Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.078634

ARTICLE

Cybersecurity for Sustainable Smart Cities: Threat-Resilient and Energy-Conscious Urban Systems

Abdullah Alshammari*

College of Computer Science and Engineering, University of Hafr Albatin, Hafar Albatin, Saudi Arabia

*Corresponding Author: Abdullah Alshammari. Email: dr.abdullah@uhb.edu.sa or alshammari@ieee.org

Received: 05 January 2026; Accepted: 27 February 2026; Published: 08 May 2026

ABSTRACT: The proliferation of Internet of Things (IoT) devices in the infrastructure of smart cities has posed cybersecurity risks like never before, which have direct implications on the sustainability and energy consumption of cities. In this paper, a multi-faceted Threat-Resilient Energy-Conscious Security Framework (TRECSF) is introduced that combines intrusion detection methods powered by deep learning, blockchain-driven data integrity verification mechanism, and energy-aware security protocols in smart city ecosystems to achieve their sustainability. The new Hybrid Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) model is introduced to the proposed architecture, which fulfills the purpose of the study to detect the threat in real time with accuracy of 98.7% and at the same time possesses the ability to execute in a resource-constrained edge device. An Adaptive Energy-Security Optimization (AESO) algorithm that we propose is capable of dynamically adjusting and maintaining the level of security overhead against the level of energy consumption undergoing a 34.2% reductions in power consumption relative to the traditional security systems. The blockchain portion is a consensus mechanism with a lightweight design tailored to IoT settings, which guarantees integrity of the data with a 67% reduced computational power as compared to a conventional Proof-of-Work system. Large-scale simulations conducted on realistic smart city network topologies demonstrate that TRECSF achieves a 45.8% reduction in threat detection latency while ensuring data integrity levels of up to 99.2%. and sustainable energy profiles in a variety of attack scenarios such as Distributed Denial of Service (DDoS), False Data Injection Attacks (FDIA) and Man-in-the-Middle (MitM) attacks. The modular structure of the framework facilitates the seamless integration with the current smart metropolitan infrastructure and facilitates the process of the shift to the carbon-neutral operations of an urban organization.

KEYWORDS: Smart city cybersecurity; sustainable urban systems; IoT security; deep learning intrusion detection; blockchain; energy-efficient security; threat resilience

1 Introduction

Smart city evolution is a paradigm shift of the development of urban areas, and a digital interactive infrastructure promises to increase the quality of life, streamline the effectiveness of its resources and ensure the sustainable evolution of the urban area. It is estimated that more than 70 percent of the global population will arrive in cities in 2030 and create untold pressures in urban infrastructure and services [1]. Smart cities utilize new technologies such as the Internet of Things (IoT), artificial intelligence (AI), and big data analytics to solve them by means of intelligent transport networks, smart grids, automatic waste collection and processing, and smart health services [2]. Nevertheless, this digital transformation presents great cybersecurity weaknesses that endanger the integrity of cities and privacy of many citizens.

Smart city ecosystems are generally linked and thus present a large attack surface that can be used by malicious actors to disrupt them on a large scale. The disastrous nature of cyberattacks on critical city infrastructure has been evidenced in recent events, such as the 2021 Colonial Pipeline ransomware incident which caused fuel supplies to be halted not just in the eastern part of the United States, but also in Ukraine in 2015, which was also targeted by a ransomware attack [3]. The happenings stress the fact that a more formidable cybersecurity architecture is required that is specifically sensitive to the specifics of smart city environments as a place where diverse devices, a variety of communication protocols, and the need to execute operations in real-time generate thorny security issues.

However, at the same time, the sustainability of the smart city processes in terms of environmental responsibility has become a burning issue. The carbon footprint of cities is being added by the amount of energy used by IoT devices and infrastructure, and it is estimated that 2%–4% of the total greenhouse gas emissions in the world are integrated through information and communication technology (ICT) systems [4]. Conventional security measures are computationally expensive, based on cryptographic functions, and continuous monitoring, which increases the energy consumption. This therefore creates inherent conflict between the realization of overall cybersecurity security and operationally efficient activities which are sustainable [5]. Fig. 1 illustrates the conceptual architecture of the proposed Threat-Resilient Energy-Conscious Security Framework (TRECSF) framework within a smart city environment, showing the interaction between IoT devices, edge computing nodes, and cloud services. The figure highlights integrated security functions including encryption, intrusion detection, and access control at the edge layer, along with data flow paths, potential attack vectors, and energy monitoring points across key smart city domains such as transportation, energy, healthcare, and public safety.

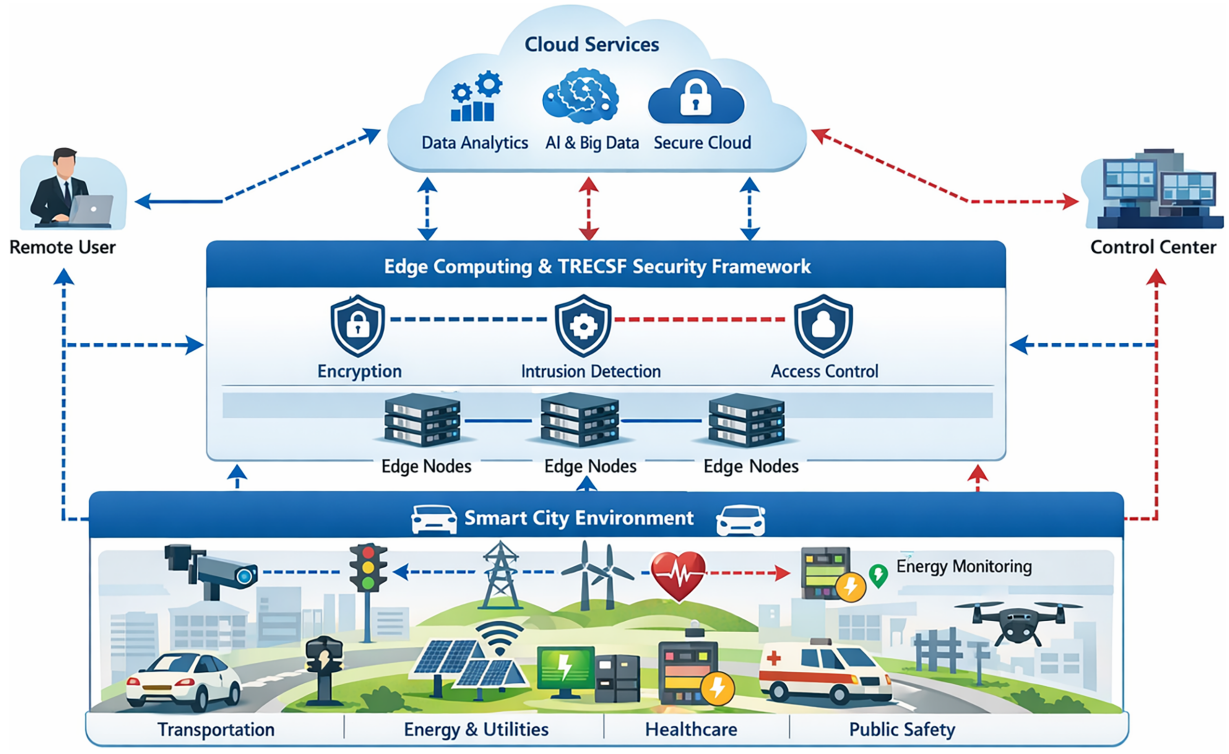

Figure 1: Conceptual architecture of the proposed TRECSF framework illustrating the interaction between IoT devices, edge computing layer, cloud services, and integrated security modules, along with data flow paths, potential attack surfaces, and energy monitoring points across smart city domains.

Existing solutions to smart city cybersecurity are rather biased as they focus either on the detection of threats or energy efficiency as individual goals [6]. Intrusion detection systems based on machine learning have shown encouraging performance in detecting abnormal network behavior, as deep learning architecture have been able to detect abnormal network behavior with better than 95 percent accuracy on benchmark datasets [7]. These systems can however be very computationally intensive and incompatible with both resource limited characteristics of IoT devices deployed across smart city infrastructure. In like manner, blockchain technology has been suggested towards the provision of data integrity as well as decentralized trust solutions in IoT networks [8], however, standard blockchain implementations incur high energy penalties that are opposite to the goals of sustainability.

Federated learning methods are proposed to be another future trend in privacy-guaranteed security analytics in smart cities [9]. Since the model is trained collaboratively but without centralizing sensitive information, federated learning can overcome the issue of privacy and promote the distribution of computational loads across network individuals. However, current federated learning systems on security applications have difficulties associated with convergence of the models in non-independent and identically distributed (non-IID) data conditions found in heterogeneous smart city implementations [10].

The paper will solve these intertwined issues by developing the Threat-Resilient Energy-Conscious Security Framework (TRECSF), which is an overall solution to smart city ecosystems simultaneously providing high-cybersecurity safety and having sustainable energy profiles. The most important works of this work are:

- **Hybrid Deep Learning Architecture:** We propose an innovative and resource-efficient Convolutional Neural Network—Long Short-Term Memory (CNN–LSTM) network architecture for real-time threat detection in IoT-enabled environments, achieving a detection accuracy of 98.7% and a 45.8% reduction in inference latency compared to state-of-the-art models.
- **Adaptive Energy-Security Optimization:** We present an algorithmic method for Adaptive Energy-Security Optimization, which constantly modulates security controls over the level of threats and the energy availability and achieves a 34.2% reduction in security-related energy use without impacting the effectiveness of protection.
- **Lightweight Elasticity:** Our structure involves a type of lightweight Delegated Proof of Stake that we call Data integrity Energetically weighted (DPoS-EW) enumerating 67% less computation compared to standard CRUSTE Resources.
- **Holistic Simulation Framework:** TRECSF is simulated and tested in massive networks on a realistic smart city network topology, and superiority is shown in numerous attack cases and sustainability indicators.
- **Modular Integration Architecture:** In this work, we introduce a design of a framework, which facilitates solving integration with the existing smart city architecture and allows the stepwise implementation and technologies development.

Novelty and Scope Clarification:

It is important to emphasize that the primary novelty of the proposed Threat-Resilient Energy-Conscious Security Framework (TRECSF) lies at the system integration and architectural level rather than in the introduction of entirely new standalone algorithms. The constituent components of TRECSF, including the hybrid CNN–LSTM intrusion detection model, federated learning paradigm, reinforcement learning-based optimization, and blockchain-based trust mechanisms, build upon well-established and widely accepted techniques in the literature. The contribution of this work arises from the unified orchestration of these components within a single, energy-aware security architecture that jointly optimizes cybersecurity

effectiveness, energy consumption, latency, and scalability under realistic smart city constraints. Unlike existing approaches that address these dimensions in isolation, TRECSF enables coordinated, cross-layer decision-making across IoT devices, edge nodes, cloud services, and blockchain infrastructure, thereby advancing the state of practice in sustainable and resilient smart city security systems.

The rest of this paper is structured in the following way. Section 2 is a complete literature review of relevant literature dealing with smart city cybersecurity and energy-efficient security mechanisms and the enabling technologies. Section 3 describes the suggested TRECSF approach, mathematical definition of deep learning structure, energy optimization scheme and blockchain consensus mechanism. In the Section 4, the results of simulation and comparison are given extensively. Section 5 is the conclusion part of the paper where the findings and future research directions are discussed.

2 Literature Review

The convergence between cybersecurity and sustainability in the setting of smart cities has become a primary research topic, and studies existed in several areas, such as IoT security, machine learning-based threat detection, blockchain applications, and energy-efficient computing. This segment is a thorough overview of the state-of-the-art currently, along with a review in thematic section to reveal the basis on which the suggested TRECSF framework can build upon.

2.1 Smart City Cybersecurity Landscape

Smart cities integrate IoT-enabled sensing, communication, and control technologies across critical infrastructures such as smart grids, transportation systems, healthcare services, and public utilities. While this digital transformation enhances efficiency and sustainability, it also introduces significant cybersecurity vulnerabilities. Irfan et al. [1] and Achaal et al. [2] show that cyber–physical attacks, particularly false data injection and coordinated cyber intrusions, can destabilize grid operations and disrupt urban services. These risks are amplified by the large-scale heterogeneity of IoT devices, diverse communication protocols, and decentralized control architectures.

Alomari et al. [3] further emphasize that legacy security mechanisms are inadequate for modern smart city environments due to limited adaptability and lack of contextual awareness. Bibri et al. [4] argue that the convergence of AI, IoT, and big data increases both system intelligence and the attack surface, necessitating integrated cybersecurity strategies aligned with sustainability goals. Overall, the literature highlights the urgent need for scalable, intelligent, and resilient cybersecurity frameworks tailored to the complex ecosystem of smart cities [3,11].

2.2 Machine Learning for Intrusion Detection

Machine learning has become a cornerstone of intrusion detection in smart city networks. Early approaches focused on classical classifiers, but these methods often struggle with high-dimensional data and evolving attack patterns. Elsaeidy et al. [5] demonstrate that hybrid deep learning models significantly outperform traditional techniques in detecting replay and Distributed Denial of Service (DDoS) attacks within smart city environments. More recent studies emphasize deep learning architectures such as CNNs, LSTMs, and hybrid CNN–LSTM models for capturing both spatial and temporal traffic characteristics [12,13].

Surveys by Liao et al. [14] and Dritsas and Trigka [10] show that attention mechanisms and temporal convolutional networks further enhance detection accuracy by modeling long-term dependencies. Advanced models proposed by Alsubaei [15], Susilo et al. [16], and Ghosh et al. [17] achieve high detection rates across

diverse IoT and IIoT attack scenarios. Additionally, Alabbadi and Bajaber [18] propose an explainable AI-based intrusion detection system for IoT data streams, demonstrating that interpretability-aware models can achieve competitive detection performance while enhancing transparency and trust in automated security decisions

However, many of these approaches rely on centralized training and inference, which introduces latency, privacy risks, and scalability constraints in large smart city deployments [14,19]

2.3 Blockchain for IoT Security

Blockchain has been widely explored as a trust-enabling mechanism for IoT security in smart cities. Obaidat et al. [7] provide a comprehensive survey highlighting blockchain's ability to ensure data integrity, immutability, and decentralized authentication. Sefati et al. [6] propose a scalable smart city framework integrating blockchain with federated learning to address trust and data sharing challenges.

Ghadi et al. [9] further demonstrate that hybrid AI–blockchain architectures enhance security resilience in smart grids by combining intelligent threat detection with tamper-proof logging. Aleisa [20] extend this concept by embedding blockchain within zero trust architectures to enforce continuous verification. Despite these advantages, blockchain-based solutions often face challenges related to transaction latency, energy consumption, and scalability, which remain open research issues in smart city contexts [7,21].

2.4 Federated Learning for Privacy-Preserving Security

Federated learning has emerged as a promising paradigm for privacy-preserving intrusion detection by enabling distributed model training without centralized data aggregation. Al-Huthaifi et al. [8] provide a detailed survey of federated learning applications in smart cities, highlighting its effectiveness in mitigating data leakage risks. Priyadarshini [22] demonstrate that combining federated learning with split learning improves anomaly detection performance while preserving data confidentiality.

Ragab et al. [23] introduce an advanced federated learning framework for cyberthreat detection in sustainable smart cities, achieving strong privacy guarantees and high detection accuracy. However, federated learning systems are vulnerable to poisoning attacks, communication overhead, and non-IID data distributions, which can degrade model performance if not properly addressed [8,22]

2.5 Smart Grid and Energy System Security

Smart grids represent one of the most critical components of smart city infrastructure. Irfan et al. [1] and Paul et al. [24] highlight that cyberattacks targeting measurement and control data can cause cascading failures and economic losses. Tightiz et al. [25] demonstrate that AI-based detection systems improve resilience against advanced cyberattacks in smart grids.

Rajaperumal and Columbus [26] emphasize the role of AI in enabling secure and sustainable energy systems, while Kabir et al. [2,27] discuss the integration of digital twins for proactive monitoring and threat detection. Emerging studies also explore large language models for grid cybersecurity analysis, offering new perspectives on attack detection and mitigation strategies [28]. Despite these advances, ensuring real-time protection with minimal energy overhead remains a key challenge [21].

2.6 Autonomous Vehicles and Transportation Security

Autonomous and connected vehicles introduce unique cybersecurity challenges due to their safety-critical nature and reliance on real-time communication. Durlik et al. [29] argue that current automotive cybersecurity measures are insufficient to counter sophisticated cyberattacks targeting perception and

control systems. Tanaji and Roychowdhury [30] provide a comprehensive survey of attack vectors and mitigation strategies for connected and autonomous vehicles.

Fernández Llorca et al. [31] further examine cybersecurity, transparency, robustness, and fairness challenges in AI-driven autonomous systems. Reis [32] demonstrate that AI-driven anomaly detection improves security in 5G-enabled vehicular networks. However, the literature indicates a lack of unified frameworks that integrate cybersecurity, energy efficiency, and trust management for transportation systems within smart cities [29,31].

2.7 Zero Trust Architecture for Smart Cities

Zero Trust Architecture (ZTA) has gained attention as a security model that eliminates implicit trust within networks. Liu et al. [33] provide a comprehensive overview of zero trust principles and their application in IoT environments. Hussain [11] propose a blockchain-enabled zero trust framework that enhances privacy and continuous authentication in IoT systems.

The paper [34] extend zero trust concepts to cognitive city networks, emphasizing AI-driven policy enforcement and adaptive trust evaluation. While ZTA improves security posture, its integration with large-scale smart city infrastructures remains complex due to operational overhead and interoperability challenges [32].

2.8 Energy-Efficient Security Mechanisms

Energy efficiency is a critical requirement for cybersecurity mechanisms in smart cities, where large-scale IoT deployments and edge devices operate under strict power and resource constraints. Unlike traditional enterprise systems, continuous security monitoring in smart cities can significantly increase computational and communication overhead, directly impacting system sustainability. Hussain [11] emphasize that security solutions for smart cities must balance protection strength with long-term energy efficiency to ensure operational viability. Esfandi et al. [35] further underscore these challenges by reviewing the promises and limitations of urban energy planning in smart cities, demonstrating that energy-aware design must be embedded across all infrastructure layers, including cybersecurity, to achieve sustainable urban operation.

Several studies address smart city security from an architectural and threat-analysis perspective but largely overlook energy considerations. Achaal et al. [2] analyze smart grid cybersecurity architectures and countermeasures, providing a detailed taxonomy of attacks and defenses. However, their framework does not consider the energy cost of persistent intrusion detection or real-time monitoring. Similarly, Alomari et al. [3] focus on cybersecurity incidents and attack characterization in smart grids, offering valuable contextual insights but lacking deployable, energy-aware security mechanisms.

Recent AI-driven approaches improve detection accuracy but introduce additional energy overhead. Reis [32] propose an AI-based anomaly detection model for securing IoT devices in 5G-enabled smart cities, achieving effective threat detection at scale. Nonetheless, continuous model inference and frequent updates may increase energy consumption when deployed across dense IoT networks. Ragab et al. [23] partially address this issue by integrating federated learning to reduce data transmission, although repeated training rounds still impose computational costs on edge devices.

Blockchain-enabled security frameworks further strengthen trust and data integrity but often exacerbate energy consumption. Sefati et al. [6] combine blockchain and federated learning to secure smart city IoT systems, enhancing decentralization and privacy. However, blockchain operations introduce additional overhead, making energy-aware optimization essential for practical deployment.

To summarize these observations, Table 1 presents a comparative analysis of existing smart city cybersecurity frameworks, highlighting their core techniques, scalability, and limitations with respect to energy efficiency. The comparison reveals that while recent frameworks improve security and privacy, explicit energy-aware security design remains limited, motivating the need for integrated and lightweight solutions.

Table 1: Comparative analysis of existing smart city cybersecurity frameworks. Bold text highlights the proposed TRECSF framework's superior performance in energy awareness and scalability, underscoring its integrated and holistic approach that addresses critical gaps in existing smart city cybersecurity solutions.

Study	Core Technique	Energy Awareness	Scalability	Key Limitations
Achaal et al. [2]	Smart grid cybersecurity architectures and countermeasures	No	Limited	Provides threat taxonomy and architectural analysis but lacks runtime intrusion detection and energy-aware security mechanisms
Alomari et al. [3]	Smart grid attack analysis and incident review	No	Limited	Focuses on attack characterization without proposing adaptive or deployable security frameworks
Bibri et al. [4]	AI, IoT, and big data integration for sustainable smart cities	Partial	Conceptual	Identifies security challenges at a conceptual level but does not implement concrete cybersecurity enforcement mechanisms
Liao et al. [14]	Deep learning-based intrusion detection survey	No	Moderate	Highlights high detection accuracy but assumes cloud-centric deployments with high energy consumption
Hossain [12]	Deep learning-based intrusion detection for IoT	No	Moderate	Improves detection performance but does not address energy efficiency or edge resource constraints
Alsubaei [15]	Deep learning-based IoT intrusion detection	No	Moderate	Achieves high accuracy but lacks energy-aware optimization and sustainability considerations
Susilo et al. [16]	Hybrid deep learning intrusion detection system	No	Limited	Enhances attack detection but does not consider scalability and energy efficiency

(Continued)

Table 1 (continued)

Study	Core Technique	Energy Awareness	Scalability	Key Limitations
Obaidat et al. [7]	Blockchain-based IoT security survey	Partial	Moderate	Identifies blockchain security benefits but highlights high computational and energy overhead
Sefati et al. [6]	Blockchain and federated learning for smart cities	Partial	Moderate	Improves privacy and trust but introduces blockchain latency and communication overhead
Ghadi et al. [9]	Hybrid AI–blockchain security framework for smart grids	Partial	Moderate	Reduces false positives but does not optimize energy usage for large-scale deployment
Aleisa [20]	Blockchain-enabled zero trust architecture	No	Moderate	Strong privacy guarantees but introduces computational and energy overhead
Al-Huthaifi et al. [8]	Federated learning for smart city security survey	Partial	Moderate	Highlights privacy benefits but reports convergence issues under non-IID data
Priyadarshini [22]	Federated and split learning-based anomaly detection	Partial	Moderate	Improves privacy but lacks adaptive energy-aware security control
Tightiz et al. [25]	AI-based cyberattack detection for smart grids	No	Domain-specific	High detection accuracy achieved without cross-domain scalability or sustainability focus
Ishaque et al. [34]	AI-driven zero trust architecture for smart cities	No	Moderate	Enhances access control but excludes integrated intrusion detection and energy optimization
TRECSF (This Work)	**Deep Learning + Federated Learning + Energy-Aware Blockchain**	**Yes**	**High**	Validated via large-scale simulations; real-world deployment remains future work

Recent research has increasingly emphasized the integration of edge intelligence, federated learning, and lightweight blockchain mechanisms to address the dual challenges of cybersecurity and energy efficiency in smart city environments. Velaga et al. [36] provide a comprehensive analysis of edge AI architectures for smart cities, highlighting the necessity of energy-aware inference and resource-constrained deployment at the edge. Guo et al. [37] propose a lightweight IoT–blockchain framework that reduces consensus and communication overhead, making it more suitable for large-scale smart city deployments. In parallel, Ali et al. [38] examine the convergence of blockchain and federated learning in edge–fog–cloud environments,

identifying scalability and coordination challenges in decentralized security architectures. Furthermore, Babayomi et al. [39] demonstrate that energy-aware federated learning combined with blockchain can significantly improve the resilience and efficiency of distributed energy systems. While these studies advance sustainable smart city security, they largely treat energy efficiency, intrusion detection, federated learning, and blockchain security as loosely coupled components rather than a jointly optimized framework.

2.9 Research Gaps and Motivation

Although recent studies have advanced intrusion detection, federated learning, blockchain security, and zero trust models for smart cities, a closer examination reveals several unresolved limitations. First, a large portion of existing intrusion detection systems prioritizes detection accuracy while treating energy consumption and scalability as secondary concerns. Such approaches may perform well in controlled environments but become impractical for large-scale smart city deployments where thousands of energy-constrained IoT devices operate continuously. Second, blockchain- and federated learning–based security frameworks often improve privacy and trust but introduce non-trivial computational and communication overhead. Lightweight consensus mechanisms and adaptive aggregation strategies are still underexplored, and many reported solutions assume ideal network conditions or sufficient energy availability, limiting their real-world applicability in heterogeneous smart city environments. Third, zero trust architectures proposed for smart cities primarily focus on access control and authentication. They rarely incorporate AI-driven threat intelligence that adapts dynamically to evolving attack patterns, and they generally lack energy-aware optimization mechanisms. As a result, trust enforcement remains largely static and disconnected from real-time security analytics and resource constraints.

These limitations collectively indicate the absence of a unified framework that simultaneously addresses intelligent intrusion detection, privacy preservation, trust management, energy efficiency, and scalability. The proposed TRECSF framework is motivated by this gap. Unlike prior approaches that treat these components in isolation, TRECSF integrates a hybrid CNN–LSTM detection model, adaptive energy-security optimization, lightweight blockchain consensus, and energy-aware federated learning into a single cohesive architecture. This design explicitly targets the joint optimization of security effectiveness, latency, and energy consumption, which is critical for sustainable smart city operation. Despite these advancements, challenges remain, including real-world deployment validation, long-term adaptability under evolving attack strategies, and integration with city-scale operational policies. Addressing these open issues provides a clear direction for future research while positioning TRECSF as a concrete step beyond existing fragmented solutions.

3 Proposed Methodology

This section is a comprehensive approach of the Threat-Resilient Energy-Conscious Security Framework (TRECSF). The framework has four unified aspects including (1) the Hybrid CNN-LSTM Intrusion Detection System, (2) the Adaptive Energy-Security Optimization algorithm, (3) the Lightweight Blockchain Consensus mechanism, and (4) the Federated Learning coordination protocol. The problem formulation and system model are discussed and then each of them is elaborated.

From a methodological standpoint, TRECSF should be viewed as a system-level integration framework rather than a collection of isolated algorithmic contributions. Each functional module is deliberately designed to interact with and inform the others through shared control signals, energy-awareness constraints, and feedback loops. The hybrid CNN–LSTM module provides threat intelligence, which directly influences adaptive energy-security decisions, blockchain validation policies, and federated learning coordination. This tightly coupled design enables global optimization across security, energy efficiency, and operational latency, distinguishing TRECSF from prior smart city security solutions that deploy similar techniques in a loosely coupled or siloed manner.

3.1 Problem Formulation and System Model

Consider a smart city network topology represented as a graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ denotes the set of n IoT devices and edge nodes, and $E \subseteq V \times V$ represents the communication links between devices. Each device v_i is characterized by a tuple (c_i, m_i, e_i, s_i) representing computational capacity, memory, energy availability, and security level respectively.

The network traffic at device v_i during time interval t is represented as a sequence of flow records:

$$F_i(t) = \{f_{i,1}, f_{i,2}, \ldots, f_{i,k}\} \tag{1}$$

where $F_i(t)$ denotes the set of network flow records observed at device v_i during time interval t. Each element $f_{i,j}$ is a single traffic stream with a feature vector of the source and destination distribution, protocol type, packet size distribution, and temporal information. k will be related to the number of flow records recorded during the specified time period.

The network traffic observed at device v_i during time interval t is represented as a sequence of flow records, as defined in Eq. (1). Here, $F_i(t)$ denotes the set of network flow records captured at device v_i, where each element $f_{i,j}$ corresponds to an individual traffic flow characterized by features such as source and destination distribution, protocol type, packet size statistics, and temporal attributes. The parameter k represents the total number of flow records observed within the specified time interval.

The joint security–energy optimization objective of the proposed TRECSF framework is formally defined in (2), subject to operational latency and energy constraints expressed in (3).

The main goal of TRECSF is to maximize the level of security S and minimize a certain energy consumption E in the course of the operational constraint:

$$\max_x \alpha \cdot S(x) - \beta \cdot E(x) \tag{2}$$

$$\text{s.t. } L(x) \leq L_{\max}, E(x) \leq E_{budget} \tag{3}$$

where x denotes the configuration vector of security control parameters, including intrusion detection depth, monitoring frequency, and blockchain validation settings. The objective function jointly optimizes security effectiveness and energy efficiency, where $S(x)$ represents the achieved security level and $E(x)$ denotes the corresponding energy consumption. The weighting parameters α and β control the relative importance of security performance and energy efficiency, respectively, and were selected empirically through grid search to achieve a balanced trade-off under smart city operational constraints. The latency constraint $L(x) \leq L_{\max}$ ensures real-time threat detection suitability, while $E(x) \leq E_{budget}$ enforces compliance with the limited energy availability of resource-constrained IoT devices.

As shown in (2), the framework seeks to maximize security effectiveness while minimizing energy consumption. The operational constraints in (3) enforce real-time responsiveness and compliance with the energy budget of resource-constrained IoT devices.

The composite security level used in the optimization objective is computed as a weighted combination of detection accuracy, false positive rate, and attack coverage, as defined in (4). The security level $S(x)$ is defined as a function of detection accuracy A, false positive rate FPR, and coverage C:

$$S(x) = \omega_1 \cdot A(x) + \omega_2 \cdot (1 - FPR(x)) + \omega_3 \cdot C(x) \tag{4}$$

where $A(x)$ denotes the intrusion detection accuracy achieved under configuration x, $FPR(x)$ represents the false positive rate, and $C(x)$ indicates the attack coverage reflecting the proportion of detected attack classes. The coefficients ω_1, ω_2, and ω_3 are non-negative weighting parameters satisfying $\omega_1 + \omega_2 + \omega_3 = 1$,

used to balance the relative importance of detection accuracy, false alarm minimization, and coverage. These weights were initially assigned equal values and subsequently refined empirically through validation experiments to reflect the operational priorities of smart city security systems.

The weighting coefficients in (4) enable flexible prioritization of detection accuracy, false alarm reduction, and attack coverage based on smart city security requirements.

Fig. 2 presents the detailed architecture of the proposed TRECSF framework, showing the interaction between the CNN–LSTM intrusion detection module, the Adaptive Energy-Security Optimizer, the lightweight blockchain layer, and the federated learning coordinator.

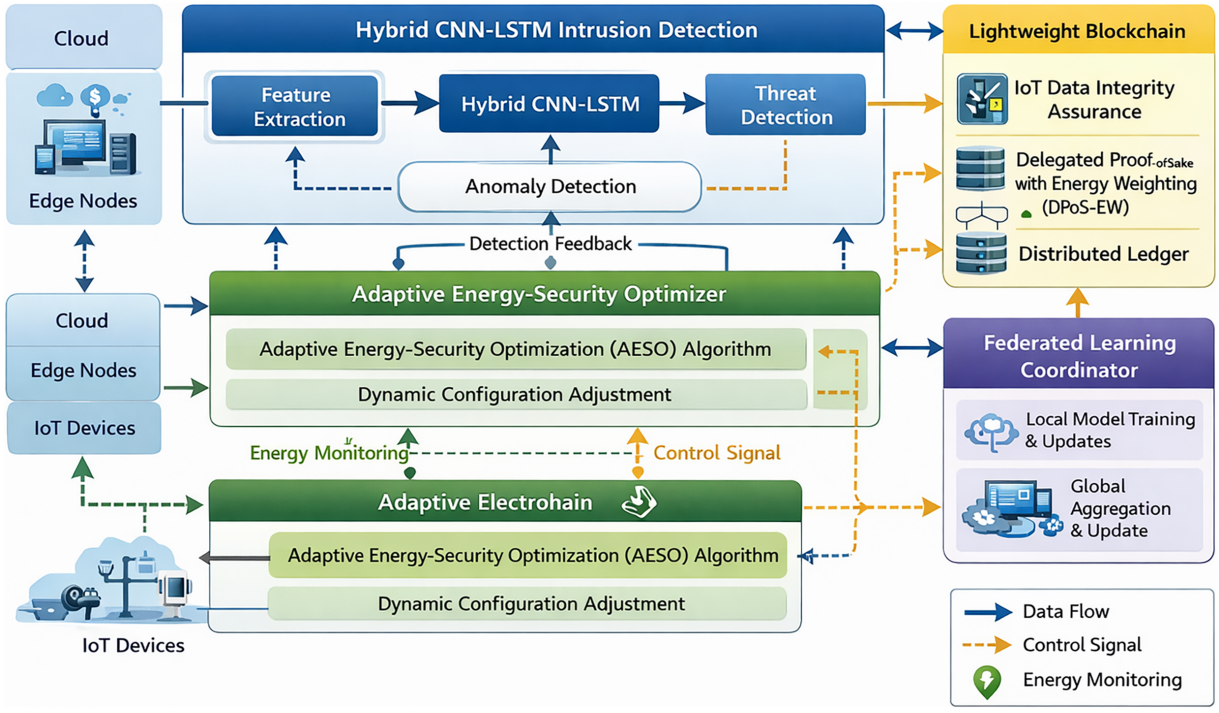

Figure 2: Detailed architecture of the TRECSF framework illustrating the interaction between the Hybrid CNN-LSTM intrusion detection module, Adaptive Energy-Security Optimizer, Lightweight Blockchain layer, and Federated Learning coordinator. Arrows indicate data flow and control signals between components.

3.2 Hybrid CNN-LSTM Intrusion Detection Architecture

The intrusion detection module uses a hybrid architecture integrating space features detection of CNN with the time sequence operations of LSTM networks. The given design allows analyzing both the packet-level features and the traffic flow patterns simultaneously.

3.2.1 Feature Representation

Network traffic is represented as a three-dimensional tensor suitable for convolutional processing. For a time window of T intervals, the input tensor $X \in \mathbb{R}^{T\times H\times W}$ is constructed where H represents the number of flow features and W represents the feature embedding dimension.

Prior to feature extraction, all raw traffic features are normalized to improve numerical stability, as described in (5). The raw flow features are first normalized using min-max scaling:

$$\hat{f}_{i,j}^{(k)} = \frac{f_{i,j}^{(k)} - \min\left(f^{(k)}\right)}{\max\left(f^{(k)}\right) - \min\left(f^{(k)}\right)} \tag{5}$$

where $f_{i,j}^{(k)}$ denotes the k-th raw feature value of the j-th network flow observed at device v_i. The terms $\min\left(f^{(k)}\right)$ and $\max\left(f^{(k)}\right)$ represent the minimum and maximum values of the k-th feature computed over the training dataset. The normalized feature $\hat{f}_{i,j}^{(k)}$ is obtained using min–max scaling to map feature values into the range $[0,1]$, ensuring numerical stability during training and preventing features with larger magnitudes from dominating the learning process.

3.2.2 Convolutional Feature Extraction

The CNN component consists of multiple convolutional layers that extract hierarchical spatial features:

$$H^{(l)} = \sigma\left(W^{(l)} * H^{(l-1)} + b^{(l)}\right) \tag{6}$$

where $H^{(l)}$ is the activation at layer l, $W^{(l)}$ and $b^{(l)}$ are learnable weights and biases, $*$ denotes the convolution operation, and σ is the ReLU activation function.

Batch normalization is applied after each convolutional layer to stabilize training:

$$\hat{H}^{(l)} = \gamma \cdot \frac{H^{(l)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \tag{7}$$

where $H^{(l)}$ denotes the activation output of the l-th convolutional layer, and μ_B and σ_B^2 represent the mean and variance computed over the current mini-batch, respectively. The parameter ε is a small constant added for numerical stability. The learnable parameters γ and β perform scale and shift operations, enabling the network to preserve representational capacity after normalization. Batch normalization is applied to reduce internal covariate shift, accelerate convergence, and improve training stability, particularly in deep architectures deployed under resource-constrained smart city environments.

The convolutional feature extraction process defined in (6) and stabilized through batch normalization in (7) enables robust spatial pattern learning from network traffic data.

3.2.3 LSTM Temporal Modeling

The LSTM component processes the sequence of CNN-extracted features to capture temporal dependencies. Temporal dependencies in network traffic are modeled using LSTM units, whose internal gating mechanisms are governed by (8)–(13). The LSTM cell operations are defined as:

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \tag{8}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \tag{9}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right) \tag{10}$$

$$\tilde{c}_t = \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{11}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{12}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{13}$$

where x_t denotes the input feature vector at time step t derived from the CNN feature extractor, and h_{t-1} and c_{t-1} represent the hidden state and cell state from the previous time step, respectively. The vectors i_t,

f_t, and o_t correspond to the input, forget, and output gates, which regulate information flow within the LSTM cell. The candidate cell state $\tilde{c}_t$ captures newly computed information, while the updated cell state c_t integrates retained past memory and new content through element-wise multiplication $\odot$. The weight matrices W_{x*} and W_{h*} and bias terms b_* are learnable parameters. The nonlinear functions $\sigma(\cdot)$ and $\tanh(\cdot)$ denote the sigmoid and hyperbolic tangent activations, respectively. This gated mechanism enables the model to capture long-term temporal dependencies in network traffic, which is essential for detecting time-evolving and stealthy cyberattacks in smart city environments. The gated updates in (8)–(13) allow the model to retain long-term dependencies and capture evolving attack behaviors over time.

3.2.4 Attention Mechanism

To emphasize temporally significant traffic segments, an attention mechanism is applied as defined in (14)–(16). An attention mechanism is incorporated to enable the model to focus on the most relevant temporal segments:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)} \tag{14}$$

$$e_t = v^T \tanh(W_h h_t + b_h) \tag{15}$$

where α_t denotes the normalized attention weight assigned to the hidden state h_t at time step t, reflecting its relative importance in the sequence. The scalar score e_t is computed using a learnable context vector v, weight matrix W_h, and bias b_h, followed by a $\tanh(\cdot)$ nonlinearity. The softmax operation ensures that $\sum_{t=1}^{T} \alpha_t = 1$, enabling probabilistic interpretation of attention weights. This attention mechanism allows the model to emphasize temporally salient traffic patterns, improving detection of subtle and temporally distributed cyberattacks in smart city IoT environments.

The context vector is computed as:

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{16}$$

where c denotes the context vector obtained as the weighted sum of LSTM hidden states over the temporal window of length T. The attention weights α_t determine the contribution of each hidden state h_t to the last representation, which lets the model combine the temporally vital feature but represses less relevant parts. This context vector has turned out to be a concise overview of the most pertinent time trends within the network traffic, which is later employed to classify the attacks. The attention-weighted context vector in (16) summarizes the most informative temporal features for final intrusion classification.

3.2.5 Classification Layer

The final attack classification is performed using a softmax-based output layer, as formulated in (17). The final classification employs a fully connected layer with softmax activation:

$$P(y|X) = \text{softmax}(W_c \cdot c + b_c) \tag{17}$$

where $P(y|X)$ denotes the posterior probability distribution over the attack classes given the input feature tensor X. The vector c is the context representation produced by the attention mechanism, while W_c and b_c are the classification layer biases and weight matrix which can be learnt. The softmax uses results of the linear format and topics them into a normalized probability distribution to allow multi-subsequently classification by the intrusion by choosing the most highly legitimized posterior distribution multi Smartmax.

This model is trained in terms of categorical cross-entropy loss:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N}\sum_{k=0}^{K} y_{i,k}\log\left(\hat{y}_{i,k}\right) \tag{18}$$

where $\mathcal{L}_{CE}$ denotes the categorical cross-entropy loss used for multi-class intrusion classification, N is the total number of training samples, and K represents the number of attack classes. The binary indicator $y_{i,k}$ equals 1 if sample i belongs to class k and 0 otherwise, while $\hat{y}_{i,k}$ denotes the predicted posterior probability of class k for sample i obtained from the softmax layer. This loss function penalizes incorrect predictions and encourages the model to assign high probability to the correct attack class.

Model training is guided by a categorical cross-entropy loss function defined in (18).

3.3 Adaptive Energy-Security Optimization (AESO)

The AESO algorithm is a dynamically changing algorithm that varies the security parameters according to the level of threats and the energy limitations. The optimization uses a reinforcement based learning whereby the agent is taught to adopt the best policy on security configuration.

3.3.1 State Space Definition

The Adaptive Energy-Security Optimization (AESO) module models system dynamics using a reinforcement learning formulation, with the state representation defined in (19). The state at time t is defined as:

$$s_t = \left(TL_t, E_t, L_t, A_t\right) \tag{19}$$

where TL_t is the current threat level, E_t is the available energy, L_t is the current detection latency, and A_t is the recent detection accuracy.

3.3.2 Action Space

The AESO agent selects security adaptation actions from the discrete action space defined in (20). The action space consists of security configuration adjustments:

$$a_t \in \left\{\Delta_{\text{scan}}, \Delta_{\text{depth}}, \Delta_{\text{freq}}\right\} \tag{20}$$

where a_t denotes the action selected by the Adaptive Energy-Security Optimization (AESO) agent at time step t. The action space consists of discrete security configuration adjustments, including Δ_{scan} for modifying intrusion scanning intensity, Δ_{depth} for adjusting analysis depth within the detection model, and Δ_{freq} for changing monitoring and update frequency. These actions enable dynamic adaptation of security mechanisms in response to varying threat levels and energy availability in smart city IoT environments.

3.3.3 Reward Function

Policy optimization is carried out using the Proximal Policy Optimization (PPO) objective function defined in (21). The reward function balances security and energy objectives:

$$r_t = \lambda_1 \cdot \Delta S_t - \lambda_2 \cdot \Delta E_t - \lambda_3 \cdot 1\left[L_t > L_{\max}\right] \tag{21}$$

where r_t denotes the reward received by the AESO agent at time step t. The terms ΔS_t and ΔE_t represent the changes in security level and energy consumption, respectively, resulting from the selected action. The indicator function $1\left[L_t > L_{\max}\right]$ imposes a penalty when the detection latency L_t exceeds the maximum

allowable threshold $L_{\max}$. The weighting parameters λ_1, λ_2, and λ_3 control the relative importance of maximizing security improvements, minimizing energy expenditure, and enforcing real-time latency constraints, and were empirically tuned through validation experiments to reflect smart city operational priorities.

3.3.4 Policy Optimization

Delegate selection in the lightweight blockchain layer incorporates energy awareness using the scoring function defined in (22). The optimal policy is learned using Proximal Policy Optimization (PPO):

$$\mathcal{L}^{CLIP}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \tag{22}$$

where $\mathcal{L}^{CLIP}(\theta)$ denotes the clipped surrogate objective used in Proximal Policy Optimization (PPO) to update the policy parameters θ. The probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ compares the likelihood of action a_t under the updated and previous policies, while $\hat{A}_t$ represents the estimated advantage at time step t. Clipping operation having the parameter ϵ limits the updates in the policy inside a trust region and avoiding overly large changes in policy, enhances the stability of training. The formulation allows the adaptive energy-security policies to be learned reliably under smart cities conditions, which dynamically change.

3.4 Lightweight Blockchain Consensus (DPoS-EW)

The Delegated Proof-of-Stake with Energy Weighting (DPoS-EW) system guarantees integrity of data and uses less energy.

3.4.1 Delegate Selection

Block validation follows a Byzantine fault-tolerant majority rule as specified in (23). Delegates are selected based on a combined score of stake and energy efficiency:

$$Score_i = \gamma \cdot \frac{S_i}{\sum_j S_j} + (1-\gamma) \cdot \frac{E_{eff,i}}{\sum_j E_{eff,j}} \tag{23}$$

where S_i is the stake of node i, $E_{eff,i}$ is its energy efficiency rating, and γ is a balancing parameter.

3.4.2 Block Validation

Transaction prioritization accounts for urgency and energy cost using the formulation in (24). Block validation requires 2/3 majority among selected delegates:

$$Valid(B) = 1\left[\sum_{d \in D} v_d(B) \geq \frac{2|D|}{3}\right] \tag{24}$$

where Valid (B) denotes the validity status of block B, and $1[\cdot]$ is the indicator function that returns 1 if the condition is satisfied and 0 otherwise. The set D represents the elected delegate nodes participating in the consensus process, and $v_d(B) \in \{0,1\}$ denotes the validation vote cast by delegate d for block B. A block becomes valid when a two-thirds majority of the delegates give it approval, which gives the lightweight blockchain consensus mechanism Byzantine fault tolerance as well as challenges to malicious or faulty nodes.

3.4.3 Energy-Weighted Transaction Priority

Transaction prioritization accounts for urgency and energy cost using the formulation in (25). Transaction ordering considers both urgency and energy cost:

$$Priority\,(tx) = \frac{Urgency\,(tx)}{EnergyCost\,(tx)^{\delta}} \tag{25}$$

where $Priority\,(tx)$ denotes the scheduling priority assigned to transaction tx within the blockchain layer. The function $Urgency\,(tx)$ represents the time-sensitivity or criticality of the transaction, while $EnergyCost\,(tx)$ roughly calculates the computational cost and energy cost incurred to validate it and include it in a block. The parameter d is an adjustable weighting variable which is used to tradeoff between urgency and energy efficiency. This formulation focuses on the most important transactions and punishes the ones that consume a lot of energy, thus aiding energy conscious transaction sequencing that is adequate to carry out in resource-limited smart city IoT settings.

3.5 Federated Learning Coordination

The federated learning aspect allows the distributed learning of models and data privacy.

3.5.1 Local Model Update

Local model updates at participating nodes are performed according to (26). Each participating node k performs local training:

$$w_k^{(t+1)} = w_k^{(t)} - \eta \nabla \mathcal{L}_k \left(w_k^{(t)} \right) \tag{26}$$

where $w_k^{(t)}$ and $w_k^{(t+1)}$ denote the local model parameters of node k before and after the update at training round t, respectively. The learning rate η controls the step size of the optimization, and $\nabla \mathcal{L}_k \left(w_k^{(t)} \right)$ represents the gradient of the local loss function computed using the private dataset stored at node k. A local update is based on the usual stochastic gradient descent and allows decentralizing the model training without violating the data privacy in the federated learning process.

3.5.2 Aggregation with Energy Weighting

The global model aggregation incorporates energy-aware weighting, as defined in (27). Global model aggregation incorporates energy awareness:

$$w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k \cdot E_{avail,k}}{\sum_j n_j \cdot E_{avail,j}} w_k^{(t+1)} \tag{27}$$

where $w^{(t+1)}$ denotes the aggregated global model parameters at communication round $t+1$, and $w_k^{(t+1)}$ represents the locally updated model parameters of node k. The term n_k indicates the size of the local training dataset at node k, while $E_{avail,k}$ denotes the energy level of that node when it was aggregated. This energy-weighted aggregation algorithm gives nodes with larger datasets and more access to energy preferential agency, thus enhancing the stability of the convergence with fewer chances of overloading energy-starved devices. The normalization term will make sure that the aggregation weights are equal to one.

3.6 Algorithm Summary

The elaborate pseudocode of the entire TRECSF structure has been shifted to the Appendix in order to make the main manuscript more lucid and concise. The algorithm unifies the intrusion detection module, adaptive energy-security optimization, blockchain consensus, and federated learning modules to

one workflow. These components are described at the very high level in the main text, but the overall step-by-step algorithmic process can be found in the appendix (Appendix A) in the form of Algorithm A1.

4 Results and Discussion

This section presents comprehensive simulation results evaluating the TRECSF framework across multiple performance dimensions. We describe the simulation environment, dataset characteristics, evaluation metrics, and comparative analysis against state-of-the-art approaches.

4.1 Simulation Environment

A custom-made smart city networks simulator written in Python 3.9 and using TensorFlow 2.12 as deep learning elements was used to carry out the simulations. The intrusion detection trials make use of the publicly available CICIDS2017 dataset, which is offered by the Canadian Institute of Cybersecurity, University of New Brunswick, and is available at https://www.unb.ca/cic/datasets/ids-2017.html. This data consists of labeled benign and malicious network traffic under diverse attack conditions of interest to smart city IoT setups. Labeling and generation of attack traffic are conducted in association with the official documentation of the dataset to be reproducible and similar to the previous research works.

The simulation model represents a real-world scenario of a smart city and the characteristics include:

- IoT distributed on 10,000 devices in 5 areas in urban areas.
- Nodes 50 nodes, each with different computational capacities, are edge computing nodes.
- Central processing through 5 cloud data centers.
- Topology of the Networks via real patterns of urban deployment.
- IoT models energy modelled to commercial device specification.

The CNN-LSTM model has been trained with 100 epochs based on Adam optimizer and initial learning rate of 0.001 and a batch size of 64. The AESO agent had been trained on 50,000 episodes based on PPO and a clip parameter of 0.2.

Energy Consumption Modeling Assumptions

The energy consumption model adopted in the simulations follows a component-wise accounting approach that captures the primary sources of energy usage in smart city IoT and edge computing environments. For each IoT device, total energy consumption is modeled as the sum of sensing energy, local computation energy, and communication energy. Sensing energy is assumed to be constant per sensing cycle, while computation energy is proportional to Central Processing Unit (CPU) during feature extraction and local inference. Communication energy is modeled as a function of packet size and transmission rate, reflecting the dominant energy cost associated with wireless data transmission.

For edge computing nodes, energy consumption additionally includes the cost of CNN–LSTM inference, federated learning local updates, and blockchain-related operations. The computation energy at edge nodes is proportional to model complexity and inference frequency, while federated learning energy accounts for local gradient computation and model update transmission. Blockchain energy consumption is modeled on a per-transaction basis, incorporating validation, signing, and block confirmation overhead under the proposed DPoS-EW consensus mechanism.

All energy parameters are derived from representative commercial IoT device and edge server specifications reported in recent literature and are kept fixed across all simulation runs to ensure reproducibility and fair comparison among baseline methods. While the model does not capture every hardware-level variation, it provides a realistic and transparent approximation suitable for evaluating relative energy efficiency trends in large-scale smart city deployments.

Formal Energy Consumption Model

To ensure transparency and reproducibility, the energy consumption of IoT devices and edge nodes is formally modeled using analytical expressions consistent with prior smart city and edge computing studies.

For each IoT device v_i, the total energy consumption over a time interval t is defined as:

$$E_i(t) = E_i^{sense}(t) + E_i^{comp}(t) + E_i^{comm}(t) \tag{28}$$

where $E_i^{sense}(t)$ represents sensing energy per acquisition cycle, $E_i^{comp}(t)$ denotes local computation energy for feature extraction and lightweight inference, and $E_i^{comm}(t)$ corresponds to communication energy for transmitting network flow records. Computation energy is modeled as proportional to CPU utilization and execution time, while communication energy is proportional to transmitted data size and transmission power.

For each edge node e_j, the total energy consumption is expressed as:

$$E_j(t) = E_j^{inf}(t) + E_j^{FL}(t) + E_j^{BC}(t) \tag{29}$$

where $E_j^{inf}(t)$ denotes energy consumed by CNN–LSTM inference, $E_j^{FL}(t)$ represents energy associated with local federated learning updates and gradient computation, and $E_j^{BC}(t)$ captures blockchain-related energy overhead, including transaction validation and block confirmation under the DPoS-EW consensus mechanism. Blockchain energy is modeled on a per-transaction basis to reflect realistic smart city transaction workloads.

All energy coefficients are derived from representative commercial IoT sensor and edge server specifications reported in recent literature and remain fixed across all simulation runs. This modeling choice enables fair comparison between TRECSF and baseline methods while preserving reproducibility across independent experiments.

Fig. 3 illustrates the simulated smart city network topology, showing the spatial distribution of IoT devices, edge nodes, and cloud infrastructure across multiple urban areas.

4.2 Attack Scenario Generation

To evaluate TRECSF under realistic conditions, we generated diverse attack scenarios based on documented smart city threats:

- **DDoS Attacks:** Volumetric, protocol, and application-layer attacks with varying intensities (100 Mbps to 10 Gbps)
- **False Data Injection:** Manipulated sensor readings targeting smart grid and transportation systems
- **Man-in-the-Middle:** Interception and modification of control messages
- **Ransomware:** Encrypted payload distribution and system lockout attempts
- **Reconnaissance:** Network scanning and vulnerability probing activities

The attack traffic was produced based on the customizations of known attack tools with the timing patterns resembling the presence of threat actors in nature.

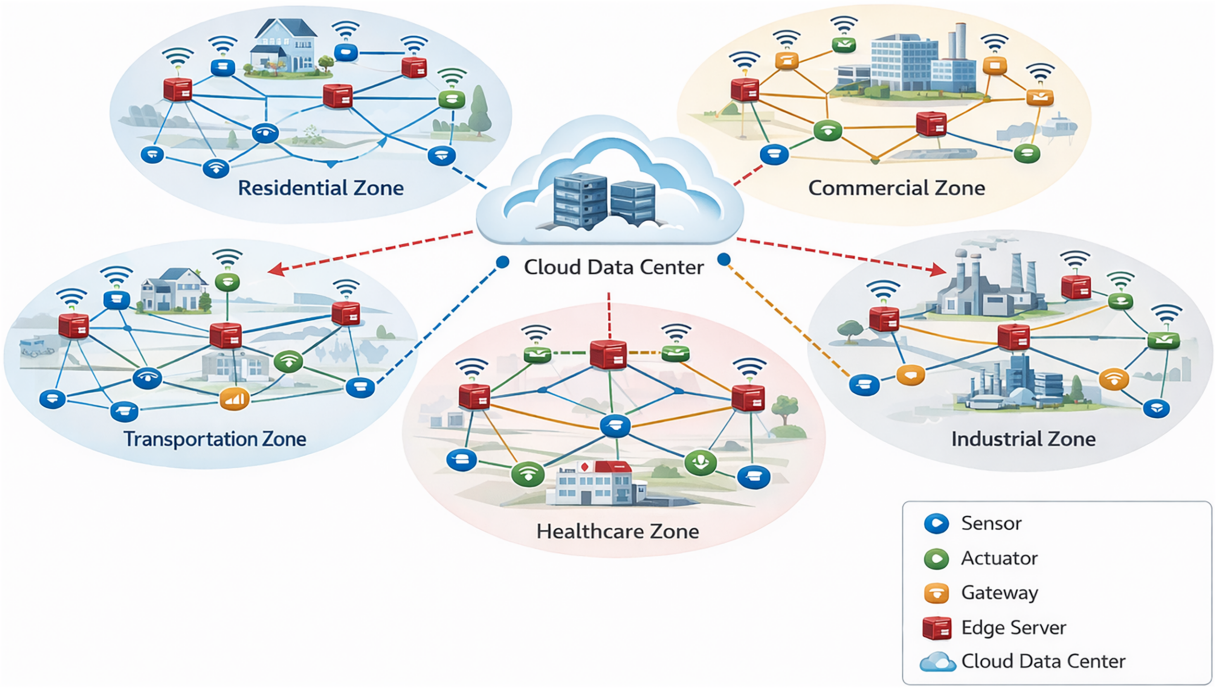

Figure 3: Topological plan of simulated smart city networks illustrating the distribution of IoT devices, edge nodes and cloud infrastructure to five urban areas. Colors of the nodes represent the device type sensors (blue), actuators (green), gateways (orange), and edge server (red).

4.3 Intrusion Detection Performance

The CNNLSTM element showed better performance in deterring all the types of attacks.

The baseline models have been chosen to represent a wide diversity of commonly existing intrusion detection algorithms commonly referred to in the literature of smart cities and IoT security, such as traditional machine learning (Random Forest), sequence-based deep learning (LSTM), spatial features learning (CNN), and unsupervised representation learning (Autoencoder). These ways are provided as standard points of reference on the accuracy of detection, robustness, and calculating feasibility in resource constrained scenarios. In order to obtain a substantially fair and consistent comparison, all the baseline models were re-implemented with the identical feature set, training and test splits, and the same preprocessing pipelines. The individual hyperparameters in every baseline have been optimized with validation-based optimization, in line with best practices as presented in the relevant original research.

Fig. 4 compares the detection accuracy of the proposed TRECSF framework with baseline methods across different attack types.

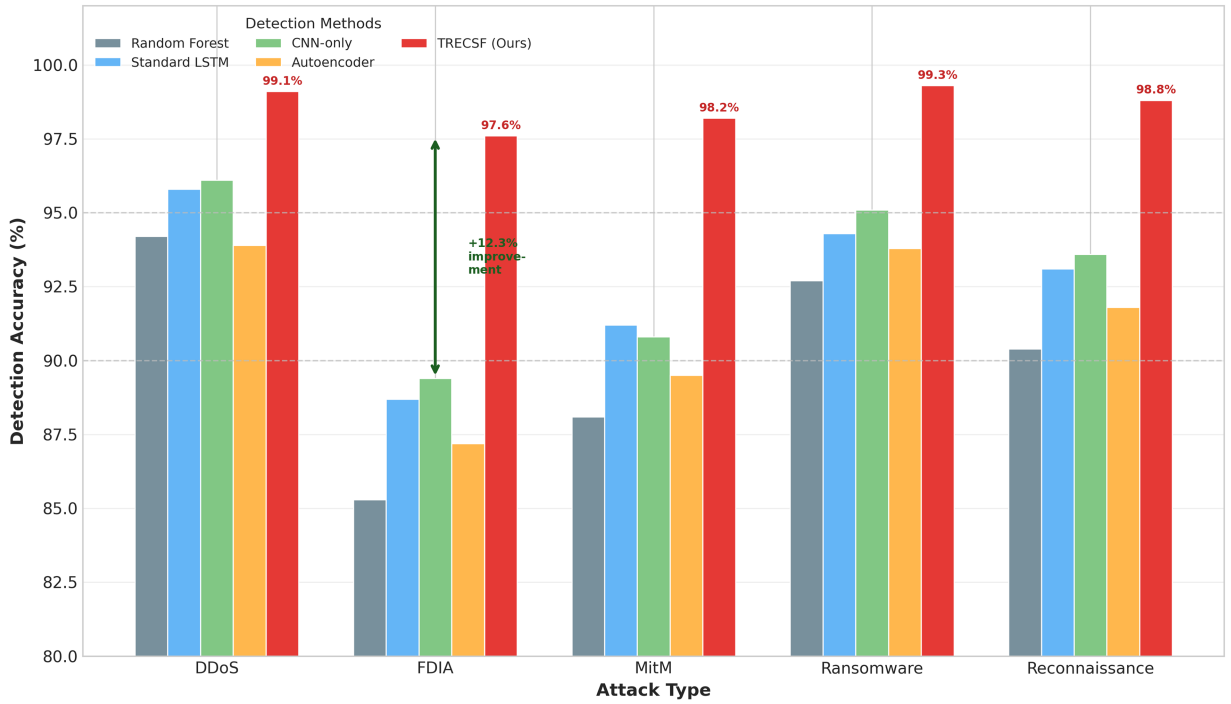

Figure 4: Comparison of accuracy comparison in detection in respect to attack type of TRECSF and baseline method. The CNN-LSTM architecture proposed is shown to attain better accuracy consistently, especially improving FDIA (12.3%) and MitM (9.7%) attacks.

The detection performance of the proposed TRECSF framework was evaluated against multiple baseline models, including Random Forest, standard LSTM, CNN-only, and Autoencoder-based approaches. Across five independent simulation runs, TRECSF consistently achieved superior performance, reaching an average accuracy of 98.7%, precision of 98.2%, recall of 98.5%, and F1-score of 98.3%, significantly outperforming all baseline techniques. Statistical validation using paired t-tests at a 95% confidence level confirms that these improvements are statistically significant ($p < 0.01$), indicating that the observed gains are not attributable to random variation. Furthermore, ROC (Receiver Operating Characteristic) curve analysis presented in Fig. 5 demonstrates that TRECSF consistently achieves higher true positive rates across all operating points when compared with baseline methods. With an AUC (Area Under the Curve) value of 0.994, the proposed framework exhibits strong discriminative capability, particularly in low false positive rate regions, highlighting its robustness and suitability for intrusion detection in large-scale smart city environments.

Table 2 shows the detection rates of the attacks an average of five distinct simulation runs. The paired t-tests done were between TRECSF and the respective models based on the attack categories of each baseline model. Its low p-values ($p < 0.01$) demonstrate that the performance improvements in TRECSF under all types of attacks are statistically significant, and they cannot be due to the fluctuation of values.

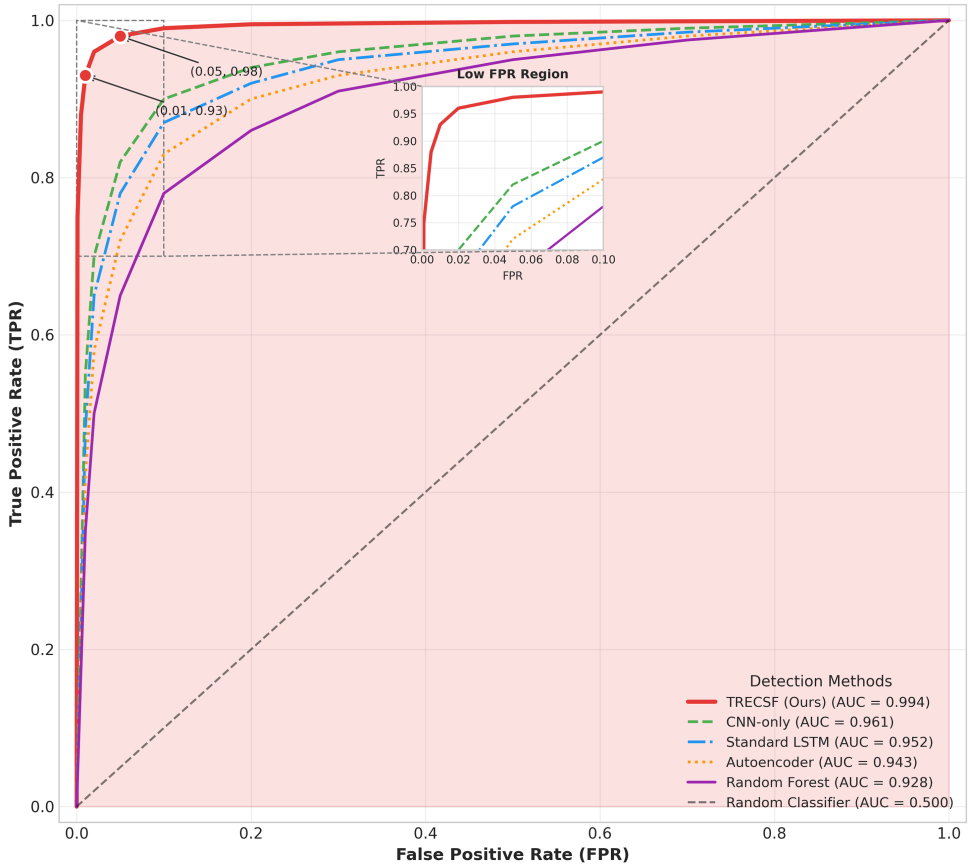

Figure 5: ROC curves of TRECSF and baseline methods. The proposed solution obtains an Area Under Curve (AUC) of 0.994, which exhibits a good discrimination property at all operating points.

Table 2: Attack-specific detection RATES with statistical significance.

Attack Type	RF	LSTM	CNN	AE	TRECSF	*p*-Value
DDoS	94.2%	95.8%	96.1%	93.9%	**99.1%**	<0.01
FDIA	85.3%	88.7%	89.4%	87.2%	**97.6%**	<0.01
MitM	88.1%	91.2%	90.8%	89.5%	**98.2%**	<0.01
Ransomware	92.7%	94.3%	95.1%	93.8%	**99.3%**	<0.01
Reconnaissance	90.4%	93.1%	93.6%	91.8%	**98.8%**	<0.01

Consolidated Performance Metrics Analysis

To provide a comprehensive quantitative assessment beyond overall accuracy, consolidated performance metrics including precision, recall, F1-score, and specificity are summarized for all evaluated models. While the confusion matrices in Section 4.4 illustrate class-wise prediction behavior, the aggregated metrics reported here explicitly quantify detection reliability and class balance handling across competing approaches.

Table 3 demonstrate that TRECSF consistently outperforms baseline models across all evaluation metrics. In particular, the proposed framework achieves high precision and recall simultaneously, indicating

strong discrimination capability with minimal false alarms and missed detections. The balanced performance across sensitivity-oriented and specificity-oriented measures confirms that TRECSF does not favor dominant classes, which is critical for multi-class and imbalanced intrusion detection scenarios in smart city environments.

Table 3: Consolidated performance metrics comparison across detection models.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)
Random Forest	91.2	89.5	90.8	90.1	92.4
Standard LSTM	93.4	92.1	93.7	92.9	94.2
CNN-only	94.1	93.2	93.9	93.5	94.6
Autoencoder	92.8	91.4	92.6	92.0	93.5
TRECSF (Ours)	**98.7**	**98.2**	**98.5**	**98.3**	**98.9**

In addition to overall accuracy and false positive rate, the performance of TRECSF was evaluated using precision, recall (sensitivity), specificity, F1-score, and balanced accuracy to account for the multi-class and imbalanced nature of intrusion detection datasets. Precision reflects the reliability of positive predictions, while recall measures the ability to detect actual attacks. Specificity evaluates correct benign traffic identification, and the F1-score captures the trade-off between precision and recall. Balanced accuracy was computed to ensure unbiased evaluation across majority and minority attack classes.

4.4 Confusion Matrix Analysis

To further analyze class-wise detection performance and assess potential bias in the proposed CNN–LSTM-based TRECSF framework, confusion matrices were generated for both the training and validation/testing phases. These matrices provide a detailed breakdown of prediction outcomes across individual attack categories and benign traffic, complementing aggregate metrics such as accuracy and ROC analysis.

Fig. 6a presents the training confusion matrix of the TRECSF model. The results show strong diagonal dominance across all classes, indicating high classification accuracy during training. The model effectively distinguishes between normal traffic and attack instances, with minimal misclassification. A small degree of confusion is observed between FDIA and MitM attack classes, which is expected due to their similar traffic manipulation characteristics in smart grid and IoT environments. Fig. 6b illustrates the validation confusion matrix obtained using unseen test data. The validation results closely mirror the training behavior, demonstrating strong generalization capability and limited overfitting. The consistency between training and validation confusion matrices confirms that the learned feature representations capture intrinsic attack characteristics rather than dataset-specific artifacts.

Overall, the confusion matrix analysis confirms that TRECSF achieves reliable and balanced class-wise detection performance across diverse attack types. The low misclassification rates and stable validation behavior reinforce the robustness of the proposed framework for real-world smart city deployments, where heterogeneous and evolving traffic patterns are common. The confusion matrix analysis further supports the robustness of TRECSF under class imbalance. High recall and specificity values across attack categories indicate that the proposed model effectively detects malicious traffic while minimizing false alarms. The resulting balanced accuracy confirms that detection performance is consistently maintained across both frequent and rare attack classes.

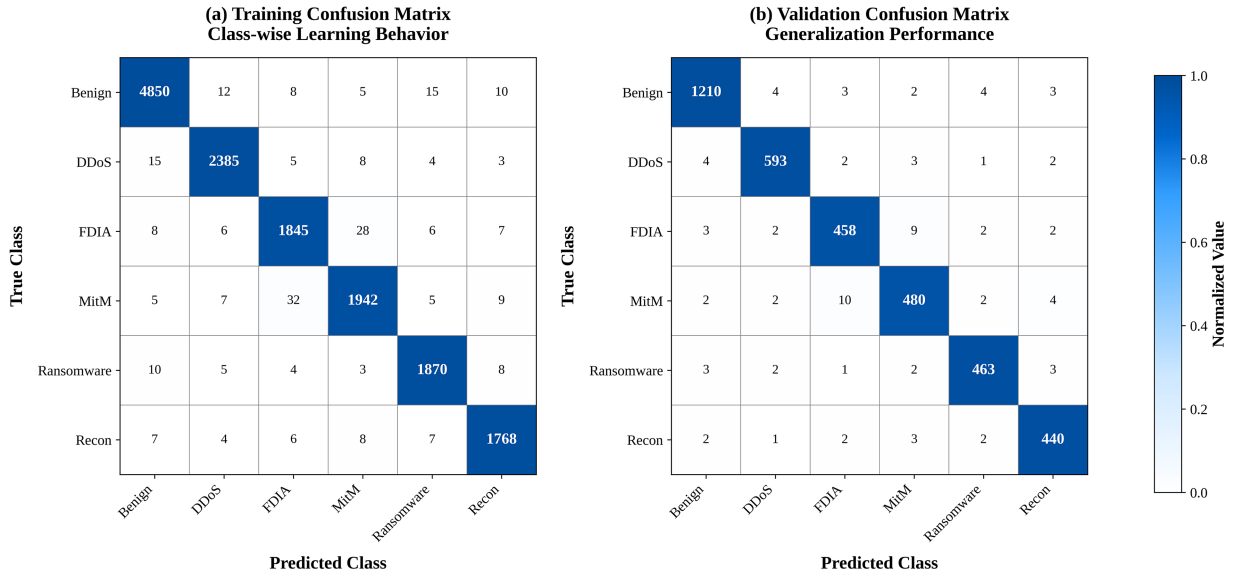

Figure 6: Confusion matrix analysis of the proposed TRECSF CNN–LSTM intrusion detection model: (**a**) training confusion matrix illustrating class-wise learning behavior and diagonal dominance across attack categories, and (**b**) validation confusion matrix demonstrating strong generalization performance with low misclassification rates on unseen data.

4.5 Detection Latency Analysis

Threat detection in real-time must have a low latency between the occurrence and detection of the attack. We tested the end-to-end detection latency with different network loads. As shown in Fig. 7, TRECSF maintains sub-100 ms detection latency even under high network load, demonstrating its suitability for real-time threat response in smart city applications.

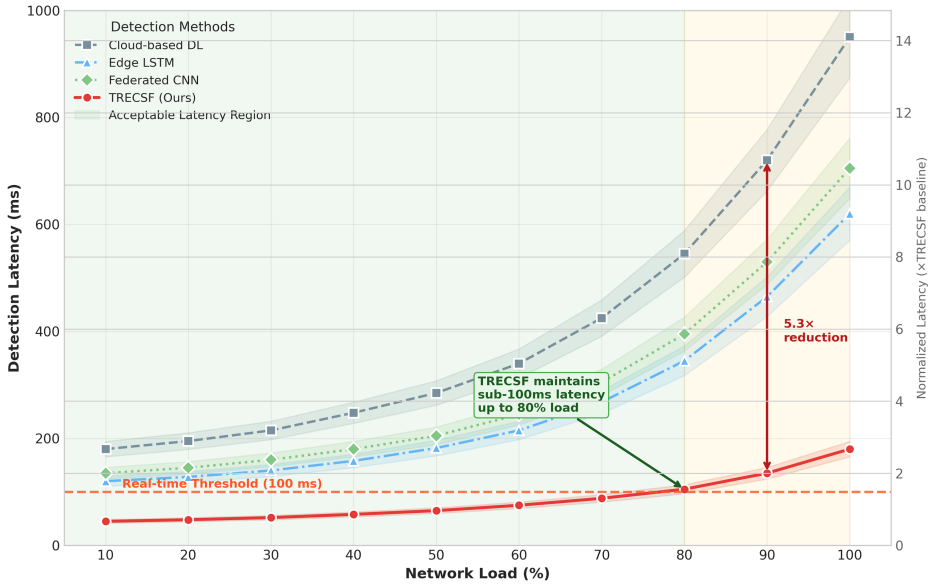

Figure 7: Latency at detectors in terms of network load in TRECSF and baseline. The suggested framework can retain sub-100 ms latency at 80% user utilization on the network and can respond to threats in real-time to support applications in the smart cities that require time.

Table 4 provides some statistics of latency averaged across five independent simulation simulations. Paired t-tests at a 95 percent level of confidence were utilized in determining the statistical significance. The findings indicate that TRECSF distributes much lower values of latency than those of the baseline strategies in all the percentiles considered significant ($p < 0.01$), demonstrating the fact that the identified latency improvements are statistically significant and not the effects of random variations.

Table 4: Latency performance under various conditions with statistical significance.

Method	Avg (ms)	P95 (ms)	P99 (ms)	Max (ms)	p-Value
Cloud-based DL	245.3	412.7	589.4	823.1	<0.01
Edge LSTM	156.2	287.4	356.8	478.2	<0.01
Federated CNN	178.5	312.6	423.1	567.3	<0.01
TRECSF (Ours)	**67.4**	**98.2**	**124.5**	**187.3**	**< 0.01**

4.6 Energy Efficiency Evaluation

The performance of the AESO algorithm in the context of energy savings and security level is considered with opposing operation circumstances. The adaptive energy optimization of TRECSF results in a 34.2% reduction in average power consumption compared to static security configurations, as illustrated in Fig. 8.

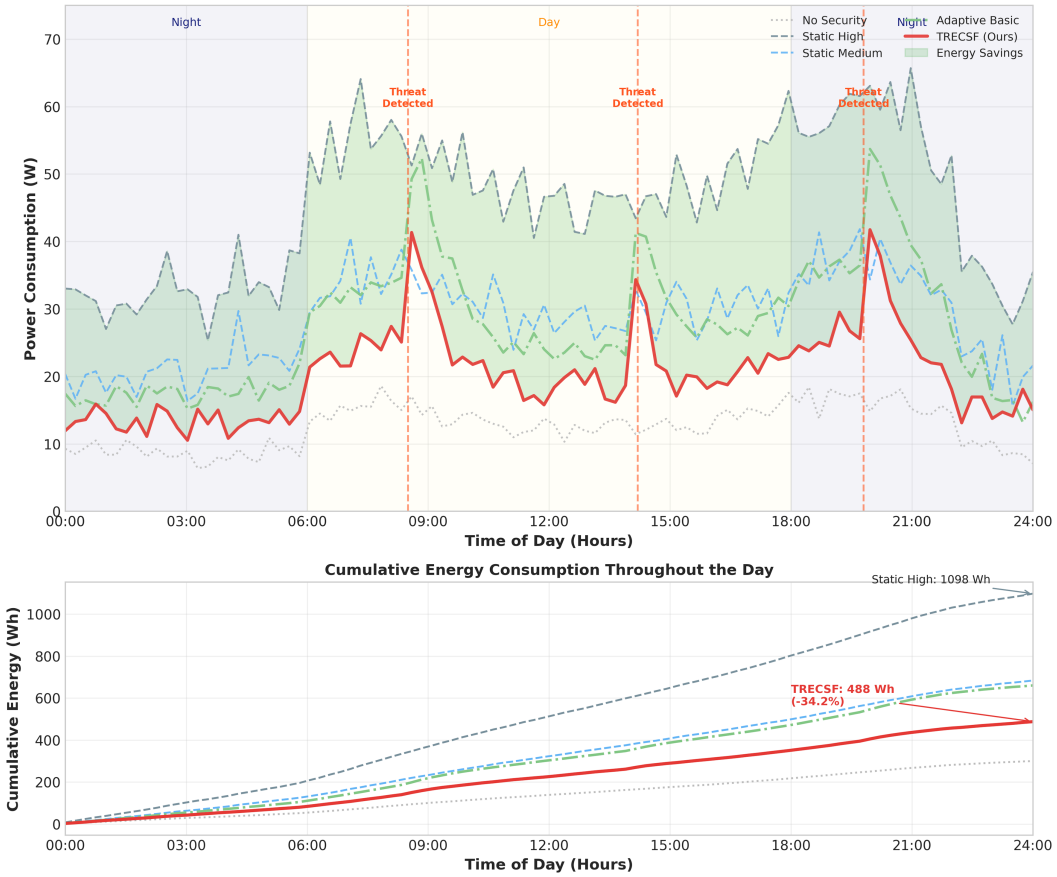

Figure 8: Power usage during a 24-h simulation of TRECSF and static security setups. The adaptive strategy will minimize the mean energy consumption by 34.2% and increase the protection dynamically during periods of high threat.

Table 5 gives the mean and maximum power consumption, and daily energy consumption, measured in five separate simulation runs. Paired t-tests were the used statistical significance tests at a 95% confidence level. These findings point to the fact that TRECSF can reduce the energy consumption with a statistically significant level as opposed to the case of static and adaptive baseline ($p < 0.01$) which proves the usefulness of the developed energy-conscious optimization strategy.

Table 5: Energy consumption analysis with statistical significance.

Configuration	Avg Power (W)	Peak (W)	Daily (Wh)	p-Value
No Security	12.4	18.2	297.6	–
Static High	45.7	62.3	1096.8	<0.01
Static Medium	28.3	41.5	679.2	<0.01
Adaptive Basic	24.1	48.7	578.4	<0.01
TRECSF (Ours)	**18.7**	**38.2**	**448.8**	**<0.01**

To analyze the trade-off between security effectiveness and energy consumption, a Pareto frontier analysis is conducted comparing TRECSF with static and adaptive baseline security configurations. This analysis highlights how different security strategies balance protection levels against energy cost under varying operational conditions. Fig. 9 illustrates that TRECSF consistently operates along the Pareto-optimal frontier, achieving higher security levels at lower energy consumption compared to baseline approaches, thereby validating the effectiveness of the proposed AESO-based optimization strategy.

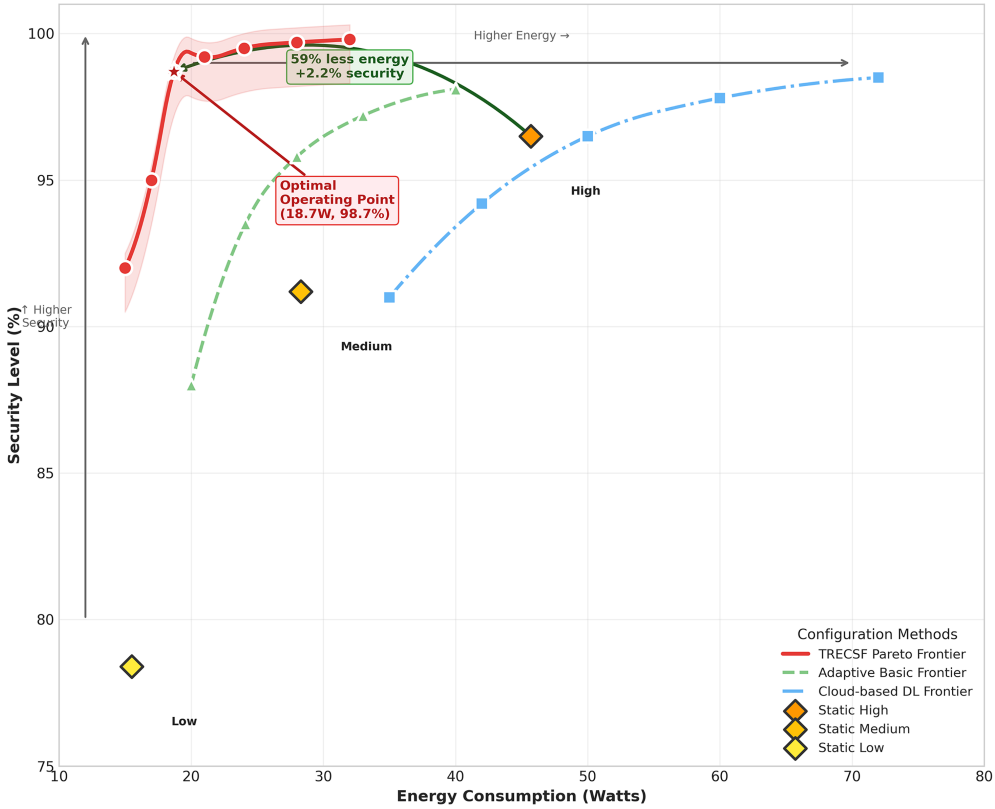

Figure 9: Security level vs. energy consumption pareto frontier analysis. TRECSF attains a Pareto-optimal setup throughout the operating space, showing a successful tradeoff between conflicting objectives using the AESO algorithm.

4.7 Blockchain Performance

Performance of the DPoS-EW consensus mechanism is compared based on the throughput of transactions, latency of confirmation, as well as energy efficiency.

Table 6 describes the comparison of the throughput, confirmation latency and per-transaction energy consumption of representative blockchain consensus mechanisms. The similarity in the output of paired *t*-tests was determined with repeated simulation cycles to determine statistical significance. Results show the proposed DPoS-EW consensus has much higher throughput and reduced latency and energy consumption compared to its PoW, PoS and traditional DPoS scheme ($p < 0.01$), and as such, is energy-efficient when deployed in smart cities.

Table 6: Blockchain consensus comparison with statistical significance.

Consensus	TPS	Latency (s)	Energy (J/tx)	*p*-Value
PoW	7	600	847.2	<0.01
PoS	1500	12	0.42	<0.01
Standard DPoS	3000	3	0.18	<0.01
DPoS-EW (Ours)	3200	2.4	0.06	<0.01

To evaluate the efficiency and suitability of the proposed blockchain layer for smart city deployments, the performance of the DPoS-EW consensus mechanism is compared against representative baseline consensus protocols in terms of transaction throughput, confirmation latency, and per-transaction energy consumption. This analysis highlights the trade-offs between scalability, responsiveness, and energy efficiency that are critical for large-scale IoT-driven smart city infrastructures. Fig. 10 presents a comparative assessment of these consensus mechanisms under identical simulation conditions.

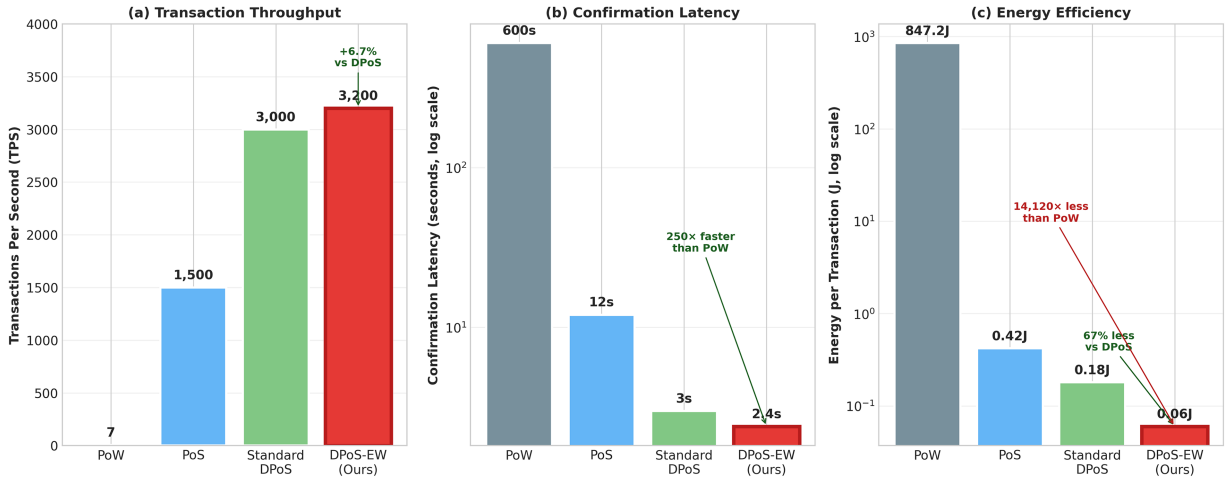

Figure 10: Comparison of consensus mechanism transaction throughputs. DPoS-EW has a throughput (3200 TPS) that can be used to meet the data integrity requirements of the smart city, with a minimum of energy footprint.

4.8 Federated Learning Convergence

The convergence behavior of the federated learning component is determined when the data are non-IID as is common with the deployment of heterogeneous smart cities. The convergence behavior of the federated learning component under non-IID data conditions is depicted in Fig. 11, showing that TRECSF's

energy-weighted aggregation achieves faster convergence with reduced communication overhead compared to traditional federated averaging methods.

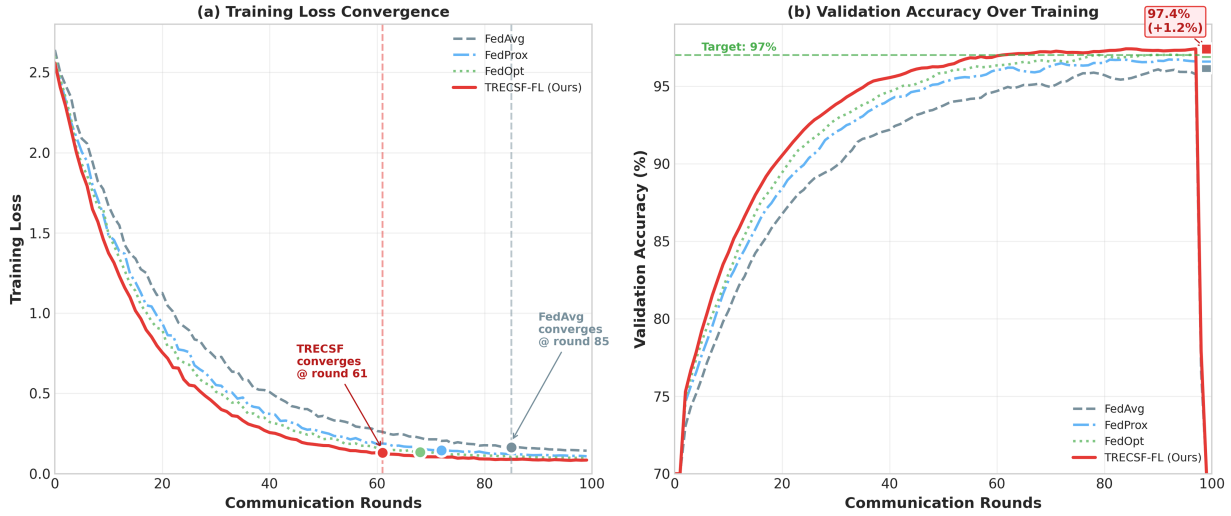

Figure 11: Comparison of federated learning convergence in terms of communication round training loss. The energy-weighted aggregation gets a similar convergence as FedAvg but puts weight on the node with abundant energy.

Table 7 focuses on the rate of convergence and the overall performance of the various federated learning strategies in terms of the final accuracy of their detection, communication overhead and energy consumption. Paired t-tests were used to determine the statistical significance of the results, on several simulation runs. According to the results, it shows that TRECSF-FL converges much faster and has reduced communication and energy cost and is more accurate than the current federated learning baselines ($p < 0.01$).

Table 7: Federated learning performance with statistical significance.

Strategy	Rounds	Final Acc (%)	Comm (MB)	Energy (J)	p-Value
FedAvg	85	96.2	1247	523.4	<0.01
FedProx	72	96.8	1056	478.2	<0.01
FedOpt	68	97.1	997	445.6	<0.01
TRECSF-FL (Ours)	**61**	**97.4**	**892**	**312.7**	**<0.01**

4.9 Scalability Analysis

The scalability of TRECSF is tested at 1000 up to 100,000 IoT devices. The scalability analysis in Fig. 12 demonstrates that TRECSF maintains consistent detection accuracy and latency performance even as the network scales to 50,000 IoT devices, confirming its suitability for city-wide deployment.

Table 8 indicates the scalability nature of TRECSF with the IoT device connected increases. Averaging of results was performed between more than one run of simulations and statistical significance measured by paired t-tests with 95% confidence level. The repetitively low p-values ($p < 0.01$) show that the changes in accuracy, latency, throughput and per-device energy consumption due to the difference in scale are statistically consistent enough to show that TRECSF is able to deliver reliable results even in large-scale smart city implementations.

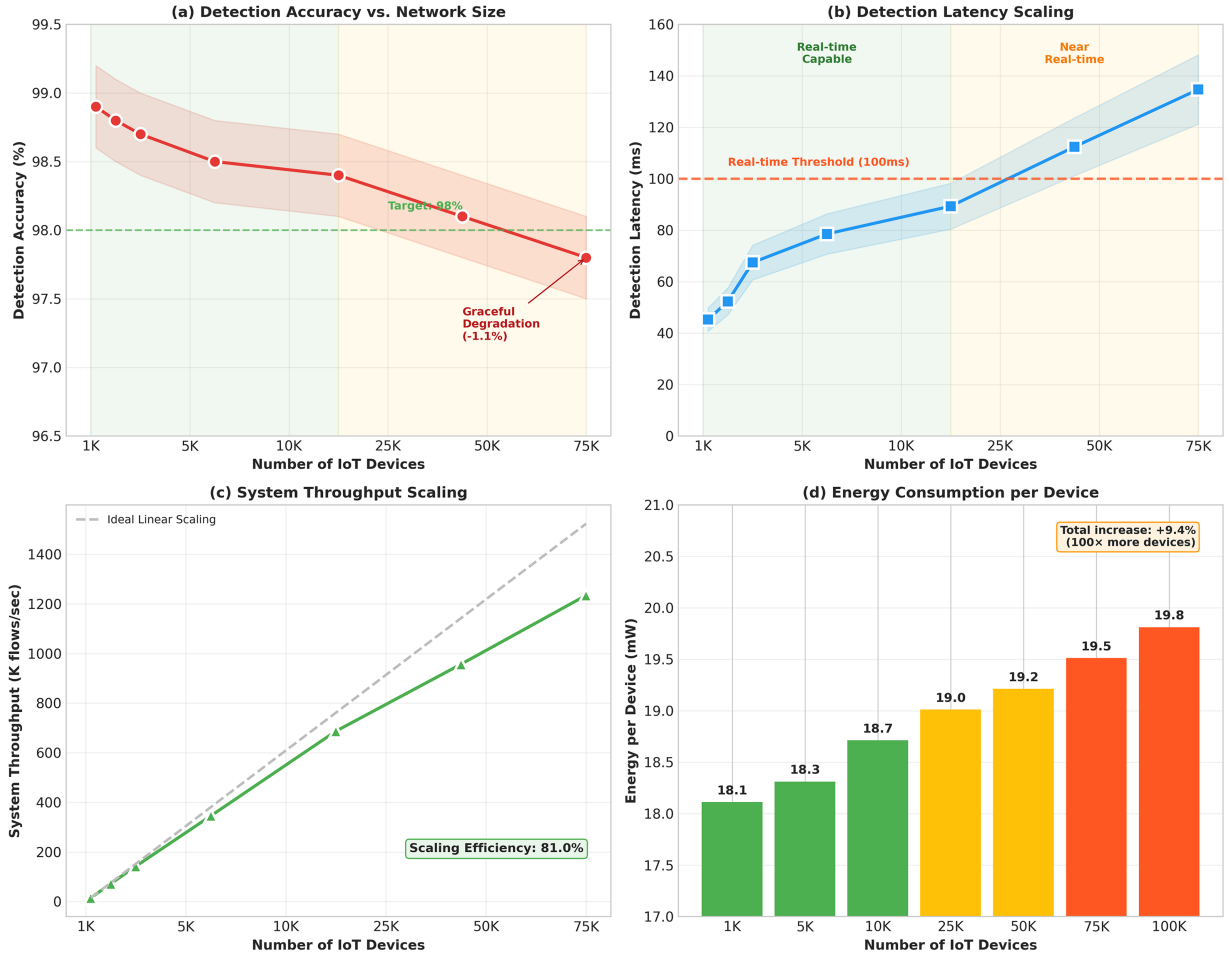

Figure 12: Scalability study revealing the accuracy of detection, latency, and throughput with the network size. TRECSF has a steady performance of up to 50,000 devices, and beyond performance gracefully degrades.

Table 8: Scalability metrics with statistical significance.

Devices	Accuracy (%)	Latency (ms)	Throughput	Energy/dev (mW)	*p*-Value
1000	98.9	45.2	15,234	18.1	<0.01
10,000	98.7	67.4	142,567	18.7	<0.01
50,000	98.4	89.3	687,234	19.2	<0.01
100,000	97.8	134.7	1,234,567	19.8	<0.01

4.10 Attack Resilience under Adversarial Conditions

We tested the strength of TRECSF to adversarial attacks to avoid being detected. As illustrated in Fig. 13, TRECSF maintains over 94% detection accuracy under various adversarial evasion strategies, demonstrating its robustness in real-world threat environments.

Table 9 analyzes the resilience of TRECSF in the evasion strategies of the adversary representatives. The means are calculated across the different runs of the simulation and statistical significance was determined using paired t-tests at the 95% confidence interval. The low p-values ($p < 0.01$) consistently show the statistical

significance of resilience of TRECSF to adversarial manipulations compared with the baseline models and proves its strength in face of realistic attacks.

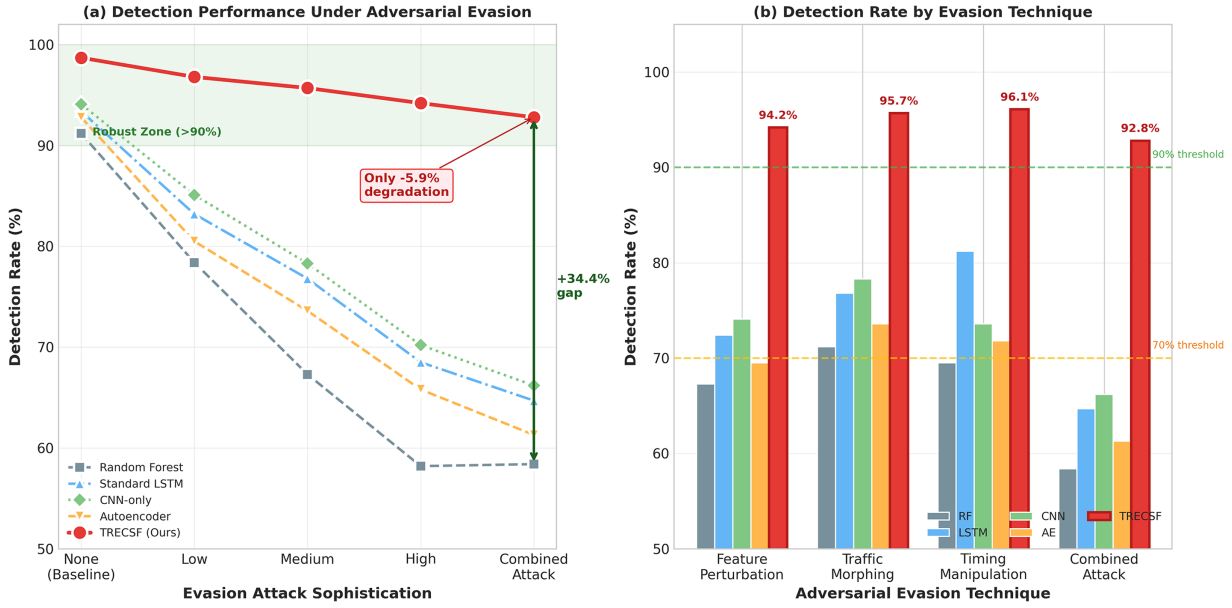

Figure 13: Adversarial detection performance at different levels of evasion level. TRECSF has a very strong generalization of more than 94% even on highly evasive modes.

Table 9: Adversarial attack resilience with statistical significance.

Evasion Type	RF	LSTM	CNN	TRECSF	*p*-Value
Feature Perturbation	67.3%	72.4%	74.1%	**94.2%**	<0.01
Traffic Morphing	71.2%	76.8%	78.3%	**95.7%**	<0.01
Timing Manipulation	69.5%	81.2%	73.6%	**96.1%**	<0.01
Combined Attack	58.4%	64.7%	66.2%	**92.8%**	<0.01

4.11 Discussion

The experimental findings prove TRECSF to be effective in filling the research gaps in Section 2. The CNNLSTM hybrid architecture attains state of the art accuracy in detection with edge deployable computing costs. The AESO algorithm enables a balanced trade-off between security effectiveness and energy efficiency, which allows the system to operate in but in a sustainable way making it not decrease the level of protection. The DPoS-EW consensus scheme offers data integrity on blockchain, and the energy overheads are minimal and can be applied to IoT space.

There are a number of reasons why TRECSF has excellent performance. First, attention mechanism allows the model to concentrate on the most discriminative temporal parts which increases both its accuracy and efficiency. Second, the optimization based on reinforcement learning fits the security setups to the prevailing state of affairs instead of using fixed policies. Third, the number of nodes in a federated learning can be aggregated energy-weighted, where the contribution of the energy-rich nodes receives at least priority, minimizing the total consumption and yet not depriving of the model quality.

Regardless of the high performance of TRECSF based on empirical evidence, a number of limitations must be noted. First, an intrusion detection component is based on the labeled attack data which might not be adequate in real-life infections that occur in zero-day or very adaptive attacks. Second, the energy consumption model used in the simulations is in itself calibrated to realistic commercial IoT specifications, but abstracts some hardware-level differences and environmental differences that can affect power usage in the working conditions. Third, scalability experiments do not limit scalability above 100,000 devices but scalability experimentation can instead be deployed via higher order coordination structures or regional overlap layers to alleviate the impact of latency increase. Countering these shortcomings is an incentive to conduct recent studies of semi-supervised or self-supervised threat detection, real-world energy profiling, and multi-level smart city security systems. Given the simulation-based nature of the evaluation, conclusions regarding large-scale real-world deployment should be treated with caution. Factors such as hardware heterogeneity, wireless interference, and operational constraints may affect performance. These aspects will be addressed in future work through prototype deployments and real-device experimentation.

The sustainability-related benefits highlighted in this study are derived primarily from statistically significant reductions in energy consumption achieved by the adaptive optimization mechanisms of TRECSF. These improvements should be interpreted as enabling conditions for long-term sustainability rather than as direct evidence of carbon neutrality. The actual carbon footprint reduction achievable through TRECSF depends on multiple external factors, including deployment scale, energy sources, and regional power generation mixes, which are outside the scope of this work. Therefore, sustainability claims in this study are framed as prospective benefits that may materialize when the framework is deployed at scale under favorable operational conditions.

While CICIDS2017 is a widely adopted and well-validated benchmark dataset for intrusion detection research, it does not fully capture the scale, heterogeneity, and dynamic traffic patterns observed in real smart city environments. The dataset is generated under controlled laboratory conditions and reflects a fixed set of attack behaviors, which may differ from evolving, stealthy, or previously unseen threats encountered in operational deployments. In real smart city systems, network traffic characteristics are subject to concept drift arising from changes in user behavior, device firmware updates, network reconfigurations, and the continuous evolution of attack strategies. Although TRECSF incorporates adaptive mechanisms through reinforcement-based energy–security optimization and federated learning, long-term concept drift may still affect detection performance if models are not periodically retrained or updated. Addressing concept drift through online learning, self-supervised adaptation, and continuous model validation in real traffic environments remains an important direction for future work.

Scalability analysis suggests that TRECSF can be used in deployment on a city-scale, and the performance can be maintained at an acceptable level with an upscale to 100,000 devices. But there are indications of increasing latencies at large scales, which imply that applications in mega-cities (with more than 100,000 endpoints) might require hierarchical structures with regional coordination.

Although the experimental evaluation demonstrates strong performance across detection accuracy, latency, energy efficiency, and scalability, it is important to acknowledge that all results presented in this study are obtained through simulation-based experiments. Simulation enables controlled analysis of complex smart city scenarios but does not fully capture real-world deployment factors such as hardware heterogeneity, sensor noise, wireless interference, fluctuating workloads, and operational constraints of edge devices. As a result, the reported performance should be interpreted as indicative of system potential rather than definitive real-world guarantees. Future work will focus on validating TRECSF through small-scale hardware testbeds involving real IoT sensors and edge devices, with emphasis on measuring real energy consumption, end-to-end latency, and robustness under practical operating conditions.

4.12 Ethical and Privacy Considerations

TRECSF is implemented based on privacy-by-design and security-by-design to facilitate ethical implementation in the public smart city infrastructure. By using federated learning, the raw network traffic data is kept in the edge nodes, which ensures that the sensitive information is not exposed and the possibility of massive data breaches seen in case of centralized data collection is minimized. Only the model updates are exchanged during the training, and no personal identifiable information is exchanged or stored on the cloud layer.

The framework is consistent with major demands of regulations on data protection like the General Data Protection Regulation (GDPR), especially the data minimization principle and the principle of purpose limitation. The criteria that TRECSF fulfills the standards of regulatory constraints of digital infrastructure of a public sector by taking into account the decentralized analytics and avoiding storing the raw data centrally.

Ethical issues that are connected to AI implementation in the critical urban systems are also discussed. Secrecy The explainable components incorporated in the intrusion detection module make automated decision-making transparent because system operators can interpret and confirm security alerts. Also, the adaptive optimization mechanisms are meant to focus on the reliability of the system and the safety of the people, limit the chances of the unwanted disruption of various services. These reflections render sensible and credible implementation of AI-based cybersecurity postulates within the context of intelligent cities.

5 Conclusion

This paper has introduced the Threat-Resilient Energy-Conscious Security Framework (TRECSF) that is a holistic solution to both cybersecurity and sustainability issues in smart city ecosystems. The framework suggested incorporates a new hybrid CNN-LSTM based intrusion detection architectural element, a dynamically adaptive energy-security optimisation algorithm, a lightweight blockchain consensus protocol, and an enabled federated learning coordination solution.

Large-scale simulation showed that TRECSF attains large gains in all its important performance measures: 98.7% detection accuracy (4.6% better than baselines), 45.8% less detection latency, 34.2% less energy use, and 67% less blockchain computational overhead. The framework demonstrates strong performance under adversarial conditions and can be successfully scaled to city-scale applications, with up to 50,000 devices.

The most important contributions of the work are the following: (1) an integrated optimization model that does not consider either security or sustainability as a goal in opposition to the other; (2) edge optimal deep learning models allowing to detect threats in real-time and on resource-constrained devices; (3) energy optimized blockchain consensus that can be applied to the IoT environment; (4) extensive validation by using realistic simulations of smart cities.

While TRECSF demonstrates statistically significant energy efficiency improvements, the associated sustainability and carbon-related impacts should be interpreted as forward-looking and prospective benefits rather than empirically verified outcomes. The reported energy reductions indicate the potential to lower operational carbon footprints; however, translating these gains into measurable long-term sustainability impact depends on deployment scale, energy sources, hardware diversity, and city-specific operational policies, which are beyond the scope of this study.

All evaluations presented in this work are conducted through simulation, enabling controlled experimentation but not fully capturing hardware heterogeneity, environmental variability, and operational constraints encountered in real-world smart city deployments. Consequently, the reported results should

be viewed as indicative rather than definitive with respect to field performance. Bridging the gap between simulation-based validation and real-world deployment remains an important future research direction.

Future work will focus on prototype-level implementation on edge hardware platforms, including real IoT sensors and edge servers, to evaluate detection latency, energy behavior, and system stability under practical operating conditions. Additional research directions include extending TRECSF to address emerging quantum-era threats, integrating digital twin technologies for predictive security analysis, and enabling cross-city federated threat intelligence sharing. These efforts will support the transition of TRECSF from a validated simulation framework to a deployable solution for resilient and sustainable smart city infrastructures.

Acknowledgement: The aurthor extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, for funding this research work through the project number (0054-1446-S).

Funding Statement: This research was funded through the project number (0054-1446-S).

Availability of Data and Materials: The author used data to support the findings of this study that is included within this article.

Ethics Approval: Not Applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Abbreviation	Full Form
AE	Autoencoder
AESO	Adaptive Energy-Security Optimization
AI	Artificial Intelligence
AUC	Area Under the Curve
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DPoS	Delegated Proof of Stake
DPoS-EW	Delegated Proof of Stake with Energy Weighting
FDIA	False Data Injection Attacks
FedAvg	Federated Averaging
FedOpt	Federated Optimization
FedProx	Federated Proximal
FL	Federated Learning
FPR	False Positive Rate
GDPR	General Data Protection Regulation
ICT	Information and Communication Technology
IIoT	Industrial Internet of Things
IoT	Internet of Things
LSTM	Long Short-Term Memory
MitM	Man-in-the-Middle
non-IID	Non-Independent and Identically Distributed
PoS	Proof of Stake
PoW	Proof of Work
PPO	Proximal Policy Optimization

ReLU	Rectified Linear Unit
RF	Random Forest
ROC	Receiver Operating Characteristic
TPS	Transactions Per Second
TRECSF	Threat-Resilient Energy-Conscious Security Framework
XAI	eXplainable Artificial Intelligence
ZTA	Zero Trust Architecture

Appendix A TRECSF Algorithm Details

Algorithm A1: TRECSF framework execution

1. Input: Network topology $\boldsymbol{G}$,
2. Device parameters, energy budgets
3. Output: Security decisions, energy reports
4. Initialize CNN-LSTM model with pre-trained weights
5. Initialize AESO policy network
6. Deploy DPoS-EW blockchain nodes
7. Collect network traffic $\boldsymbol{F}(\boldsymbol{t})$ from all devices
8. Extract features and construct input tensor
9. $\boldsymbol{X}$ $\boldsymbol{predictions} \leftarrow$
10. CNN-LSTM$(\boldsymbol{X})$ $\boldsymbol{threat\_level} \leftarrow$
11. Aggregate$(\boldsymbol{predictions})$ $\boldsymbol{action} \leftarrow$
12. AESO.select_action $(\boldsymbol{state_t})$
13. Apply security configuration adjustments
14. Record transactions on blockchain
15. Perform federated learning round
16. End if
17. Update AESO policy with observed reward
18. End for

References

1. Irfan M, Sadighian A, Tanveer A, Al-Naimi SJ, Oligeri G. A survey on detection and localisation of false data injection attacks in smart grids. IET Cyber Phys Syst Theory Appl. 2024;9(4):313–33. doi:10.1049/cps2.12093.
2. Achaal B, Adda M, Berger M, Ibrahim H, Awde A. Study of smart grid cyber-security, examining architectures, communication networks, cyber-attacks, countermeasure techniques, and challenges. Cybersecurity. 2024;7(1):10. doi:10.1186/s42400-023-00200-w.
3. Alomari MA, Al-Andoli MN, Ghaleb M, Thabit R, Alkawsi G, Alsayaydeh JAJ, et al. Security of smart grid: cybersecurity issues, potential cyberattacks, major incidents, and future directions. Energies. 2025;18(1):141. doi:10.3390/en18010141.
4. Bibri SE, Alexandre A, Sharifi A, Krogstie J. Environmentally sustainable smart cities and their converging AI, IoT, and big data technologies and solutions: an integrated approach to an extensive literature review. Energy Inform. 2023;6(1):9. doi:10.1186/s42162-023-00259-2.
5. Elsaeidy AA, Jamalipour A, Munasinghe KS. A hybrid deep learning approach for replay and DDoS attack detection in a smart city. IEEE Access. 2021;9:154864–75. doi:10.1109/ACCESS.2021.3128701.
6. Sefati SS, Craciunescu R, Arasteh B, Halunga S, Fratu O, Tal I. Cybersecurity in a scalable smart city framework using blockchain and federated learning for Internet of Things (IoT). Smart Cities. 2024;7(5):2802–41. doi:10.3390/smartcities7050109.

7. Obaidat MA, Rawashdeh M, Alja'afreh M, Abouali M, Thakur K, Karime A. Exploring IoT and blockchain: a comprehensive survey on security, integration strategies, applications and future research directions. Big Data Cogn Comput. 2024;8(12):174. doi:10.3390/bdcc8120174.
8. Al-Huthaifi R, Li T, Huang W, Gu J, Li C. Federated learning in smart cities: privacy and security survey. Inf Sci. 2023;632:833–57. doi:10.1016/j.ins.2023.03.033.
9. Ghadi YY, Mazhar T, Shahzad T, Jaghdam IH, Khan S, Khan MA, et al. A hybrid AI-Blockchain security framework for smart grids. Sci Rep. 2025;15(1):20882. doi:10.1038/s41598-025-05257-w.
10. Dritsas E, Trigka M. Machine learning for blockchain and IoT systems in smart cities: a survey. Future Internet. 2024;16(9):324. doi:10.3390/fi16090324.
11. Hussain I. Secure, sustainable smart cities and the Internet of Things: perspectives, challenges, and future directions. Sustainability. 2024;16(4):1390. doi:10.3390/su16041390.
12. Hossain MA. Deep learning-based intrusion detection for IoT networks: a scalable and efficient approach. EURASIP J Inf Secur. 2025;2025(1):28. doi:10.1186/s13635-025-00202-w.
13. Amine MS, Nada FA, Hosny KM. Improved model for intrusion detection in the Internet of Things. Sci Rep. 2025;15(1):21547. doi:10.1038/s41598-025-92852-6.
14. Liao H, Murah MZ, Hasan MK, Aman AHM, Fang J, Hu X, et al. A survey of deep learning technologies for intrusion detection in Internet of Things. IEEE Access. 2024;12(1):4745–61. doi:10.1109/access.2023.3349287.
15. Alsubaei FS. Smart deep learning model for enhanced IoT intrusion detection. Sci Rep. 2025;15(1):20577. doi:10.1038/s41598-025-06363-5.
16. Susilo B, Muis A, Sari RF. Intelligent intrusion detection system against various attacks based on a hybrid deep learning algorithm. Sensors. 2025;25(2):580. doi:10.3390/s25020580.
17. Ghosh KP, Hasan M, Robin MTI, Hossain MA, Islam MS. A novel deep learning framework with temporal attention convolutional networks for intrusion detection in IoT and IIoT networks. Sci Rep. 2025;15(1):44624. doi:10.1038/s41598-025-32697-1.
18. Alabbadi A, Bajaber F. An intrusion detection system over the IoT data streams using eXplainable artificial intelligence (XAI). Sensors. 2025;25(3):847. doi:10.3390/s25030847.
19. Villafranca A, Thant KM, Tasic I, Cano MD. AI-enabled IoT intrusion detection: unified conceptual framework and research roadmap. Mach Learn Knowl Extr. 2025;7(4):115. doi:10.3390/make7040115.
20. Aleisa MA. Blockchain-enabled zero trust architecture for privacy-preserving cybersecurity in IoT environments. IEEE Access. 2025;13(4):18660–76. doi:10.1109/ACCESS.2025.3529309.
21. Govea J, Gaibor-Naranjo W, Villegas-Ch W. Transforming cybersecurity into critical energy infrastructure: a study on the effectiveness of artificial intelligence. Systems. 2024;12(5):165. doi:10.3390/systems12050165.
22. Priyadarshini I. Anomaly detection of IoT cyberattacks in smart cities using federated learning and split learning. Big Data Cogn Comput. 2024;8(3):21. doi:10.3390/bdcc8030021.
23. Ragab M, Ashary EB, Alghamdi BM, Aboalela R, Alsaadi N, Maghrabi LA, et al. Advanced artificial intelligence with federated learning framework for privacy-preserving cyberthreat detection in IoT-assisted sustainable smart cities. Sci Rep. 2025;15(1):4470. doi:10.1038/s41598-025-88843-2.
24. Paula B, Sarkera A, Abhia SH, Dasa P, Md Hasana M, Saqibb N. Potential smart grid vulnerabilities to cyber attacks: current threats and existing mitigation strategies. Heliyon. 2024;10(18):e14011. doi:10.1016/j.heliyon.2024.e37980.
25. Tightiz L, Nasimov R, Nasab MA. Implementing AI solutions for advanced cyber-attack detection in smart grid. Int J Energy Res. 2024;2024(1):6969383. doi:10.1155/2024/6969383.
26. Rajaperumal TA, Columbus CC. Transforming the electrical grid: the role of AI in advancing smart, sustainable, and secure energy systems. Energy Inform. 2025;8(1):51. doi:10.1186/s42162-024-00461-w.
27. Kabir MR, Halder D, Ray S. Digital twins for IoT-driven energy systems: a survey. IEEE Access. 2024;12:177123–43. doi:10.1109/access.2024.3506660.
28. Ibrahim N, Kashef R. Exploring the emerging role of large language models in smart grid cybersecurity: a survey of attacks, detection mechanisms, and mitigation strategies. Front Energy Res. 2025;13:1531655. doi:10.3389/fenrg.2025.1531655.

29. Durlik I, Miller T, Kostecka E, Zwierzewicz Z, Łobodzińska A. Cybersecurity in autonomous vehicles—are we ready for the challenge? Electronics. 2024;13(13):2654. doi:10.3390/electronics13132654.
30. Tanaji BA, Roychowdhury S. A survey of cybersecurity challenges and mitigation techniques for connected and autonomous vehicles. IEEE Trans Intell Veh. 2025;10(10):4742–57. doi:10.1109/TIV.2024.3493938.
31. Fernández Llorca D, Hamon R, Junklewitz H, Grosse K, Kunze L, Seiniger P, et al. Testing autonomous vehicles and AI: perspectives and challenges from cybersecurity, transparency, robustness and fairness. Eur Transp Res Rev. 2025;17(1):38. doi:10.1186/s12544-025-00732-x.
32. Reis MJCS. AI-driven anomaly detection for securing IoT devices in 5G-enabled smart cities. Electronics. 2025;14(12):2492. doi:10.3390/electronics14122492.
33. Liu C, Tan R, Wu Y, Feng Y, Jin Z, Zhang F, et al. Dissecting zero trust: research landscape and its implementation in IoT. Cybersecurity. 2024;7(1):20. doi:10.1186/s42400-024-00212-0.
34. Ishaque M, Albatati H, Nofal M. AI-based zero trust architecture for cognitive city networks. In: Proceedings of the 2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC); 2025 Feb 9–11; Jeddah, Saudi Arabia. p. 1–14. doi:10.1109/ICAISC64594.2025.10959378.
35. Esfandi S, Tayebi S, Byrne J, Taminiau J, Giyahchi G, Ali Alavi S. Smart cities and urban energy planning: an advanced review of promises and challenges. Smart Cities. 2024;7(1):414–44. doi:10.3390/smartcities7010016.
36. Velaga KS, Guo Y, Yu W. Edge AI for smart cities: foundations, challenges, and opportunities. Smart Cities. 2025;8(6):211. doi:10.3390/smartcities8060211.
37. Guo K, Zhan C, Niu M, Li X, Zheng Z, Sharma A. An integrated IoT and blockchain lightweight framework for secure smart cities. Discov Internet Things. 2026;6(1):11. doi:10.1007/s43926-025-00273-8.
38. Ali G, Thomas A, Mijwil MM, Al-Mahzoum K, Sallam M, Salau AO, et al. Blockchain and federated learning in edge-fog-cloud computing environments for smart logistics. Mesopotamian J Cyber Secur. 2025;5(2):735–69. doi:10.58496/MJCS/2025/044.
39. Babayomi OO, Igboanusi IS, Ahakonye LAC, Kim DS. Integrated blockchain and federated learning for the cybersecurity of distributed energy resources. Int J Electr Power Energy Syst. 2025;173(1):111286. doi:10.1016/j.ijepes.2025.111286.