Journal of Marketing & Social Research

ISSN (Online): 3008-0711

Volume: 02 | Issue 06 | 2025
Journal homepage: https://jmsr-online.com/

Research Article

An Intelligent IoT Security System: Cloud-Native Architecture with Real-Time AI Threat Detection and Web Visualization

Ravi Shankar Garapati
Sr. Software Engineer, ORCID ID: 0009-0002-1945-5796

Received: 30/06/2025; Revision: 12/07/2025; Accepted: 18/07/2025; Published: 12/08/2025

Corresponding author: Ravi Shankar Garapati *(raviishankargarapti@gmail.com)

Abstract: The rapid evolution of the Internet of Things (IoT) interconnects billions of devices and systems servicing numerous applications. However, connectivity increases the risk of cyber-attacks and the overall impact of any security attack can be catastrophic. Keeping IoT systems, applications, and services secured and resilient is a growing challenge with security risks increasingly difficult to detect. Artificial intelligence, coupled with Cloud-based services and Web interfaces, can provide a powerful tool for building intelligent real time threat detection and defence systems. Key architectural challenges are identified and addressed. The use case of IoT WiFi network breach real time threat detection is presented. An AI Engine is developed and trained using supervised deep learning classifiers with time domain extracted features. The AI Engine architecture is Cloud-native in both deployment and operation. A Web-based UI that allows for the visualisation of live alerts completes the smart threat detection system. The resulting classification accuracy of 95.6% was shown to outperform existing similar solutions. The architecture can be adapted for a vast number of IoT threat detection use cases.

Keywords: Intelligent IoT security system; cloud-native architecture; real-time threat detection; AI; web visualization.

INTRODUCTION

Application programmers in both industry and academia are frequently tasked with developing a security architecture that uses multiple sensors for network monitoring. These sensors collect data from a network, perform analysis, and generate reports or alerts in case of suspicious activity. The IoT Security System project provides a data analysis architecture based on sensors collecting data such as network traffic and system logs for relation analysis and causing multi-level response. Based on these events, a security decision is made and response is automatically deployed with firewall and SDN controller rules.

With Cloud Native technologies on the rise in popularity together with the growing importance of security monitoring, this project presents an intelligent Cloud Native IoT security system with real-time AI analysis and Web Visualization. The system gathers data based on security detection rules using IoT honeypots, then detects threats using a NIDS powered by an AI model. The detections are shown on a Web Dashboard with an organized topology view. Finally, the solution is Cloud Native, which means it can run on Kubernetes and scale out with multi-replicas.

The lost ticket detection system is an example of a double safeguard. Users authenticate to use the infrastructure in the IoT environment. The allotted resources are further protected with a lost ticket detection system. Besides monitoring the environment, an intelligent IoT security system should also protect the IoT assets. The lost ticket scenario represents a leaked resource allocation in the environment. For example, if a train ticket is lost, any strangers will probably use it. A ticket is valuable information after the authentication phase. The success of gaining unauthorized resource utilization is unnatural behavior. Detecting unnatural behavior in resource control also requires an intelligent neural network for the intelligent IoT security system. In summary, the intelligent IoT security system protects both the infrastructure and the users.

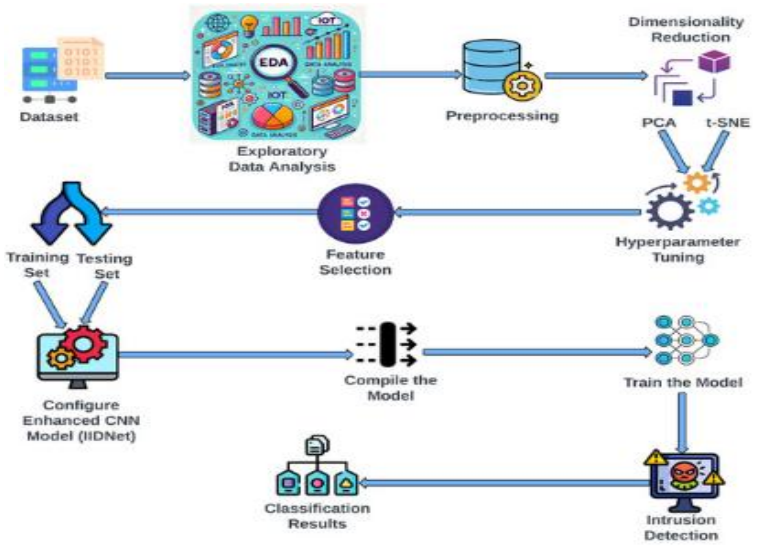

Fig 1: Real-Time AI Threat Detection and Web Visualization

Background and Significance

The intelligent Internet of Things (IoT) security system, which efficiently secures daily life, industry, agriculture, energy, traffic, etc., is designed in a cloud-native architecture with real-time artificial intelligence (AI) threat detection and web visualization. The modern IoT environment adopts artificial intelligence of things (AIoT) by deploying AI models, including intrusion detection systems, on IoT devices. However, designing an intelligent IoT security system usually suffers from scalability issues and design complexity. As the IoT infrastructure is the foundation of the AIoT environment, the core function of the intelligent IoT security system is protecting the IoT infrastructure, such as providing an intrusion detection system. Because the intelligent IoT security system can observe human behaviors, it can also protect users' execution environments (such as virtual machines and IoT devices). Therefore, the intelligent IoT security system is also designed as a safeguard for both ends.

Equ 1: Maintenance Decision Trigger

$$\delta(\hat{y}(t)) = \begin{cases} 1, & \text{if } \hat{y}(t) \leq \tau \\ 0, & \text{otherwise} \end{cases}$$

Where:

- τ: maintenance threshold (e.g., RUL < 10 hours)
- $\delta = 1$: maintenance action recommended

Background and Motivation

Recent advances in technology have a significant impact on the four main components of the Internet of Things (IoT) paradigm: Things (heterogeneity, miniaturization), networks (5G, ZigBee), Intelligent Systems (DL, MTs, Fuzzy logic), and cloud computing (IaaS, PaaS, SaaS). As the IoT ecosystem expands, security remains its Achilles' heel, necessitating solutions that can identify and thwart evolving cyberattacks.

A research paper introduces the design and implementation of a cloud-native intelligent IoT security system. Leveraging Kubernetes for deployment and resource management, this architecture supports a hybrid AI-based Security Information and Event Management (SIEM) powered by a custom Security Orchestration, Automation, and Response (SOAR) system, which serves as a real-time AI threat detector. Specialized AI models, including an ensemble of BERTweet for textual analysis and CNN models such as MobileNet, DenseNet, and EfficientNet for image analysis, classify detected threats during reconnaissance and analysis phases. The results feed into a cloud-native web application developed to visualize, build, and publish responsive, customizable spiders or crawlers for web content extraction.

Research design

The research follows an applied, qualitative, and explanatory design, guided by bibliographic development within the frameworks of cloud-native architecture and real-time threat detection through intelligent IoT security. Through this analysis of specialized databases and the synergies of the studied topics, an intelligent IoT Security System is designed on a cloud-native architecture, capable of detecting external threats to an infrastructure. The system measures external attack probabilities against the infrastructure and offers an accessible control for users through web visualization.

Today's IoT devices are characterized by low computational capacity and memory, which inhibits the installation of intelligent antivirus programs on the devices. The vast majority of IoT devices connect to cloud servers, offering excellent scalability. By storing intelligent detection models in the cloud, a solution arises for protecting the devices. Attack detection in real time must meet the requirements of Best Practice policies and amperometric models, providing faster and more versatile responses to

threats affecting a company's security.

IoT Security Challenges

As the Internet of Things (IoT) continues to expand, providing increased convenience for humans and machines, it has created a threat landscape for users, vendors, companies, and host governments—a landscape that is growing more dangerous with the proliferation of connected devices. This problem is especially apparent for devices relying on the cloud for operation, remote management, or software updates. Additionally, devices designed and programmed with one country or region in mind deploy in politically opposed locations; locations where the original manufacturer may be located in a country or region where cyberwarfare campaigns or tactics are common. This mix of cultural and functional challenges creates a scenario where supplies for IoT products can be disrupted, their updates may be unavailable, or their remote functions may be used maliciously. Naturally, users, companies, or agencies tend to disable the external, cloud-to-device communication for these devices in an attempt to mitigate potential future attacks, but this reduces the convenience originally provided.

The inherent vulnerability of individual devices worsens when considering the "switching network" nature of the Internet. The Internet facilitates communications between distinct devices on disparate physical networks; devices which would otherwise pose little threat to each other through their daily operations. These mediated communications are abused in a variety of ways; for instance, by forcing a device on Network A to poll an address on Network B in order to identify the secret key needed to communicate (or simply to exist). One of the most famous uses of this technique is the Low Orbit Ion Cannon (LOIC), which requires a substantial base of "bots" from many distinct networks to successfully create a Distributed Denial of Service attack. The principal security challenges highlighted here are: (i) switching networks allow attacks between devices, (ii) devices often lack physical security, thus can be exploited and used in these attacks, and (iii) devices often lack defensive capabilities, thus cannot mitigate or fend off these attacks.

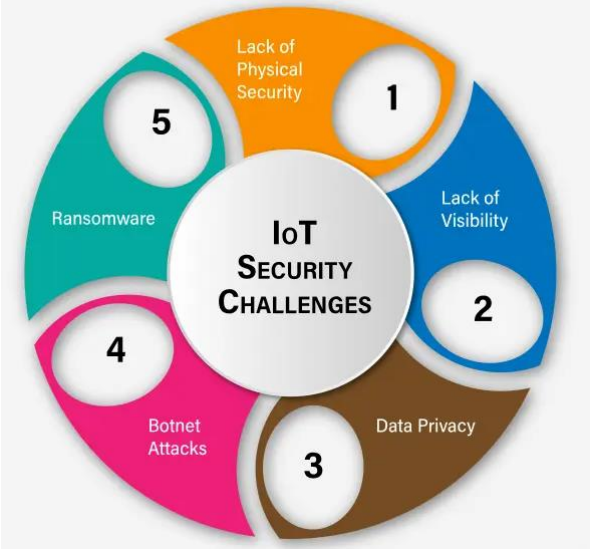

Fig 2: IoT Security Challenges

Threat Landscape Overview

Advances in information and communication technologies also result in a growing number of devices connected to the Internet, referred to as the Internet of Things (IoT). Society's rapid digital transformation and the incorporation of sensors have made smart homes and smart cities a reality. In this context, the adoption of smart devices has grown exponentially to improve the quality of life of users, enhancing communication and allowing faster data transfers, greater information transparency and, consequently, more efficient inferences by users. However, the high degree of dependence on IoT-connected devices is an enabler for the emergence of new types of attacks, with varying levels of intensity and frequency. This scenario brings even greater concern regarding the privacy of IoT users, since the information contained in the device can be accessed and manipulated by cybercriminals. Consequently, this also impacts the financial security of companies, considering the damage that DDoS attacks can cause to their services.

The threat landscape for IoT systems has been changing from artifacts that were mere network scanners to high-performance scanning machines based on multiple attack vectors. In a smart home context, various types of botnets have proliferated and exploited smart homes by recruiting IoT devices and launching a wide range of cyberattacks, from DDoS attacks to crypto-mining attacks. Among these botnets, Mirai is one of the most dangerous, exploiting IoT devices through brute-force attacks using a hardcoded list of removed default credentials. Persirai is another significant actor, scanning devices with the goal of creating a botnet army to perform DDoS attacks by exploiting weak Telnet credentials. Sneakyv2 is a very prolific variant of the Gafgyt family, having a large volumetry of files involved, in addition to the use of TCP port 8080 to perform HTTP flood attacks; this change allows the performance of a proxy network, enhancing the launching of SYN flooding attacks. Bashlite is a family performing DDoS attacks using asymmetric reflection and amplifier techniques to expand its attack power. Lastly,

Torii is a malware developed in the Go language, with cross-platform capability and the ability to perform different attacks, from launching DoS to obtaining access credentials for services on the hacked machine.

Common Vulnerabilities in IoT Devices

Despite their name, IoT devices rarely operate in isolation. Indeed, IoT interactions can be important, necessitating what is essentially an IoT type of group in which the required things can communicate and operate with each other to achieve a common goal. As a consequence, adequate risk assessment is necessary; otherwise, a compromised IoT device could enable an attacker to take control of other related devices and the whole ecosystem. Moreover, IoT ecosystems today include three main "things" that constitute the main attack vector; more specifically—(i) User, who manages everything and acts as a controller, (ii) IoT device and associated infrastructure, which gather data from the environment and are controlled by the user, and (iii) IoT platform component, which integrates user, device, and IoT device manager. These "things" provide a clear bi-directional communication channel by which to exchange data, but also creates connections between the external and internal network, which can be utilized by attackers inside the network.

Additionally, ignoring basic security strategies also contributes to the insecurity within an IoT ecosystem. For instance, DSDV and SCP carried out a survey of 200 IoT devices for vulnerability analysis in 2017. Their results demonstrated that encryption was supported in only 57% of devices; 90% did not require a complex password, 60% revealed their functionality through descriptive device names, and 60% did not limit login attempts. After analyzing these results, one can conclude that these security implementations are essential in building a robust ecosystem that can withstand attacks.

Equ 2: Web-Based User Interface for Visualization and Control

$$\mathbf{U} = \mathcal{G}(\mathbf{X}(t), \hat{y}(t), \delta)$$

Where:

- $\mathcal{G}$: rendering function (e.g., dashboards, graphs)
- $\mathbf{U}$: user-facing output (visual alerts, diagnostics)

4.

Cloud-Native Architecture

Cloud-native architectures are recognized as the most modern and flexible way to develop and deploy applications. They are designed to address the challenges of hybrid cloud and multicloud, which are faced by many organizations today. In simple terms, cloud-native architectures allow applications to be configured for optimal scalability and efficiency in any cloud environment.

The Smart Detective solution incorporates such an architecture, implemented through Amazon Web Services (AWS). This is important, as it allows the application to leverage the different capabilities provided at low-level from the AWS platform. Implementing a security system in this way benefits the automation of features such as auto-scaling, monitoring, backups, and automatic software updates. This is implemented through a set of managed and serverless tools provided by the AWS platform. The serverless philosophy allows a developer to focus on the core project by leaving operational tasks to the cloud provider.

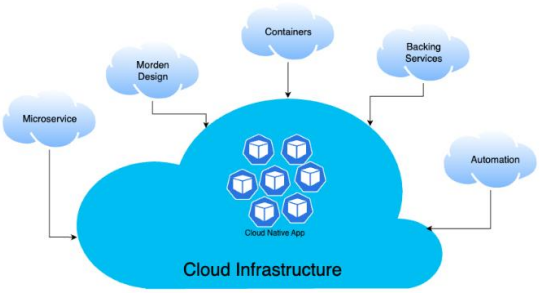

Fig 3: Embracing Cloud Native

Definition and Key Principles

The concept of Cloud-Native application security is built upon the foundations of cloud computing, particularly Platform as a Service (PaaS), Software as a Service (SaaS), application containers, microservices, dynamic orchestration, and declarative APIs. A mutual thread among these concepts is the extensive use of public APIs, which enable advanced policy control, immersive analytics, and declarative information exchange. These public APIs allow broad integrations but also open up the potential for new security attack vectors.

The API-enabled environment of Cloud-Native applications presents a flexible architecture in which every aspect of the system is secured, thereby reducing the probability of application insecurity and promoting easier maintenance in the future. Cloud computing addresses four major security challenges: regulation and compliance, infrastructure security, data security and

privacy, and incident response and recovery. Cloud providers have established dedicated teams to tackle these challenges effectively. Protecting APIs is crucial because they represent the primary interface for system interaction. The security of Cloud-Native applications can be enhanced by securing these public APIs.

Benefits of Cloud-Native Solutions

Cloud-native solutions offer the advantage of scalability: more computing resources can be allocated automatically if necessary [9]. Furthermore, these solutions have significant benefits in the area of overall system management. The automated deployment process speeds up updates and reduces downtimes. Monitoring is also automated by connecting the current and historical status of every service, including all log information.

Cloud-native applications support distributed locations and users in a very efficient way. Through global presence of providers, the computing resources toilet desired can be very close to the location where the services are used. Additionally, the elastic approach of resource allocation leads to an efficient usage of the services. Therefore, the transaction costs for the users are reduced. The cloud providers also benefit from this approach in the business context, because many different users share the same infrastructure, foreseen by a multi-tenancy architecture.

REAL-TIME AI THREAT DETECTION

A subsystem for real-time threat detection is also included in the architecture. Raw network packets captured by Suricata (a high-performance IDS, IPS, and network security monitoring engine) are forwarded to a WebSocket to JSON dumper/sender, which abstracts Suricata's expert-derived network threat detection capabilities and relays them to the WebSocket / JSON API Server.

The AI subsystem is a micro-service that receives packets and introspects them through a series of scikit-learn supervised-learning models and custom codes to detect anomalous behavior that could indicate a cybersecurity threat. Results generated by AI models, along with intrusion-detection system output from the Suricata packet sender, are forwarded on through the WebSocket to JSON Server to a WebSocket / JSON API Server ahead of web browser visualization.

Fig 4: Real-Time AI Threat Detection

Machine Learning Algorithms for Threat Detection The data is analyzed through machine learning algorithms for threat detection models within a cloud-native architecture. Two models are trained for detecting threats. A duration by traffic model is based on network traffic and attack duration information. The ports and traffic model focuses on inbound and outbound ports and traffic. A web application is created for administrators or other users to visualize the findings. The WebSocket protocol provides real-time streaming of traffic details to Wazuh, which further provides alerts to EDK. The alert details are passed from EDK to the browser service to be displayed via the browser for the administrator.

Rules-based detection is a way of using machine learning for threat detection. Wazuh—the open-source fork of OSSEC—uses a rules-based detection system or a signatures-based attack detection system. Wazuh, along with the WebSocket streaming setup, and threat detection and blocking models for the IDS cluster is implemented. A GCP VM creates a real-time streaming connection between Wazuh and the IDS cluster. The streaming service continuously passes traffic details to Wazuh. These details can be optimized, filtered or edited using StreamAlert, which runs on the Streaming Service VM.

Data Processing Techniques

Streaming data processing requires significant computing power and has real-time constraints not present in batch processing. Processing streaming data for an intelligent IoT security system lowers the risk from attack events and allows fast decision-making. The cloud-native architecture for an intelligent IoT security system described in earlier sections uses a data processing technique on streaming data.

Streaming data processing detects real-time AI-generated threat events before they reach near-real-time consoles, allowing security teams to respond quickly. Data streaming services AWS Kinesis Data Streams, Google Cloud Pub/Sub, and Microsoft Azure Event Hubs integrate scalable services for real-time and near-real-time data processing. Services Amazon Kinesis Data Analytics, Google Cloud Dataflow, and Azure Stream Analytics provide data processing that filters and modifies raw Apache Kafka event messages for visualization. With this approach, the event message structure used for visualization can be different

from the raw Kafka event message structure.

Equ 3: Data Security and Access Control (Optional)

$$\text{Access} = \begin{cases} \text{Granted}, & \text{if } T_u \in \mathcal{A} \\ \text{Denied}, & \text{otherwise} \end{cases}$$

Where:

- T_u: user token
- $\mathcal{A}$: set of valid tokens or roles (RBAC model)

Web Visualization Techniques

An intelligent Internet of Things (IoT) security system inherently necessitates aspects of web visualization. Specifically, web visualization offers the IoT operator the ability to quickly monitor all devices connected to the cloud and detect abnormalities. In the context of cyber-security, web visualization can intuitively present real-time cyber-attack information, thereby assisting cyber-security analysts in promptly responding to attacks and safeguarding critical systems.

The employed Zabbix method for web visualization relies on the Web services API of Zabbix to extract information related to the IoT security system's detection results. Zabbix acts as a bridge between the IoT security system and the developed web visualization, enabling data queries based on various parameters. Internally, Zabbix utilizes the MySQL query language to operate on its database. For illustration, using the SQL query language to analyze the Zabbix database can reveal the attack mode with the highest volume of cyber-attack detection within a specified time range.

Fig 5: Best Data Visualization Techniques

User Interface Design Principles

In a modern intelligence IoT security system, the user interface is not just a simple evolution of design but a deliberate, comprehensive approach intended to enhance information presentation and management. The emphasis goes beyond aesthetics, centering on a full-stack design that optimizes how information is communicated, facilitates user control over it, and supports easy modification and addition of features. Deep comprehension of user needs and consistent improvement of the user experience are paramount for ensuring the interface meets the demands of the moment and remains resilient to the rapidly changing security landscape.

The web page of the client web service is the most direct access point for users. Beginning with principles of simplicity, efficiency, recognition, popularity, unusually interesting elements, and post-processing hints, the visual style and color scheme are clearer and more reasonable than in previous iterations. The new design for area and skill selection significantly enriches operation methods. Carefully designed elements—such as operation controls, search functionality, area and skill selection, message browsing, and location information—contribute to a well-rounded interactive experience. Icons for operation controls—occlusion, pause, play, and refresh—and an integrated search bar enhance user convenience. The area and skill selection column allows all related comments and the To-Do List to be filtered by area and skill dimension. A message column categorizes items within the To-Do List, facilitating focused handling, while a location information column prepares the view for Google Map display.

Data Visualization Tools

In their basic form, these graphs are error bars, with the central tendency displayed as the dot and the uncertainty represented as the width of the bar. For boxplots, the center line depicts the median, the dot corresponds to the mean, and the shape indicates the range between the 25th and 75th percentiles. Similarly, violin plots utilize a mirrored kernel density estimation (KDE) to illustrate the full distribution and its skewness.

Equipped with a web interface, the cloud-native DPS enables user-friendly data visualization by plotting specified indices and estimates from the database through a graphical user interface (GUI). The Plotly Dash web interface is designed to be simple, flexible, and easy to develop, featuring drop-down menus for dataset name, data index name, estimate name, plot type, confidence interval type, and (optionally) data grouping.

CONCLUSION

The rapid adoption of cloud-native architecture in the development of intelligent Internet of Things (IoT) security systems enables an efficient connection between a heterogeneous group of self-configured distributed edge devices and cloud resources for data collection, training, and storing. In this particular use case, each edge device equipped with Omron surveillance cameras streams video to the cloud for real-time intelligent analysis. The system

employs several AI models running inference processes to detect multiple events of interest, such as vehicle presence, license plate (LP) recognition, intrusion detection, chronologically identified license plates, unsafe behaviors, and loitering. Analysis metadata, together with LP images and blocked car images, are stored in the cloud. They are subsequently displayed in a user-friendly web portal that offers live analytics information linked to the corresponding LP image, including details such as LP number, time detected, and car-blocked time.

The system for studying human behavior as a response to the COVID-19 pandemic is divided into the front-end and the back-end. The front-end provides the user interface for displaying information, while the back-end consists of edge devices, a message bus, and the cloud, collectively responsible for managing the entire process. The software on the front-end is built using Vue.js, and analysis metadata are fetched through RESTful API connectors that communicate with the back-end. The back-end employs a serverless function on AWS Lambda to handle metadata received from an MQTT message bus, which is managed by Mosquitto—an open-source MQTT broker running on an AWS EC2 instance. Samples of human-behavior-related images and videos received from edge devices are uploaded and stored in Amazon S3. Data collection and subsequent processes for responses to the COVID-19 pandemic are made possible by a distributed-heterogeneous-device connection layer, allowing edge devices from different manufacturers to connect and stream information through a self-configuration mechanism.

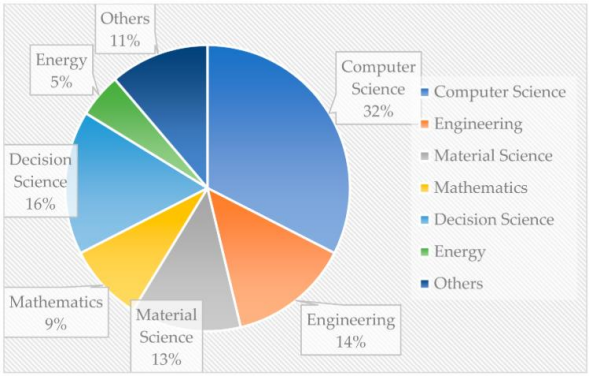

Fig: Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence

Emerging Trends

Industry 4.0 (I4.0) aims to digitize and transform production and manufacturing processes by embedding sensors into every physical asset in the factory. Consequently, software intelligence and scripts in the cloud will automatically monitor, report outcomes, and adjust production lines for efficiency and other business goals. The rapidly evolving IoT is at the heart of this digital transformation toward smart manufacturing and smart factories. One need is to gather data from every process and asset in near real-time and with high availability. Within buildings, the IoT extends across all applications for building management, comfort, security, and surveillance. Intelligent IoT sensors enable smart buildings where every physical aspect can be measured and controlled through software intelligence.

From an IT perspective, the cloud is also evolving into a very capable and efficient resource for operating systems, machine learning models, storage, computing, industrial process management, workflows, and even control and execution systems. The beauty of cloud computing is that these resources are available within seconds; scaling up Another emerging aspect of IoT is Area 51 IoT, where the IoT is installed in remote regions of earth and other planets for scientific, exploratory, and militaristic reasons. The common theme across all is the need to have secure, resilient, accessible, and reliable IoT networks with full situational awareness.

REFERENCES

1. Recharla, M., Chakilam, C., Kannan, S., Nuka, S. T., & Suura, S. R. (2025). Harnessing AI and Machine Learning for Precision Medicine: Advancements in Genomic Research, Disease Detection, and Personalized Healthcare. American Journal of Psychiatric Rehabilitation, 28(1), 112-123.
2. Chakilam, C., Sunil, D. T. K., Sivakami, R., Suresh, K., Negi, A. S., & Suresh, T. (2025, May). Immersive Augmented Reality Collaboration Platforms for Future Workplace Productivity Team Innovation and Virtual Co Working Spaces. In International Conference on Sustainability Innovation in Computing and Engineering (ICSICE 2024) (pp. 1583-1593). Atlantis Press.
3. Nagamani, A., Nuka, S. T., Iyyanar, P., Suzana, A. A., Nayak, M., & Kumar, S. S. (2025, May). Artificial Intelligence Optimized Citizen Engagement Platforms for Resilient Urban Futures Inclusive Policy Making and Smart Community Driven Governance. In International Conference on Sustainability Innovation in Computing and Engineering (ICSICE 2024) (pp. 221-231). Atlantis Press.
4. Manikandan, K., Pamisetty, V., Challa, S. R., Komaragiri, V. B., Challa, K., & Chava, K. (2025). Scalability and Efficiency in Distributed Big Data Architectures: A Comparative Study. Metallurgical and Materials Engineering, 31(3), 40-49.
5. Peruthambi, V., Pandiri, L., Kaulwar, P. K., Koppolu, H. K. R., Adusupalli, B., & Pamisetty, A. (2025). Big Data-Driven Predictive Maintenance for Industrial IoT (IIoT) Systems. Metallurgical an
6. Koppolu, H. K. R., Nisha, R. S., Anguraj, K., Chauhan, R., Muniraj, A., & Pushpalakshmi, G. (2025, May). Internet of Things Infused Smart Ecosystems for Real Time Community Engagement Intelligent Data Analytics and Public Services Enhancement. In International Conference on Sustainability Innovation in Computing and Engineering (ICSICE 2024) (pp. 1905-1917). Atlantis Press.
7. Singireddy, J. (2025). Smart Finance: Harnessing Artificial Intelligence to Transform Tax, Accounting, Payroll, and Credit Management for the Digital Age. Deep Science Publishing.
8. Kummari, D. N., Challa, S. R., Pamisetty, V., Motamary, S., & Meda, R. (2025). Unifying Temporal Reasoning and Agentic Machine Learning: A

Framework for Proactive Fault Detection in Dynamic, Data-Intensive Environments. Metallurgical and Materials Engineering, 31(4), 552-568.
9. Somu, B., & Inala, R. (2025). Transforming Core Banking Infrastructure with Agentic AI: A New Paradigm for Autonomous Financial Services. Advances in Consumer Research, 2(4).
10. Yellanki, S. K. (2025). Behavioral Intelligence and Operational Design: Exploring Modern Service Models, Customer-Centric Platforms, and Sustainable Digital Infrastructure. Deep Science Publishing.
11. Krishnaprasath, V. T., Pamisetty, V., Sharma, V., Nayak, M., Baalakumar, N. N., & Aravindh, S. (2025, May). Federated Learning Based Artificial Intelligence Systems with Blockchain Security for Global Healthcare Collaboration and Patient Centric Data Privacy. In International Conference on Sustainability Innovation in Computing and Engineering (ICSICE 2024) (pp. 1277-1290). Atlantis Press.
12. Motamary, S. (2025). A Deep Dive into CI/CD Pipelines Tailored for Telecom: Orchestrating Cloud-Native 5G Services with DevOps and Infrastructure Automation. CD Pipelines Tailored for Telecom: Orchestrating Cloud-Native 5G Services with DevOps and Infrastructure Automation (May 04, 2025).
13. Dodda, A. (2025). Artificial Intelligence and Financial Transformation: Unlocking the Power of Fintech, Predictive Analytics, and Public Governance in the Next Era of Economic Intelligence. Deep Science Publishing.