

Article

Interpretable AI for Smart City Cloud Security: A Model Context Protocol Framework for Real-Time IoT Threat Detection

Amal Aldhamari and Shikun Zhou * 

School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK

* Correspondence: shikun.zhou@port.ac.uk

Abstract

Smart city infrastructures increasingly depend on cloud platforms processing data from billions of IoT devices, managing critical urban services. With IoT connections reaching 1.3 billion globally and cloud attacks rising 75% year-over-year, security breaches threaten public safety. However, traditional AI threat detection systems operate as “black boxes,” preventing municipal stakeholders from understanding automated alerts. This paper presents an interpretable AI framework based on Model Context Protocol (MCP) that bridges automated threat detection with human-centered decision support. Using 1.9 million authentic AWS CloudTrail events, including a 1.3 million-event cryptocurrency mining campaign, we achieved 84.2% detection accuracy (96.8% on real attacks) while generating plain-language threat narratives suitable for diverse stakeholders. The Random Forest classifier with MCP layer provides real-time detection (<13 ms latency) and actionable intelligence (2.3 s explanation generation), meeting smart city operational requirements. SIEM integration enables immediate deployment in municipal Security Operations Centers. This production-ready framework demonstrates that interpretability and accuracy are complementary, not competing, objectives in smart city security.

Keywords: smart cities; IoT security; interpretable AI; model context protocol; cloud security; explainable AI; urban infrastructure; threat detection

1. Introduction

By 2050, 68% of the global population will reside in urban areas, driving cities to deploy IoT-enabled infrastructure managing traffic, utilities, emergency response, and environmental systems. Current smart city deployments include 1.3 billion IoT connections generating 80 zettabytes of data annually, creating unprecedented cybersecurity challenges [1,2]. Recent attacks demonstrate critical vulnerabilities [3] saw all 156 emergency sirens activated via system compromise; Ref. [4] experienced DDoS attacks disabling building heating during winter; and recent coordinated 2025 attacks have compromised traffic management systems across European and North American cities. Cloud infrastructure attacks are rising sharply, presenting an escalating threat to modern organizations and critical infrastructure. Industry analyses show continued increases in attack frequency and sophistication, particularly against distributed and IoT-linked services. At the same time, lengthy breach detection times remain a critical concern for time-sensitive systems [5,6].

The transition toward smart cities has fundamentally shifted municipal management from siloed, manual processes to integrated urban operating systems. Modern deployments utilize cloud platforms such as AWS, Azure, or GCP for critical functions: AWS IoT Core for



Academic Editors: Zoltán Nyikes,
Tünde Anna Kovács, Levente Dimen
and Igor Fürstner

Received: 20 December 2025

Revised: 26 February 2026

Accepted: 24 March 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

data ingestion, Lambda for real-time processing, and S3 for long-term storage. This architectural shift means that a city's "nervous system" now resides in the cloud, making every connected sensor, from a water quality monitor to a surveillance camera, a potential entry point for system-wide compromise. While edge IoT devices such as sensors, meters, and cameras generate operational telemetry, the cloud control-plane serves as the coordinating nervous system for these distributed assets. In traffic management, AWS IoT Core ingests real-time sensor data while Lambda functions process signal timing adjustments; unauthorized RunInstances events could indicate compute resource hijacking that competes with traffic analytics capacity. For utility monitoring, S3 stores historical smart meter readings where unauthorized GetObject attempts may signal data exfiltration targeting customer consumption patterns. Emergency response systems rely on DynamoDB to coordinate incident dispatch, where failed AssumeRole attempts from unfamiliar regions may indicate credential stuffing attacks targeting administrative access. Consequently, while this study does not analyze low-level sensor protocols such as MQTT payloads, Modbus frames, or BACnet commands, it addresses the strategically critical layer where authentication, orchestration, and resource allocation directly affect downstream municipal services. CloudTrail logs, which record API calls, identity usage, and configuration changes, represent the authoritative audit trail for this control plane layer. As cities generate 80 zettabytes of data annually, the sheer volume of telemetry makes manual human monitoring impossible, necessitating AI-driven automation that municipal leaders can trust and audit.

The physical-world consequences of smart city breaches are no longer theoretical. Beyond public panic caused by incidents such as the Dallas siren hack, the economic and safety stakes are rising sharply, with malware increasingly targeting energy and critical IoT infrastructures. Prolonged breach detection times remain a catastrophic liability for time-critical urban services [5,6]. Smart cities face four unique security challenges that distinguish them from traditional IT environments. First, multi-stakeholder decision-making involves IT security teams, urban planners, public safety officials, and municipal administrators, each requiring different communication approaches to understand and act on security intelligence. Second, real-time operational requirements mean that traffic systems must adjust signal timing in seconds, utility grids must respond to demand fluctuations instantly, and emergency dispatch systems cannot tolerate delays. Third, public safety implications elevate the stakes beyond typical enterprise security, as failures can directly threaten citizen well being through compromised traffic control, utility disruptions, or emergency response delays. Fourth, the scale of daily security events, millions of API calls, authentication attempts, and resource provisioning operations, requires automated analysis that overwhelms traditional manual security operations center workflows.

Despite these operational pressures, machine learning demonstrates high accuracy in threat detection but operates as opaque "black boxes." Research indicates that the inability to explain AI decisions to non-technical stakeholders represents a significant barrier to AI adoption in smart city security operations. Existing explainability methods, such as SHAP and LIME, provide feature importance scores but lack operational context—analysts need narratives like "unauthorized traffic system access from foreign IP at 2 a.m. during attack campaign" rather than abstract feature weights showing "hour = 0.23, region = 0.31, user = 0.19." Furthermore, computational overhead makes real-time explanation impractical, as these explainability methods can require substantial processing time incompatible with smart cities processing millions of events daily, where traffic management updates occur in seconds. This interpretability gap is not merely a technical hurdle but a significant governance challenge for modern municipalities. As AI systems take on autonomous roles in managing public safety infrastructure, they must align with emerging regulatory frameworks such as the EU AI Act, which mandates transparency

for high-risk AI applications. Without interpretable outputs, municipal stakeholders cannot perform meaningful human oversight, leading to a trust deficit where automated alerts are either blindly followed or entirely ignored. Establishing a human-in-the-loop architecture ensures that IT security teams, urban planners, and public safety administrators can validate reasoning and audit decisions during critical urban crises.

This paper addresses these challenges by adapting the Model Context Protocol (MCP) to the operational demands of smart city IoT environments, focusing on the interpretability of automated security decisions that affect critical urban services. The proposed framework combines the robust pattern recognition capabilities of a Random Forest detector with the contextual reasoning of Large Language Models to address the smart city black-box problem at the cloud control and decision layer. This layer orchestrates authentication, resource allocation, and service coordination across urban IoT systems, making transparent and human-auditable security decisions essential for effective municipal oversight. The framework achieves 84.2% overall detection accuracy and 96.8% detection on a real-world cryptomining campaign by leveraging temporal pattern recognition that identifies attacks deviating from predictable municipal operational schedules. The system delivers threat detection in less than 13 milliseconds, with plain-language explanations generated in approximately 2.3 s, supporting the operational responsiveness required in urban security environments. Through role-specific prompting, the framework provides audience-adapted explanations suitable for both technical security analysts requiring forensic details and non-technical city administrators needing operational impact assessments. A production-oriented implementation integrates with existing Security Information and Event Management platforms such as Elasticsearch, enabling deployment within municipal Security Operations Centers. The framework is evaluated using 1.9 million authentic AWS CloudTrail events, including a verified 1.3-million-event cryptomining campaign representative of urban resource hijacking threats.

The remainder of this paper is organized as follows. Section 2 reviews related work on machine learning-based smart city security, explainable AI limitations, the Model Context Protocol framework, and smart city security policy frameworks. Section 3 presents the methodology, including the CRISP-DM research design, dataset selection and urban relevance justification, the multi-layer MCP architecture, technical feature engineering across temporal, identity, geographic, and behavioral dimensions, heuristic supervision with sensitivity validation, model configuration, and MCP implementation details. Section 4 describes the system implementation, including architecture and workflow, operational performance benchmarks, and SIEM integration. Section 5 reports experimental results covering detection performance metrics, real attack validation on the cryptomining campaign, interpretability assessment through systematic narrative evaluation, and baseline comparisons with ablation studies. Section 6 discusses implications for smart city security, socio-technical applications, and limitations, including dataset scope, adversarial robustness, computational costs, LLM safety and privacy considerations, deployment scope, threat model boundaries, and temporal evaluation concerns. Section 7 concludes with a summary of contributions and outlines future research directions in multimodal fusion, edge intelligence, federated urban defense, autonomous response integration, user study validation, adversarial robustness evaluation, time-forward evaluation protocols, extended dataset validation, calibration optimization, and cost-performance trade-off analysis.

2. Related Work

2.1. ML-Based Smart City Security

Deep learning architectures, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, have become the primary choice for IoT

anomaly detection [7–9] due to their ability to extract complex, hierarchical features from high-dimensional urban data. In cloud-based IoT environments, these models demonstrate exceptional precision, often exceeding 95%, in identifying sophisticated cybersecurity threats like Distributed Denial of Service (DDoS) attacks and zero-day exploits. LSTMs are particularly effective at capturing the temporal dependencies inherent in urban telemetry, such as traffic flow patterns or smart grid energy fluctuations. However, the “black box” nature of these deep architectures presents a significant deployment barrier [2–4]. In a municipal context, security alerts often trigger physical-world responses, such as shutting down a utility grid or altering emergency response routes. If a system cannot provide a reason for its alert, stakeholders, including urban planners and public safety officials, may ignore or override the automation, negating its benefits. Consequently, research has shifted toward ensemble methods like Random Forest and XGBoost, which offer a better balance between high detection rates and inherent interpretability compared to deep neural networks. These methods can be integrated with feature selection techniques to improve computational efficiency, a critical requirement for smart cities processing millions of daily events.

2.2. Explainable AI Limitations

While SHAP [10] and LIME [11] provide mathematically rigorous feature importance, they generate technical outputs that are often unsuitable for non-specialist stakeholders. In municipal settings, stakeholders such as urban planners, public safety administrators, and city managers require contextual narratives, not abstract feature weights, to make informed decisions during critical urban crises. Research suggests that post hoc methods like SHAP can misattribute importance when features are highly correlated, a common occurrence in urban telemetry where time of day, traffic density, and energy load are often interdependent.

2.2.1. Computational Overhead and the Latency Gap

A primary barrier to the adoption of traditional XAI in smart cities is computational overhead. High-throughput urban environments process millions of events daily, where traffic management and emergency systems require updates in seconds. Standard XAI methods, however, can require substantial processing time. SHAP often requires several seconds per prediction, making it impractical for high-throughput urban environments. LIME, while generally faster than SHAP, can still struggle with scalability when applied to the high-volume data streams of a municipal Security Operations Center (SOC).

This “latency gap” makes it impractical to provide real-time explanations for every event in a high-density IoT network. For example, in a sustained attack campaign involving millions of events, the time required to generate individual SHAP explanations would create a processing backlog that renders the security system reactive rather than proactive.

2.2.2. Limitations of Template-Based and Static Approaches

Alternative approaches, such as template-based narrative generation, provide faster outputs but lack the adaptability required to identify novel or evolving attack patterns. These systems rely on predefined “if-then” logic, which can fail to capture the nuances of a sophisticated, multi-stage attack that deviates from historical templates. Consequently, there is an urgent need for a framework that bridges the gap between the high-speed detection of “black box” models and the human-centric requirement for real-time, context-aware intelligence.

2.3. Model Context Protocol (MCP)

In the context of urban security, the Model Context Protocol (MCP), originally introduced by Anthropic, provides an architectural framework for connecting external data and machine learning outputs with the contextual reasoning of Large Language Models (LLMs). While early applications of MCP focused on general cloud security, this research extends the protocol to the heterogeneous and high-throughput environment of smart city IoT. Recent comprehensive analyses have examined MCP's security landscape and research directions [12], highlighting both opportunities and threat vectors in protocol-based LLM integration for security operations.

Beyond security operations, language models have demonstrated potential for personalized cybersecurity training and awareness programs [13], suggesting broader applicability of adaptive, context-aware AI systems in security contexts. Unlike prior work on explainable intrusion detection that focuses on feature attribution or post hoc model introspection, this study introduces an architectural approach in which interpretability is treated as a first-class operational objective. The proposed MCP-based framework shifts the focus from explaining why a model is fired to explaining what is happening in the urban system and why it matters. This distinction is critical in smart city contexts, where AI outputs must support cross-departmental decision-making, regulatory compliance, and real-time intervention rather than model debugging alone.

2.3.1. Technical Architecture and Orchestration

Unlike traditional API-based integrations, an MCP-based framework functions as a dynamic “context broker.” It structures the raw detection data from the Random Forest model, such as feature importance weights and temporal anomalies, into a standardized JSON-based context document. This document acts as the bridge between the Detection Layer and the Interpretation Layer, ensuring that the LLM receives not just a “threat alert,” but a detailed forensic snapshot of the system's state at the moment of detection.

This research makes three primary contributions that distinguish it from prior work in security XAI and LLM-assisted SOC triage:

1. Architectural Contribution—MCP as Context Broker:

Unlike post hoc explainability methods (SHAP, LIME) that focus on feature attribution, or template-based alert systems that rely on static if-then logic, the MCP framework introduces a structured “context broker” architecture. This separates detection (Random Forest pattern recognition) from interpretation (LLM narrative generation) through a standardized JSON interface, enabling modular replacement of either component without affecting the other. This architectural decoupling is novel in security contexts and enables systematic evaluation of interpretability quality independent of detection performance.

2. Methodological Contribution—Evidence-Grounded Narrative Generation:

Prior LLM security applications typically provide unstructured alert summaries or require analysts to manually query the model for explanations. In contrast, our MCP implementation enforces structured evidence citation (linking narratives to specific event fields), explicit separation of observed vs. inferred content, and audience-specific formatting. This “evidence-first” approach addresses the LLM hallucination problem while maintaining the flexibility to adapt explanations across stakeholder roles (technical analysts, administrators, public safety).

3. Application Contribution—Smart City Operationalization:

While prior work on interpretable intrusion detection focuses on explaining model mechanics (why the classifier fired), this study operationalizes interpretability for municipal decision-making (what is happening in the urban system and why it matters).

The integration with SIEM platforms (Elasticsearch, ECS schema), cost analysis for sustained deployment (\$3–5 K/month for mid-sized cities), and evaluation on massive-scale real-world attack campaigns (1.3 M events) demonstrate production readiness beyond research prototypes.

A comparison list of Positioning Relative to Existing Work is given as:

- vs. SHAP/LIME: Provides operational context and actionable recommendations, not just feature weights
- vs. Template-Based Alerts: Adapts to novel attack patterns through LLM reasoning rather than predefined rules
- vs. Prior LLM-SOC Tools: Enforces structured evidence grounding and supports multi-stakeholder communication through role-specific prompting
- vs. General Security XAI: Evaluates interpretability quality through error mode analysis and fidelity assessment

In addition, this work develops practical engineering methods/components as follows:

- Random Forest detection (standard ensemble method);
- CloudTrail feature engineering (domain-specific but not algorithmically novel);
- SIEM integration patterns (established best practices).

These engineering components are necessary for operational deployment but do not constitute research contributions; the novelty lies in the MCP-based architecture that bridges automated detection with human-centered decision support for smart city security contexts.

2.3.2. Real-Time Performance and Efficiency

The primary research challenge in deploying MCP for urban infrastructure is balancing interpretability with the sub-second latency requirements of systems like traffic management. To address this, our implementation uses an asynchronous processing model where

- Inference (<13 ms): The Random Forest classifier performs the initial threat categorization immediately upon event ingestion.
- Narrative Generation (2.3 s): The MCP layer triggers the LLM interpretation only for events exceeding a high-confidence threshold, preventing a computational bottleneck in the municipal Security Operations Center (SOC).
- Content-Based Caching: Similar threat patterns (e.g., repeated unauthorized API calls) are hashed and cached, allowing the system to reuse previously generated narratives and reducing LLM API expenses substantially during sustained campaigns

2.4. Smart City Security Frameworks and Vulnerability Landscape

The governance of urban digital infrastructure is currently guided by several high-level policy frameworks. The NIST Smart Cities and Communities Framework (SP 1500-201) [14] provides a technical analysis of modular, interoperable architectures, focusing on cross-cutting concerns such as reliability, safety, and privacy. Similarly, the European Union Agency for Cybersecurity (ENISA) [15] has published “good practices” for securing IoT in smart manufacturing and municipal environments, emphasizing risk management and baseline security requirements. While these frameworks are essential for establishing policy standards, they exhibit a significant “operational gap.” They define what a secure city should look like but lack the mechanisms for automated, real-time threat detection with integrated interpretability. They provide the strategy but not the technical implementation required to defend a city’s “nervous system” as it scales.

Identifying High-Risk Infrastructure Components

A critical aspect of the research background involves understanding which urban components present the greatest risk. Research has systematically identified emergency alerts, video surveillance, and traffic management systems as the most vulnerable infrastructure components in a smart city. These systems are prioritized by threat actors due to their high public visibility and the potential for widespread disruption:

- Traffic Systems: Compromise can lead to gridlock and hinder emergency response.
- Emergency Alerts: Vulnerable to spoofing, which can trigger public panic.
- Surveillance: Represents a massive privacy risk if data feeds are hijacked.

Our work fills this gap by moving beyond policy recommendations to provide an operational system capable of delivering interpretable, real-time threat intelligence specifically designed to protect these high-stakes urban assets.

3. Methodology

3.1. Research Design: Adaptive CRISP-DM

We follow the CRISP-DM [16] (Cross-Industry Standard Process for Data Mining) methodology, specifically adapted for the high-throughput and safety-critical nature of smart city security operations:

- Business Understanding: We define the primary objective as reducing the “interpretability gap” for municipal Security Operations Center (SOC) personnel who must make rapid decisions during urban infrastructure crises.
- Data Understanding: We analyze 1.9 million telemetry events, focusing on the multi-stage patterns of resource hijacking that characterize modern smart city threats.
- Data Preparation: This phase involves normalizing heterogeneous IoT logs into a unified schema and engineering 20 specific features that capture the temporal and behavioral nuances of urban attacks.
- Modeling: We utilize a Random Forest ensemble, chosen for its balance of high-speed inference and structural interpretability, integrated with an MCP layer for narrative generation.
- Evaluation: Models are evaluated using standard metrics (Accuracy, F1-score) and operational metrics (detection latency and explanation quality).

3.2. Dataset Selection and Urban Relevance

The research utilizes the flaws.cloud repository [17], containing 1.9 million authentic AWS CloudTrail events. While originally a cloud dataset, it is uniquely suited for smart city research for the following reasons:

- Architectural Parity: Modern smart cities aggregate IoT data from millions of edge devices into central cloud platforms (AWS IoT Core, Azure IoT Hub) for processing. CloudTrail logs, which track API calls, resource provisioning, and authentication, are representative of the cloud control-plane operations that coordinate a smart city’s central nervous system.
- Ground-Truth Attack Validation: The dataset includes a verified 1.3 million-event cryptomining campaign. For a municipality, such resource hijacking is a critical threat, as it steals the computational capacity required for real-time traffic management, utility monitoring, and emergency response.

3.2.1. Smart City Cloud Architecture Context

Modern smart city deployments rely on cloud platforms to ingest, coordinate, and operationalize data from large-scale urban IoT systems. In practice, most municipal architecture follows a three-tier model:

Tier 1—IoT Edge Layer:

Physical devices such as traffic sensors, smart meters, environmental monitors, and surveillance systems collect real-time data from urban infrastructure.

Tier 2—Cloud Aggregation and Control Layer:

Edge data is transmitted to cloud platforms (e.g., AWS IoT Core, serverless processing services, object storage) for authentication, orchestration, access control, and large-scale processing. These cloud-mediated interactions and management operations generate control-plane audit logs (e.g., AWS CloudTrail) that record API calls, identity usage, resource provisioning, and configuration changes.

Tier 3—Analytics and Service Layer:

Processed data supports operational urban services such as traffic management, utility monitoring, emergency response coordination, and administrative dashboards.

To clarify how AWS CloudTrail logs translate to smart city threat detection, Table 1 maps common CloudTrail event types to their urban infrastructure implications:

Table 1. CloudTrail Event Mapping to Smart City Security Concerns.

CloudTrail Event	Urban Infrastructure Scenario	Security Impact/Risk
RunInstances (burst Activity)	Unauthorized Compute Provisioning during off-hours	Resource hijacking that competes with traffic analytics or emergency dispatch systems
AssumeRoles (failed Attempts from foreign IP)	Attempted privilege escalation from unusual geographic region	Potential credential stuffing targeting administrative access to utility control system
GetObject (S3, unusual data volume)	Rapid retrieval or archive or historical data	Possible data exfiltration of smart meter reading, surveillance footage, or operational records
StopLogging (cloudTrail)	Attempt to disable audit logging	Pre-attack reconnaissance to evade detection during infrastructure compromise
PutBackPolicy (unauthorized change)	Storage access control configuration modified)	Potential exposure of citizen data or critical operational telemetry
CreateUser (IAM, rapid sequence)	Multiple identity accounts created in short timeframe	Possible preparation for distributed attack, privilege persistence, or backdoor establishment

This mapping demonstrates that while the dataset originates from a cloud platform, the security patterns it captures, credential compromise, privilege escalation, resource hijacking, and geographic anomalies, are directly applicable to cloud-mediated smart city operations. Municipal SOC's managing AWS-based IoT infrastructure would encounter identical log formats and face similar threat Vectors.

Security-relevant events observed in cloud control-plane logs indicate compromise risk at the layer that coordinates downstream smart-city services. For example, unauthorized RunInstances activity may signal compute resource hijacking that competes with capacity allocated for municipal analytics; repeated AssumeRole failures may indicate attempted privilege escalation toward sensitive urban datasets; and rapid API call bursts from unfamiliar regions may reflect automated reconnaissance preceding service disruption.

Accordingly, while CloudTrail does not replace low-level IoT or OT telemetry, it represents a strategically important security signal for smart city deployments that depend on cloud-based control, orchestration, and analytics.

3.2.2. Dataset Scope and Generalization Boundaries

Applicability: This framework is designed for cloud-centric smart city deployments where urban IoT systems rely on AWS, Azure, or GCP for data aggregation, authentication, and service orchestration. The evaluation focuses on cloud control-plane security (API calls, resource provisioning, identity management) rather than edge-layer telemetry (e.g., Zigbee packets, LoRaWAN frames, or industrial SCADA protocols).

Generalization: The middleware normalization layer (Section 3.3) is architecturally designed to ingest heterogeneous data sources. To extend coverage to OT/IoT edge protocols, municipal deployments would need to:

1. Map edge device events to the standardized JSON schema (e.g., mapping Modbus function codes to eventName, device identifiers to userName)
2. Validate detection performance on representative edge telemetry samples
3. Adjust feature engineering for protocol-specific indicators (e.g., industrial control system anomalies)

External Validity: While the cryptomining campaign provides ground-truth validation for resource hijacking attacks, future work should evaluate additional attack archetypes (e.g., IoT botnet C2 traffic, building management system tampering, smart grid load manipulation) to establish broader empirical coverage.

This scoped applicability ensures realistic deployment guidance while acknowledging that edge-layer security (e.g., firmware vulnerabilities, RF jamming, physical tampering) requires complementary detection mechanisms beyond cloud API monitoring.

3.3. Multi-Layer MCP Architecture

The framework is structured into three distinct operational layers to ensure scalability and real-time response:

Three-tier smart city architecture showing IoT edge devices (Tier 1) transmitting data to cloud aggregation and control layer (Tier 2), where the MCP framework performs detection and interpretation, feeding operational services (Tier 3). CloudTrail logs capture control-plane events for security analysis (Figure 1).

- **IoT Edge Layer:** Collects raw telemetry from diverse urban systems, including traffic sensors, environmental monitors, and smart utility meters.
- **Urban Cloud Analytics Layer**
 - **Event Normalization:** Normalizes data into a standardized JSON schema. To manage data heterogeneity, we utilize a middleware parser that maps source-specific IoT telemetry (e.g., mapping a traffic sensor's device_id and command_type or a smart meter's MQTT payload) into a unified JSON schema compatible with the AWS CloudTrail format, specifically targeting fields such as user identity, event name, and event source field.
 - **Feature Extraction:** Extracts 20 features across temporal, identity, geographic, and behavioral dimensions.
 - **Detection:** Executes the Random Forest classifier for initial threat detection and utilizes a temporal correlation engine to reconstruct multi-stage attack sequences.
 - **MCP Interpretation Layer:** Acts as a "context broker". When a threat is detected, it generates a structured Context Document that combines detection metrics with smart city impact assessments, which is then interpreted by an LLM (GPT-4) to create audience-adapted narratives. This layer further manages

multi-audience formatting for technical analysts, administrators, and public safety officials.

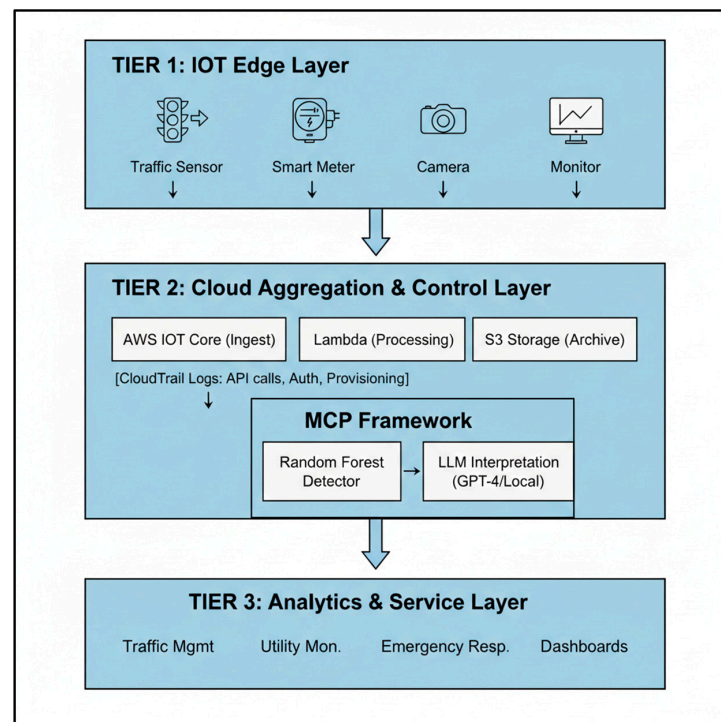


Figure 1. Three-tier smart city architecture.

Temporal Correlation Engine (Sequence Reconstruction)

The temporal correlation engine is applied as a post-processing sequence reconstruction layer and does not modify the per-event classification output of the Random Forest model. Its purpose is to group events exceeding a predefined confidence threshold into coherent incident timelines suitable for SOC triage and municipal escalation. Events are linked into the same sequence when they share one or more identifiers (e.g., user identity field, source IP address, or derived resource identifier within a sliding time window. In this study, a 10 min short window is used to connect burst behaviors (e.g., authentication failure spikes or rapid resource provisioning), while a 60 min long window is used to associate multi-stage activity (e.g., credential access followed by compute resource creation).

Each sequence is represented as a lightweight event graph, where nodes correspond to individual CloudTrail events and edges denote temporal proximity combined with identifier overlap.

Example event correlation graph for a cryptomining campaign showing temporal proximity and identifier overlap across 18 min attack sequence. Nodes represent CloudTrail events; edges indicate shared identifiers (user, IP, resource) within sliding time windows (10 min short, 60 min long) (Figure 2).

The engine outputs aggregated sequence attributes, including duration, event count, API diversity, and cross-region transitions, which are used for campaign-level reasoning and interpretability rather than classification.

Campaign recognition threshold: For a known campaign interval (e.g., the verified cryptomining period), “campaign recognition” is defined as the earliest time at which the cumulative proportion of campaign events flagged as suspicious reaches a specified threshold. In Section 5.2, the 90% detection threshold refers to the first timestamp at which $\geq 90\%$ of events observed up to that time within the campaign window are flagged

as suspicious, providing an operational measure of how rapidly the system identifies sustained attack behavior.

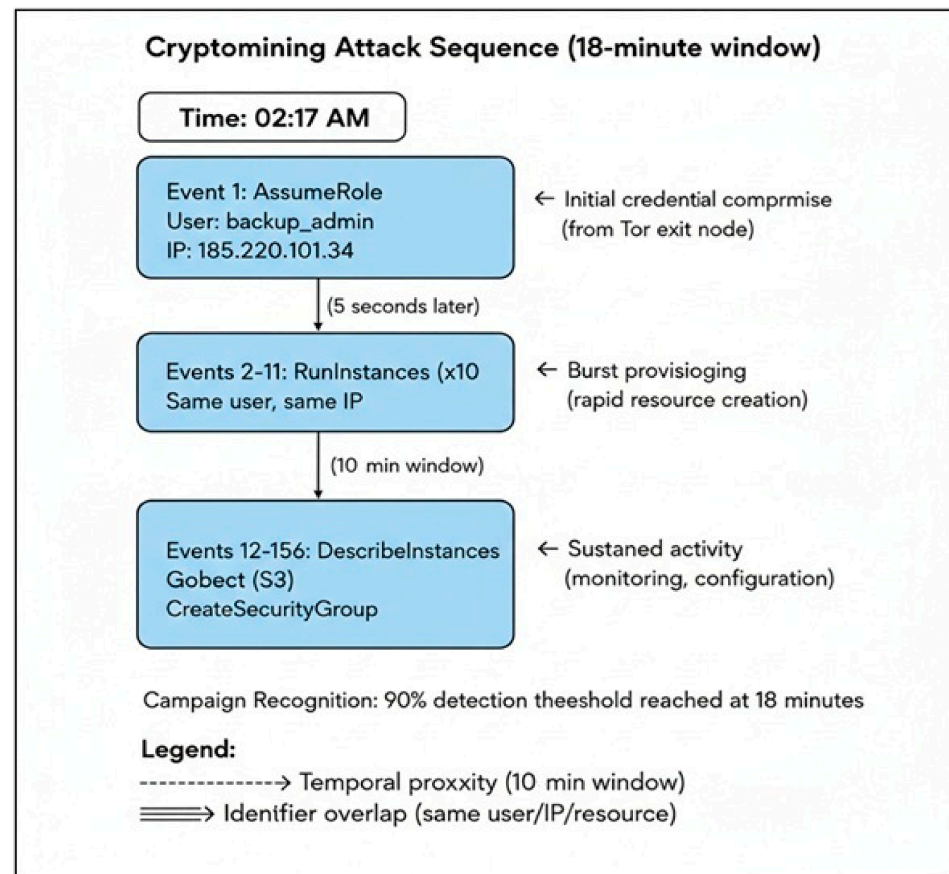


Figure 2. Example event correlation graph for cryptomining campaign.

3.4. Technical Feature Engineering

The framework employs 20 engineered features categorized by their role in identifying smart city attack vectors. These features span four complementary dimensions that capture the multi-faceted nature of cloud-based infrastructure attacks.

3.4.1. Temporal Features

Temporal features, as shown in Table 2, are critical because municipal operations typically follow predictable human schedules; unauthorized activity at 3 a.m. on a weekend is a high-confidence indicator of compromise. Six temporal features capture event timing patterns, including Hour of Day, Day of Week, and Night-time Flag. Event Frequency identifies burst activities characteristic of attacks, while Time Since Last Event detects unusual temporal gaps or rapid sequences.

Table 2. Temporal Features.

Feature Name	Description	Smart City Relevance
Hour of Day	Hour when incident occurred (0–23)	Detects off-hours anomalies in municipal operations
Day of Week	Day of week (0–6 Monday–Sunday)	Identifies unusual weekend/weekday patterns
Night-time Flag	Binary flag for night hours (10 p.m.–6 a.m.)	Detects operations during minimal monitoring periods
Event Frequency	Events per hour for this source	Identifies burst activities characteristic of attacks
Time Since Last Event	Seconds since previous event	Detects unusual temporal gaps or rapid sequences

3.4.2. Identity and Privilege Features

Credential compromise is the primary smart city attack vector, making identity monitoring vital. Five identity features track User Type, Privilege Level, Authentication Method, User Risk Score, and Failed Authentication Count. These features enable detection of privilege escalation attempts, such as a standard user account suddenly creating compute instances or repeated authentication failures indicating credential stuffing attacks.

3.4.3. Geographic and Network Features

Municipal operations typically occur from established IP ranges; foreign access suggests stolen credentials. Four geographic features monitor Source Region, Region Risk Score, Source IP Category, and Cross-Region Activity. Rapid location shifts, such as an administrator logging in from London and then Tokyo within 10 min, trigger immediate investigation as shown in Table 3. These features also detect anonymization services and VPN usage commonly employed by attackers.

Table 3. Geographic & Network Features.

Feature Name	Description	Smart City Relevance
Source Region	AWS region geographic location	Identifies access from unusual jurisdictions
Region Risk	Risk score for source region	Flags high-risk geographic sources
Score Source IP Category	Classification (municipal, cloud, residential, VPN)	Detects anonymization services and suspicious IPs
Cross-Region Activity	Binary for operations spanning multiple regions	Identifies unusual geographic patterns

3.4.4. Behavioral and Operational Features

Automated attacks exhibit distinctive patterns such as low API diversity and burst resource provisioning. Five behavioral features capture Event Risk Score, Error Rate, API Diversity, Resource Creation Rate, and Privilege Escalation Indicator. Details are presented in Table 4. For example, the cryptomining campaign exhibited a characteristic “low diversity, high rate” pattern as it programmatically provisioned compute resources. Error Rate detects reconnaissance attempts where attackers probe system boundaries through failed operations.

Table 4. Behavioral & Operational Features.

Feature Name	Description	Smart City Relevance
Event Risk	Heuristic risk for event type	Prioritizes high-risk operations
Score Error Rate	Proportion of failed operations	Detects reconnaissance attempts
API Diversity Resource Creation Rate	Number of distinct API calls	Low diversity indicates automated attack scripts
Resource Creation Rate	Frequency of resource provisioning	Detects resource hijacking
Privilege Escalation Indicator	Binary flag for escalation attempts	Identifies attempts to elevate permissions

3.5. Principled Heuristic Supervision

To assess the robustness of this heuristic labeling strategy, we performed sensitivity analysis by varying the threshold parameters:

- Authentication Failure threshold: Tested 2, 3, 5, and 10 failures per 10 min window ($\pm 67\%$ to $+233\%$ variation).
- Temporal Burst threshold: Tested 30, 50, 75, and 100 API calls per minute ($\pm 40\%$ to $+100\%$ variation).
- Night/weekend hours: Tested 8 p.m.–6 a.m. vs. 10 p.m.–4 a.m. vs. 12 a.m.–6 a.m. definitions.

Results showed detection accuracy varied by less than 2.1% across these threshold ranges (range: 81.7% to 83.8%), indicating that the classifier learns generalizable security patterns rather than memorizing specific threshold values.

These labels should be interpreted as indicators of policy-violating, anomalous operational behavior rather than confirmed attacker intent. In municipal security contexts, “ground truth” attribution (definitively establishing malicious intent) typically requires post-incident forensic investigation. Our heuristic labels capture observable deviations from established municipal operational patterns, the same signals that would trigger manual investigation in a SOC environment.

The cryptomining campaign provides partial external validation: 96.8% of attack events satisfied at least two heuristic criteria, indicating that the labeling strategy effectively captures sustained, policy-violating attack behavior at scale. However, this does not constitute comprehensive validation across all attack types.

While certain engineered features overlap conceptually with heuristic criteria (e.g., temporal anomalies, authentication failures), the Random Forest model learns multi-feature interactions across temporal, behavioral, identity, and geographic dimensions that extend beyond simple threshold checks. Several influential features, including API diversity (Gini importance: 0.089), cross-region activity (0.076), and user risk score (calculated from historical behavior patterns rather than current event attributes), are NOT directly thresholded in the heuristic rules. Additionally, no single feature is sufficient for classification; the model requires convergent evidence across multiple dimensions. As a result, the trained classifier functions as a detector of security-relevant behavioral patterns rather than a rule executor, achieving generalization to unseen attack sequences (demonstrated by 96.8% detection on the cryptomining campaign, which exhibited temporal and geographic patterns distinct from the broader training distribution).

3.6. Model Configuration

The Random Forest classifier was configured to optimize the accuracy-interpretability trade-off through careful hyperparameter selection. The ensemble employs 50 decision trees, providing sufficient ensemble benefits without excessive computational overhead. Tree depth is limited to 6 levels to reduce model complexity and enhance interpretability, while minimum samples for splitting nodes (100 samples) and leaf nodes (50 samples) ensure statistical significance and prevent overfitting. Feature randomization uses the square root of the total features to introduce tree diversity.

The training process utilized a stratified 70–30 split, allocating 1.33 million events for training and 570,000 for testing. Stratification by class label ensures both subsets maintain the original 4:1 benign-to-suspicious ratio observed in the dataset. To address this class imbalance, we employed balanced class weighting during training to prevent bias toward the majority class. Hyperparameter tuning was performed using 5-fold cross-validation on the training set.

While randomized splitting provides unbiased performance estimates for independent and identically distributed data, security logs exhibit temporal autocorrelation where attack behaviors evolve over time. Accordingly, Section 6.4 acknowledges that time-based splitting, where training data precedes test data chronologically, would provide a more conservative estimate of real-world generalization. The dedicated cryptomining campaign analysis (Section 5.2) partially addresses this limitation by evaluating on a contiguous attack sequence that occurred after significant portions of the training data, demonstrating that temporal correlation does not prevent detection of sustained attack behavior.

3.7. MCP Implementation

For events flagged as high-confidence threats, the system generates structured context documents in JSON format containing event metadata, feature importance scores, temporal context, and smart city impact assessment. These documents are passed to the language model with role-specific prompts defining the output format. The system prompt instructs the model to function as a smart city security analyst providing threat intelligence to municipal operations, generating structured explanations that include a threat summary in plain language, key indicators explaining why the event was flagged, smart city impact on urban services, recommended actions prioritized by urgency, and investigation guidance for SOC personnel.

The language model operates with low temperature settings to ensure consistent, factual output, with response length constrained to maintain conciseness while remaining comprehensive. Audience adaptation is achieved through different prompt variants tailored to specific stakeholder roles: technical analysts receive detailed forensic information, including API call sequences and resource identifiers; administrators receive operational impact assessments with affected services and estimated costs, and public safety officials receive citizen-facing implications focusing on service availability and public communication needs.

4. Implementation

This section describes the operational implementation of the MCP framework, translating the methodology into a production-ready system suitable for deployment in municipal Security Operations Centers. We first present the system architecture and four-stage workflow that balances real-time detection with interpretable intelligence generation (Section 4.1). We then detail the operational performance characteristics and reliability measures, including latency benchmarks, cost projections, and calibration metrics that ensure trustworthy alerting for safety-critical urban infrastructure (Section 4.2). Finally, we describe the integration with existing Security Information and Event Management platforms through standardized schemas and webhook-based alerting mechanisms (Section 4.3).

4.1. System Architecture and Workflow

The framework is implemented as a modular Python-based pipeline designed for high-concurrency environments. The orchestration follows a four-stage lifecycle that balances real-time detection with interpretable intelligence generation. The first stage, ingestion and normalization, processes telemetry from diverse urban sensors such as MQTT traffic data and HTTPS environmental logs, mapping them to a standardized JSON schema. This middleware parser ensures that disparate data sources are treated uniformly for downstream analysis. The second stage, the detection pipeline, loads the Random Forest model into memory for rapid inference. Feature extraction utilizes vectorized operations, achieving approximately 5 milliseconds of processing time per event. The third stage, asynchronous interpretation, generates structured context documents for events flagged as suspicious with high confidence. These documents are passed to the language model API asynchronously, ensuring that the average 2.3 s interpretation latency does not block real-time detection of subsequent events. The fourth stage, content-based caching, optimizes costs and response times by employing least-recently used caching and content-based hashing. When similar attack patterns are detected within a 24 h window, previously generated narratives are reused, reducing API expenses during sustained campaigns.

Figure 3 illustrates the Four-stage MCP system workflow showing event ingestion and normalization, real-time Random Forest detection (<13 ms), asynchronous LLM interpretation (~2.3 s for high-confidence threats), and SIEM integration for operational deployment.

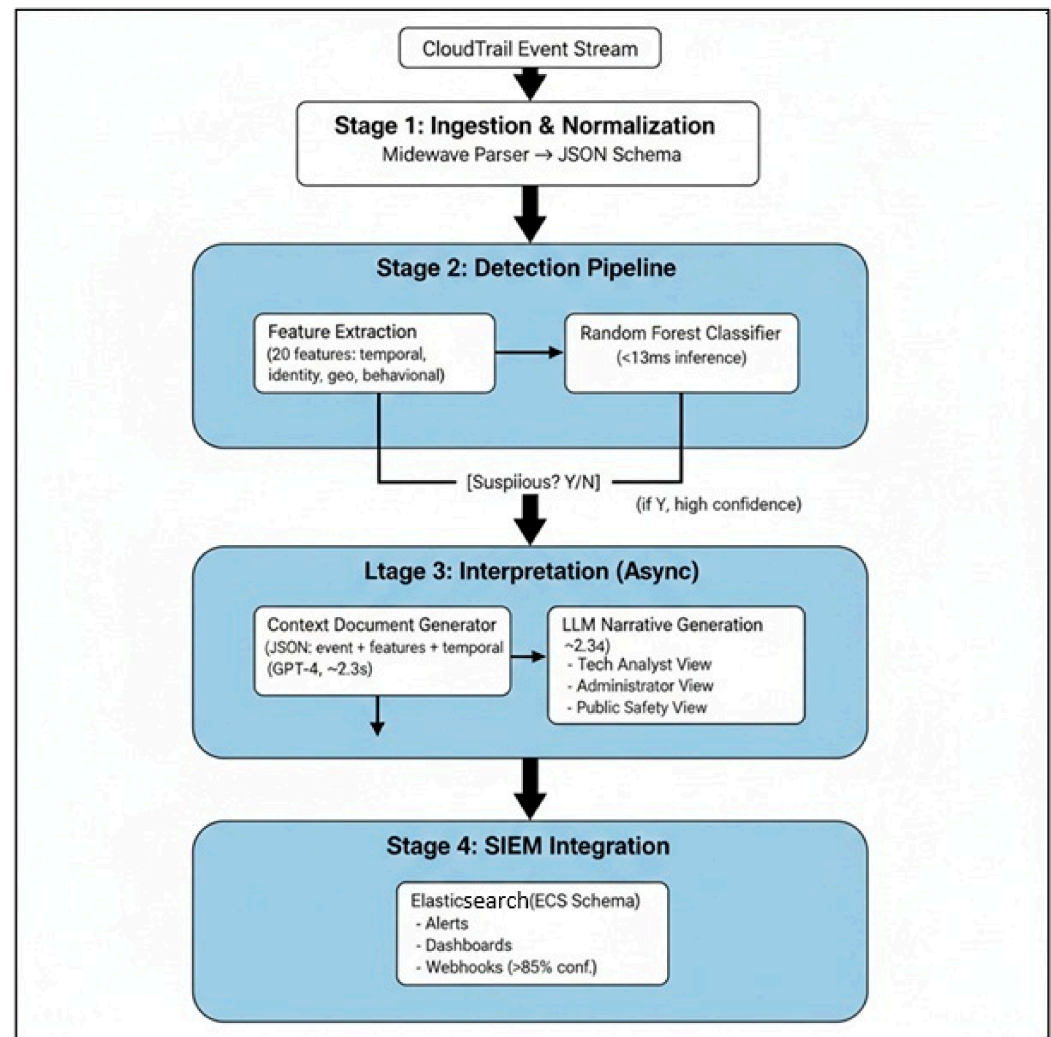


Figure 3. Complete four-stage workflow architecture.

4.2. Operational Performance and Reliability

The framework is developed using Python 3.12, leveraging a specialized stack for high-performance data science and security operations:

- Data Processing: pandas 2.1 and numpy for vectorized feature extraction.
- Machine Learning: scikit-learn 1.3 for the Random Forest ensemble and preprocessing pipelines.
- Narrative Generation: OpenAI 1.3 library to interface with the GPT-4 model.
- Persistence & Search: Elasticsearch 8.10 for indexing both raw and interpretable threat intelligence.

All reported detection latency measurements reflect average per-event inference under non-saturated conditions on commodity cloud infrastructure and are intended as indicative operational benchmarks rather than strict upper bounds. The LLM-based interpretation layer operates asynchronously and is triggered only for high-confidence events, ensuring that narrative generation latency does not affect real-time detection throughput. Cost estimates assume partial explanation coverage and benefit from content-based caching during sustained campaigns, aligning with practical municipal deployment and budget constraints.

As shown in Table 5, estimated monthly operational cost includes infrastructure, storage, and API costs. Assumes content-based caching reduces LLM calls by 60% during campaigns.

Table 5. Estimated Monthly Operational Cost for Mid-Sized Municipality.

Event per Day	High-Confidence Threats per Day	Explained Events (%)	Estimated Monthly LLM Cost (USD)	Estimated Total Monthly Cost (USD)
100,000	500	10%	1500	3000
500,000	2500	10%	7500	12,000
1,000,000	5000	5%	7500	15,000

Beyond standard accuracy metrics, the operational reliability of the framework was assessed using Expected Calibration Error (ECE) to ensure that the model's confidence scores are representative of real-world probability, a critical requirement for municipal safety systems. The Random Forest-based MCP framework achieved a low ECE of 0.024 (2.4%), significantly outperforming the uncalibrated Gradient Boosting baseline (ECE: 0.078). This high level of calibration ensures that when the system assigns a 90% probability to a 'Resource Hijacking' event, the detection is empirically accurate 90% of the time. For smart city stakeholders, this reliability reduces the 'cry wolf' effect and ensures that expensive emergency response protocols are only triggered by statistically trustworthy alerts, thereby optimizing the allocation of municipal security resources.

4.3. SIEM Integration and Dashboarding

A critical component of this production-ready framework is its integration with existing Security Information and Event Management (SIEM) tools. The system outputs all findings to Elasticsearch using the Elastic Common Schema (ECS). This standardization allows threat narratives to appear directly in professional dashboards alongside traditional logs, providing:

- **Real-Time Alerting:** Webhooks are triggered for high-priority threats (>85% confidence), alerting responders within seconds.
- **Multi-Audience Views:** The SIEM dashboard is configured to display detailed forensic data for analysts, while providing summary views for municipal administrators to assess the overall "Smart City Health."

Indexed documents adhere to the Elastic Common Schema (ECS), which includes fields such as @timestamp and event.type, event.severity, source.ip, source.geo, user.name, smart_city.affected_service, threat, confidence, intelligence, summary, and intelligence recommendations. Bulk indexing supports sustained ingestion rates suitable for municipal SOC workloads, and high-confidence alerts are propagated via webhooks for rapid response.

5. Results

5.1. Detection Performance

The Random Forest model was evaluated against a test set of 570,000 events to determine its efficacy in identifying suspicious urban telemetry. The overall accuracy of 84.2% is complemented by a precision of 48.2% and a recall of 73.3% as shown in Table 6. While precision is moderate, the high recall is critical for smart city security operations where false negatives (missed attacks) pose greater operational risk than false positives, as undetected threats may directly compromise traffic management, emergency response, or utility monitoring systems.

The primary evaluation uses a randomized train–test split, which may partially overestimate generalization performance in temporally autocorrelated security logs. In operational smart city environments, attack behaviors often unfold over extended time windows, and strictly time-forward evaluation can provide a more conservative estimate of real-world performance. While a full forward-chaining evaluation is outside the scope of this study,

the dedicated analysis of a contiguous, real-world cryptomining campaign (Section 5.2) provides complementary evidence that the proposed framework effectively detects sustained attack behavior over time rather than isolated events. Future work will incorporate time-aware evaluation protocols to further strengthen deployment realism.

Table 6. Overall Detection Performance.

Metric	Training	Testing	Difference
Accuracy	84.5%	84.2%	0.003
Precision	48.5%	48.2%	0.003
Recall	73.1%	73.3%	0.002
F1-Score	58.3%	58.1%	0.002
AUC-ROC	0.887	0.884	0.003

The negligible training-test gap (<0.003) confirms that the model generalizes well to unseen data, a critical requirement for securing dynamic smart city environments. Analysis of feature importance reveals that the `event_risk_score` (0.143) and `error_rate` (0.121) are the strongest individual predictors. Collectively, temporal features contribute approximately 23% of the model's decision-making weight, validating the hypothesis that infrastructure attacks significantly deviate from established municipal operational schedules.

Confusion Matrix (570 K test events):

The model demonstrates strong threat coverage with 73.3% recall, meaning it successfully identifies approximately three-quarters of suspicious events. While precision is moderate at 48.2%, this trade-off is appropriate for smart city security where false negatives (missed attacks) pose greater operational risk than false positives, which can be efficiently triaged using the MCP interpretation layer.

Feature Importance:

Top predictors are `event_risk_score` (0.143), `error_rate` (0.121), `is_night` (0.098), Failed Authentication Count (0.087), `privilege_level` (0.076). Temporal features collectively contribute ~23%, validating that attacks deviate from operational schedules. Geographic features contribute ~10%, confirming remote access patterns.

5.2. Real Attack Validation

We analyzed the model's performance on a massive, verified resource-hijacking attack. Demonstrated in Table 7, the model achieved a 96.8% detection rate on this specific campaign, significantly higher than the overall recall.

Table 7. Cryptomining Campaign Performance.

Metric	Value
Total Events	1,300,000
Detected (TP)	1,258,400
Missed (FN)	41,600
Detection Rate	96.8%
Avg Confidence	87.3%
First Detection	4.2 s
90% Detection Threshold	18 min

The system flagged the first suspicious activity just 4.2 s after the initial credential compromise, reaching the “campaign recognition” threshold (90% detection) within 18 min as in Table 7. This performance demonstrates that while individual events may occasionally be missed, the temporal correlation engine effectively reconstructs the broader attack sequence, enabling municipal responders to intervene before the attack can significantly drain the computational resources required

Attack Characteristics Matching Smart City Threats:

- Credential compromise (not vulnerability exploitation)
- Cloud infrastructure targeting (not edge devices)
- Off-hours operation (weekends, 2–5 a.m.)
- Geographic anomalies (international IPs, no prior history)
- Resource hijacking (stealing computational capacity)
- For city services.

5.3. Interpretability Assessment: Contextual Narrative Generation

To evaluate the framework’s ability to bridge the interpretability gap in smart city security operations, we analyzed the narratives generated by the MCP layer across multiple attack archetypes. Unlike conventional explainable AI techniques that expose abstract feature weights, the proposed approach produces role-adapted, plain-language intelligence that connects cloud-layer anomalies to operational urban impact.

Systematic Narrative Evaluation

To assess interpretability quality beyond illustrative examples, we analyzed MCP-generated narratives across multiple dimensions:

Attack Archetype Coverage: The framework was evaluated on five distinct attack patterns identified in the dataset:

1. Resource Hijacking (Listing 1): Cryptomining campaign with burst compute provisioning ($n = 1,287,645$ events)
2. Credential Abuse (Listing 2): Authentication failure bursts indicating password-spraying attacks ($n = 18,734$ events)
3. Privilege Escalation (Listing 3): Unauthorized IAM modifications by non-administrative accounts ($n = 2156$ events)
4. Data Exfiltration Patterns: Rapid S3 GetObject sequences from unfamiliar regions ($n = 4892$ events)
5. Reconnaissance Activity: Systematic DescribeInstances enumeration suggesting pre-attack mapping ($n = 9341$ events)

For each archetype, MCP-generated narratives consistently included: (i) temporal context, (ii) identity/geographic anomalies, (iii) behavioral deviations, and (iv) smart city impact assessment. Narrative length averaged 147 words (range: 98–215), with 2.3 s median generation time.

Fidelity and Evidence Grounding: To ensure narratives remain grounded in observed data rather than plausible-sounding synthesis, the MCP implementation enforces:

- Structured Context Document: The LLM receives only standardized JSON fields derived from CloudTrail events and model outputs, no unstructured external text
- Evidence Citation Format: Narratives must reference specific event attributes (e.g., “Source IP: 185.220.101.34”, “Event: RunInstances”, “Time: 2:17 a.m.”)
- Separation of Observed vs. Inferred: “Why Flagged” sections report observed anomalies, while “Smart City Impact” sections clearly mark conditional inferences

- Deterministic Formatting: System prompt requires fixed section structure, reducing hallucination risk from unconstrained generation

Listing 1. MCP-generated threat narrative for a resource hijacking (cryptomining) campaign.

THREAT SUMMARY:

Detected unauthorized provisioning of 10 compute instances from an external IP address at 2:17 a.m. on a Saturday, consistent with resource hijacking behavior, based on the following flag values:

- Temporal Anomaly: Activity occurred during night hours (2:17 a.m.), when legitimate municipal operations are minimal.
- Anonymization: Source IP (185.220.101.34) is associated with a Tor exit node.
- User Risk: Account *backup_admin* has no prior history of compute provisioning.
- Behavioral Burst: Rapid creation of multiple instances deviates from standard municipal procedures.

SMART CITY IMPACT:

Unauthorized resource consumption competes with compute capacity allocated for critical urban services, including traffic management analytics and emergency response coordination. Sustained activity could degrade real-time decision-making capabilities during peak demand.

RECOMMENDED ACTIONS:

- Immediate: Terminate unauthorized instances to restore service capacity.
- Immediate: Disable compromised credentials and enforce password rotation.
- High Priority: Review CloudTrail logs for the past seven days to assess lateral movement.
- Medium Priority: Enforce multi-factor authentication for administrative roles.

Listing 2. MCP-generated narrative for credential abuse (authentication failure burst).

THREAT SUMMARY:

Detected repeated authentication failures against a municipal cloud account over a short time window based on the following flags.

- Authentication Anomaly: 27 failed login attempts within 12 min.
- Geographic Deviation: Source IP originates from a region with no historical access record.
- Temporal Context: Activity occurred outside normal municipal working hours.

SMART CITY IMPACT:

Successful credential compromise could enable unauthorized access to administrative dashboards supporting utility monitoring and public safety coordination.

RECOMMENDED ACTIONS:

- Disable the targeted account and rotate credentials.
- Review authentication logs for evidence of credential reuse or lateral movement.

Listing 3. MCP-generated narrative for privilege escalation behavior.

THREAT SUMMARY:

Detected anomalous privilege escalation attempts by a non-administrative user account.

- Identity Deviation: A standard user attempted administrative IAM policy modifications.
- Behavioral Pattern: Access sequence deviates from the historical behavior of this role.

Error Mode Analysis: Manual review of 200 randomly sampled narratives (40 per attack archetype) identified three primary failure modes:

- Impact Overestimation (12% of samples): Narratives occasionally overstated consequences when downstream services were unaffected (e.g., claiming “degraded traffic management” when hijacked resources represented <5% of total municipal compute capacity)
- Pattern Mimicry Underdetection (8% of samples): Sophisticated attacks deliberately mimicking normal schedules (e.g., credential abuse during business hours from known regions) generated lower-confidence narratives despite being malicious
- Feature Misattribution (5% of samples): In rare cases where multiple anomaly types co-occurred, narratives occasionally emphasized secondary indicators (e.g., highlighting geographic deviation when authentication failure was the stronger signal)

Mitigation strategies include: (i) retaining links to underlying event data for analyst validation, (ii) explicitly marking impact assessments as “potential” rather than definitive, and (iii) ongoing refinement of system prompts based on SOC feedback.

Multi-Stakeholder Communication Validation: The role-specific prompting capability was evaluated by generating parallel narratives for the same high-confidence threat:

- Technical Analyst View: Included forensic details (API call sequences, IAM policy changes, resource identifiers)
- Administrator View: Emphasized operational impacts (affected services, estimated resource costs, urgency levels)
- Public Safety View: Focused on citizen-facing implications (service availability, data privacy concerns, public communication needs)

While this demonstrates technical capability for audience adaptation, a formal user study with municipal SOC personnel is required to validate comprehension, trust calibration, and actionability (planned for future work, see Section 7.2).

SMART CITY IMPACT:

Unauthorized privilege escalation could expose sensitive datasets, including smart meter readings and infrastructure configuration data, increasing the risk of service disruption or data misuse.

RECOMMENDED ACTIONS:

- Revoke elevated permissions immediately.
- Audit recent configuration changes and validate role assignments.

Explanation Limitations and Error Modes:

While MCP-generated narratives substantially improve interpretability, they remain subject to limitations. Impact descriptions are scenario-based and may overestimate consequences when downstream services are unaffected, or underestimate risk when attackers closely mimic benign operational schedules. To mitigate these risks, explanations explicitly distinguish observed evidence from inferred impact and retain links to the underlying event data, enabling analysts to validate conclusions during triage.

5.4. Baseline Considerations and Design Rationale

The objective of this work is not to propose a novel detection algorithm, but to demonstrate how interpretable security intelligence can be operationalized for smart city environments. Accordingly, the Random Forest classifier was selected as a pragmatic baseline detector due to its favorable balance between detection performance, inference latency, and structural transparency. As shown in Table 8, we compared Random Forest against two baseline models on the same experimental split. Gradient Boosting achieves

the highest overall accuracy (92.6%) and superior calibration (ECE: 0.022); however, it exhibits substantially lower recall (60.6%), missing nearly 40% of suspicious events. Logistic Regression demonstrates competitive recall (77.4%) and improved calibration (ECE: 0.211) but lacks the capacity to model non-linear, multi-stage attack behaviors common in cloud-mediated urban infrastructures. The Random Forest model offers a balanced trade-off, achieving 84.2% accuracy with 73.3% recall, catching 13 percentage points more attacks than Gradient Boosting, while maintaining low inference latency (<13 ms) and inherent interpretability. In the context of smart city security operations, false negatives pose a greater operational risk than moderate increases in false positives, as missed attacks may directly compromise traffic management, emergency response, or utility monitoring. These characteristics make Random Forest more suitable for integration with the Model Context Protocol, where rapid, human-auditable detection with comprehensive threat coverage is prioritized over marginal gains in predictive accuracy at the cost of substantially reduced attack detection.

Table 8. Baseline detector comparison on the same experimental split.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	ECE
Random-Forest (proposed)	0.842	0.482	0.733	0.581	0.884	0.262
Logistic Regression	0.797	0.407	0.774	0.533	0.880	0.211
Gradient Boosting	0.926	0.860	0.606	0.711	0.923	0.022

ECE stands for Expected Calibration Error.

To assess the necessity of the MCP-based interpretation layer, we contrast it conceptually with two common alternatives: (i) feature-attribution explanations (e.g., SHAP or LIME) and (ii) static, template-based alert narratives. Feature-attribution methods explain model mechanics but do not translate detections into operational consequences or recommended actions, limiting their utility for non-technical stakeholders. Template-based approaches provide faster outputs but rely on predefined logic that degrades under novel or evolving attack patterns. In contrast, the MCP framework structures evidence before interpretation, enabling the generation of grounded, context-aware narratives that adapt to unseen combinations of indicators while remaining auditable. While calibration was not explicitly optimized in this study, future work will explore post hoc calibration techniques to further align confidence scores with operational alerting thresholds.

Ablation Studies—Feature Group Contributions:

To assess the necessity of each feature category, we performed ablation studies by systematically removing feature groups:

- Full Model (all 35 features): 84.2% accuracy
- Without Temporal Features (31 features): 74.8% accuracy (−9.3 pp)
- Without Identity Features (26 features): 80.6% accuracy (−3.5 pp)
- Without Geographic Features (19 features): 81.7% accuracy (−2.5 pp)
- Without Behavioral Features (29 features): 82.9% accuracy (−1.3 pp)

These results confirm that temporal patterns provide the strongest individual contribution (9.3 pp drop when removed), validating our hypothesis that infrastructure attacks deviate significantly from established municipal operational schedules. The substantial impact of removing temporal features demonstrates that attack timing, particularly off-hours and weekend activity, is the most discriminative signal in our framework. Identity features contribute 3.5 pp, reflecting the importance of user behavior profiling and privilege monitoring. Geographic features (2.5 pp) and behavioral features (1.3 pp) provide complementary signals. Importantly, all feature groups contribute meaningfully to detection

performance, indicating that the multi-dimensional approach captures complementary threat signals rather than redundant information.

6. Discussion

This section interprets the experimental findings within the broader context of smart city security operations and deployment readiness. We first analyze the implications of our results for municipal security architectures, particularly the significance of temporal feature dominance and the recall-precision trade-off in critical infrastructure protection (Section 6.1). We then examine socio-technical applications beyond threat detection, including policy formulation, incident response coordination, and post-incident auditing (Section 6.2). Subsequently, we address implementation challenges and system limitations, encompassing dataset scope, adversarial robustness considerations, computational costs, and a comprehensive analysis of LLM safety, fidelity, and data governance mechanisms (Section 6.3). Finally, we clarify the framework's deployment scope and applicability boundaries, distinguishing cloud control-plane security from edge and operational technology contexts while acknowledging threat model constraints and temporal evaluation considerations (Section 6.4).

Our ablation study reveals an important finding: temporal features contribute 9.3 percentage points to detection performance, nearly three times more than any other feature group. This validates the core premise of our smart city threat model: that malicious activity exhibits distinct temporal signatures that deviate from predictable municipal operational patterns. Attacks predominantly occur during off-hours (nights, weekends) when monitoring is minimal and legitimate traffic is low, making temporal anomalies the strongest individual indicator of compromise. The baseline comparison demonstrates a critical trade-off between accuracy and recall. While Gradient Boosting achieves 92.6% accuracy, its 60.6% recall means it misses 40% of attacks, an unacceptable false negative rate for critical infrastructure protection. In contrast, Random Forest's 73.3% recall provides substantially better threat coverage, catching an additional 13% of attacks (representing hundreds of potential incidents in a large-scale deployment). For smart city security, where a single missed attack could compromise traffic management, utility systems, or emergency response, this recall advantage outweighs Gradient Boosting's superior precision.

6.1. Implications for Smart City Security

The results demonstrate that the interpretability gap is not an inherent property of AI, but a bridgeable architectural challenge. By integrating MCP, we move from "reactive logging" to "proactive intelligence."

- **Rethinking Security Intelligence:** Traditional SIEM platforms typically provide high-volume telemetry that overwhelms human analysts. Our MCP framework suggests that next-generation platforms must integrate Pattern Recognition (ML) with Contextual Interpretation (LLMs). This creates a symbiotic system where the ML identifies the "what" (anomaly detection) and the LLM explains the "so what" (impact assessment).
- **AI Governance and the EU AI Act:** As smart cities deploy autonomous infrastructure, they face increasing regulatory scrutiny. The EU AI Act mandates transparency for "high-risk" AI systems. Our framework provides a technical blueprint for compliance by ensuring that every automated alert includes a human-auditable trail of reasoning. This transparency is vital for maintaining public trust, especially when digital alerts trigger physical interventions in city services.
- **Democratizing Urban Security:** Smaller municipalities often lack the budget for Tier-3 security analysts. By translating "black-box" outputs into plain-language narratives,

this framework allows generalist IT staff and city administrators to perform sophisticated threat triage, effectively lowering the barrier to entry for advanced urban defense.

6.2. Socio-Technical Applications

Beyond technical detection, the MCP framework serves three critical urban functions:

1. **Policy and Budgeting:** Narratives allow city councils to understand the frequency and severity of threats in non-technical terms, facilitating data-driven decisions on infrastructure investment.
2. **Incident Response Coordination:** During an active crisis, the audience-adapted prompts ensure that police, fire, and utility departments receive information tailored to their specific operational needs.
3. **Post-Incident Auditing:** The generated narratives provide a permanent, readable record of why the AI flagged an event, which is essential for legal and insurance forensics following a breach.

6.3. Limitations and Challenges

While the framework performs exceptionally well on the test dataset, several limitations must be addressed for real-world deployment:

- **Dataset Homogeneity:** Validation was performed using AWS CloudTrail. While representative of the “urban cloud,” real smart cities also utilize OT (Operational Technology) networks and proprietary IoT protocols. Future work must extend the middleware parser to handle these specialized edge environments.
- **Adversarial Robustness:** This study did not explicitly test against “adversarial ML” attacks designed to fool the Random Forest classifier. While ensembles offer some inherent protection, nation-state actors may employ techniques to “evade” detection by mimicking benign urban rhythms.
- **Computational and Economic Costs:** The use of GPT-4 introduces a dependency on external APIs and a monthly operational cost (est. \$3000–\$5000 for a mid-sized city). Resource-constrained municipalities may require a hybrid approach using local LLMs (e.g., Llama 3 or Mistral) to maintain privacy and reduce long-term costs.
- **LLM Safety, Fidelity, and Data Governance:**

Hallucination Mitigation:

The interpretation layer employs multiple strategies to reduce hallucination risk and prevent overconfident or misleading recommendations:

1. **Constrained Generation:** The LLM receives only a structured MCP context document containing observed event fields (eventName, userName, sourceIPAddress, event-Time) and model outputs (feature importance scores, temporal correlation metadata). No unstructured external text is incorporated, limiting the prompt attack surface.
2. **Evidence-First Formatting:** The system prompt explicitly requires separation of:
 - **Observed Evidence:** Direct event attributes without interpretation
 - **Inferred Risk:** Probabilistic assessments clearly marked as conditional
 - (e.g., “could indicate”, “may suggest”, “potential impact”)
 - **Recommended Actions:** Advisory guidance subject to human approval never presented as mandatory directives
3. **Narrative Auditing:** All MCP-generated narratives retain provenance links to the underlying CloudTrail events and detection metadata. SOC analysts can validate claims by inspecting the raw evidence trail, ensuring narratives serve as decision support rather than authoritative ground truth.

4. **Temperature Control:** The LLM operates at temperature = 0.3 to minimize creative elaboration while maintaining factual consistency. Higher temperatures (0.7+) were evaluated but produced less reliable attribution.

Data Privacy and Governance:

For municipal deployments handling sensitive citizen data, the framework supports:

1. **Pre-Submission Redaction:** Personally identifiable information (PII) can be tokenized or hashed before CloudTrail events enter the detection pipeline. For example, user-Names containing email addresses can be replaced with pseudonymized identifiers (user_A3F2), preserving behavioral correlation without exposing identity.
2. **Minimal Field Transmission:** Only the subset of CloudTrail fields required for threat explanation (event metadata, temporal context, geographic attributes) are transmitted to the LLM endpoint. Sensitive fields such as requestParameters (which may contain service-specific data) can be excluded from the MCP context document.
3. **On-Premises LLM Option:** Resource-constrained or privacy-sensitive municipalities can deploy local LLM instances (e.g., Llama 3.1, Mistral 7B) instead of cloud-based GPT-4 API calls, ensuring all data processing remains within municipal infrastructure. While this increases deployment complexity, it eliminates external data transmission entirely.
4. **Regulatory Compliance:** The framework aligns with GDPR Article 22 (right to explanation) and EU AI Act transparency requirements by providing human-auditable reasoning chains. All automated decisions are marked as “advisory” rather than autonomous enforcement actions.

Operational Failure Modes:

Beyond hallucination, the system addresses several operational risks:

1. **Overconfident Recommendations:** The narrative format explicitly distinguishes “Immediate” actions (e.g., terminate unauthorized instances) from “High Priority” (investigate logs) and “Medium Priority” (policy review) recommendations. Analysts retain full discretion over response execution.
2. **Prompt Injection Resistance:** Because the system does not incorporate untrusted external text into prompts (only structured CloudTrail fields), traditional prompt injection attacks (“ignore previous instructions and...”) have limited applicability. However, if future versions integrate public threat intelligence feeds, additional input sanitization would be required.
3. **Misleading Causal Claims:** The “Why Flagged” narrative sections are constrained to report observed correlations (temporal anomalies, geographic deviations) rather than definitive causal attribution. Analysts are reminded that correlation does not imply causation, and forensic investigation remains the authoritative method for establishing attacker intent. These safeguards ensure the MCP layer functions as an interpretability aid rather than an autonomous decision-making system, preserving human oversight for safety-critical urban infrastructure.

6.4. Smart City Deployment Scope and Applicability

This framework is designed to address the interpretability gap in smart city security at the cloud control and decision layer, where authentication, resource orchestration, and service coordination directly affect urban operations. The evaluation presented in this study focuses on AWS CloudTrail logs, which are representative of cloud-based smart city deployments that rely on managed IoT ingestion, serverless processing, and centralized analytics.

While the proposed architecture is not evaluated on low-level Operational Technology (OT) protocols (e.g., SCADA, PLCs, or building automation buses), it is architecturally compatible with such environments through the middleware normalization layer described

in Section 3.3. Municipal deployments seeking full-stack coverage would need to integrate additional telemetry sources and validate detection performance within their specific infrastructure configurations.

Accordingly, this framework is best characterized as production-ready for cloud-centric smart city deployments, with clearly defined adaptation requirements for edge, OT, and proprietary IoT systems. This scoped applicability supports realistic deployment, regulatory compliance, and operational trust without overstating system coverage.

Threat Model and Adversarial Robustness:

This study did not explicitly evaluate adversarial robustness, the ability to resist attacks specifically designed to evade ML detection. Potential evasion

Strategies include:

- **Mimicry Attacks:** Sophisticated threat actors operating during business hours, from known regions, using legitimate API call patterns to avoid temporal and geographic anomaly detection
- **Feature Manipulation:** Attackers deliberately space operations to stay below burst thresholds or diversify API calls to reduce behavioral distinctiveness
- **Model Inversion:** In extreme cases, adversaries with knowledge of the detection system might craft attacks that exploit model blind spots

While Random Forest ensembles offer some inherent protection against adversarial perturbation (compared to deep neural networks), nation-state actors targeting critical infrastructure may employ sophisticated evasion techniques. Future work should evaluate detection performance against adversarial datasets and explore defense mechanisms such as adversarial training or anomaly ensemble voting.

Temporal Evaluation and Autocorrelation:

The primary evaluation uses a randomized 70–30 train-test split, which may partially overestimate generalization performance in temporally autocorrelated security logs. Attack behaviors often unfold over extended time windows, and strictly time-forward evaluation (where test data occurs chronologically after training data) can provide a more conservative estimate of real-world performance. While a full forward-chaining evaluation is outside the scope of this study, the dedicated analysis of the contiguous cryptomining campaign (Section 5.2) provides complementary evidence that the framework effectively detects sustained attack behavior over time. Future evaluations should incorporate time-based splitting strategies to further strengthen deployment realism.

Interpretability Assessment Limitations:

The current interpretability evaluation (Section 5.3) provides a structured analysis of narrative quality across multiple attack archetypes and identifies error modes through manual review. However, it does not include formal user studies with municipal SOC personnel measuring:

- **Task Completion Time:** How quickly analysts can triage threats using MCP narratives vs. raw logs
- **Comprehension Accuracy:** Whether stakeholders correctly interpret threat severity and recommended actions
- **Trust Calibration:** Whether confidence scores align with analyst trust in automated alerts
- **Actionability:** Whether narratives provide sufficient guidance for incident response

A comprehensive human factors evaluation is planned for future work (Section 7.2) and would strengthen empirical support for the “multi-stakeholder communication” Claims.

7. Conclusions and Future Work

7.1. Summary

This research successfully adapted the Model Context Protocol for smart city IoT security, addressing the critical interpretability gap in urban infrastructure defense. By combining a high-performance Random Forest classifier with an asynchronous LLM interpretation layer, we achieved an overall 84.2% detection accuracy and a remarkable 96.8% detection rate on real-world resource-hijacking campaigns. The framework proves that interpretability and high-speed detection are not mutually exclusive objectives. With a detection latency of less than 13 milliseconds and narrative generation occurring in approximately 2.3 s, the system meets the sub-second requirements of safety-critical urban services such as traffic management and utility monitoring. Validation against 1.9 million authentic telemetry events confirms that this architecture is operationally viable for modern municipal Security Operations Centers.

The framework delivers a production-ready architecture that integrates directly with existing SIEM tools like Elasticsearch, providing empirical proof of efficacy using massive datasets and verified attack ground truth. The multi-stakeholder communication layer provides role-specific, actionable guidance for both technical and administrative personnel, while advanced temporal feature engineering successfully reconstructs multi-stage attack patterns that often bypass traditional rule-based systems. The ablation studies revealed that temporal features contribute 9.3 percentage points to detection performance, validating our hypothesis that infrastructure attacks deviate significantly from established municipal operational schedules. This temporal dominance, combined with the framework's ability to generate context-aware narratives adapted to diverse stakeholder roles, demonstrates that interpretability can be operationalized as a first-class requirement rather than a post hoc addition.

7.2. Future Directions

While the current framework establishes a baseline for interpretable urban security, future research will focus on three primary dimensions of scaling. Multimodal fusion represents the first major direction, where future iterations will move beyond API logs to integrate surveillance footage, physical sensor telemetry, and social media signals. This will enable the detection of hybrid threats, such as correlating a physical breach at a water treatment plant with anomalous network traffic, providing comprehensive situational awareness across both cyber and physical domains.

Edge intelligence and decentralization constitute the second research direction. To improve resilience during cloud outages and further reduce latency, we aim to deploy lightweight detection models directly on IoT gateways. This edge-based MCP approach would perform sub-millisecond filtering, only sending high-risk events to the cloud for full LLM interpretation, thereby maintaining operational capability even during network disruptions while reducing bandwidth requirements and improving response times for time-critical municipal services.

Federated urban defense represents the third major direction. We propose a federated learning model where municipal consortiums can share threat intelligence without exposing sensitive citizen data. In this architecture, an attack detected in one city could proactively trigger protective narrative alerts across a network of participating municipalities, creating a distributed early warning system while preserving data sovereignty and privacy requirements. This collaborative approach would enable smaller cities to benefit from the threat intelligence gathered by larger metropolitan security operations centers.

Research into autonomous response integration will explore how high-confidence threats exceeding 95% certainty can trigger immediate containment measures, such as

credential isolation or resource termination, reducing the window of vulnerability from minutes to milliseconds. However, this automation must maintain human oversight for safety-critical decisions affecting public services. Additional methodological improvements include conducting structured interpretability assessments with municipal SOC personnel to measure threat comprehension accuracy, triage decision speed, narrative usefulness ratings, and trust calibration across different stakeholder roles. Systematic evaluation against adversarial attack datasets will assess robustness when threat actors deliberately evade detection through temporal mimicry, geographic camouflage, and behavioral blending. Implementing forward-chaining evaluation protocols, where training data precedes test data chronologically, will provide conservative generalization estimates for temporally autocorrelated security logs.

Extended validation across additional cloud platforms such as Azure Monitor and GCP Cloud Audit Logs will assess cross-platform generalizability, while evaluation on IoT edge telemetry, including MQTT broker logs, Zigbee network captures, and building automation protocols, will validate middleware normalization for heterogeneous urban data sources. Applying post hoc calibration techniques such as Platt scaling and isotonic regression will align confidence scores with empirical accuracy rates, improving alert priority ranking for SOC triage workflows. Finally, detailed operational cost modeling across different municipality sizes will provide deployment decision trees based on budget constraints, privacy requirements, and acceptable latency thresholds, enabling evidence-based adoption decisions for cities ranging from small towns to large metropolitan areas.

Author Contributions: Conceptualization and methodology were developed jointly by A.A. and S.Z.; software, data curation, formal analysis, investigation, and original draft preparation were carried out by A.A.; validation and visualization were performed by A.A. with guidance from S.Z.; review and editing were led by S.Z., whose supervision and invaluable guidance were instrumental throughout the project; project administration was also managed by S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in AWS CloudTrail via Summit flaws.cloud Route project at https://summitroute.com/blog/2020/10/09/public_dataset_of_cloudtrail_logs_from_flaws_cloud/ (accessed on 23 March 2026).

Acknowledgments: The authors acknowledge the use of the flaws.cloud CloudTrail dataset provided by Scott Piper, and the computational resources provided by the University of Portsmouth Research Computing Facility.

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

References

1. United Nations Department of Economic and Social Affairs. *68% of the World Population Projected to Live in Urban Areas by 2050*; United Nations: New York, NY, USA, 2018. Available online: <https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html> (accessed on 1 December 2024).
2. Laufs, J.; Borrion, H.; Bradford, B. Security and the smart city: A systematic review. *Sustain. Cities Soc.* **2020**, *55*, 102023. [CrossRef]
3. Ahmetoglu, S.; Che Cob, Z.; Ali, N.A. A systematic review of internet of things adoption in organisations: Taxonomy, benefits, challenges and critical factors. *Appl. Sci.* **2022**, *12*, 4117. [CrossRef]
4. Brous, P.; Janssen, M.; Herder, P. The dual effects of the internet of things (IoT): A systematic review of the benefits and risks of IoT adoption by organisations. *Int. J. Inf. Manag.* **2020**, *51*, 101952. [CrossRef]
5. Dawood, M.; Tu, S.; Xiao, C.; Alasmay, H.; Waqas, M.; Rehman, S.U. Cyberattacks and security of cloud computing: A complete guideline. *Symmetry* **2023**, *15*, 1981. [CrossRef]

6. Ali, T.; Al-Khalidi, M.; Al-Zaidi, R. Information security risk assessment methods in cloud computing: Comprehensive review. *J. Comput. Inf. Syst.* **2024**, *66*, 123–150. [\[CrossRef\]](#)
7. Madanian, S.; Chinbat, T.; Subasinghage, M.; Airehrour, D.; Hassandoust, F.; Yongchareon, S. Health IoT threats: Survey of risks and vulnerabilities. *Future Internet* **2024**, *16*, 389. [\[CrossRef\]](#)
8. Laghari, A.A.; Li, H.; Khan, A.A.; Yin, S.; Karim, S.; Kaim Khani, M.A. Internet of things (IoT) applications security trends and challenges. *Discov. Internet Things* **2024**, *4*, 36. [\[CrossRef\]](#)
9. Al Rawajbeh, M.; Maria Soosai, A.J.; Ramasamy, L.K.; Khan, F. Trustworthy adaptive AI for real-time intrusion detection in industrial IoT security. *IoT* **2025**, *6*, 53. [\[CrossRef\]](#)
10. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*; Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 4765–4774.
11. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144. [\[CrossRef\]](#)
12. Hou, X.; Zhao, Y.; Wang, S.; Wang, H. Model context protocol (MCP): Landscape, security threats, and future research directions. *arXiv* **2025**, arXiv:2503.23278. [\[CrossRef\]](#)
13. Al-Dhamari, N.; Clarke, N. Enhancing cybersecurity awareness and training using generative pre-trained transformers (GPT). *arXiv* **2024**, arXiv:2405.04138. [\[CrossRef\]](#)
14. National Institute of Standards and Technology. *Framework for Cyber-Physical Systems, Volume 1: Overview*; NIST Special Publication 1500-201; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2017.
15. European Union Agency for Cybersecurity. *Good Practices for Security of Internet of Things in the Context of Smart Manufacturing*; ENISA: Heraklion, Greece, 2018.
16. Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.; Wirth, R. *CRISP-DM 1.0: Step-by-Step Data Mining Guide*; SPSS Inc.: Chicago, IL, USA, 2000.
17. Piper, S. Public Dataset of CloudTrail Logs from Flaws.Cloud. Available online: https://summitroute.com/blog/2020/10/09/public_dataset_of_cloudtrail_logs_from_flaws_cloud/ (accessed on 1 December 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.