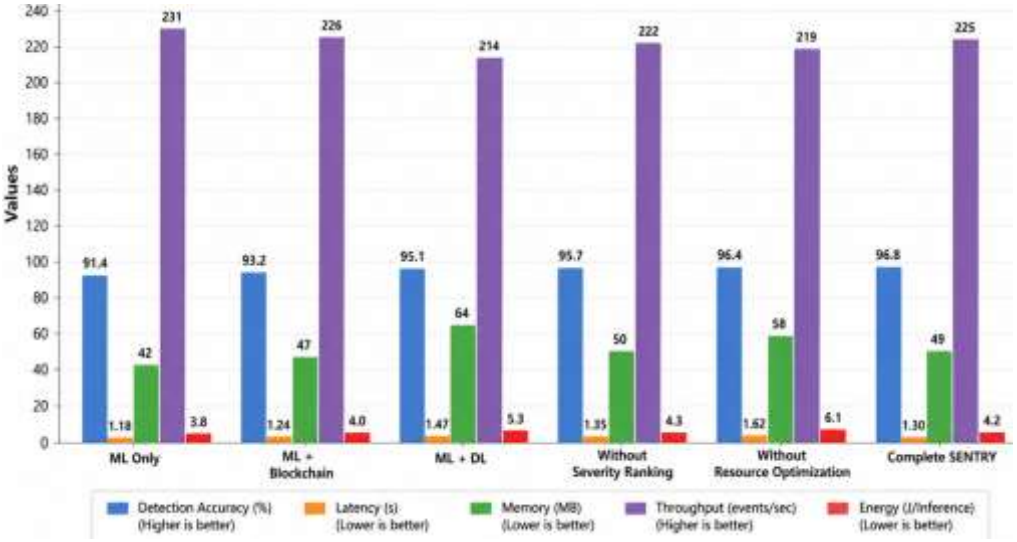


Fig 6. Comparison of Component Contribution Analysis

Performance decrease is highest in the case where the adaptive resource optimization scheme is removed. The continuous running of the BiLSTM model causes latency to increase from 1.30 s to 1.62 s and energy from 4.2 J to 6.1 J, which proves that the proposed adaptive scheduling scheme is effective. The entire SENTRY framework performs best as depicted in (Fig 6.) by ensuring optimal performance in security and high computational efficiency, which ensures the highest accuracy and the lowest memory, latency, and energy use.

Statistical Validation: For the purpose of establishing the statistical significance of the suggested SENTRY framework, all experiments have been conducted five times independently under the same conditions, and the mean values along with standard deviations have been calculated for each of the performance metrics. In addition, paired t-tests have been conducted on the strongest baseline (HMLB) with a 95% confidence interval. The p-values obtained (< 0.01)

prove that the gain in the performance of SENTRY is statistically significant, and not due to any experimental error.

Table IV. Statistical Validation

Metric	Mean ± SD	95% CI	p-value
Accuracy (%)	96.80 ±0.31	96.42–97.18	<0.001
Latency (s)	1.30±0.05	1.24–1.36	<0.001
Memory (MB)	49±1.1	47.8–50.2	<0.001
Throughput	225±2.5	222–228	<0.001
Energy (J)	4.20±0.08	4.10–4.30	<0.001

Smart Contact Validation: The proposed SENTRY-based solution was able to achieve up to 99.0% of smart contract patch success, thus proving its efficiency in identifying common blockchain security risks before the deployment process as shown in (Fig 7.).

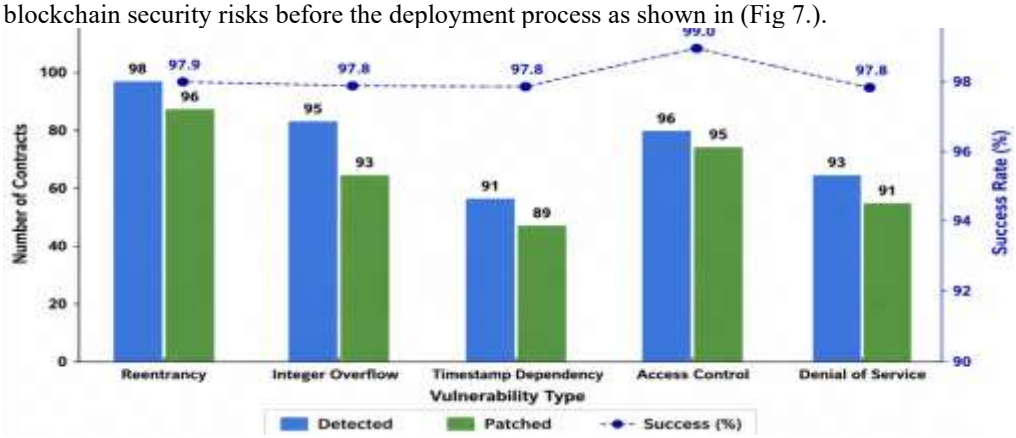


Fig 7. Smart Contact Validation

Scalability Analysis: Complete execution time is given by: $T(N) = \alpha N + \beta$ (16)
Here 'N' denotes the total no. of sensor nodes; α is Processing time per node; and β is Constant overhead for initialization (seconds), comprising the loading of features, initializing of the model, setting up of the blockchain, etc., independent of the number of sensor nodes.

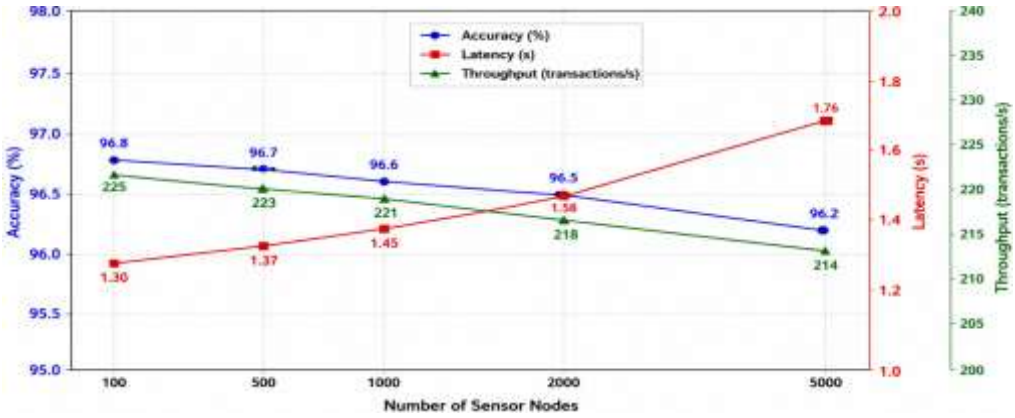


Fig 8. Scalability Analysis

Sensitivity Analysis: Sensitivity index (SI) is defined as:

$$SI = \frac{\Delta Accuracy}{\Delta Parameter} \qquad (17)$$

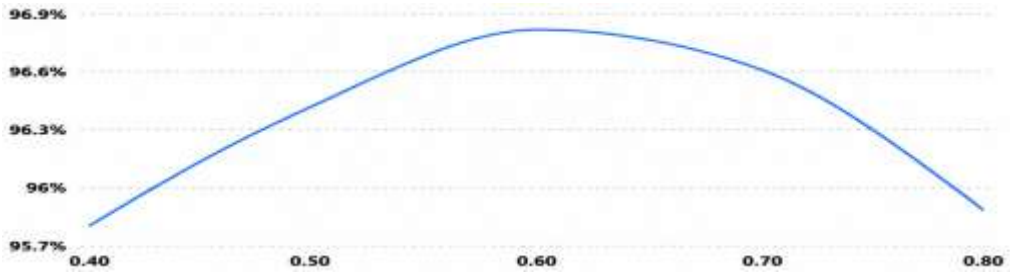


Fig 7. Effect of threat severity threshold on detection accuracy

The sensitivity analysis is represented in (Fig 7.) was performed to determine the effect of important hyperparameters such as the severity of threat, learning rate, and the number of decision trees in the Gradient Boosted Decision Tree framework. Experimental analysis reveals that small changes in these parameters lead to minimal changes in detection accuracy and energy usage, which confirms the effectiveness of the developed methodology. The chosen set of parameters represents the optimal combination for detection accuracy and energy use.

5 Conclusion and Future Work

The proposed work introduces an innovative computational framework known as SENTRY (Secure Energy-efficient Network Threat Response and Yield) that aimed for securing data exchanges in collaborative environment of WSN and IoT. The study combines Ethereum with AI-driven strategy for improving detection of threat accurately, adaptive smart contract management, and enhanced data integrity. The ML model is responsible for determining the threats associated with privacy key compromise and data privacy while the DL model assists in dynamic monitoring followed by revising the management of smart contract.

An extensive analysis of the outcome states SENTRY to accomplish potentially higher detection accuracy with minimal latency. It also exhibited higher resource efficiency while these performance score supports SENTRY to easily withstand adversaries like cryptographic misuse (private key), off-chain risk, defective logic in smart contract, behavioural anomalies (e.g., data injection and zero day attack), as well as network-level attacks (e.g. replay attack, Sybil attack, DDoS attack, etc.). The significant novely is that SENTRY contributes to a attacker profiling mechanism that is capable of ranking threat severity.

This research work has introduced SENTRY which is a new hybrid approach towards dealing with the cybersecurity issues associated with WSN-IoT ecosystems. In this security architecture, Ethereum blockchain has been combined with the gradient boosted decision tree machine learning and BiLSTM deep learning models for efficient threat detection and contract vulnerability analysis. Besides, a ranking scheme for the priority of threats as well as an adaptive resource optimization approach has been introduced. Unlike the existing methods, which only make use of blockchain technology or AI approach separately, SENTRY presents

a complete solution to deal with the cyber threats, secure private keys, detect contract vulnerabilities, and mitigate off-chain attacks in WSN-IoT environment.

The effectiveness of the proposed approach was shown through experimental evaluation by employing the benchmark dataset of Gotham2025 and ToN-IoT. SENTRY scored 96.8% for detection accuracy, 1.30s for latency, 49MB for memory utilization, 225 events/sec for throughput, and 4.2J per inference for energy utilization, surpassing the performance of all baseline approaches. Security evaluation also showed robustness in the face of Denial of Service attack, data injection, private key compromise, off-chain attacks, and smart contract attacks. All in all, through statistical validation, computational complexity analysis, scalability analysis, sensitivity analysis, and smart contract validation, it can be seen that the proposed approach is computationally efficient, energy efficient, scalable, and feasible for real-time implementation in large-scale WSN-IoT systems.

In contrast to multiple state-of-art model, SENTRY has been noted to outperform in terms of adaptability of smart contract, efficiency, threat coverage. However, SENTRY also has a shortcoming as it is not currently design to withstand certain physical form of attacks e.g. node capture attack, physical tampering, and side-channel attacks. This limiting factors will be included in future direction of work. Further, BoT-IoT and Edge-IioTSet can be considered towards permissioned blockchain systems, using federated and explainable learning techniques, and applying the system in real-life implementation through embedded IoT devices. These improvements will help further enhance the scalability, interoperability, and robustness of the system along with enabling applications for cybersecurity and Industry 5.0 enabled WSN-IoT environments.

Authors Contribution: The authors Nandini Prasad K S-Writing Original Paper and methodology; Chaya Puttaswamy - Formal Analysis, Coding; Pavani and Mumtaz - Review & Editing; Suma and Divyashree H B- Conceptualization.

Funding: There has been no significant financial support for this work that could have influenced its outcome

Data Availability: The data availability statement is not applicable

Code Availability: Not applicable

Declarations

Conflict of Interest: We wish to confirm that no known conflicts of interest are associated with this publication.

Ethical Approval: Not applicable

Consent to Participate: Not applicable.

Consent for Publication: Not applicable

References

1. Majid, M., Habib, S., Javed, A. R., Rizwan, M., Srivastava, G., Gadekallu, T. R., & Lin, J. C.-W. (2022). Applications of wireless sensor networks and Internet of Things frameworks in the industry revolution 4.0: A systematic literature review. Sensors (Basel, Switzerland), 22(6), 2087. doi:10.3390/s22062087
2. Poddar, H., & Singh, V. (2022). An analysis of the Internet of Things in wireless sensor network technologies. Retrieved from http://arxiv.org/abs/2209.11675
3. Faris, M., Mahmud, M. N., Salleh, M. F. M., & Alnoor, A. (2023). Wireless sensor network security: A recent review based on state-of-the-art works. International Journal of Engineering Business Management, 15. doi:10.1177/18479790231157220
4. Naskar, S. (2023). Wireless sensor networks challenges and solutions. In J. Sen, M. Yi, F. Niu, & H. Wu (Eds.), Wireless Sensor Networks - Design, Applications and Challenges. London, England: IntechOpen
5. Kandris, D., & Anastasiadis, E. (2024). Advanced wireless sensor networks: Applications, challenges and research trends. Electronics, 13(12), 2268. doi:10.3390/electronics13122268
6. Moslehi, M. M. (2025). Exploring coverage and security challenges in wireless sensor networks: A survey. Computer Networks, 260(111096), 111096. doi:10.1016/j.comnet.2025.111096
7. Ravikumar, K. C., Chiranjeevi, P., Manikanda Devarajan, N., Kaur, C., & Taloba, A. I. (2022). Challenges in internet of things towards the security using deep learning techniques. Measurement. Sensors, 24(100473), 100473. doi:10.1016/j.measen.2022.100473
8. Chaurasia, N., & Kumar, P. (2023). A comprehensive study on issues and challenges related to privacy and security in IoT. E-Prime - Advances in Electrical Engineering, Electronics and Energy, (100158), 100158. doi:10.1016/j.prime.2023.100158
9. Choudhary, A. (2024). Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. Discover Internet of Things, 4(1). doi:10.1007/s43926-024-00084-3
10. Mazhar, T., Talpur, D. B., Shloul, T. A., Ghadi, Y. Y., Haq, I., Ullah, I., … Hamam, H. (2023). Analysis of IoT security challenges and its solutions using artificial intelligence. Brain Sciences, 13(4), 683. doi:10.3390/brainsci13040683
11. Martens, C. D. P., Silva, L. F. da, Silva, D. F., & Martens, M. L. (2022). Challenges in the implementation of internet of things projects and actions to overcome them. Technovation, 118(102427), 102427. doi:10.1016/j.technovation.2021.102427
12. Sabri, O., Al-Shargabi, B., Abuarqoub, A., & Hakami, T. A. (2025). A lightweight encryption method for IoT-based healthcare applications: A review and future prospects. IoT, 6(2), 23. doi:10.3390/iot6020023
13. Radhakrishnan, I., Jadon, S., & Honnavalli, P. B. (2024). Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices. Sensors (Basel, Switzerland), 24(12), 4008. doi:10.3390/s24124008
14. Valluri, B. P., & Sharma, N. (2024). Exceptional key based node validation for secure data transmission using asymmetric cryptography in wireless sensor networks. Measurement. Sensors, 33(101150), 101150. doi:10.1016/j.measen.2024.101150
15. Almarri, S., & Aljughaiman, A. (2024). Blockchain technology for IoT security and trust: A comprehensive SLR. Sustainability, 16(23), 10177. doi:10.3390/su162310177

16. Abrar, I., & Sheikh, J. A. (2024). Current trends of blockchain technology: architecture, applications, challenges, and opportunities. Discover Internet of Things, 4(1). doi:10.1007/s43926-024-00058-5
17. Arshad, Q.-U.-A., Khan, W. Z., Azam, F., Khan, M. K., Yu, H., & Zikria, Y. B. (2023). Blockchain-based decentralized trust management in IoT: systems, requirements and challenges. Complex & Intelligent Systems, 9(6), 6155–6176. doi:10.1007/s40747-023-01058-8
18. Punia, A., Gulia, P., Gill, N. S., Ibeke, E., Iwendi, C., & Shukla, P. K. (2024). A systematic review on blockchain-based access control systems in cloud environment. Journal of Cloud Computing Advances Systems and Applications, 13(1). doi:10.1186/s13677-024-00697-7
19. Zhang, Y., Tang, Y., Li, C., Zhang, H., & Ahmad, H. (2024). Post-quantum secure identity-based signature scheme with lattice assumption for Internet of Things networks. Sensors (Basel, Switzerland), 24(13), 4188. doi:10.3390/s24134188
20. Prajapat, S., Gautam, D., Kumar, P., Jangirala, S., Kumar Das, A., & Sikdar, B. (2025). Secure lattice-based signature scheme for internet of things applications. IEEE Access: Practical Innovations, Open Solutions, 13, 75985–75999. doi:10.1109/access.2025.3565200
21. Zhang, P., Zhang, Q., Ma, T., Liu, M., Zhang, J., & Wang, J. (2025). Lattice-based group signature scheme and its application in IoMT. Journal of Communications and Networks, 27(2), 59–69. doi:10.23919/jcn.2025.000017
22. Shim, K.-A. (2024). Cryptanalysis of Compact Certificateless Aggregate Signature Schemes for HWMSNs and VANETs. IEEE Access: Practical Innovations, Open Solutions, 12, 137634–137641. doi:10.1109/access.2024.3416954
23. Ahmed, A. A., & Barukab, O. M. (2022). Unforgeable digital signature integrated into lightweight encryption based on effective ECDH for cybersecurity mechanism in Internet of Things. Processes (Basel, Switzerland), 10(12), 2631. doi:10.3390/pr10122631
24. Khan, M. S., Chen, T. M., Sathiyanarayanan, M., Mujeerulla, M., & Raja, S. P. (2025). Application of Lenstra–Lenstra–Lovasz on Elliptic Curve Cryptosystem Using IOT Sensor Nodes. Journal of ICT Standardization, 12(4), 381–408. doi:10.13052/jicts2245-800x.1242
25. Nouma, S. E., & Yavuz, A. A. (2024). Lightweight and Resilient Signatures for cloud-assisted embedded IoT systems. Retrieved from http://arxiv.org/abs/2409.13937
26. Thankappan, M., Rifà-Pous, H., & Garrigues, C. (2024). A signature-based wireless intrusion detection system framework for multi-channel man-in-the-middle attacks against protected WI-fi networks. IEEE Access: Practical Innovations, Open Solutions, 12, 23096–23121. doi:10.1109/access.2024.3362803
27. Nallakaruppan, M. K., Somayaji, S. R. K., Fuladi, S., Benedetto, F., Ulaganathan, S. K., & Yenduri, G. (2024). Enhancing security of host-based intrusion detection systems for the internet of things. IEEE Access: Practical Innovations, Open Solutions, 12, 31788–31797. doi:10.1109/access.2024.3355794
28. Ismail, S., Dandan, S., & Qushou, A. (2025). Intrusion Detection in IoT and IIoT: Comparing Lightweight Machine Learning Techniques Using TON_IoT, WUSTL-IIOT-2021, and EdgeIIoTset Datasets. IEEE Access: Practical Innovations, Open Solutions, 13, 73468–73485. doi:10.1109/access.2025.3554083
29. Zachos, G., Mantas, G., Porfyrakis, K., Manuel Camões Sobral de Bastos, J., & Rodriguez, J. (2025). Anomaly-based intrusion detection for IoMT networks: Design, implementation, dataset generation, and ML algorithms evaluation. IEEE Access: Practical Innovations, Open Solutions, 13, 41994–42028. doi:10.1109/access.2025.3547572

30. Saleh, H. M., Marouane, H., & Fakhfakh, A. (2024). Stochastic gradient descent intrusions detection for wireless sensor network attack detection system using machine learning. IEEE Access: Practical Innovations, Open Solutions, 12, 3825–3836. doi:10.1109/access.2023.3349248
31. Kivanç Eren, K., Küçük, K., Özyurt, F., & Alhazmi, O. H. (2025). Simple yet powerful: Machine learning-based IoT intrusion system with smart preprocessing and feature generation rivals deep learning. IEEE Access: Practical Innovations, Open Solutions, 13, 41435–41455. doi:10.1109/access.2025.3547642
32. Popoola, S. I., Tsado, Y., Ogunjinmi, A. A., Sanchez-Velazquez, E., Peng, Y., & Rawat, D. B. (2025). Multi-stage deep learning for intrusion detection in industrial internet of things. IEEE Access: Practical Innovations, Open Solutions, 13, 60532–60555. doi:10.1109/access.2025.3557959
33. Berguiga, A., Harchay, A., & Massaoudi, A. (2025). HIDS-IoMT: A deep learning-based intelligent intrusion detection system for the internet of medical things. IEEE Access: Practical Innovations, Open Solutions, 13, 32863–32882. doi:10.1109/access.2025.3543127
34. Bhavsar, M. H., Bekele, Y. B., Roy, K., Kelly, J. C., & Limbrick, D. (2024). FL-IDS: Federated learning-based intrusion detection system using edge devices for transportation IoT. IEEE Access: Practical Innovations, Open Solutions, 12, 52215–52226. doi:10.1109/access.2024.3386631
35. Bouzeraib, W., Ghenai, A., & Zeghib, N. (2025). Enhancing IoT intrusion detection systems through horizontal federated learning and optimized WGAN-GP. IEEE Access: Practical Innovations, Open Solutions, 13, 45059–45076. doi:10.1109/access.2025.3547255
36. Shahid, U., Zunnurain Hussain, M., Zulkifl Hasan, M., Haider, A., Ali, J., & Altaf, J. (2024). Hybrid intrusion detection system for RPL IoT networks using machine learning and deep learning. IEEE Access: Practical Innovations, Open Solutions, 12, 113099–113112. doi:10.1109/access.2024.3442529
37. Srinivasan, M., & Senthilkumar, N. C. (2025). Intrusion detection and prevention system (IDPS) model for IIoT environments using hybridized framework. IEEE Access: Practical Innovations, Open Solutions, 13, 26608–26621. doi:10.1109/access.2025.3538461
38. Duraibi, S., & Mujawib Alashjaee, A. (2024). Enhancing cyberattack detection using dimensionality reduction with hybrid deep learning on internet of things environment. IEEE Access: Practical Innovations, Open Solutions, 12, 84752–84762. doi:10.1109/access.2024.3411612
39. Aburasain, R. Y. (2024). Enhanced black widow optimization with hybrid deep learning enabled intrusion detection in internet of things-based smart farming. IEEE Access: Practical Innovations, Open Solutions, 12, 16621–16631. doi:10.1109/access.2024.3359043
40. Castiglione, A., Esposito, J. G., Loia, V., Nappi, M., Pero, C., & Polsinelli, M. (2025). Integrating post-quantum cryptography and blockchain to secure low-cost IoT devices. IEEE Transactions on Industrial Informatics, 21(2), 1674–1683. doi:10.1109/tii.2024.3485796
41. Khan, Z. A., Amjad, S., Ahmed, F., Almasoud, A. M., Imran, M., & Javaid, N. (2023). A blockchain-based deep-learning-driven architecture for quality routing in wireless sensor networks. IEEE Access: Practical Innovations, Open Solutions, 11, 31036–31051. doi:10.1109/access.2023.3259982
42. García, L., Cancimance, C., Asorey-Cacheda, R., Zúñiga-Cañón, C.-L., Garcia-Sanchez, A.-J., & Garcia-Haro, J. (2025). Lightweight blockchain for data integrity and traceability in IoT networks. IEEE Access: Practical Innovations, Open Solutions, 13, 81105–81117. doi:10.1109/access.2025.3567773
43. Dhaka, S., Chen, Y.-J., De, S., & Wang, L.-C. (2023). A lightweight stochastic blockchain for IoT data integrity in wireless channels. IEEE Open Journal of Vehicular Technology, 4, 765–781. doi:10.1109/ojvt.2023.3317640

44. Amin Almaiah, M., Ali, A., Alkhdour, T., & Lutfi, A. (2024). Security risk and breach detection approach based blockchain for medical applications. IEEE Access: Practical Innovations, Open Solutions, 12, 171876–171896. doi:10.1109/access.2024.3487217
45. Kang, T., Woo, N., & Ryu, J. (2024). Enhanced lightweight medical sensor networks authentication scheme based on blockchain. IEEE Access: Practical Innovations, Open Solutions, 12, 35612–35629. doi:10.1109/access.2024.3373879
46. Liu, B., Yao, X., Guo, K., & Zhu, P. (2023). Consortium blockchain based lightweight message authentication and auditing in smart home. IEEE Access: Practical Innovations, Open Solutions, 11, 68473–68485. doi:10.1109/access.2023.3293401
47. Subramani, J., Azees, M., Rajasekaran, A. S., Aljaedi, A., Bassfar, Z., & Jamal, S. S. (2024). Blockchain-enabled secure data collection scheme for fog-based WBAN. IEEE Access: Practical Innovations, Open Solutions, 12, 38287–38297. doi:10.1109/access.2024.3351844
48. Anwar, M., Tariq, N., Ashraf, M., Moqurrab, S. A., Alabdullah, B., Alsagri, H. S., & Almjally, A. (2024). BBAD: Blockchain-backed assault detection for cyber physical systems. IEEE Access: Practical Innovations, Open Solutions, 12, 101878–101894. doi:10.1109/access.2024.3404656
49. Sabrina, F., Li, N., & Sohail, S. (2022). A blockchain based secure IoT system using device identity management. Sensors (Basel, Switzerland), 22(19), 7535. doi:10.3390/s22197535
50. Vaiyapuri, T., Shankar, K., Rajendran, S., Kumar, S., Acharya, S., & Kim, H. (2023). Blockchain assisted data edge verification with consensus algorithm for machine learning assisted IoT. IEEE Access: Practical Innovations, Open Solutions, 11, 55370–55379. doi:10.1109/access.2023.3280798
51. Latif, S., Ilyas, M. S. B., Imran, A., Abosaq, H. A., Alzubaidi, A., & Jr., V. K. (2024). Machine learning empowered security and privacy architecture for IoT networks with the integration of blockchain. Intelligent Automation & Soft Computing, 39(2), 353–379. doi:10.32604/iasc.2024.047080
52. Mishra, S. (2023). Blockchain and machine learning-based hybrid IDS to protect smart networks and preserve privacy. Electronics, 12(16), 3524. doi:10.3390/electronics12163524
53. Liu, Z., Jiang, M., Zhang, S., Zhang, J., & Liu, Y. (2023). A smart contract vulnerability detection mechanism based on deep learning and expert rules. IEEE Access: Practical Innovations, Open Solutions, 11, 77990–77999. doi:10.1109/access.2023.3298048
54. Mani Rahnama, S. (2025). DeFiSentinel: AI-enhanced decentralized finance architecture with advanced cryptographic smart contracts for data integrity and threat resilience. IEEE Access: Practical Innovations, Open Solutions, 13, 83107–83122. doi:10.1109/access.2025.3568842
55. Almuqren, L., Mahmood, K., Aljameel, S. S., Salama, A. S., Mohammed, G. P., & Alneil, A. A. (2023). Blockchain-assisted secure smart home network using gradient-based optimizer with hybrid deep learning model. IEEE Access: Practical Innovations, Open Solutions, 11, 86999–87008. doi:10.1109/access.2023.3303087
56. Alruwaili, F. F., Alohali, M. A., Aljaffan, N., Alhashmi, A. A., Mahmud, A., & Assiri, M. (2024). A decentralized approach to smart home security: Blockchain with red-tailed hawk-enabled deep learning. IEEE Access: Practical Innovations, Open Solutions, 12, 14146–14156. doi:10.1109/access.2024.3352502
57. Alami, R., Biswas, A., Shinde, V., Almogren, A., Rehman, A. U., & Shaikh, T. (2024). Blockchain enabled federated learning for detection of malicious internet of things nodes. IEEE Access: Practical Innovations, Open Solutions, 12, 188174–188185. doi:10.1109/access.2024.3511272
58. Tschand, A., Rajan, A. T. R., Idgunji, S., Ghosh, A., Holleman, J., Király, C., … Reddi, V. (2024). MLPerf Power: Benchmarking the energy efficiency of machine learning systems from μWatts to

MWatts for sustainable AI. International Symposium on High-Performance Computer Architecture, 1201–1216. doi:10.1109/HPCA61900.2025.00092

59. Aymone, F. M., & Pau, D. P. (2024). Benchmarking in-Sensor Machine Learning Computing: An extension to the MLCommons-Tiny suite. Information (Basel), 15(11), 674. doi:10.3390/info15110674
60. Belarbi, O., Spyridopoulos, T., Anthi, E., Rana, O., Carnelli, P., & Khan, A. (2025). Gotham dataset 2025: A reproducible large-scale IoT network dataset for intrusion detection and security research. Retrieved from http://arxiv.org/abs/2502.03134