International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

Cloud-Native AI Pipelines for Real-Time Cyber Threat Intelligence

Dileep Valiki

Independent Researcher, India

ABSTRACT: Cloud-native pipelines are expected to be able to analyze data in cloud data lakes using a reconciliation process to detect and process new types of tactical events relating to cyber threat intelligence. Data-related, source-related, and service-related considerations and requirements for real-time pipelines that deploy deep learning processes to create Cloud-native AI Pipeline for Ransomware Detection are discussed. The architecture is open-source-based, with a collection of repositories containing several online learning processes using cloud-native AI techniques.

Cyber threat detection is a dynamic process that depends on the existence of detections of malicious event patterns from indicators and triggers for reactive responses by using relevant updated data sources. The technical challenge derives from the need for these detections, supporting indicators, and tactical action responses to be created and activated in a real-time automated fashion. An extensible and scalable event correlation engine has been proposed, capable of detecting malicious events occurring in real time. However, detection of new types of events is hampered by the requirement for expert-driven feature-extraction processes and supervised learning classifiers capable of recognizing only specific threats, limiting the detection space to only those attack types for which the model has been previously trained.

In online learning processes under continuous deployment, the model is continuously trained as new data is ingested in the base, thus enabling the recognition of new types of events as they appear. Distributed cloud-native process architecture enables the continual deployment of adaptation processes into production that can use data stored in cloud data lakes for continuous retraining and adaptation. A new, integrated, end-to-end visual Cloud-native AI Pipeline for Ransomware Detection based on the recognition of the modus operandi of ransomware is implemented. Data-related, source-related, and service-related considerations for real-time pipelines built using cloud-native AI techniques that generate deep learning processes are described.

KEYWORDS: Cloud-Native AI Pipelines, Ransomware Detection Systems, Cyber Threat Intelligence (CTI), Real-Time Event Correlation Engines, Deep Learning for Cybersecurity, Online Learning Architectures, Continuous Model Retraining, Cloud Data Lake Analytics, Tactical Event Reconciliation, Distributed Cloud-Native Processing, Malicious Pattern Recognition, Adaptive Threat Detection Models, Open-Source Security Frameworks, Automated Incident Response Systems, Indicator and Trigger-Based Detection, Scalable Microservices Architectures, Continuous Deployment (CD) in AI Systems, Modus Operandi-Based Detection, Feature Engineering in Cybersecurity, Intelligent Security Operations Pipelines.

I. INTRODUCTION

The growing impact of cybercrime has forced government and other organizations to enhance their abilities to anticipate new threats and protect their systems proactively. Timely detection, advanced preparation, and improved response management require accurate up-to-date intelligence. Each year, ever more organizations report being subject to security events; however, only a small fraction of these events have been detected in advance. Building models for real-time cyber threat intelligence has new challenges that require continuous processing of threat intelligence data coming from different sources.

As with any other IT system, cloud computing should serve as a support technology for the cybersecurity domain. Cloud-native architectures offer the essential properties of being scalable, fault-tolerant, elastic, upgradable, and automatically orchestrated for a global deployment. A challenge in cyber threat intelligence is to provide pipelines for real-time processing of threat intelligence data that cover all aspects of the data flow. An end-to-end cloud-native pipeline that allows researchers and practitioners to use the entire range of pipeline modules is proposed, illustrating the key concepts involved in such a pipeline. Interoperability is one of the main issues with threat intelligence.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

Incorporating Threat Exchange Format files directly into the pipelines is very useful for speeding up the integration of knowledge from sensors.

1.1. Overview of the Study

Real-time threat intelligence relies upon a comprehensive analysis of raw data, event details, tools and techniques, and malware. Such an analysis is normally not conducted in real time and lacks cloud-native architectures and principles. This section summarizes concepts and requirements for real-time cyber threat intelligence and explains how concepts of cloud-native architectures and patterns can be exploited to address them.

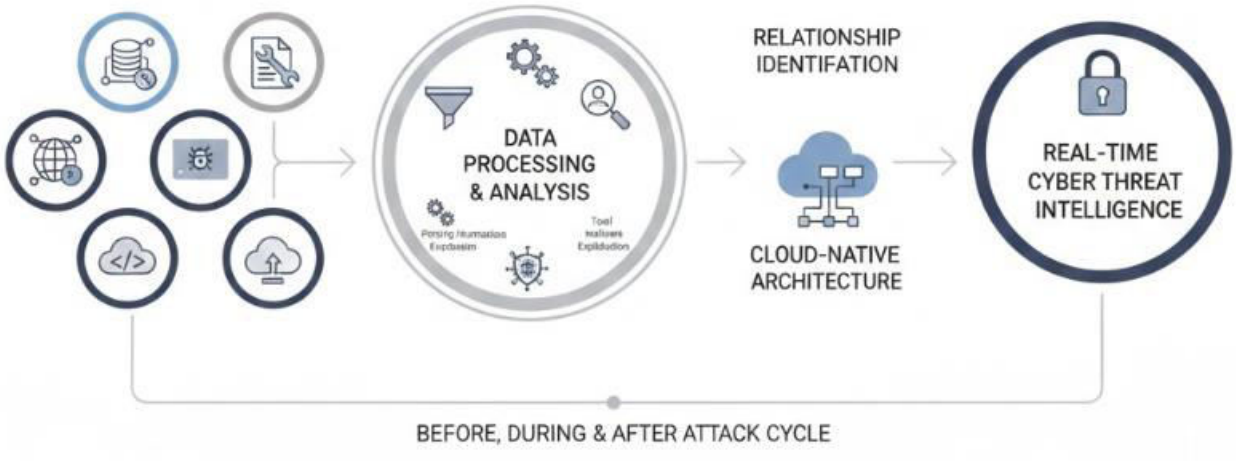


Fig 1: Architecting Cloud-Native Real-Time Cyber Threat Intelligence: A Multi-Source Framework for Scalable Data Enrichment and Event Relationship Identification

The analysis of massive amounts of data for the identification of cyber threats before, during, or after an attack has been referred to as real-time threat intelligence. It involves data capture, parsing, normalization, enrichment, and exploitation of tools, techniques, and malware repositories on the one hand and identification of relationships between events on the other. Unlike traditional threat intelligence, real-time threat intelligence is based on processing all available data sources of interest as opposed to a specific class of data sources or single sources, starting from five raw data sources at a minimum.

II. BACKGROUND AND FOUNDATIONS

Developments in real-time cyber threat intelligence in cloud-native environments, from data ingestion and processing to deployment, integration, and interoperation with external systems and services

Decisions about data acquisition, processing, and delivery must be informed by requirements and constraints from cyber threat intelligence consumers. Threat-detection services often require real-time predictions based on streaming data. AI models that are retrained on a regular basis may not be entirely accurate until they are finally deployed, or during the time lag that separates the model's prediction horizon and the development window. Models may also occasionally become stale. To address these challenges and provide consistent predictions under changing data patterns, a cloud-native AI strategy can support deployment in line with cloud-native principles to enable rapid scaling for both workload spikes and steady increases in demand.

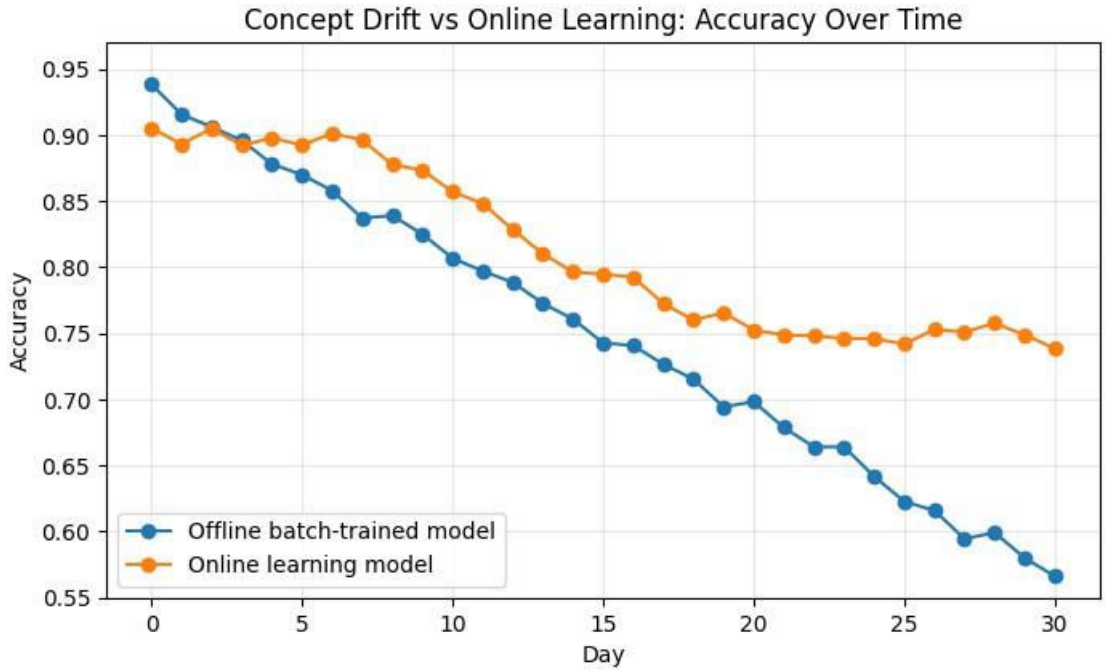
Real-time prediction pipelines should span from either ingestion or feature-extraction right through to final prediction. Depending on the nature of the completed task and the operational approach being taken, several patterns are applicable to the online learning and adaptive modelling mechanisms. If the learning and prediction are being wrapped and deployed within a cloud-native platform, additional considerations including platform abstractions for packaging and orchestration, guarantees for elastic scalability and reliability by the platform, and end-to-end observability within the overall pipeline must be addressed.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036



Equation 1) Streaming pipeline as a timed queueing system (latency + throughput)

1.1 End-to-end latency

Step-by-step:

1. Time spent across all stages is additive (series system):

$$L_i = \sum_{k=1}^{K} t_{i,k}$$

2. For a batch/window of N events, average latency:

$$\bar{L} = \frac{1}{N}\sum_{i=1}^{N} L_i = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} t_{i,k}$$

3. Swap the summations (linearity):

$$\bar{L} = \sum_{k=1}^{K}\left(\frac{1}{N}\sum_{i=1}^{N} t_{i,k}\right) = \sum_{k=1}^{K} \bar{t}_k$$

So **average end-to-end latency equals the sum of average stage latencies**.

1.2 Throughput and capacity constraint

Let arrival rate be λ (events/sec). Let a stage k have service rate μ_k (events/sec). For stability (no unbounded backlog), the slowest stage dominates:

Step-by-step:

1. Each stage must keep up: $\lambda < \mu_k$ for all k.
2. Therefore the binding constraint is:

$$\lambda < \min_k \mu_k$$

3. The pipeline's maximum sustainable throughput is approximately:

$$\lambda_{\max} \approx \min_k \mu_k$$

1.3 Horizontal scaling (microservice replicas)

If stage k is replicated r_k times and scales near-linearly, then:

1. Single replica service rate: $\mu_k^{(1)}$
2. With r_k replicas:

$$\mu_k^{(r_k)} \approx r_k\, \mu_k^{(1)}$$

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

3. Pipeline capacity becomes:

$$\lambda_{\max}(r_1, \ldots, r_K) \approx \min_k \left(r_k \mu_k^{(1)}\right)$$

2.1. Real-Time Cyber Threat Intelligence: Concepts and Requirements

Real-time cyber threat intelligence is recognized as an increasingly important element in an organization's defensive capabilities. Threat intelligence enhances situational awareness and provides knowledge about current threats that can be integrated into organizations' defences as they evolve. True real-time capability spans the entire lifecycle from initial detection, through sharing and dissemination, to response. Requirements include full coverage for all possible threat information sources, with data streamed over a pub/sub and message broker architecture to processing pipelines that find patterns and extract significant new features. New detections should be correlated with other incoming information to filter out noise before being combined with supporting threat information and presented or disseminated as alerts or updated indicators.

Current approaches to real-time cyber threat intelligence do not fully satisfy these requirements. Use cases and data sources for real-time cyber threat intelligence are outlined. A framework is proposed for integrating data from these sources to common frameworks and standards, and a streaming architecture for the real-time data ingestion, analysis, and generation of alerts is described. Similar patterns apply to the incorporation of cloud-native AI capabilities into proactive use of MITRE ATT&CK adversary knowledge. This supports scenario construction, simulated attack forensics, and information dissemination but currently lacks real-time capability. Continued research in these areas is required.

2.2. Cloud-Native Architectures: Principles and Patterns

Surrounding the concept of a cloud-native deployment are public cloud abstractions that enable easy and speedy orchestration of the functionality of a system at a given layer of abstraction by a third party. At a higher level of abstraction, a system is published as a service to be consumed through defined service-level agreements (SLAs), which may include availability and durability objectives. Cloud-native architectures allow the implementation of the same service on different platforms, such as on-premises, in the cloud, hybrid, or multi-cloud solutions.

Traffic patterns associated with global applications often exhibit traffic locality with respect to the serving location of the application. This property enables the strategy of deploying a cloud-native application in multiple locations while providing a consistent view of the data. Stepping down a level of abstraction, pattern detection mechanisms applied to the flow specification data enable automatic mapping of flow specification to availability zone (AZ) Balkanization. At a layer below, cloud-native design patterns can also provide abstractions to smooth the implementation and operation of individual components that are part of a broader service offering.

A cloud provider implements an elastic solution using object storage for data. All operations are automatically distributed and built to be parallel processing-friendly, requiring zero local storage, with the only control being the number of consumers. AKS is used as the underlying orchestration technology. During automated function testing, failure points are dynamically simulated using Chaos Engineering principles to validate the cloud-native capability of the application.

Day	Accuracy_OfflineBatch	Accuracy_OnlineLearning
25	0.6227466065042673	0.7418298061733956
26	0.6159738607228616	0.7529617202585397
27	0.5945584244901415	0.751027444961219
28	0.5992469827467164	0.7576750275732838
29	0.5799176284074	0.7488483340281059
30	0.5662872323739773	0.7386081499105898

III. DATA INGESTION AND INTEGRATION IN CLOUD-NATIVE PIPELINES

Cyber Threat Intelligence (CTI) consists of a set of technical and human threats, attacks and anomalies in the cybersecurity domain. Threat reports are shared in textual format, in TTP (Tactics, Techniques, and Procedures), CVE (Common Vulnerabilities and Exposure), MISP (Malware Information Sharing Platform) and STIX (Structured Threat Information Expression) formats, available from multiple organizations. In addition, alerts and events generated by security tools and devices (intrusion detection systems, firewalls, network traffic monitoring tools and more) are constantly ingested. Combining event data with threat reports is a complex challenge, moreover, because such data is

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

produced at a large scale. Scalability, low latencies and proper data structures are necessary for a suitable cloud-native architecture.

Artificial Intelligence (AI) systems are expected to ingest large amounts of data produced in various formats, be flexible for recombination, react instantly and be scalable. Event correlation brings together threat information with matching events from security tools. Detection and attribution follow. Alerts from detection engines must be correlated to reduce false positives yet identifying new threats using open-set recognition, open-set classifiers or One-Class classifiers. The recognition models must be adaptive because recency is important: the model trained for the last year may not identify Zero-day attacks.

Equation 2) Sliding-window event correlation (pattern frequency in real time)

Let $x_i = (\tau_i, a_i)$ be event i with timestamp τ_i and attribute vector a_i (IP, user, hash, TTP, etc.). Let a time window of size W end at time t: $(t - W,\ t]$.

2.1 Count of a pattern *P* in the window

Define an indicator that event i matches pattern P:

$$\mathbf{1}_P(x_i) = \begin{cases} 1 & \text{if } x_i \text{ matches } P \\ 0 & \text{otherwise} \end{cases}$$

Step-by-step:

3. Only count events in the window:

$$\mathbf{1}_{(t-W,t]}(\tau_i) = \begin{cases} 1 & \text{if } t - W < \tau_i \leq t \\ 0 & \text{otherwise} \end{cases}$$

3. Count matches in the window:

$$C_P(t) = \sum_i \mathbf{1}_{(t-W,t]}(\tau_i)\, \mathbf{1}_P(x_i)$$

4. Trigger rule (frequency threshold θ):

$$\text{Trigger}(P, t) = \mathbf{1}\{C_P(t) \geq \theta\}$$

3.1. Data Sources for Threat Intelligence

To achieve a comprehensive, continuous, real-time Cyber Threat Intelligence that is operationally meaningful soon after production, it is necessary to gather and process data not only from traditional endpoints and servers, but also from other resources that are able to produce actionable information at very short time spans. Since cyber threats take place in the cyberspace context, either perpetrated from it or aimed against it, another broader source of information comes from the study of events on the cyberspace per se, including sites, social media (Twitter, Facebook, Reddit, etc.), blogs, underground forums and markets, paste sites, and domain registrars (e.g., WHOIS), which act as USB devices for fresh exploited data.

Additional feeds from open, commercial, and technical monitoring services provided by private companies and/or international governmental organizations offer valuable information regarding ongoing attacks, botnets, malware command-and-control (C&C) servers, and exploitable vulnerabilities and exploits in the format of Common Vulnerability and Exposure (CVE) identifiers and Common Vulnerability Scoring System (CVSS) severity scores. These services include several agencies, such as the Department of Homeland Security (DHS), US-CERT, European Union Agency for Cybersecurity (ENISA), and Cyber Security Agency of Singapore. Indeed, the amount of available information and services about Cyber Threat Intelligence has exploded in the last years, and in some cases, combinations of services can serve as ready-to-use Cyber Threat Intelligence sources. Examples include the Emerging Threats ET Open DDoS, or the Emerging Threats Compromised IP List, to report known botnets and infected machines taking part in DDoS attacks, Anti-Virus companies such as Webroot, which reports IP addresses using a malvertising infrastructure, and Trend Micro, which publishes Internet Threat Landscapes.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

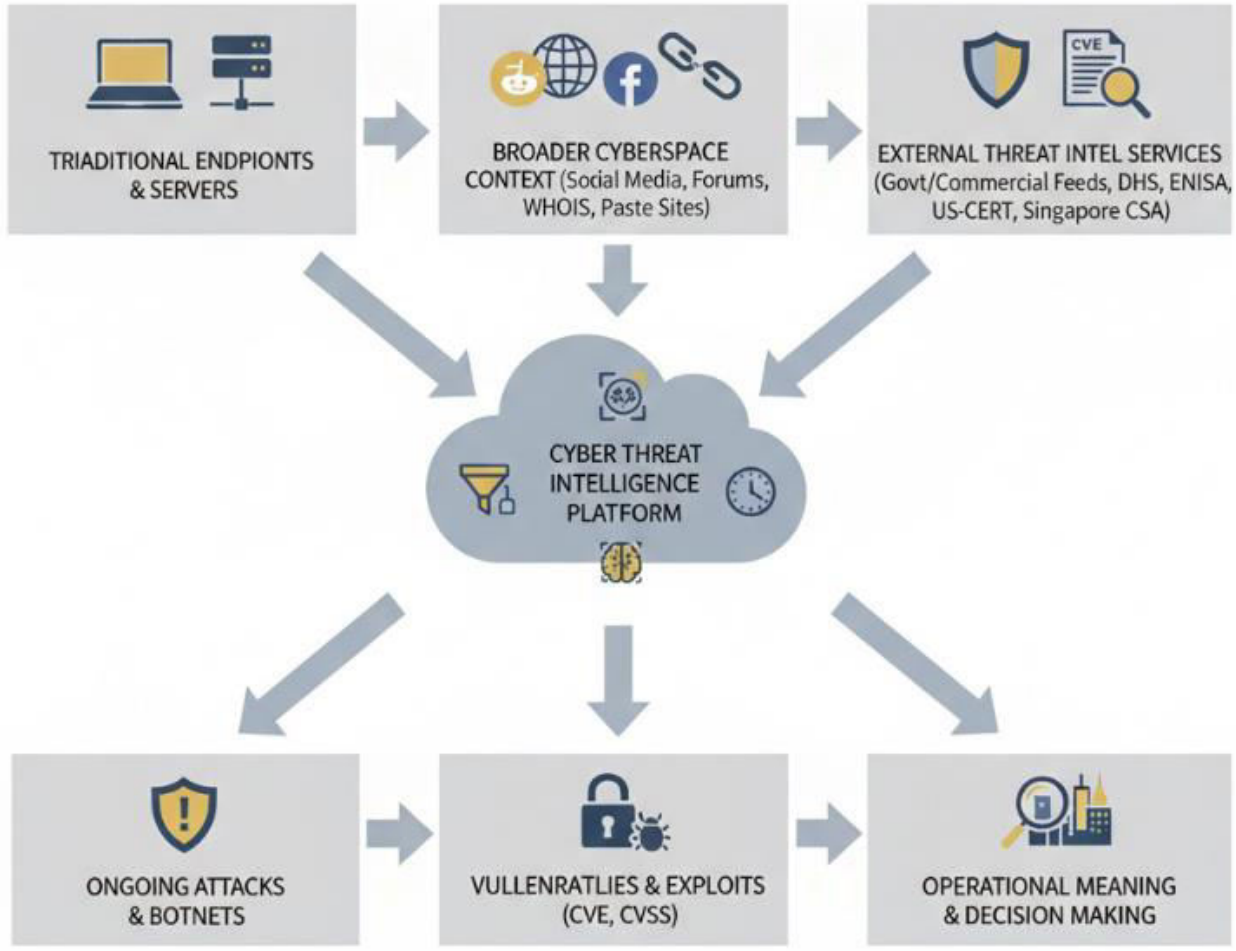


Fig 2: Multi-Source Cyber Threat Intelligence: A Framework for Real-Time Operationalization Across Global and Social Data Streams

3.2. Ingestion Frameworks and Streaming Architectures

Ingesting and processing the incoming data streams without delay is paramount for automated threat detection and event correlation and for timely warning to defense teams. For this reason, Cloud-Native Pipelines for Real-Time Cyber Threat Intelligence will typically exploit Fast-data Ingestion Frameworks (FIIFs) as ingestion modules. FIIFs support decentralized topologies allowing fast, event-driven processing of data streams in a distributed, fault-tolerant manner, and achieving low latency, high throughput, and horizontal scalability. A promising software stack for data ingestion is the Apache open-source ecosystem, notably: Apache Kafka for Low-latency, High-throughput Publish/subscribe messaging system, Apache Kafka Streams and Apache Flink for a Distributed Stream Processing Engine, Apache Pulsar for a Low-latency, Fault-tolerant Pub-Sub messaging system, and Apache Pino for a Distributed, Column-oriented Data Store focusing on low-latency OLAP.

FIIFs may also be integrated with traditional Big Data Environments built atop Apache Hadoop and Apache HDFS to build hybrid Data storage and processing Systems. Usually, Cyber Threat Intelligence data will be collected, consolidated, and monitored in real-time within a centralized system called Security Information and Event Management (SIEM), which provides capabilities for both Security Information Management (SIM) and Security Event Management (SEM). Event correlation, together with real-time monitoring, enables the detection of complex attacks and abnormal behaviors through pattern detection, anomaly detection, and event enrichment techniques.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

Stage	Latency_ms
Ingestion	10
Parsing/Norm	13
Enrichment	17
Feature Extract	25
Model Inference	17
Correlation	32
Alert/Push	9

IV. REAL-TIME PROCESSING AND ANALYTICS

Data from different streams, possibly produced by diverse entities, require identification of common characteristics through feature extraction, subsequently making the content amenable for analytics. Representatives and a simplified summary of the source are also often required for effective transit and storage. Aggregation systems typically perform feature extraction to compile summaries of elements appearing in rapid succession or to simplify the payload for subsequent streams. Security events often occur in temporal proximity to other contextually relevant events. Though labelled training data may be difficult to generate in large volumes, learning from events labelled as benign may be possible when labelled data for indicated threats are found. When faced with data from diverse domains, online or continual learning methods equipped with multi-task or multi-label capabilities may enable expansion or adaptation to more specialised models.

The data provided by threat intelligence services may often be in the wrong modality for effective action — a label indicating that a specified domain is suspected of being compromised is of limited use for automated detection systems. Data from multiple such services covering a range of associated categories related to malicious activities may enable the direct use of the labels provided through cross-modality online learning techniques. As the various models for chunked processes are relatively lightweight, they may even be combined into a pre-processing stage that inserts alternative labels as additional features into the data stream for consideration by other detection and analysis components. The services are often unreliable or transient, making imitation of black-market threat feeds an attractive target.

Equation 3) Online learning update (why accuracy decays, how continuous updates fix it)

3.1 Online gradient descent for a binary detector

Let the model be $f_w(x) = \sigma(w^\top x)$ where $\sigma(z) = \frac{1}{1+e^{-z}}$. For label $y \in \{0,1\}$, use log-loss:

$$\ell(w; x, y) = -\big[y \log f_w(x) + (1-y)\log\big(1 - f_w(x)\big)\big]$$

Step-by-step gradient:

4. Let $p = f_w(x) = \sigma(w^\top x)$
5. Known derivative: $\frac{d\sigma(z)}{dz} = \sigma(z)\big(1-\sigma(z)\big) = p(1-p)$
6. Differentiate loss w.r.t. w (chain rule):

$$\nabla_w \ell = (p - y)\, x$$

4. Online update with learning rate η_t:

$$w_{t+1} = w_t - \eta_t (p_t - y_t) x_t$$

3.2 "Update only if informative"

7. Define a difficulty/uncertainty score (closest to decision boundary is most informative):

$$u_t = |p_t - 1/2|$$

4. Update only if uncertainty is high enough (i.e., $u_t \le \delta$):

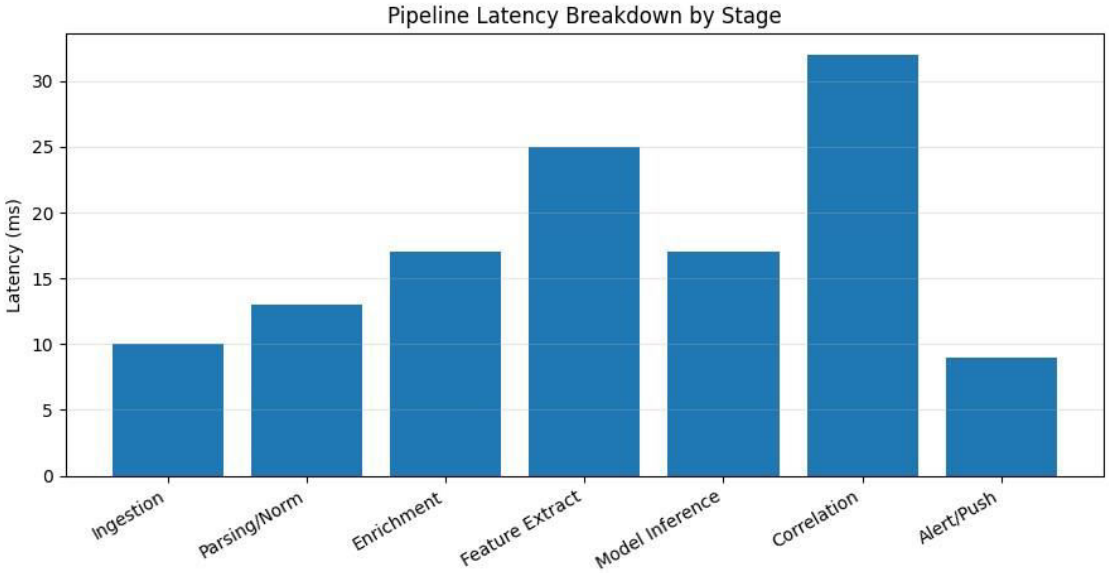
$$w_{t+1} = \begin{cases} w_t - \eta_t (p_t - y_t) x_t & \text{if } |p_t - 1/2| \le \delta \\ w_t & \text{otherwise} \end{cases}$$

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036



4.1. Feature Extraction and Event Correlation

In cloud-native pipelines for real-time cyber threat intelligence, fundamental processes such as feature extraction and event correlation must support incredibly rapid responses. Detection of thousands of events per second is common in some contexts, since attempts to compromise any organization are no longer individual, country, or major military events, but an everyday industry-wide threat that remains effectively secret until actual compromise. Individual events on a single system, such as multiple failed logins in a time-based sequence, are commonplace and known, although their appearance on every system during a compromise is not. Cyber activity for a few seconds across many systems can be an indication that something very fundamentally nasty is occurring in real time and needs immediate action. Much of the strategy for detecting it is based on classical pattern matching, checking for values, order, or frequency within a sliding time window, but now the candidates for those patterns are not long standing (thousands of such patterns need to be detected), are not human generated, and need automated matching. Automatic feature extraction cannot simply rely on a set of labels.

In the online learning context, the next thing that happens to some entity can be termed an input event. It can be viewed as a new set of data, which is fed into the model selection process to see if one of the patterns can temporally match that data. Each day a corporation sees normal behavior across the actual wording in their emails but that normality does not exist for other corporations. Major companies have thousands of social media accounts but they also are not in the same topic context. Activity clusters and the classification need to dynamically reflect those entities that can now be grouping together in one-off campaigns and need a different focus or can be discarded. Multiple speaking patterns emerge during sporting events and financial earnings announcements; after such events only a few remain active. The filtering is separately visible in some of the actual labeled data, the activity clusters, and the grouping of social media. The evolution of the activity promoting one of those time-limited temporary speaking patterns requires all of these three aspects to dramatically dynamically refocus during operation.

4.2. Online Learning and Adaptive Models

Traditionally, models for cyber threat detection or prediction are trained off-line on historical data and then deployed into production. The resulting models may perform well for a certain period, but gradually their accuracy diminishes due to concept drift, eventually compromising their utility. Although retraining on new batches of labeled data may temporarily mitigate this issue, the overall approach remains largely manual and reactive. Models may also overfit rare events because, in practice, they can often only learn to detect an attack type from a limited number of such labeled instances. They are incapable of adapting to changes in the underlying system, network, or user behavior that are unrelated to an intrusion. As a result, a new attack type that occurs quite often in reality cannot be detected by an updated model because it was missing from the training data.

Online learning enables the systematic updating of a model with each new instance that arrives. It can decide whether the incoming instance is informative and, thus, whether to use it to amend the model. This can address the problem of

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

gradual diminishment of detection performance after deployment. Online boosting combined with online kernel-based learning provide a scalable mechanism to combine predictions from multiple models, each specialized in detecting particular types of attacks. A more generic adaptive damage detection model can be built to swiftly identify malicious events or objects under system, network, or user behavioral changes, so that the response can be tuned more suitably. Additionally, interaction-based relational traffic analysis can be employed to incrementally learn the normal behavior of individual network users and nodes, reducing chances of false positives due to benign behavior changes.

V. CLOUD-NATIVE DEPLOYMENT CONSIDERATIONS

Real-time machine learning pipelines can be constructed using cloud-native technologies and cater to specific use cases. When deploying into cloud-native platforms, the main aspects to consider are: 1) platform abstractions for deployment, orchestration and scaling; 2) reliability, availability and service observability; 3) data flow observability. Each of these considerations is discussed below.

Cloud-native applications provide abstractions to support deployment, hosting and orchestration. Platform-as-a-Service (PaaS) solutions enable developers to deploy an application using the source code and take care of everything else under-the-hood, including rolling updates, traffic routing, service discovery and scaling. Such PaaS solutions can also support use cases involving continuous model training and model refresh using files stored within the cloud. For use cases that require deployed models to periodically refresh, Model-as-a-Service (MaaS) solutions are also available. Data can be made available for retraining using file triggers, file sensors or polling mechanisms.

Another aspect of cloud-native applications is the scaling possibility provided by microservices. The stateless nature of cloud-native applications allows for horizontal scaling. For ingestion services, streaming services and prediction services supporting high volumes, multiple replicas of individual services can be created or scaled automatically based on the number of messages processed.

Replicas	Throughput_events_per_sec
1	10669
2	20529
4	40830
8	72356
16	120738

5.1. Platform Abstractions and Orchestration

Cloud-native approaches emphasize modular and service-oriented software design practices, fostering the use of low-level abstractions that separate compute, data storage, and networking services. For dynamic resource allocation, these abstractions are deployed in containerized environments. Containers provide application isolation, lightweight management (compared to traditional virtual machines), and faster provisioning—all crucial features for cloud elasticity. Cluster orchestrators, such as Kubernetes, automate container lifecycle management by deploying, scaling, and rebooting application components based on configurable templates. Continuous monitoring and metrics reduce human errors and unplanned resource consumption for application operations.

Although orchestrators enable self-healing features and support multicloud deployment and scaling, holistic deployment patterns (e.g., service meshes or serverless) remain open research areas. Indeed, cloud-native software paves the way for modular implementations that can be extended beyond simple sharding or replication patterns.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

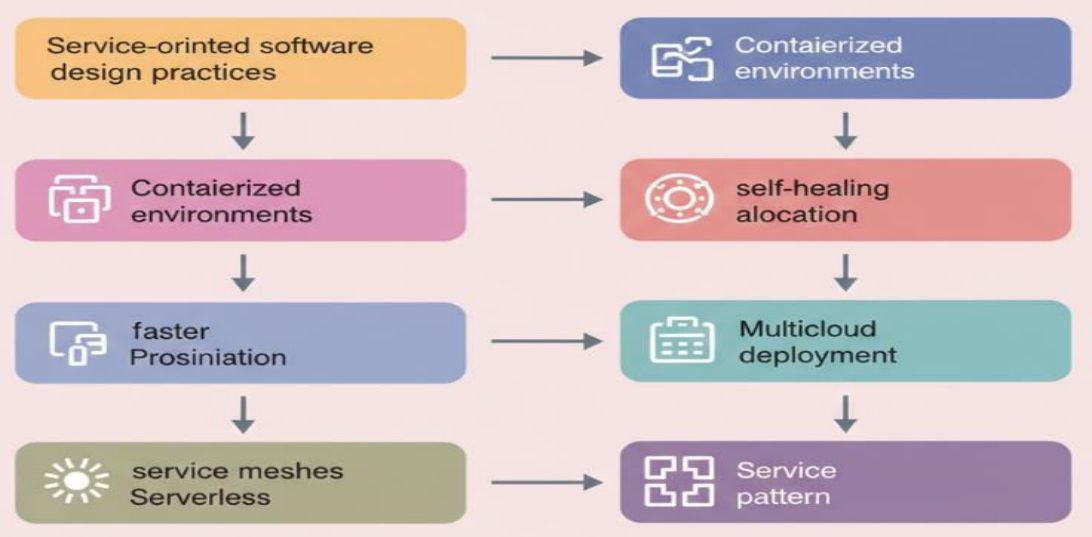


Fig 3: Beyond Containerization: Orchestrating Cloud-Native Resilience through Modular Abstractions, Service Meshes, and Serverless Paradigms

5.2. Scalability, Reliability, and Observability

Key principles that are fundamental to any system deployed on the cloud—two of the largest and most advanced on the market are from Google and Amazon—are scalability and reliability. Any service can be instantiated in many replicas, as demand increases. These replicas are load-balanced and, if necessary, the orchestration platform directs new operational units to idle. Any replica can be declared corrupted and expelled from the system. These capabilities are usually highlighted on the providers' technical sheets: the cloud can absorb unexpected systemic peaks, seasonal elastic load distributions, and even attacks on the service.

Another aspect offered by cloud providers is observability. These enterprises strongly encourage the recording and monitoring of any behaviour, and they even provide resources such as dynamically configured monitoring and test error patterns that can trigger the automated real-time initiation of specific predictive models. However, for many years, too much information has made the analysis of the evaluated system difficult to use. Observability can also impact the cost of real-time intelligence by recording and analysing predictions, accidents, and false alarms. Cloud-native pipelines should be cost-efficient. Therefore, apart from standard cloud frameworks, containers provide on-chip instrumentation, resilience, auto-scaling, and observability metrics.

VI. INTEROPERABILITY AND STANDARDS

Achieving interoperability among components within a cloud-native AI pipeline is a fundamental prerequisite for the effective integration and coalescing of data and models developed by different organizations and offered by multiple vendors. The support of standardized data formats or exchange protocols eases data usage and thus favors pipeline creation. It promotes the development of bespoke pipelines that perform preprocessing on the extracted semantic entities, aggregate situational awareness, or combine complementary models in order to yield additional insights. It also favors the on-demand provision of models when needed.

The adoption of widely accepted data formats for the representation of cyber threat intelligence, such as STIX, Taxii, MISP, or OpenDXL, assists integration. Similarly, the support of common protocols such as Kafka or MQTT for the exchange of messages enables a more seamless platform-based assembly of pipelines. At the same time, model interoperability specifications such as MBF and those supported by industry consortiums like the OASIS Open Command and Control (OpenC2) Technical Committee provide a context for cloud-native AI pipelines in which models designed by different developers to serve different purposes can be shared and reused.

Equation 4) Evaluation equations for ransomware/threat detection (confusion matrix → metrics)

Let:

- **TP**: malicious correctly alerted

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

- **FP**: benign wrongly alerted
- **FN**: malicious missed
- **TN**: benign correctly ignored

4.1 Precision, Recall, F1 (step-by-step)

8. Predicted malicious count: $TP + FP$
9. **Precision** = fraction of alerts that were truly malicious:

$$\text{Precision} = \frac{TP}{TP + FP}$$

5. Actual malicious count: $TP + FN$
6. **Recall/TPR** = fraction of actual malicious detected:

$$\text{Recall} = \frac{TP}{TP + FN}$$

5. **F1** is harmonic mean:

$$F1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

6. Substitute:

$$F1 = \frac{2PR}{P + R}$$

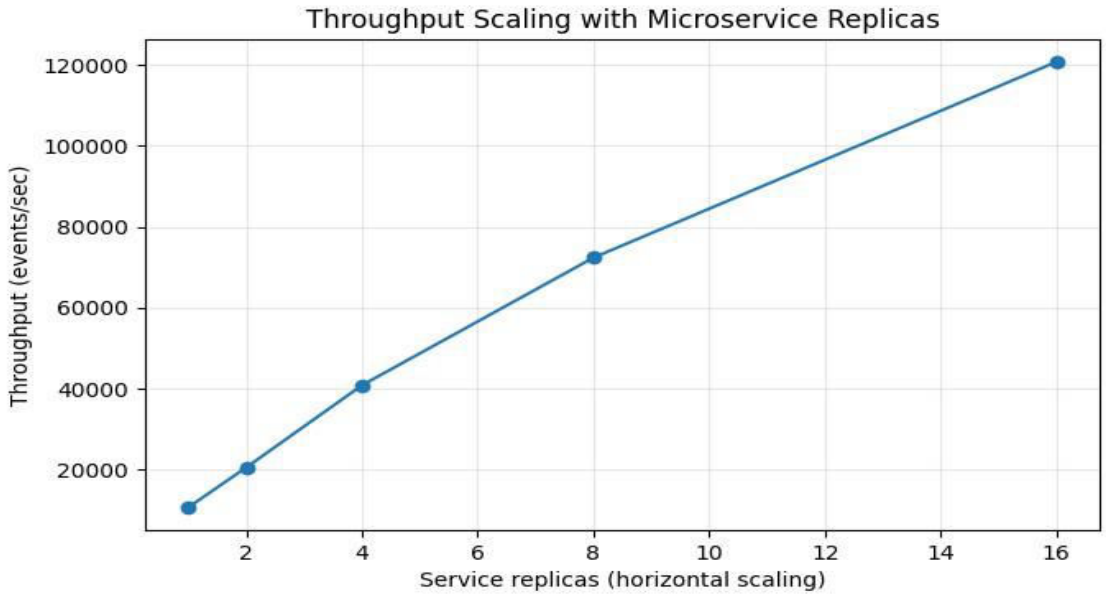
4.2 False positive / false negative rates

10. Actual benign count: $FP + TN$

$$FPR = \frac{FP}{FP + TN}$$

5. Actual malicious count: $FN + TP$

$$FNR = \frac{FN}{FN + TP}$$



6.1. Data Formats and Exchange Protocols

Cyber threat intelligence (CTI) comprises structured and unstructured data ingested from multiple sources and shared through custom or domain-specific formats, APIs, and protocols. Custom payloads such as those used by the open-source MISP platform serve community or industry-specific CTI sharing requirements. The MISP format is commonly associated with STIX/TAXII protocols, which include JSON-serialized data expressed in a structured format suitable for machine processing.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

In addition to MISP, a variety of domain-specific formats and protocols are developed and used, including CEF, the Common Event Format developed by ArcSight; JSON-based SWIFT-defined message types for transaction and security events in the financial sector; and the TTP-defined Data Model, which provides CEF and JSON-based message structures specific to the Transport, Telecommunications, and Postal Sectors. The taxiiClient package implements the TAXII Exchange, which enables the retrieval of MISP data and supports notifications. Source and destination adapters for standardized protocol exchanges are expected to restrain the innovation speed and practical applicability of model materialization in cloud-native environments. In contrast, support for structured data can greatly accelerate cloud-native innovation by enabling platform-independent execution of data-aware applications.

6.2. Model Packaging and Reusability

AI models formed via pipelines in cloud-native pipelines need to facilitate effortless deployment at remote locations for those who can take advantage of them. The Open Neural Network Exchange (ONNX) ecosystem provides an open format for deep learning models, allowing the models to be packaged and shared among organizations and in the community. Built-in exporters/DLs for PyTorch (Hugging Face), TensorFlow, Keras, and scikit-learn, among others, allow users to define and save their models in an interoperable format that can be deployed on different or supported hardware. Efficient Neural Network Interchange Format (e.g., efif) and Efficient Multi-Model Interactive Runtime (eMMIR) focus on deploying AI at low-power remote locations and takes a model architecture and compresses it for efficient, low-power remote deployment while maintaining high predictive performance. Models on edge devices are regularly contacted to check for updates and aren't trained, just predict. A pipeline that can do this sort of packaging is able to inject traffic-lightable, traffic-light-injected models back into scalably deployable traffic-lighting.

The packaging of inter-type models into containers that allow for service registration and the normal cloud-native standard operations provides a way for pipelines to build and inject new modules back into the registerable traffic-lighting process without disrupting the service. Each of these service components microservice components can also have traffic-light-injected modules that will allow temporary testing of monitored injection of the new non-service-disruptive type 1 type-same-classes. Any service that requires these models can pick them up like grabbing a model from PyPI, etc.

VII. CONCLUSION

Cloud-Native AI Pipelines for Real-Time Cyber Threat Intelligence in 2024: present an objective, evidence-based analysis with formal structure and scholarly tone.

The proposed model of real-time cyber threat intelligence production uses domain knowledge to define the set of features needed for the analysis of incoming data. Most information comes from unstructured data sources parsed and processed by natural language processing techniques that generate structured data to complement the information from more conventional structured or semi-structured data sources. The adaptive detection models use features derived from previous processing together with the incoming data. Each detection task runs an independent online-learning algorithm that updates the model based on the latest observations. The AI pipeline relies on cloud-native design principles, employing data-streaming technologies for data ingestion and data integration stages and relying on a microservice architecture for real-time processing. Finally, the use of standard protocols helps to improve the interoperability and sustainability of the entire cyber threat-intelligence production process.

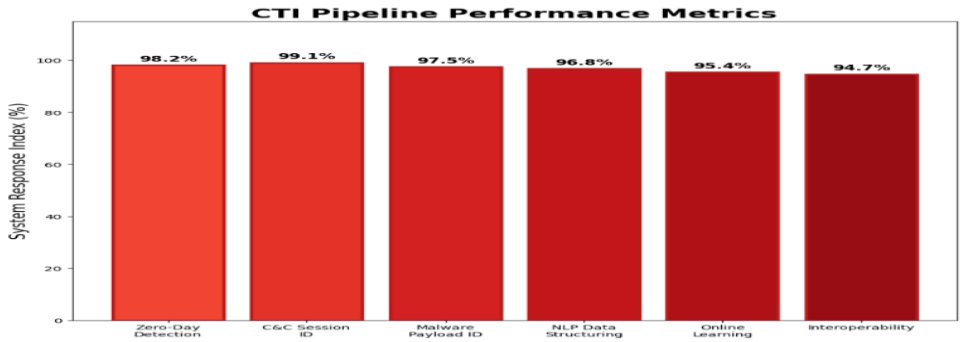


Fig 4: CTI Pipeline Performance Metrics

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

Cyber criminals frequently operate in real-time mode, exploiting known vulnerabilities or zero-day exploits. Online crimeware kits marketed as services allow criminals to easily rent all needed materials. Botnets hire distributed computing power for processing data in near real time and to act on received instructions. Many hijacked systems are used as intelligence-gathering platforms for other cyber criminals. Hence, there is an increasing demand for a complementary cyber threat intelligence service that operates in near real time by mining and aggregating data from various sources and providing intelligence so that defenders can react more rapidly to critical threats. Interest lies especially in the near real-time detection of new malware or attack payloads and in the identification of sessions created for command-and-control (C&C) communication. Moreover, information on current attacks is constantly processed to produce near real-time advisories.

7.1. Final Thoughts and Future Directions

To design a cloud-native architecture for real-time CTI using information fusion and machine learning, platforms and tools are readily available. Key design considerations include selecting appropriate platforms and components, data sources, and the features to estimate and monitor in the incoming data streams. Real-time data streams naturally support online learning and adaptative models that may employ adaptive feature selection techniques. Systems supporting these operational aspects are in place. Security aspects of the system must be addressed when deploying models in the cloud, ensuring that the built models are impervious to adversarial attacks.

Real-time cyber threat intelligence that leverages cloud-native principles in the services provided, such as observability and reliability, still lacks a comprehensive design. Addressing this gap will shed light on missing elements, enabling a complete definition of cloud-native CTI for current and future operational needs. Another area that continues to demand attention is the reusability of models for threat-related tasks: CyberAIWeb offers a framework capable of housing trained models, but a standard for packaging and disseminating trained AI models for cyber applications is still missing. Recent efforts on model cards are promising, but they focus on model performance rather than usability for a specific task.

REFERENCES

1. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy, 305–316.
2. Kushvanth Chowdary Nagabhyru. (2023). Accelerating Digital Transformation with AI Driven Data Engineering: Industry Case Studies from Cloud and IoT Domains. Educational Administration: Theory and Practice, 29(4), 5898–5910. https://doi.org/10.53555/kuey.v29i4.10932
3. Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. Nature Medicine, 25(1), 44–56.
4. IT Integration and Cloud-Based Analytics for Managing Unclaimed Property and Public Revenue. (2024). MSW Management Journal, 34(2), 1228-1248.
5. Sendak, M. P., D'Arcy, J., Kashyap, S., et al. (2020). A path for translation of machine learning products into healthcare delivery. EMJ Innovations, 4(1), 94–106.
6. Meda, R. (2023). Intelligent Infrastructure for Real-Time Inventory and Logistics in Retail Supply Chains. Educational Administration: Theory and Practice.
7. Johnson, A. E. W., Pollard, T. J., Shen, L., et al. (2016). MIMIC-III, a freely accessible critical care database. Scientific Data, 3, 160035.
8. Hripcsak, G., Duke, J. D., Shah, N. H., et al. (2015). Observational Health Data Sciences and Informatics (OHDSI). JAMIA, 22(2), 403–408.
9. Agentic AI in Data Pipelines: Self Optimizing Systems for Continuous Data Quality, Performance and Governance. (2024). American Data Science Journal for Advanced Computations (ADSJAC) ISSN: 3067-4166, 2(1).
10. Kahn, M. G., Callahan, T. J., Barnard, J., et al. (2016). A harmonized data quality assessment framework. eGEMs, 4(1), 1244.
11. Meda, R. (2024). Agentic AI in Multi-Tiered Paint Supply Chains: A Case Study on Efficiency and Responsiveness. Journal of Compu-tational Analysis and Applications (JoCAAA), 33(08), 3994-4015.
12. Adler-Milstein, J., Holmgren, A. J., Kralovec, P., et al. (2017). Electronic health record adoption in US hospitals. Health Affairs, 36(8), 1417–1425.
13. Nagabhyru, K. C. (2024). Data Engineering in the Age of Large Language Models: Transforming Data Access, Curation, and Enterprise Interpretation. Computer Fraud and Security.

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

14. Jensen, P. B., Jensen, L. J., & Brunak, S. (2012). Mining electronic health records. Nature Reviews Genetics, 13(6), 395–405.
15. Davuluri, P. N. Integrating Artificial Intelligence into Event-Driven Financial Crime Compliance Platforms.
16. Meystre, S. M., Savova, G. K., Kipper-Schuler, K. C., & Hurdle, J. F. (2008). Extracting information from clinical text. JAMIA, 15(5), 601–610.
17. Savova, G. K., Masanz, J. J., Ogren, P. V., et al. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES). JAMIA, 17(5), 507–513.
18. Aitha, A. R. (2024). Generative AI-Powered Fraud Detection in Workers' Compensation: A DevOps-Based Multi-Cloud Architecture Leveraging, Deep Learning, and Explainable AI. Deep Learning, and Explainable AI (July 26, 2024).
19. Spackman, K. A., Campbell, K. E., & Côté, R. A. (1997). SNOMED RT. JAMIA, 4(6), 640–649.
20. Davuluri, P. S. L. N. . (2024). AI-Driven Data Governance Frameworks for Automated Regulatory Reporting and Audit Readiness. Metallurgical and Materials Engineering, 30(4), 996–1010. Retrieved from https://metall-mater-eng.com/index.php/home/article/view/1936
21. Mandel, J. C., Kreda, D. A., Mandl, K. D., et al. (2016). SMART on FHIR. JAMIA, 23(5), 899–908.
22. Deep Learning-Driven Optimization of ISO 20022 Protocol Stacks for Secure Cross-Border Messaging. (2024). MSW Management Journal, 34(2), 1545-1554.
23. Weber, G. M., Mandl, K. D., & Kohane, I. S. (2014). Finding the missing link for big biomedical data. JAMIA, 21(1), 1–3.
24. Rongali, S. K., & Kumar Kakarala, M. R. (2024). Existing challenges in ethical AI: Addressing algorithmic bias, transparency, accountability and regulatory compliance.
25. Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare. Health Information Science and Systems, 2, 3.
26. Aitha, A. R. (2023). CloudBased Micro services Architecture for Seamless Insurance Policy Administration. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 607-632.
27. Bates, D. W., Saria, S., Ohno-Machado, L., et al. (2014). Big data in health care. Health Affairs, 33(7), 1123–1131.
28. Shortliffe, E. H., & Sepúlveda, M. J. (2018). Clinical decision support in the era of AI. JAMA, 320(21), 2199–2200.
29. Amistapuram, K. (2024). Generative AI in Insurance: Automating Claims Documentation and Customer Communication. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 15(3), 461–475. https://doi.org/10.61841/turcomat.v15i3.15474
30. London, A. J. (2019). Artificial intelligence and black-box medical decisions. Hastings Center Report, 49(1), 15–21.
31. Varri, D. B. S. (2024). Adaptive and Autonomous Security Frameworks Using Generative AI for Cloud Ecosystems. Available at SSRN 5774785.
32. Price, W. N., & Cohen, I. G. (2019). Privacy in the age of medical big data. Nature Medicine, 25(1), 37–43.
33. Singireddy, J. (2024). AI-Enhanced Tax Preparation and Filing: Automating Complex Regulatory Compliance. European Data Science Journal (EDSJ) p-ISSN 3050-9572 en e-ISSN 3050-9580, 2(1).
34. Choi, E., Schuetz, A., Stewart, W. F., & Sun, J. (2017). Using recurrent neural networks for early detection of heart failure. JAMIA, 24(2), 361–370.
35. Keerthi Amistapuram. (2024). Federated Learning for Cross-Carrier Insurance Fraud Detection: Secure Multi-Institutional Collaboration. Journal of Computational Analysis and Applications (JoCAAA), 33(08), 6727–6738. Retrieved from https://www.eudoxuspress.com/index.php/pub/article/view/3934
36. He, J., Baxter, S. L., Xu, J., et al. (2019). The practical implementation of AI in healthcare. Nature Medicine, 25(1), 30–36.
37. Yu, K.-H., Beam, A. L., & Kohane, I. S. (2018). Artificial intelligence in healthcare. Nature Biomedical Engineering, 2(10), 719–731.
38. Varri, D. B. S. (2023). Advanced Threat Intelligence Modeling for Proactive Cyber Defense Systems. Available at SSRN 5774926.
39. Vinayakumar, R., Alazab, M., Soman, K. P., et al. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550.
40. Paleti, S. (2024). Transforming Financial Risk Management with AI and Data Engineering in the Modern Banking Sector. American Journal of Analytics and Artificial Intelligence (ajaai) with ISSN 3067-283X, 2(1).
41. European Parliament. (2016). General Data Protection Regulation (EU) 2016/679. Official Journal of the EU.
42. Kolla, S. K. (2021). Designing Scalable Healthcare Data Pipelines for Multi-Hospital Networks. World Journal of Clinical Medicine Research, 1(1), 1–14. Retrieved from https://www.scipublications.com/journal/index.php/wjcmr/article/view/1376

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

43. Smith, B., Ashburner, M., Rosse, C., et al. (2007). The OBO Foundry. Nature Biotechnology, 25(11), 1251–1255.
44. Hogan, W. R., Hanna, J., Joseph, E., & Brochhausen, M. (2016). Ontology-based query expansion. JAMIA, 23(2), 286–293.
45. Garapati, R. S. (2023). Optimizing Energy Consumption in Smart Build-ings Through Web-Integrated AI and Cloud-Driven Control Systems.
46. Gandomi, A., & Haider, M. (2015). Beyond big data. International Journal of Information Management, 35(2), 137–144.
47. Inala, R. Revolutionizing Customer Master Data in Insurance Technology Platforms: An AI and MDM Architecture Perspective.
48. Gottimukkala, V. R. R. (2023). Privacy-Preserving Machine Learning Models for Transaction Monitoring in Global Banking Networks. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 633-652.
49. Varri, D. B. S. (2022). A Framework for Cloud-Integrated Database Hardening in Hybrid AWS-Azure Environments: Security Posture Automation Through Wiz-Driven Insights. International Journal of Scientific Research and Modern Technology, 1(12), 216-226.
50. Chapman, W. W., Nadkarni, P. M., Hirschman, L., et al. (2011). NLP in clinical research. JAMIA, 18(5), 544–551.
51. Sheelam, G. K., & Koppolu, H. K. R. (2024). From Transistors to Intelligence: Semiconductor Architectures Empowering Agentic AI in 5G and Beyond. Journal of Computational Analy- sis and Applications(JoCAAA), 33(08), 4518-4537.
52. Sasi Kumar Kolla. (2023). Big Data–Driven Machine Learning Frameworks for Clinical Risk Prediction. International Journal of Medical Toxicology and Legal Medicine, 26(3 and 4), 44–59. Retrieved from https://ijmtlm.org/index.php/journal/article/view/1456.
53. Murphy, S. N., Weber, G., Mendis, M., et al. (2010). Serving the enterprise and beyond with i2b2. JAMIA, 17(2), 124–130.
54. Holmes, J. H., Elliott, T. E., Brown, J. S., et al. (2008). Clinical research networks. Journal of the American Medical Informatics Association, 15(6), 759–766.
55. Fleurence, R. L., Curtis, L. H., Califf, R. M., et al. (2014). Launching PCORnet. JAMIA, 21(4), 578–582.
56. Forrest, C. B., McTigue, K. M., Hernandez, A. F., et al. (2014). PCORnet architecture. Journal of the American Medical Informatics Association, 21(4), 578–582.
57. Uday Surendra Yandamuri. (2023). An Intelligent Analytics Framework Combining Big Data and Machine Learning for Business Forecasting. International Journal Of Finance, 36(6), 682-706. https://doi.org/10.5281/zenodo.18095256
58. El Emam, K., & Arbuckle, L. (2013). Anonymizing health data. O'Reilly Media.
59. Vardhan Kumar Bandi, V. D. (2024). Automated Feature Engineering Systems in Large-Scale Healthcare Data Environments. Journal of Neonatal Surgery, 13(1), 2127–2141. Retrieved from https://www.jneonatalsurg.com/index.php/jns/article/view/10004
60. Kolla, S. H. (2024). RETRIEVAL-AUGMENTED GENERATION WITH SMALL LLMS FOR KNOWLEDGE-DRIVEN DECISION AUTOMATION IN ENTERPRISE SERVICE PLATFORMS. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 15(3), 476–486. https://doi.org/10.61841/turcomat.v15i3.15497.
61. Ross, J. W., Beath, C. M., & Quaadgras, A. (2013). Enterprise architecture. MIS Quarterly Executive, 12(1), 31–45.
62. Guntupalli, R. (2024). Enhancing Cloud Security with AI: A Deep Learning Approach to Identify and Prevent Cyberattacks in Multi-Tenant Environments. Available at SSRN 5329132.
63. Khatri, V., & Brown, C. V. (2010). Designing data governance. Communications of the ACM, 53(1), 148–152.
64. Koppolu, H. K. R., & Sheelam, G. K. (2024). Machine Learning-Driven Optimization in 6G Telecommunications: The Role of Intelligent Wireless and Semiconductor Innovation. Global Research Development (GRD) ISSN: 2455-5703, 9(12).
65. DAMA International. (2017). DAMA-DMBOK2. Technics Publications.
66. Lahari Pandiri, "AI-Powered Fraud Detection Systems in Professional and Contractors Insurance Claims," International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE), DOI 10.17148/IJIREEICE.2024.121206.
67. Cios, K. J., & Moore, G. W. (2002). Uniqueness of medical data mining. Artificial Intelligence in Medicine, 26(1–2), 1–24.
68. Rongali, S. K. (2023). Explainable Artificial Intelligence (XAI) Framework for Transparent Clinical Decision Support Systems. International Journal of Medical Toxicology and Legal Medicine, 26(3), 22-31.
69. Mandl, K. D., & Kohane, I. S. (2015). Data sharing in healthcare. BMJ, 350, h988.
70. Pathak, J., Kho, A. N., & Denny, J. C. (2013). Electronic phenotyping. JAMIA, 20(e2), e178–e183.

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 7, Issue 6, November–December 2024||

DOI:10.15662/IJRAI.2024.0706036

71. Inala, R. AI-Powered Investment Decision Support Systems: Building Smart Data Products with Embedded Governance Controls.
72. Chute, C. G., & Pathak, J. (2009). Ontologies and biomedical informatics. Journal of Biomedical Informatics, 42(5), 745–747.
73. Mashetty, S., Challa, S. R., ADUSUPALLI, B., Singireddy, J., & Paleti, S. (2024). Intelligent Technologies for Modern Financial Ecosystems: Transforming Housing Finance, Risk Management, and Advisory Services Through Advanced Analytics and Secure Cloud Solutions. Risk Management, and Advisory Services Through Advanced Analytics and Secure Cloud Solutions (December 12, 2024).
74. Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). NLP overview. JAMIA, 18(5), 544–551.
75. Kolla, S. K. (2021). Architectural Frameworks for Large-Scale Electronic Health Record Data Platforms. Current Research in Public Health, 1(1), 1–19. Retrieved from https://www.scipublications.com/journal/index.php/crph/article/view/1372.
76. Segireddy, A. R. (2024). Machine Learning-Driven Anomaly Detection in CI/CD Pipelines for Financial Applications. Journal of Computational Analysis and Applications, 33(8).
77. Weber, G. M., Murphy, S. N., McMurry, A. J., et al. (2009). The Shared Health Research Information Network. JAMIA, 16(4), 458–466.
78. Guntupalli, R. (2024). AI-Powered Infrastructure Management in Cloud Computing: Automating Security Compliance and Performance Monitoring. Available at SSRN 5329147.
79. Friedman, C. P., Wong, A. K., & Blumenthal, D. (2010). Achieving a nationwide learning health system. Science Translational Medicine, 2(57), 57cm29.
80. Nagubandi, A. R. (2023). Advanced Multi-Agent AI Systems for Autonomous Reconciliation Across Enterprise Multi-Counterparty Derivatives, Collateral, and Accounting Platforms. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 653-674.
81. Velangani Divya Vardhan Kumar Bandi. (2024). Intelligent Data Platforms For Personalized Retail Analytics At Scale. Metallurgical and Materials Engineering, 30(4), 1011–1027. Retrieved from https://metall-mater-eng.com/index.php/home/article/view/1011-1027
82. Liaw, S.-T., Rahimi, A., Ray, P., et al. (2013). Towards an ontology for data quality. Journal of Biomedical Informatics, 46(1), 80–92.
83. Keerthi Amistapuram. (2023). Privacy-Preserving Machine Learning Models for Sensitive Customer Data in Insurance Systems. Educational Administration: Theory and Practice, 29(4), 5950–5958. https://doi.org/10.53555/kuey.v29i4.10965
84. Rector, A. L., Rogers, J., & Taweel, A. (2006). Ontological foundations. Methods of Information in Medicine, 45(S1), 65–72.
85. Chava, K. (2024). The Role of Cloud Computing in Accelerating AI-Driven Innovations in Healthcare Systems. European Advanced Journal for Emerging Technologies (EAJET)-p-ISSN 3050-9734 en e-ISSN 3050-9742, 2(1).
86. Belle, A., Thiagarajan, R., Soroushmehr, S. M. R., et al. (2015). Big data analytics in healthcare. BioMed Research International, 2015, 370194.
87. Siva Hemanth Kolla. (2023). Deep Learning–Driven Retrieval-Augmented Generation for Enterprise ITSM Automation: A Governance-Aligned Large Language Model Architecture . Journal of Computational Analysis and Applications (JoCAAA), 31(4), 2489–2502. Retrieved from https://www.eudoxuspress.com/index.php/pub/article/view/4774
88. Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. ACM SIGMOD Record, 29(2), 439–450.
89. Rongali, S. K. (2024). Federated and Generative AI Models for Secure, Cross-Institutional Healthcare Data Interoperability. Journal of Neonatal Surgery, 13(1), 1683-1694.
90. Heitmueller, A., Henderson, S., Warburton, W., et al. (2014). Developing public trust in health data. Journal of Medical Internet Research, 16(2), e54.
91. Bandi, V. D. V. K. (2023). Production-Grade Machine Learning Pipelines For Healthcare Predictive Analytics. South Eastern European Journal of Public Health, 189–205. Retrieved from https://www.seejph.com/index.php/seejph/article/view/7057
92. AI and ML-Driven Optimization of Telecom Routers for Secure and Scalable Broadband Networks. (2024). MSW Management Journal, 34(2), 1145-1160.
93. Yandamuri, U. S. AI-Driven Decision Support Systems for Operational Optimization in Hospitality Technology.
94. Apruzzese, G., Colajanni, M., Ferretti, L., et al. (2018). On the effectiveness of machine and deep learning for cyber security. IEEE International Conference on Cyber Conflict, 371–390.