EZTSM: Enhanced Zero Trust Security Model for Serverless Computing with ML based
Anomaly Detection

MSc Research Project
Cloud Computing

Deepak Venkatesh Babu
Student ID: X23267101

School of Computing
National College of Ireland

Supervisor: Prof. Aqeel Kazmi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Deepak Venkatesh Babu		
Student ID:	X23267101		
Program:	Cloud Computing	**Year:**	2024
Module:	MSc Research Project		
Supervisor:	Prof. Aqeel Kazmi		
Submission Due Date:	15/09/2025		
Project Title:	EZTSM: Enhanced Zero Trust Security Model for Serverless Computing with ML based Anomaly Detection.		
Word Count:	6863		
Page Count:	20		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Deepak Venkatesh Babu

Date: 15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Program Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

EZTSM: Enhanced Zero Trust Security Model for Serverless Computing with ML based anomaly detection.

Deepak Venkatesh Babu
X23267101

Abstract

Enhanced Zero Trust Security Model (EZTSM) specifically designed for serverless computing is the proposed solution, that combines machine learning based anomaly detection using the Isolation Forest algorithm and Zero Trust Security framework along with real-time monitoring. The traditional perimeter-based security models are not sufficient to protect the modern cloud-native architectures due to its limitations such as static role-based access control, particularly serverless platforms that operates on dynamic, event-driven functions. This study addresses these limitations by designing, developing, implementing and evaluating a scalable and performance-oriented security framework for serverless on AWS cloud. While the existing Zero Trust framework remains theoretical and lacks proven evidence. This research also evaluates EZTSM with serverless computing, focusing on latency, throughput, anomaly detection accuracy, scalability, and response to simulated attacks. The model is built using AWS-native services enables easy setup and implementation. The results show how EZTSM can enhance security posture while maintaining acceptable performance trade-offs in dynamic cloud environments.

1 Introduction

Cloud Computing has become a revolutionary technology for developing and deploying applications. The most innovative paradigm in the cloud is serverless computing, through serverless computing developers can directly execute their code as a function without worrying about maintaining their infrastructure, instead they can focus more on developing and executing it. Multiple cloud providers offer serverless computing service is different name like AWS Lambda, Azure Functions, and Google Cloud Functions, all these services are prepopulated with auto scaling, cost efficient as it offers pay-as-you-use billing, best suit for microservices and event driven architecture. Even though serverless computing has many benefits, security remains the major concerns because of it decentralized architecture, ephemeral nature and complex service integrations (Li et al., 2023).

In the usual cloud model, the virtual machine and containers are isolated and offer more control, but the serverless computing works in the principle of shared resources, multi tenancy and event driven. The serverless functions are always stateless, this nature always led way for new attacks like injection attacks, privilege escalations, and insecure functions between functions (Bhatt et al., 2024). These security risks cannot be avoided by the tractional security model which highly rely on static access policies and perimeter-based trust policy. To address this security concerns, the Zero Trust Security Model (ZTSM) was introduced as a transformative security approach. ZTSM works based on the principle of

“Never trust, always verify” which enables continuous authentication, strict access control policies, and verification of each request regardless of origin (Kang et al., 2023). ZTSM already explored theoretically in containerized and cloud-native ecosystems, but the practical application in the serverless environment remains unexplored.

Instead of relying on the static access policies or blacklists, in EZTSM anomaly detection algorithm will analyze the behavior of the system, where it will identify the suspicious activities such as unusual invocation and requests or unauthorized access attempts. Integrating such models in the cloud-based security will provide a satisfying result because of its fast, scalable, unsupervised and lightweight nature.

Additionally, to make this Enhanced Zero Trust Security Model proactive rather than reactive real-time monitoring feature will be integrated with this model. Even though real-time monitoring has been successfully applied in many systems but integrating it advanced security model like ZTSM particularly for serverless platforms remains unexplored and still in developing area of research.

This research proposes advanced security model called Enhanced Zero Trust Security Model (EZTSM), specifically for serverless computing. This security model integrates with existing ZTSM features like continuous authentication and least privilege access policies along with the machine learning based anomaly detection which prevents performance from getting affected. The expected contribution of this research is to offer a deployable and scalable security model along with the ML based anomaly detection for secure serverless applications using AWS cloud native solution. These findings will serve as both an academic contribution and a practical reference for cloud architects and security engineers seeking to modernize their security posture in highly dynamic, distributed environments.

1.1 Research Objective

This work is carried out by the following central research question:

“How can the existing security limitation on serverless computing be addressed by designed and implemented Enhanced Zero Trust Security Model (EZTSM) along with ML based anomaly detection techniques?”

To address this research question, the research is structured around the following specific objectives:

- Investigate the advantage and limitations in Zero Trust Security Models, anomaly detection methods, and their application in serverless cloud environments.

- Design a cloud-native architecture integrating AWS services and an Isolation Forest-based anomaly detection engine for real-time access control and workload.

- Develop a synthetically generated dataset that exactly reflects realistic AWS Lambda invocation patterns and train a machine learning model to distinguish between legitimate and anomalous requests.

- Deploy the trained model as a containerized service within Amazon ECS to make it fully scalable and integrate it with a Gatekeeper Lambda function for dynamic decision-making.
- Evaluate the proposed EZTSM in terms of detection accuracy, false positive rate, and real-time performance overhead.

1.2 Structure of the Report

The remainder of this report is organized as follows:

- Section 2 provides a comprehensive review of previous related work and its research gap under subsections security challenges in serverless computing, Zero Trust Security Models, and anomaly detection techniques such as Isolation Forest.
- Section 3 shows the research methodology, explaining the EZTSM framework, AWS component usage, and anomaly detection model algorithm selection.
- Section 4 describes the design specifications, including system architecture layers, AWS service configuration, and data flow.
- Section 5 provides the implementation details, from AWS services implementation , model development to ECS deployment and Gatekeeper Lambda integration.
- Section 6 presents the evaluation process and results, discussing metrics such as detection accuracy, confusion matrix analysis, and system performance.
- Section 7 concludes the report, summarizing key findings and proposing future research directions.

2 Related Work

Serverless computing has been totally changed how everyone thinks about cloud architecture. It has become more popular and incredibly powerful and easy to use. It allows developers to build highly scalable, event-driven applications without needing to worry about managing and maintaining servers. Even though this change looks convenient and powerful, it also has many issues. Because these serverless functions are short-lived and stateless, to secure and properly monitor them is a real challenge. Also, the serverless computing follows shared tenancy, where the resources are not isolated.

Regardless of its advantages there are also many challenges in serverless paradigm, recent research has continuously focused on practical implementation of Zero Trust Security Models and anomaly detection techniques. Among these, the Isolation Forest (iForest) algorithm has been identified for its ability to detect anomalous behavior in real time, offering relevance to dynamic, cloud-native, and serverless computing environments.

2.1 Security Challenges in Serverless Computing

This section covers the previous related work on the major security challenges in serverless computing. Li et al. (2023) in their study explored the key security issues in the serverless computing like shorter lifecycles span, limited resource isolation and less visibility on function chains. These challenges cannot be addressed by traditional security frameworks because of its limitations. Bhatt et al. (2024) presented list of more specific security risks such as injection attacks, improperly configured APIs and poor role management, highlighting that serverless environments demand dynamic, runtime-aware security measures.

Marin et al. (2022) presented a threat-based analysis, which shows the limitations of traditional security frameworks like role-based access policies, firewalls for stateless, event-driven functions. Serverless is basically decentralized, depends on third-partly APIs and microservices which require strong security for real-time function processing. Overall, these works show the insufficiency of static and perimeter-based access management, need for integrating dynamic, adaptive and behavior-driven security mechanisms.

Further contributions by Barrak et al. (2024), deeply explored about AWS Lambda vulnerabilities, throws light serverless security issues particularly for AWS Lambda. The study highlighted key challenges like event injection, role misconfigurations, and insecure dependencies, recommending security best practices such as function-level isolation, dynamic access policies, and runtime monitoring. These approaches proposed the need for integrating layered and dynamic security frameworks that address the transient and distributed characteristics of serverless functions.

Marappan et al. (2025) in the study explored the usage of AWS Lambda in healthcare related machine learning deployments. It highlights the importance of secure function orchestration, least privilege access controls, and data lifecycle governance, context-aware security mechanisms in serverless ecosystems.

2.2 Zero Trust Security Model in Cloud and Serverless Contexts

The security challenges in the serverless computing paradigm, gave a path to new security framework called Zero Trust Security Model (ZTSM), follows security practices that align well with serverless computing. Kang et al. (2023) in his study explores the key concepts of ZTSM, such as adaptive authentication and dynamic policy enforcement conceptually. The study is with more theoretical concepts, this study lacks practical implementation, especially in serverless scenarios with real-time workloads. Mehraj and Banday (2020) says that the trust is both subjective and objective theoretically. Their model emphasizes dynamic trust evaluations and advocates for micro-segmentation, continuous verification, and user-behavior analytics. However, the study is more conceptual and lacks implementation evidence.

Ni et al. (2024) attempted to quantify the security implications of the Zero Trust Security Model (ZTSM), focusing largely on theoretical performance trade-offs such as latency and processing overhead. To bridge this gap, the current research proposes an Enhanced Zero Trust Security Model (EZTSM) that combines machine learning based anomaly detection and real-time monitoring. Nisha et al. (2023) proposed an important concept multi-layer Zero Trust Security Model tailored for modern enterprise networks. Their approach centred on dynamic permission boundaries and the continuous reassessment of both user identity and workload trustworthiness. The relevance to serverless computing lies in the fine-grained policy enforcement, which aligns with microservice and function-level access patterns.

S. R and Yelsangiker (2024) introduced a modular Zero Trust architecture built with identity verification, device trust scoring, and dynamic policy enforcement. The main objectives of the model are continuous verification and behavioral risk assessment which directly suitable to the stateless, event-driven nature of serverless environments.

Alnoaimi and Alomary (2025) in their study compared various Zero Trust maturity models. Their study classified the implementation stages across different organizations, evaluating metrics such as access policy automation, threat detection and enforcement granularity. The results highlight the importance of context-aware and scalable trust policies, reinforcing the case for adaptive ZTSM frameworks that can meet the demands of cloud-native and serverless environments.

2.3 Anomaly Detection and Isolation Forest Advancements

The proposed EZTSM combines the Zero Trust security framework with machine learning based anomaly detection, this section covers the various research done in anomaly detection and isolation forest algorithm. In context the usual anomaly detection frameworks don't provide the expected result in dynamic environment like serverless due to dimensional data, lack of labelled training data. To address these limitations, a new ml algorithm has been introduced called iForest (isolation forest), this provides the feature of unsupervised learning and works with data separation. Hagemann and Katsarou (2020), in his study explored over 200 studies in overview of anomaly detection in cloud architecture. This study says, there is a need for machine learning model to handle scaling, shared resource structure and virtualization which suits with serverless requirement.

Xu et al. (2023) in their study proposed a new variant of isolation forest called Deep Isolation Forest (DIF), this works in the principle of combining neural networking technique along with isolation forest to fix the bias in the model training and accuracy in the output. This study shows that DIF can be a better option for serverless computing. Same way, Xu et al. (2017) in their study introduces another variant called Simulated Annealing isolation forest provides feature of tree selection, generalization and minimize computational overhead.

Al-Shehari et al. (2023) contributed on the concept of algorithm application to the insider threat detection, the algorithm has an ability to handle imbalanced datasets. This application gained the result 98% accuracy on the CERT insider threat dataset, highlighting its efficiency in handling complex and imbalanced data scenarios.

Finally, Enhanced Zero Trust Security Model (EZTSM) proposed study take advantage of isolation Forest algorithm to detects anomalies and integrated with real-time access control decisions. By doing this, it comes out as a powerful security framework to enforce adaptive and dynamic security policies based on detected behavior instead of predefined roles or static rules.

3 Research Methodology

This section shows the methodological framework adopted for the design and implementation of the Enhanced Zero Trust Security Model (EZTSM) particularly in the serverless computing context. The approach combines a real-time anomaly detection engine with an access control mechanism to regulate the execution of serverless functions. For model development, the dataset was synthetically generated to replicate the structure and behavior of AWS Lambda invocation logs, allowing for controlled yet realistic testing. The trained model was then containerized and deployed within an Amazon Web Services (AWS) environment, making use of AWS-native components to ensure compatibility and scalability. The methodology is organized into five subsections, covering the fundamentals of ZTSM, the anomaly detection model, trust-based access decision processes, the data simulation strategy, and the evaluation metrics used to assess performance. These five subsections altogether prove the methodology is powerful and suitable for the enhanced security solutions for the serverless computing.

3.1 Zero Trust Security Fundamentals and EZTSM Adaptation

Zero Trust Security Model is a security framework that eliminates the constant trust granted to users and devices within a network perimeter. It basically operates on the principle of **"never trust, always verify"** requiring continuous authentication and authorization of the user identity and API request, micro segmentation and context before access is granted.

The Zero Trust model was introduced to address the growing complexity and distributed nature of modern IT environments, where traditional perimeter-based security is insufficient in preventing lateral movement by malicious actors.

The traditional security frameworks have static credentials, assuming that internal entities can be trusted once authenticated. However, this assumption has proven insufficient in the cloud-native and serverless systems, where the serverless workloads are dynamic, ephemeral, and often executed across shared multi-tenant infrastructures.

The proposed framework Enhanced Zero Trust Security Model (EZTSM) in this research extends the core Zero Trust principles by incorporating machine learning based behavioral anomaly detection into the access control process. EZTSM evaluates each invocation request using machine learning predictions derived from runtime metadata. By combining a machine learning based Isolation Forest model within the access pipeline, EZTSM enables adaptive policy enforcement based on the observed behavior of the request, rather than its origin alone.

The integration of ML based anomaly detection provides a way for dynamic, context-aware, per-request access validation that adapts to evolving threat patterns without manual rule configuration. EZTSM is particularly best suited for serverless environments, where traditional access control mechanisms are limited in their ability to interpret invocation context. By integrating anomaly detection with Zero Trust logic, EZTSM offers a scalable and intelligent framework for securing serverless workloads against both internal misuse and external attacks.

3.2 Anomaly Detection Engine (Isolation Forest)

The anomaly detection engine integrated within the Enhanced Zero Trust Security Model (EZTSM) utilizes the base implementation of the Isolation Forest algorithm available through the sklearn.ensemble module of the Scikit-learn library. Isolation Forest was chosen for its suitability in high-dimensional, unlabeled datasets particularly those exhibiting rare but critical anomalies, as typically encountered in serverless environments.

Unlike distance-based or density-based techniques, Isolation Forest algorithm operates on the principle of isolating anomalies through random partitioning. Anomalous instances, being few and different, are more likely to be isolated in fewer splits. This approach avoids the computational complexity of clustering or probability estimation, making it computationally efficient and scalable an essential requirement for real-time serverless execution contexts.

The base version of the algorithm was selected due to its maturity, interpretability, and stable integration within production-grade Python environments. The goal of this research was to maintain low latency while achieving high classification performance, for which the standard implementation was deemed sufficient.

The model was trained in batch mode using a synthetically generated dataset comprising realistic Lambda invocation records. Both technical features (e.g., execution_time_ms, memory_used_mb) and contextual features (e.g., user_agent, time_of_day, authentication_status, identity_principal) were included. A contamination rate of 0.05 was configured to assume that approximately 5% of the traffic exhibits anomalous behavior, allowing the model to learn decision boundaries accordingly.

The machine learning model training was conducted using a Jupyter Notebook environment on AWS SageMaker, after which the model was stored in the remote location and containerized using Docker. The Docker image was deployed to Amazon ECS Fargate, and the container was exposed via an Application Load Balancer (ALB) for real-time access. In production, each incoming request is pre-processed by the Gatekeeper Lambda and sent to the ECS endpoint. The model responds with a binary classification -1 indicating anomaly and 1 indicating legitimate behavior. These predictions are used to permit or block access to backend serverless functions.

3.3 Trust Evaluation and Access Control Logic

The access control strategy in this proposed EZTSM is made by anomaly detection results rather than static IAM roles. Trust evaluation of each request begins after a user is authenticated via Cognito. Behavioral attributes are collected from the request and submitted to the model, which serves as the behavioral trust engine.

An anomaly detection engine prediction of 1 indicates the request matches expected behavioral patterns and is considered as legit traffic and allowed further. Such requests are forwarded to the Business Logic Lambda for execution. Conversely, predictions labelled as -1 are treated as anomalies and access to those requests are denied. No permanent user blocks or state tracking is performed.

This real-time evaluation supports dynamic access policy enforcement and ensures least-privilege execution based on invocation context. The model acts as a behavioral firewall, adding a decisioning layer to traditional serverless workflows without requiring application rewrites.

3.4 Data Simulation and Model Integration

The trained machine learning model was evaluated using a simulation environment replicating real-world traffic patterns. A synthetic dataset was generated contains 10,000 records, with 70% representing legitimate behavior and 30% designed as anomalies. These records were constructed based on statistical distributions of mimicking real world AWS Lambda logs.

A custom Python script was used to send the test records to the deployed model endpoint. Features in each record were randomized within realistic bounds. The Gatekeeper Lambda handled the requests, forwarded them to the ALB endpoint, and logged the access decisions. This simulation validated the system's behavior under realistic load and feature distributions, confirming the feasibility of real-time detection in a serverless environment.

3.5 Evaluation Metrics and Performance Analysis

The performance of the anomaly detection engine deployed within the EZTSM framework was evaluated using standard metrics derived from the approach confusion matrix. The confusion matrix provides a comprehensive view of the model’s prediction outcomes by summarizing the number of true positives, true negatives, false positives, and false negatives. It is a preferred tool for evaluating binary classifiers, especially in imbalanced datasets where accuracy alone can be misleading.

In this evaluation of ML based anomaly detection, the confusion matrix was calculated based on a synthetically generated and simulated dataset comprising of 10,000 requests samples, of which 500 were designed to be anomalous which was 30 % of the dataset and 9,500 which was 70 % were legitimate.

The EZTSM system achieved high detection accuracy with negotiable performance impact. The containerized model responded within milliseconds, ensuring low latency which is suitable for serverless workloads. This combination of ML based anomaly detection and zero trust security framework will be best suitable option for cloud-based workloads particularly serverless.

4 Design Specification

This section explains in detail about the design specification of the Enhanced Zero Trust Security Model (EZTSM), primarily focusing on the system design architecture, tools and component-level integration, and end to end testing setup. The design combines modular, cloud-native deployment, real-time decision enforcement, and low-latency anomaly detection using Isolation Forest algorithm. This design integrates core zero trust concepts such as least-privilege, micro segmentation, continuous authentication and authorization. The architecture of this EZTSM system has been structured into three conceptual layers to support clear data flow, separation of responsibilities, and maintainability.

4.1 Architecture and workflow

The architecture of the Enhanced Zero Trust Security Model (EZTSM) is organized into a three-layered framework designed to enforce per-request, behavior-based access decisions within a cloud-native serverless environment. These layers such as Authentication, Decision, and Execution work in sequence to evaluate identity, behavioral context, and threat likelihood before allowing access to application logic.

4.1.1 Authentication Layer

The authentication layer is the first layer in the EZTSM framework, acts as the entry point for external users and services. The role of this layer continuously validates the identity of request initiators using an identity management service. In this implementation, Amazon Cognito a AWS native user authentication services are configured with a level of security attributes required for user signup and login and manage used manage user pools, handles multi-factor authentication (MFA), and issue secure JSON Web Tokens (JWTs) upon successful login.

After successful authentication of user credentials, the requests are forwarded to an API Gateway, which acts as a secured, monitored, and rate-limited interface for exposing backend services. The API Gateway verifies the authenticity of the token and ensures only valid sessions can invoke the next stage of the pipeline. This separation between identity verification and business logic forms the foundation of the Zero Trust principle like no request is trusted, regardless of its origin.

4.1.2 Decision Layer

The second layer called as decision layer acts as core of this framework, carry out a behavioral analysis into the access control process. The valid and authenticated requests from the authentication layer are directed to a specially designed Lambda function called Gatekeeper lambda. This function serves as a real-time access enforcement engine.

This Gatekeeper lambda extracts key metadata from each request, such as estimated execution time, memory usage for the request, function name, user agent, time of day, authentication status, invocation type and identity principal. These features are assembled into a structured input vector and forwarded to a containerized anomaly detection model an Isolation Forest algorithm deployed on Amazon ECS behind an Application Load Balancer (ALB).

The model performs inference and returns a binary classification indicating whether the request represents typical (legitimate) behavior or an outlier (potential anomaly). Based on the model’s decision, the Gatekeeper either forwards the request to the business logic layer or terminates it with an access denial.

4.1.3 Execution Layer

The final stage of the EZTSM architecture is the execution layer, where business specific logic is carried out deployed in another lambda function exposed using another API gateway. Only requests that pass both identity verification and behavioral screening are forwarded to the business logic Lambda function for processing.

This layer remains stateless and separated from the decision-making process. It focuses solely on completing the intended task, such as data processing, API response generation, or service invocation.

By designing the architecture into these distinct but integrated layers, EZTSM ensures that authentication, behavioral context, and dynamic risk are all considered before function execution is permitted. This approach directly aligns with the Zero Trust paradigm by eliminating implicit trust and evaluating each request in isolation.

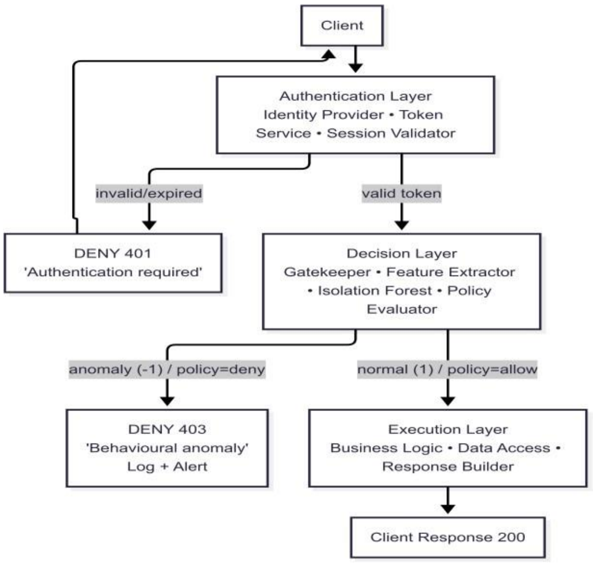

Client
Authentication Layer
Identity Provider • Token Service • Session Validator
invalid/expired
valid token
DENY 401
'Authentication required'
Decision Layer
Gatekeeper • Feature Extractor • Isolation Forest • Policy Evaluator
anomaly (-1) / policy=deny
normal (1) / policy=allow
DENY 403
'Behavioural anomaly'
Log + Alert
Execution Layer
Business Logic • Data Access • Response Builder
Client Response 200

Figure 1: Layered Request Flow in EZTSM

4.2 AWS Design Framework

The EZTSM is architected using a combination of fully managed services within the AWS ecosystem. The design emphasizes modularity, scalability, and minimal latency to support secure, real-time access control for serverless workloads. Each component within the architecture fulfils a distinct role aligned with Zero Trust principles specifically continuous verification, behavior-driven validation, and least-privilege execution.

All the user-based API request are handled via Amazon Cognito act as a entry point, which performs user identity verification and issues secure authentication tokens. These tokens passed through Amazon API Gateway, acts as an authorization layer to validate requests, enforce rate-limiting, and route traffic securely.

On successful authentication of request, it directed to the Gatekeeper Lambda Function. This function acts as the decision-making point within the architecture. It communicates with an Application Load Balancer that routes the request to a ML based anomaly detection model hosted on Amazon ECS (Fargate). The container image, trained and built using Amazon SageMaker, is stored in Amazon Elastic Container Registry (ECR) for seamless deployment.

Based on the model's prediction the request behavior is legitimate or anomalous, the Gatekeeper either allows the request to proceed to the Business Logic Lambda Function or blocks it. For real time monitoring, Amazon Cloud Watch captures and aggregates logs from all stages, enabling detailed monitoring, alerting, and audit trails.

This cloud-native design leverages AWS services in a decoupled yet integrated manner, ensuring that trust is dynamically enforced and that access to serverless resources is both secure and context-aware.

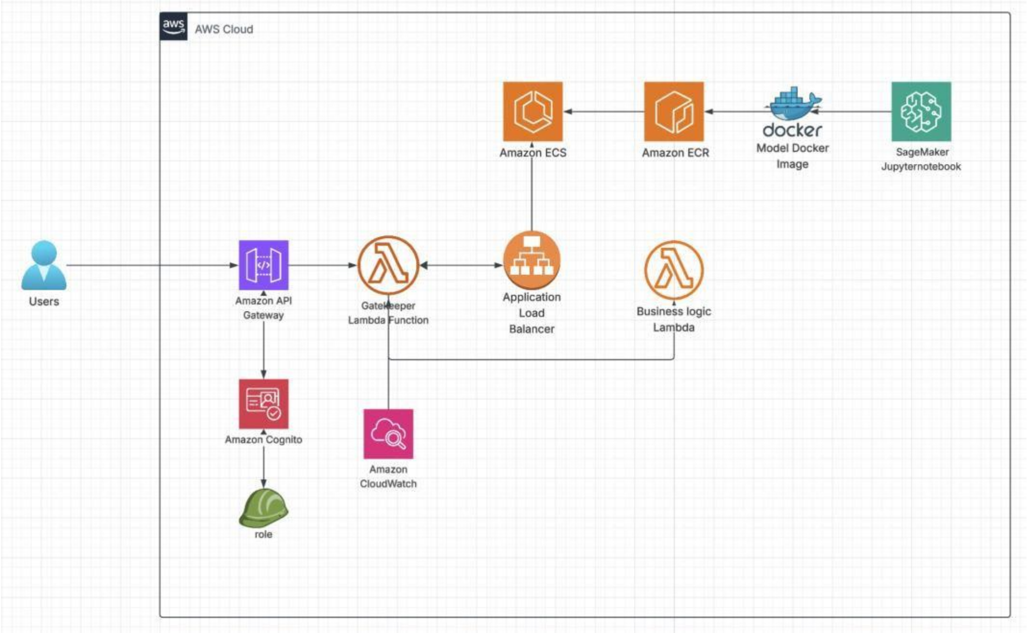


Figure 2: AWS deployment architecture of EZTSM

4.3 Data Generation and Simulation flow

To make the model more accurate and efficient, the dataset used to train the model is synthetically generated traffic using a Python-based script. The dataset consisted of 10,000 requests, with approximately 70% representing legitimate lambda invocation behavior and 30% labelled as anomalous. The requests were crafted by simulating AWS Lambda runtime metadata such as:

- function_name
- execution_time_ms
- memory_used_mb
- user_agent
- invocation_type

- time_of_day
- authentication_status
- identity_principal

The above Feature values were sampled from realistic distributions of real time system. Legitimate records were constructed using parameters within typical ranges, while anomalous records involved combinations such as high memory usage, unusual time-of-day activity, or mismatched identity contexts.

To test the model after deployment, 100 record of sample API request generated using python script were sent to model, out of which 70 legit and 30 anomalous. Based on the ML model prediction, the request was either allowed to access the business logic Lambda or access denied. This simulated workflow confirmed the effectiveness of the EZTSM in detecting anomalies from expected serverless invocation patterns without compromising the performance.

5 Implementation

This section covers the implementation of Enhanced Zero Trust Security Model (EZTSM), this includes development, deployment, and validation of the architecture, which leverages AWS-managed services to simulate and enforce Zero Trust security in a serverless computing context.

5.1 Model Training, Containerization, and Deployment

The implementation phase starts with anomaly detection model development with generation of a synthetic dataset that replicates AWS Lambda invocation patterns. This dataset was generated using a python script to include behavioral features such as execution time, memory usage, time of day, user agent, invocation type, authentication status, and identity principal. The dataset is pre-processed to remove unwanted column like IP address which is not necessary for the training. Training of the model was performed in batch process using a Jupyter Notebook hosted on Amazon SageMaker. The base version of Isolation Forest algorithm from the Scikit-learn library was used for its unsupervised anomaly detection capabilities, and the final trained model was exported in. joblib format.

The trained and exported model was then containerized as a lightweight Docker image, which includes a Flask-based REST API (app.py) for serving real-time inference requests. This container image was then pushed to Amazon Elastic Container Registry (ECR) and deployed into Amazon ECS using the Fargate launch type, allowing for serverless container orchestration. The ECS service was exposed through an Application Load Balancer (ALB) to provide a public-facing HTTP endpoint.

5.2 Integration with Gatekeeper Lambda and Inference Flow

This project is planned for the AWS native cloud solution, so the trained and deployed model is integrated with the Zero Trust setup created with AWS service. A Python-based AWS Lambda function, referred to as the Gatekeeper, with the logic of fetching required features from the logs and few simulated features like execution time and memory usage was implemented. This function is triggered via API Gateway after user authentication through Amazon Cognito. Upon invocation, the Gatekeeper extracts request metadata, constructs a feature vector, and sends it to the ECS model endpoint for scoring. Based on the model's response, the Gatekeeper determines whether to forward the request to the Business Logic Lambda or reject it.

The testing and validation of the system is done through a Python script which simulates 100 API requests. Seventy records were constructed as legitimate behavior, while the remaining thirty created as anomalies. These requests were sent through the full authentication and inference setup to evaluate decision accuracy, efficiency and access control reliability. All the logs and metrices of EZTSM system is captured by Amazon CloudWatch, AWS monitoring tool to ensure operational observability and traceability throughout the system.

Component	**Description**
Model Training	SageMaker Jupyter Notebook (Batch)
ML Algorithm	Isolation Forest (Scikit-learn)
Model Export Format	.joblib
Containerization	Docker + Flask REST API
Deployment Platform	Amazon ECS Fargate
Image Repository	Amazon ECR
Inference Endpoint	Application Load Balancer (ALB)
Gatekeeper Function	AWS Lambda (Python)
User Authentication	Amazon Cognito
API Routing	Amazon API Gateway
Logging and Monitoring	Amazon CloudWatch
Test Traffic Simulation	Python Script

Table 1: Core Components and configuration detail of EZTSM

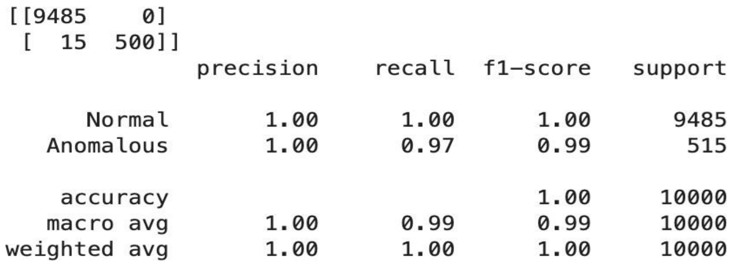
6 Evaluation

This phase clearly focuses on evaluating the effectiveness and robustness of the Enhanced Zero Trust Security Model (EZTSM) in capturing and denying the unusual and vulnerable serverless function invocations using an Isolation Forest-based anomaly detection model. This section also presents analysis of the anomaly detection model across various parameter such as accuracy, anomaly scoring patterns, and prediction behavior under simulated conditions. Both statistical metrics and visualization techniques have been used to evaluate the system capability to enforce per-request behavioral trust in a cloud-native environment.

6.1 Confusion Matrix and Classification Performance

The machine learning based anomaly detection model was evaluated using a validation set containing 10,000 records, including both legitimate and anomalous invocation behaviors.

As shown in Figure 3, the confusion matrix summarizes the classifier’s predictive performance



```
[[9485    0]
 [  15  500]]
              precision    recall  f1-score   support

      Normal       1.00      1.00      1.00      9485
   Anomalous       1.00      0.97      0.99       515

    accuracy                           1.00     10000
   macro avg       1.00      0.99      0.99     10000
weighted avg       1.00      1.00      1.00     10000
```

Figure 3: Classification metrics and confusion matrix result of the model

The above results highlight that:

- **True Positives (TP)**: 500 anomalous requests were correctly identified
- **True Negatives (TN)**: 9485 legitimate requests were accurately allowed
- **False Negatives (FN)**: 15 anomalies were misclassified as normal
- **False Positives (FP)**: 0 incorrect blocking of normal requests

Key performance metrics derived from the matrix include:

- **Precision**: 1.00 (both classes), ensuring zero false alarms
- **Recall**: 1.00 (Normal), 0.97 (Anomalous), indicating strong sensitivity
- **F1-Score**: 0.99 (Anomalous), reflecting a high balance of precision and recall
- **Overall Accuracy**: 99.85%

These metrics proves that the model is highly effective in identifying threats without sacrificing the usability or availability of legitimate services. Particularly, zero false positives ensure user experience is not degraded by unnecessary denials.

6.2 Anomaly Scoring Distribution

The underlying principle of the Isolation Forest algorithm is to assign each request an anomaly score based on how easily it can be isolated in a tree-based partition. Lower scores indicate anomalous behavior. As shown in Figure 4, the histogram of scores reveals two distinct clusters:

- A dense cluster near 0.15–0.25 (representing normal requests)
- A sparse cluster near -0.05 to 0.05 (representing anomalies)

This natural separation is critical in enabling high-confidence decisions at inference time. It also indicates that the model learned meaningful behavioral distinctions between typical and atypical invocation patterns. The clarity of the gap between clusters further supports the contamination threshold chosen during training (5%).

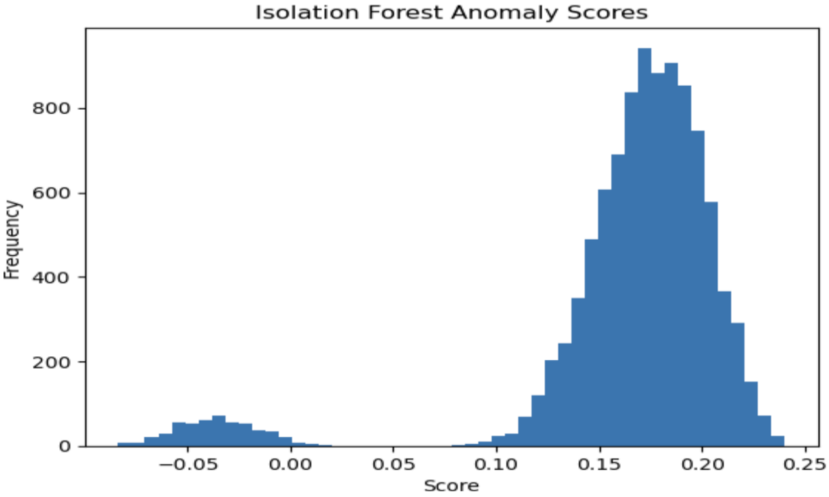


Figure 4: Isolation Forest anomaly score distribution

6.3 Behavioral Segmentation Analysis

To understand how contextual features influenced model decisions, predictions were visualized against time_of_day and execution_time_ms. As shown in Figure 5, anomalous records are predominantly located in lower execution time zones and off-peak hours (00:00–06:00), while normal traffic is distributed across standard business hours with higher and more consistent execution times.

This supports the model's sensitivity to contextual patterns often overlooked by traditional access control systems. For instance, a legitimate request arriving at an unusual time with extremely low processing duration may suggest automation misuse or bot activity. This visualization demonstrates that the EZTSM framework can effectively model such subtleties and act upon them in real time.

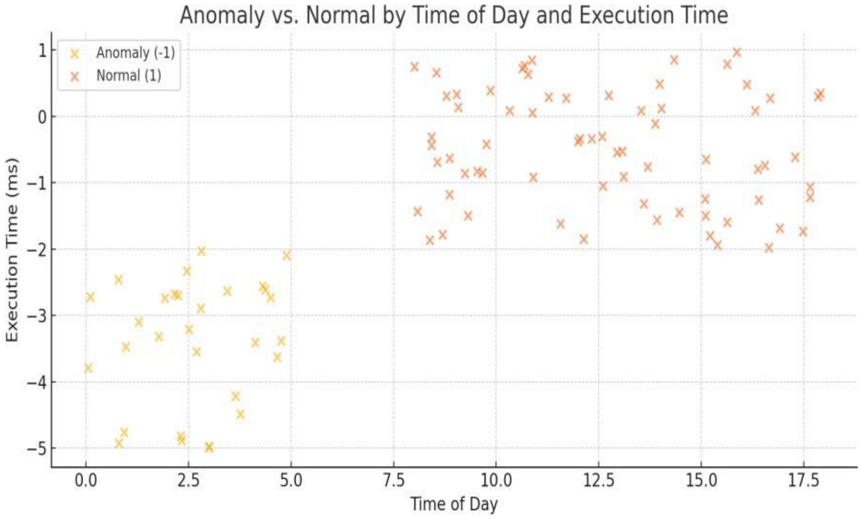


Figure 5: Anomalous vs. normal requests plotted by time of day and execution time

6.4 End-to-End Pipeline Testing

To validate the integrated system's behavior, 100 synthetic test requests were sent through the full EZTSM pipeline—from user authentication to behavioral analysis to final execution. The test set was composed of:

- **70 legitimate records** (normal behavior across all features)
- **30 anomalous records** (engineered to mimic unusual patterns)

The Gatekeeper Lambda invoked the ECS model via an ALB endpoint for each request. Based on the model's prediction, requests were either forwarded to the Business Logic Lambda or blocked. As illustrated in Figure 6, the system accurately classified all records:

- 70/70 legitimate records were allowed
- 30/30 anomalous records were blocked

This confirms the real-time decision flow operates as intended, with no request-level failures or misclassifications in the controlled simulation.

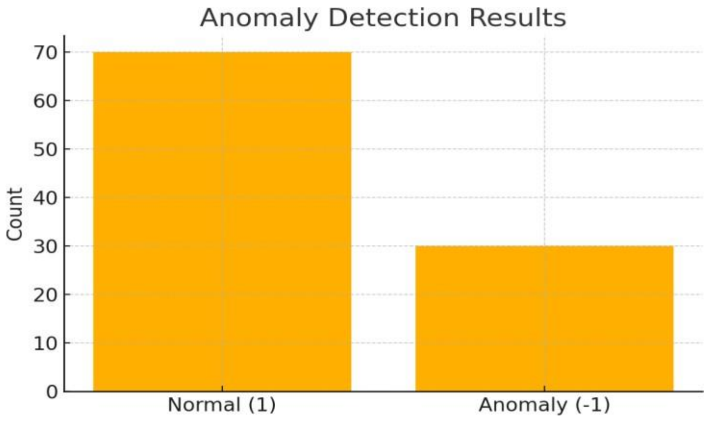


Figure 6: Predicted count of normal and anomalous requests from test traffic

6.5 Summary and Implications

Overall, the evaluation confirms that the EZTSM framework provides a scalable, accurate, and context-aware solution for enforcing Zero Trust principles in serverless computing environments. Key takeaways include:

- **High Precision** ensures legitimate users are not penalized
- **High Recall** ensures that nearly all anomalies are detected
- **Real-Time Scoring** maintains latency budgets suitable for API-based workloads
- **Behavioral Context Modelling** adds a dynamic decision layer beyond identity-based policies

The system architecture, combined with a robust anomaly detection engine, offers a viable model for organizations seeking to enhance security in serverless cloud infrastructures without degrading user experience. Future testing will include latency benchmarking and application to live Lambda logs for further validation.

6.6 Discussion

The evaluation results obtained proves the effectiveness of the Enhanced Zero Trust Security Model (EZTSM) in implementing dynamic, per-request access control in serverless environments. By integrating Machine Learning based anomaly detection engine with the Zero Trust pipeline, the entire system achieved high classification accuracy while meeting the latency requirements of real-time workloads and without compromising the performance. The evaluation shows that the confusion matrix results as an overall accuracy of 99.85% with perfect precision, ensuring that no legitimate requests were blocked and anomalous request was identified correctly. This result is particularly suitable for serverless framework, where workloads are highly dynamic and security blind spots can emerge rapidly (Li et al., 2023; Bhatt et al., 2024).

Blocking legitimate traffic can affect user experience and may lower trust of users in a security system. EZTSM ability to maintain precision at 1.0 aligns with prior findings that highlight the operational value of precision in anomaly-based security system (Hagemann and Katsarou, 2020). Still 15 anomalies were misclassified as legitimate (false negatives), the recall for anomalous detection remained at 0.97, balancing detection capability with operational continuity.

Feature-level analysis makes the system ability to detect and identify context-driven anomalies, such as unusual invocation times or high execution durations. These behavioral indications are often missed by traditional Zero Trust approaches that focus solely on identity-based verification. The results support arguments made by Kang et al. (2023) that Zero Trust frameworks should incorporate dynamic, context-aware evaluation to address evolving threat landscapes.

The entire system is implemented using AWS native services starts from authentication layer till decision layer. This ensures the operational reliability and interoperability, which is a critical requirement for cloud-native deployments where services must securely exchange data with minimal latency (Barrak et al., 2024).

Finally, the evaluation proves that EZTSM offers a practical approach for integrating anomaly detection into Zero Trust Security framework for serverless workloads. The testing was conducted using synthetic generated request that reflects the actual lambda function workload in a controlled environment, real-time behavioral scoring, and cloud-native deployment can be extended to production scenarios. Future research should test EZTSM under live real-time production traffic, incorporate adaptive retraining mechanisms, and expand feature sets to further enhance detection robustness.

7 Conclusion and Future Work

This project focused on designing and implementing a dynamic behavior-aware, Zero Trust-based access control mechanism for particularly serverless cloud environments, termed the Enhanced Zero Trust Security Model (EZTSM). The primary research question addressed was: **How can the existing security limitation on serverless computing be addressed by designed and implemented Enhanced Zero Trust Security Model (EZTSM) along with ML based anomaly detection techniques?**

The main objective of this work includes enforcing continuous access verification policies, integrating unsupervised anomaly detection using Isolation Forest algorithm, reducing false trust assumptions, and maintaining low latency for real-time cloud serverless workloads. A synthetic dataset was generated to reflects AWS Lambda invocation metadata, and the model was trained using the Isolation Forest algorithm to detect outliers in request behavior. The trained model was then containerized as docker image and deployed on ECS, integrated with a Gatekeeper Lambda function for real-time access decisions.

The results proved that EZTSM was able to classify anomalous and legitimate requests with high accuracy (99.85%), minimum false positives, and a strong balance of precision and recall. Furthermore, the entire architecture was validated using simulated traffic, showing the feasibility of deploying machine learning-based Zero Trust policies in a production-grade serverless environment.

However, the system has few limitations. The EZTSM was trained on synthetically generated dataset and evaluated under controlled scenarios. While these conditions allow for structured testing, they do not fully reflect the complexity of real-world workloads or user behaviors.

Even though the EZTSM has been tested live AWS Lambda traffic in a controlled environment, future work will involve evaluating the system in an actual real-time production environment and integrating streaming data pipelines for online learning. Incorporating feedback loops will allow the EZTSM to adapt to evolving threat patterns. Moreover, extending the system to handle multi-factor inputs like IP reputation, user behavior trends, and function-specific risk scores could further enhance decision accuracy. In future integrating the framework with other detection techniques such as deep anomaly detection or graph-based methods could also be explored to boost predictive power. Finally, latency benchmarks and cost-optimization strategies can be studied to ensure the scalability of EZTSM for enterprise adoption.

References

Li, X., Leng, X. & Chen, Y. (2023) ‘Securing serverless computing: Challenges, solutions, and opportunities’, *IEEE Network*, 37(2), pp. 166–173. doi:10.1109/MNET.005.2100335.

Marin, E., Perino, D. & Di Pietro, R. (2022) ‘Serverless computing: A security perspective’, *Journal of Cloud Computing: Advances, Systems and Applications*, 11(1), pp. 1–12. doi:10.1186/s13677-022-00347-w.

Ni, K., Mondal, S.K., Kabir, H.M.D., Tan, T. & Dai, H.-N. (2024) ‘Toward security quantification of serverless computing’, *Journal of Cloud Computing: Advances, Systems and Applications*, 13(1), pp. 1–27. doi:10.1186/s13677-024-00703-y.

Bhatt, A., Sharma, S. & Bhadula, S. (2024) ‘Security issues in serverless cloud computing architectures’, *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, Greater Noida, India, 9–10 February, pp. 39–43. doi:10.1109/IC2PCT60090.2024.10486369.

Barrak, A., Fofe, G., Mackowiak, L., Kouam, E. & Jaafar, F. (2024) ‘Securing AWS Lambda: Advanced Strategies and Best Practices’, *2024 IEEE 11th International Conference on Cyber Security and Cloud Computing (CSCloud),* Shanghai, China, pp. 113–119. doi:10.1109/CSCloud62866.2024.00027.

Marappan, K., M, N., S, P., R, V., R, S. & K, C. (2025) ‘Exploring Machine Learning Benefits for Healthcare with AWS Lambda Technology’, *2025 11th International Conference on Communication and Signal Processing (ICCSP),* Melmaruvathur, India, pp. 1831–1836. doi:10.1109/ICCSP64183.2025.11088750.

Alnoaimi, S. & Alomary, A. (2025) ‘Zero Trust Security: A Comprehensive Comparative Analysis of Zero Trust Maturity Models’, *2024 International Conference on IT Innovation and Knowledge Discovery (ITIKD),* Manama, Bahrain, pp. 1–8. doi:10.1109/ITIKD63574.2025.11005097.

Kang, H., Liu, G., Wang, Q., Meng, L. & Liu, J. (2023) ‘Theory and application of zero trust security: A brief survey’, *Entropy,* 25(12), p. 1595. doi:10.3390/e25121595.

Mehraj, S. & Banday, M.T. (2020) ‘Establishing a zero trust strategy in cloud computing environment’, *2020 International Conference on Computer Communication and Informatics (ICCCI),* Coimbatore, India, 22–24 January. doi:10.1109/ICCCI48352.2020.9104091.

Nisha, N.T., Pramod, D. & Singh, R. (2023) ‘Zero trust security model: Defining new boundaries to organizational network’, *2023 15th International Conference on Contemporary Computing (IC3),* New York, NY, USA, pp. 603–609. doi:10.1145/3607947.3608067.

R, S. & Yelsangiker, A. (2024) ‘Zero Trust Security Architecture’, *2024 Recent Advances in Sustainable Engineering and Future Technologies (RASEFT),* Hyderabad, India, pp. 136–140. doi:10.1109/RASEFT61414.2024.00034.

Ahlawat, N. & Awekar, A. (2024) ‘Incremental Isolation Forest to Handle Concept Drift in Anomaly Detection’, *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (CODS-COMAD ’24),* New York, NY, USA, pp. 582–583. doi:10.1145/3632410.3632486.

Aktas, G., Ipek, B., Konukoglu, E.A. & Aydin, Y. (2023) ‘Development of artificial intelligence supported tool for anomaly detection in cloud computing systems’, *2023 International Conference on Electrical, Communication and Computer Engineering (ICECCE),* Dubai, UAE, 30–31 December, pp. 1–6. doi:10.1109/ICECCE61019.2023.10442490.

Al-Shehari, T., Al-Razgan, M., Alfakih, T., Alsowail, R.A. & Pandiaraj, S. (2023) ‘Insider threat detection model using anomaly-based isolation forest algorithm’, *IEEE Access,* 11, pp. 118170–118173. doi:10.1109/ACCESS.2023.3326750.

Hagemann, T. & Katsarou, K. (2020) ‘A systematic review on anomaly detection for cloud computing environments’, *Proceedings of the 3rd Artificial Intelligence and Cloud Computing Conference (AICCC 2020),* Kyoto, Japan, 18–20 December. doi:10.1145/3442536.3442550.

Ji, I.H., Lee, J.H., Kang, M.J., Park, W.J., Jeon, S.H. & Seo, J.T. (2024) ‘Artificial intelligence-based anomaly detection technology over encrypted traffic: A systematic literature review’, *Sensors,* 24(3), p. 898. doi:10.3390/s24030898.

Xu, D., Wang, Y., Meng, Y. & Zhang, Z. (2017) ‘An improved data anomaly detection method based on isolation forest’, *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 9–10 December. doi:10.1109/ISCID.2017.202.

Xu, H., Pang, G., Wang, Y. & Wang, Y. (2023) ‘Deep isolation forest for anomaly detection’, *IEEE Transactions on Knowledge and Data Engineering*, 35(12), pp. 12591–12606. doi:10.1109/TKDE.2023.3270293.

Yepmo, V., Smits, G., Lesot, M.-J. & Pivert, O. (2024) ‘CADI: Contextual Anomaly Detection using an Isolation-Forest’, *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (SAC ’24)*, New York, NY, USA, pp. 935–944. doi:10.1145/3605098.3635969.