# Project Report

MSc Research Project
MSc Cloud Computing

Maitreyee Mulay
Student ID: 23298227

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

# National College of Ireland
# Project Submission Sheet
# School of Computing

| | |
|---|---|
| **Student Name:** | Maitreyee Mulay |
| **Student ID:** | 23298227 |
| **Programme:** | MSc Cloud Computing |
| **Year:** | 2025 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Yasantha Samarawickrama |
| **Submission Due Date:** | 10/08/2025 |
| **Project Title:** | Project Report |
| **Word Count:** | 5864 |
| **Page Count:** | 21 |

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:** | Maitreyee Mulay |
| **Date:** | 10th August 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 Programme Coordinator office must be placed into the assignment box located outside the office.

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

# Project Report

Maitreyee Mulay
23298227

**Abstract**

In modern world sophisticated cyber threats are emerging for which traditional perimeter based security systems are proving inadequate to fight such threats. The solution proposed in this research is zero trust security model which follows the rule of 'never trust always verify', this offers a more robust paradigm. This paper presents the implementation of a self adaptive zero trust framework that ustilises machine learning to provide dynamic, real-time policy enforcement in a cloud environment. The dataset CSE-CIC-IDS2018 network traffic is used for model development. An end to end ML operational pipeline is created on AWS (Amazon Web Services). The data pre-processing and segregation into internal and external threat classification is done initially. Machine learning models - Random Forest , SVM (Support Vector Machine) and logistic regression are used in Sagemaker AI notebook. The lambda functions are created to trigger based on simulation which provides scalable and isolated inference endpoints. Cloudwatch is used log the events of the simulation. The results of the model demonstrates high model accuracy. The classifies for external threat models have 89 and 91 percent accuracy while internal threat model achieved is 99 percent. The system successfully indicates the threat detected with recommended action based on which the lambda applies the policy to the user appropriate to the threat level. This research provides a complete blueprint for operating an intelligent Zero Trust system which confirms and feasibility and effectiveness using ML-driven predictions with explainability for adaptive security policy enforcement.

# 1 Introduction

There is a paradigm shift in the digital security landscape. In the past few years the security paradigm was known as the perimeter-based or castle-and-moat model. This concentrated more on constructing a good solid impenetrable protective perimeter over the inner network of an organization. Such an idea means that one thing or an entity could be trusted when situated within a boundary and would not be trusted otherwise. Nevertheless, this model has become more obsolete than ever considering the complexities that modern IT ecosystems. The experience in the contemporary world that are increasing the security concerns through the use of cloud computing, mobile workforces and the Internet of Things (IoT). The network perimeter is not well defined as well. Once breached, such weak borders almost always reveal the presence of an internal trust within the system that permits intruders to move laterally with minimal obstruction. This exponentially enhances the possibility of extented damage produced by the intrusion. Additionally, the cyber-type threats are evolving with time. Attackers make use of most advanced, diverse and automated approaches as techniques that cannot be detected by the signature-based systems. This has created a condition where Security Operations Centers (SOCs) are overwhelmed with the large number of alerts that can lead to the phenomenon of so called alert fatigue when critical threats are left undetected (Tariq et al.; 2025).

It is no longer a question of breach prevention but rather the ability to quickly detect and respond to the breach. The new reality requires a more dynamic, detailed, and smarter way of security one that is based on the idea of explicit verification over implicit trust. Adaption of Intrusion Detection Systems (IDS) are essential, particularly in upcoming areas such as smart car networks since the assault territory is immense and an attack is catastrophic (Almehdhar et al.; 2024)

### 1.0.1 Research Question

This thesis addresses this gap by seeking answer of the problem: How can a functional, self-adaptive Zero Trust framework be designed, implemented and evaluated to enforce security policies in real-time while ensuring transparency based on machine learning predictions within a cloud infrastructure?

To investigate this problem, the following research questions (RQs) are posed:

- RQ1: How an architecture could be designed for an end-to-end functional pipeline on a public cloud platform that supports the entire lifecycle of a security-focused machine learning model, from data ingestion and preprocessing to model training, deployment?

- • RQ2: How effective are machine learning models accurately classifying different types of network threats using a large network detailed, realistic dataset like CSE-CIC-IDS2018?[1]

- • RQ3: How can the probabilistic outputs of these machine learning models be programmatically translated into automated policy enforcement actions within a serverless computing framework to create a truly adaptive security response system while maintaining transparency?

### 1.0.2 Research Objectives and Hypothesis

Based on the research questions, the primary objectives of this work are as follows:

- To Create a functioning pipeline by leveraging AWS services like Sagemkaer, Lambda, Amazon S3, CloudWatch

- Segrating threats into external and internal type. And to train, evaluate, and compare the performance of multiple machine learning models for threat detection.

- To automate the policy enforcement based on results of machine learning pipeline and maintaining transparency

[1] CSE-CIC-IDS2018 Dataset. Canadian Institute for Cybersecurity. Available at: `https://www.unb.ca/cic/datasets/ids-2018.html`

Hypothesis on which this research is based is as follows:

A self adaptive system that leverages machine learning models to be trained on network traffic data in order to make real time predictions and can dynamically enforce Zero Trust access policies with high accuracy and automation. By providing mode effective defense against both internal and external threats compared to static perimeter based security models.

# 2 Related Work

This section conducts the critical review of the current academic sources associated with a focus on the technology of machine-learning based security and zero trust systems automation. This review is anchored on three major pillars that are the foundation of this thesis. The first is the evolution of security paradigms towards Zero Trust Architecture. Second is the application of machine learning for intelligent threat detection and the third is the operationalization of these systems through automation. By analyzing the strengths and limitations of prior work and highlights the specific gap that this thesis aims to fill.

## 2.1 Zero Trust Security

The traditional models assumed that network security was up to its end. It operated on a binary principle of trust: entities within the perimeter are trusted while outside are not. However, the emerging IT environment negatively affects the concept of a defensible perimeter, making security a primary concern (Jeyalakshmi et al.; 2025) Among the most severely affected by this are specialized data-rich artificial intelligence domains for example, artificial intelligence of things, which deals with surveillance Khurshid et al. (2025) as well as artificial intelligence of medical things affecting health systems, where security and privacy are critical (Awotunde et al.; 2023)

On the other hand Zero Trust has principle of 'never trust always verify' here the least required access is provided to do the tasks assigned to the user (Ahmadi; 2024) A primary strength of the ZT model is its focus on granular, identity-centric controls where access is granted on a per-session, least-privilege basis—a core tenet of next-generation secure authentication architectures (Kumar; n.d.). The focus is on explicit verification provides a strong foundation for advanced Data Loss Prevention (DLP) strategies against both internal and external threats Abid (2020). ZTA can be utilised with transport access control and the authentication inside the cloud networks Rodigari et al. (2021)

However,literature indicates that one of the main constraints of ZT is that of being operationally complex. Manually setting up the policies of each user and device is practically impossible in large dynamic organizations. This vulnerability helps in understanding why automation and intelligence are needed to render Zero Trust scalable. As (Bashaa et al.; 2025) explicitly state in the review, the convergence of ZT and machine learning is a vital advance towards the safeguarding of contemporary infrastructures such as Software-Defined Networking (SDN). The work is good in terms of conceptual framework, but as a review it lacks the empirical implementation.Something which leaves a big gap between the theory and the testing true implementation.

## 2.2 Machine Learning in Security

Machine learning has emerged as a powerful tool to overcome the challenges faced in modern security such as scalability and dynamism (Dritsas and Trigka; 2025) By learning from large quantities of data the ML models can identify complex patterns, detect anomalies and automate decision-making process at such a scale that would be highly difficult for human analysts.

Machine learning has also transformed the cybersecurity by making systems learn the patterns and real-time anomaly detection. It is the key factor of the experimental approach because in real-time the testing of models is maintained, and data logs can be monitored in real-time, and threat detection could become instant(Dhanushkodi and Thejas; 2024).

### 2.2.1 Machine Learning for Intrusion Detection Systems

The application of ML to Intrusion Detection Systems (IDS) is a well established field of research. Unlike signature based IDS which fail against novel attacks the ML based systems can learn a baseline of normal behavior and flag deviations. A comprehensive survey by (Almehdhar et al.; 2024) explores the use of deep learning for advanced IDS in the highly dynamic context of intelligent vehicle networks. The strength of their work lies in its thoroughness. But as a survey it highlights the need for such systems without presenting a deployable architecture. The versatility of ML in predictive analytics reinforces the maturity of the underlying techniques. Its principles have been successfully applied across diverse domains, from crime forecasting (Shah et al.; 2021) to managing the reliability of critical energy infrastructure (Duchesne et al.; 2020) and optimizing supply chains (Vlachos and Reddy; 2025). These different studies confirm the fundamental assumption that ML is a credible method of predictive model in a complex setting. Network threat detection can be directly compared with the rule of projecting the past to the future. One weakness of most such work is that it does not involve real-time, automatized enforcement of policies; the work ends in a successful prediction but does not publish the future step of actualizing this prediction within a closed-loop response system.

## 2.3 Explainable AI XAI in Security

The most crucial barrier to the adoption of ML in critical security systems is the 'black box' problem. It is a significant drawback of many well-performing models because providing practically no information about their decision-making, they fail to secure the stage in which an analyst needs to know why a certain operation was performed. Based on which XAI has come in the picture. The work of (Wang et al.; 2024) provides a seminal overview of the role of XAI in future 6G networks, arguing cogently that for use cases like security, explainability is a core requirement for accountability and trust. Their main shortcoming is that they are mostly concerned with characterizing the problem space rather than offering a working piece of implementation. This research addresses this by incorporating SHAP (SHapley Additive exPlanations) to provide tangible, feature-level explanations for model predictions.

Counterfactual explanation that gives observations on how the minimal change could have made a difference in the policy enforcement. To supply insight data about specific

AI model so that custom fit with any AI model can be transversely feasible in the system, model agnostic explanations would be provided (Ahmadi; 2024).

## 2.4 Policy Enforcement Automated Mechanisms

In intelligent security system is the goal is Zero Touch Management, where routine operations are fully automated.technical way of addressing it. (DeCusatis and Zhang; 2016) discuss the possibility of Zero Trust Access Control Policy with regard to cloud nativity through the integration of IAM and real-time threats detection. The automation is a direct answer to 'alert fatigue' in Security Operations Centers (SOCs), a problem detailed by (Tariq et al.; 2025). The paper has highlighted the human factors challenge, but it fails to suggest a

This gap allows the enforcement mechanisms represented by ML to gain massive benefit in terms of automaticity of rules maintenance based on context analysis of threats (Shaik and Gudala; 2021)

## 2.5 Architectures for ML Deployment: Cloud, Edge, and Serverless

The operationalization of ML models depends heavily on infrastructure. Modern architectures are moving towards distributed models leveraging cloud and edge computing. Research by (Pujol et al.; 2023) explores the opportunities for 'edge intelligence'. (Walia et al.; 2023) review AI-empowered resource management at the fog/edge. The strength of these works is their forward-looking perspective, but their focus is broad, centered on general distributed computing challenges rather than a specific, deployable security application.

The work presented in this research utilizes a serverless architecture (AWS Lambda). This offers automatic scaling and reduced overhead ideal for event-driven security tasks. While concepts like multi-cloud resilience are important for enterprise-grade systems (Grace; 2025), this thesis provides a concrete, single-cloud implementation as a foundational building block.

## 2.6 Research Niche

The existing literature confirms a clear and converging trajectory. It establishes that Zero Trust is the necessary strategy (Bashaa et al.; 2025), machine learning is the key enabling technology Dritsas and (Tariq et al.; 2025) and automation is the path to operationalization (Coronado et al.; 2022). Where there is automation it lacks in elaboration of the automated decisions in a format that is understandable by a human being. Due to the absence of real time explainability in ML powered zero trust implementations which then causes regulatory vulnerabilities and lack of auditability(Shaik and Gudala; 2021). However, a critical analysis of these papers reveals a significant gap: the components are often studied in isolation. There are excellent surveys on ZTS, detailed analysis of ML algorithms (Almehdhar et al.; 2024), forward-looking papers on XAI (Wang et al.; 2024) and (Shaik and Gudala; 2021), and architectural reviews of edge computing (Pujol et al.; 2023); (Walia et al.; 2023).

What is absent is one study that combines all these elements into one, useful and repeatable system. The existing solutions fail to meet the requirements since they are either of high level of abstraction singly limited to one component of algorithm without specifying the deployment conditions or architecture possibilities with no actual implementation. This research gap is addressed in this thesis. It goes beyond the theory to show a whole reproducable implementation. Which includes serverless deployment, automatic, real-time policy application and enforcement and transperancy in decision making. This paper becomes the connection points of the concept on the one side with the actual functioning of a ZTS system on the another side

# 3 Methodology

The methodology for this research includes the design, implementation and deployment of a cloud native, serverless architecture for real-time, ML-driven policy enforcement, embodying the principles of a self adoptive zero trust model. The system is constructed within the AWS cloud platform. The system has leveraged variety of services to build rapid development, scalability and operational effeciency. The implementation process is detailed from initial environment setup to the deployment and simulation.

## 3.1 Phase 1: Environment and Infrastructure Setup

The foundational phase focused on establishing a secure and functional AWS environment, including identity management and infrastructure setup.

### 3.1.1 AWS Environment and Identity Management

The project is developed within a single AWS account (405045611860) and geographically scoped to the eu-north-1 that is Stockholm region. For programmatic access by AWS services, a distinct IAM Role named SageMakerExecutionRole is established. This role-based approach is a security best practices. Permissions are granted to the services rather than embedding credentials in code. The trust policy of the role is defined in a JSON format to allow the AWS SageMaker service. This follows the principle of least privilege. The policies are assigned to this role to grant the necessary permissions they are as follows: AmazonS3FullAccess: To read the datasets and write model artifacts. AmazonSageMakerFullAccess To create and manage notebook instances in sagemaker. Created inline policy for invoking lambda action. This role is central to the architecture, later being re-used by AWS Lambda functions to ensure they had the necessary permissions to interact with other AWS services, demonstrating a unified permissions model.

### 3.1.2 ML Development Environment

For the Integrated Development Environment (IDE) all the data science tasks an Sagemaker notebook is created named ZTMLNotebook. The instance type is ml.t3.medium was selected as a cost effective choice and based on the need with sufficient computational resources for data preprocessing and model training. The role created earlier SageMaker-ExecutionRole is assigned to the instance with permissions to connect with S3. The development work is done in JupyterLab notebboks running a conda python3 kernel.

## 3.2 Data Selection and Preprocessing

### 3.2.1 Initial preparation

This research has utilized the CSE-CIC-IDS2018 dataset1. It is a comprehensive and acknowledged benchmark for evaluating intrusion detection systems. This is suitable for this project for following reasons:

- Real and Divers data: The data is in large volume of network traffic including a wide variety of internal and external attack type such as Brute-force, DoS, DDoS, Botnet, Infiltration and benign traffic.
- The Features: The data includes total 79 network flow features extracted by the CIC FlowMeter tool. This provides rich, high dimensional representation of traffic patterns suitable for ML analysis.
- Labeled Data: All traffic is particularly labeled. This is essential for the supervised learning approach adopted in this thesis.

To provide a secure, scalable data storage solution for the dataset for ML model training AWS simple storage service is used for this purpose. An S3 bucket named 'zero-trust- ml-dataset' is created in the eu-north-1 region using the AWS CLI. The raw dataset is then uploaded from the local development environment to the root of this S3 bucket. A local Jupyter Notebook file 'dataset preprocessing.ipynb' is created to define the data cleaning, feature engineering and transformations process needed to have the data in a consumable state for ML algorithms. This notebook is the roadmap to the model building phase that is next.

### 3.2.2 Data Cleaning and Balancing

The dataset selected is imbalanced as it is a raw network traffic datasets. The benign traffic vastly outnumber malicious traffic and few attack types are frequent than others. If model is trained on such imbalanced data it can lead to creating a classifier that is heavily biased towards the majority class. IT would perform poorly on rare but critical attack types.To handle this a multi-step cleaning and balancing procedure was executed within a SageMaker notebook:

- Handling Missing Values: All rows containing any missing values (NaN) are dropped to ensure data quality and prevent errors during model training.
- Class Selection and Sampling: The class distribution is analyzed and the top seven most frequent classes were selected for the study. This was a pragmatic decision to create a focused, multi-class problem that was computationally manageable while still representing a diverse set of threats This is done to create a focused multi class problem that is manageable computationally and still represents a diverse set of threats.
- Stratified Sampling: A stratified sampling have to be designed so that the data set could be balanced. In every one of the seven classes used, samples would be fixed Out of (12, 000) it is randomly selected. This would make each of the classes equal in number of representation in the last dataset where it is important that the model will learn what is distinctive in each kind of threat as opposed to using the frequency of a class. The result of balanced and shuffled dataset is saved in S3 bucket again.

### 3.2.3 Contextual Data Segregation

One of the hypotheses in this study is that the Zero Trust system can be developed more successfully by training specific models to tackle threat situations in different contexts. To check this further the balanced data set was divided into two separate subsets depending on the probable source of threat:

- External Threats: The threat types that were included in this dataset are those that have the origins of attack usually outside.The wide range of attacks that are aimed at attacking the network perimeter, i.e., DDoS, DoS, and SSH-Bruteforce with traffic being benign

- Internal Threats: The attack types that are more representative of an internal system compromise or malicious insider were included in this dataset, including Botnet activity and Infiltration.

The segregation enables one to come up with specific models. An example is external facing can be deployed at network perimeter and internal model can monitor internal flows within the network. The two new databases 'internal threats.csv' and 'external threats.csv' have been stored in a special folder on the S3 bucket.

## 3.3 Model Development and Training

After the data is in order, the next step involves using it to build and train and evaluate machine learning models developed to meet the internal and external threat scenarios. The development is made in Python and all the libraries used were Scikit-learn, Pandas, and SHAP developed in the SageMaker environment.

### 3.3.1 Feature Engineering and Preprocessing

The preprocessing steps were standardized both in the case of the internal dataset and the external dataset:

- Label Separation: The column called Label was put apart as a feature: By setting (X) to form the target variable (y)

- Label Encoding: The classification level string labels like Benign, DDoS are transformed into numerical representation with label encoder of Scikit-learn. The ML algorithms require such a step to make predictions. A copy of the object of the internal model encoder was saved to be used later to decode the predictions.

- Feature Scaling: StandardScaler was used on the numeric features to scale them. The feature is standardized in this technique by subtracting the mean and centreing by unit variance. This is an important method of the algorithms which is considered sensitive to the scale of input features such as the SVM and Logistic Regression and it also holds advantages to the tree based models with an advantage such as the Random Forest.

### 3.3.2 Train Test Split

In order to have a strong and objective performance of a model, the preprocessed data of every context was divided into training and testing sets. This was in a ratio of 70/30. so 70 percent of the data will be used to train the model and save the other 30 percent testing. Critically, the stratify parameter was used during the split to ensure that the class proportions in both the training and testing sets are identical to the proportions in the original dataset. This prevents a scenario where the test set accidentally contains a different distribution of classes than the training set.

### 3.3.3 Model Selection and Training

Based on the literature review and the nature of the problem several well-established supervised learning algorithms are chosen:

- Random Forest: It is a powerful ensemble method that builds multiple decision trees and merges their outputs. It is robust to overfitting and handles high-dimensional data well. It also captures non-linear relationships. Hence it is chosen for the complex, multi-class external threat analysis.

- Support Vector Machine (SVM): It is a strong binary and multi-class classifier that works by finding the optimal hyperplane to separate classes. It is trained on a subset of the external data to evaluate its performance.

- Logistic Regression: It is a reliable and highly interpretable linear model. It is used for internal threat analysis as it has high computational efficiency and clear outputs.This is well-suited for the simpler binary classification task (Bot vs. Infiltration).

To streamline the process a Scikit-learn pipeline is used for each model. This encapsulates the preprocessing steps such as scaling. The classifier into a single object which prevents data leakage from the test set into the training process and simplifies the model saving and deployment process. The trained pipeline objects are serialized and saved as .pkl files using the joblib library.

### 3.3.4 Model Evaluation

The performance of the trained models is evaluated on the test set using a suite of standard classification metrics:

- Accuracy: The proportion of correctly classified instances.

- Precision, Recall, and F1-Score: These metrics provide a more detailed view of performance especially in a multi class settings. Precision measures the accuracy of positive predictions, recall measures the model's ability to find all positive instances and the F1-score provides their harmonic mean. These are calculated on a per class basis.

- Classification Report: A consolidated text report is created from Scikit-learn which shows the main classification metrics: F1 Score, precision, recall, accuracy for each class.

- Model Explainability (SHAP): To address the 'black box' concern, the SHAP (SHapley Additive exPlanations) library was used to analyze the Random Forest model. By calculating SHAP values, it is possible to determine the contribution of each feature to a specific prediction.This provides crucial insights into the model's decision-making process. This aligns with the need for XAI in security as specified by Wang et al. (2024)

## 3.4 Model Deployment and Operationalization

The model is deployed to make predictions on new data. This research implemented a serverless, event-driven architecture to operationalize the models.

### 3.4.1 Trigger Script and Serverless Deployment with AWS Lambda

For external and internal threat simulation section is created within the ML model script. It loads the trained model pipeline. It includes the SHAP where all the feature values influencing the decision are recorded. The sample from dataset is taken and prediction is made for the same. For the simulation a threat level is defined and data is sent over to the lambda function.

Two lambda functions 'Internal-test' and 'External-test' are created for simulation of both types of threats. The lambda function includes python script which receives the data from the simulation script. Based on the data the script runs and checks if the user has any policies applied before. If yes it will remove existing policies and apply newly recommended action based on threat level. If the recommended policy is already applied it will skip the application. The policy is applied based on threat level. If score is low 0.5 the action is 'allow', if score is moderate 0.5 to 0.8 the action recommended be 'require MFA' and for high score 0.8 the action is 'revoke session or deny'. The image 1 shows the automated simulation flow.

This serverless architecture is highly cost-effective and scalable, as AWS automatically manages the underlying compute resources, and the functions are only executed (and billed for) when they are invoked.

### 3.4.2 Logging

The function logs the input data from the simulation script based on model prediction and resulting policy enforcement is logged to AWS CloudWatch Logs.This provides a complete audit trail of the decision making of the system. Which is used to verify the successful operation of the entire pipeline from end to end. The detailed diagram of the simulation flow is shown in figure 2

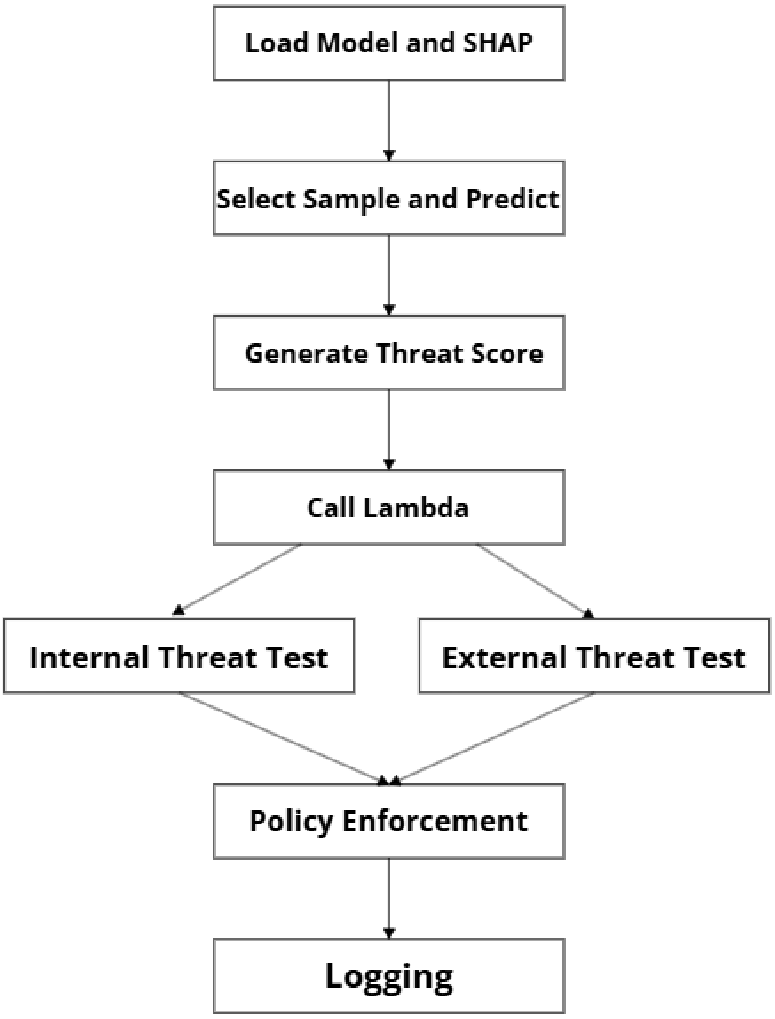


Figure 1: Simulation Flow

# 4 Design Specification

The framework implemented in this thesis is intended to be an event-driven, cloud native modular system which provides real-time security enforcement. The architecture is designed to answer some of the primary research questions. It represents a total end-to-end MLOps pipeline that converts machine learning predictions into automated Zero Trust policy actions. There is a deliberate segmentation of the design into three. there are logical planes: Model Development Plane, Deployment Plane and Real Time Inference Plane. Using this segregation modularity, scalability, and maintainability assured. Block diagram 2 below presents the design plan for the system:

The Model Development Plan can be considered 'brain' of the system as data is processed and utilized here. It includes the use of Amazon S3 where data is stored for the project artifacts. Amazon SageMaker which gives the integrating development environment of all data science activities. It starts with the raw CSE-CIC-IDS2018 being ingested into S3. The data within a SageMaker notebook endures a thorough preprocessing, clean-

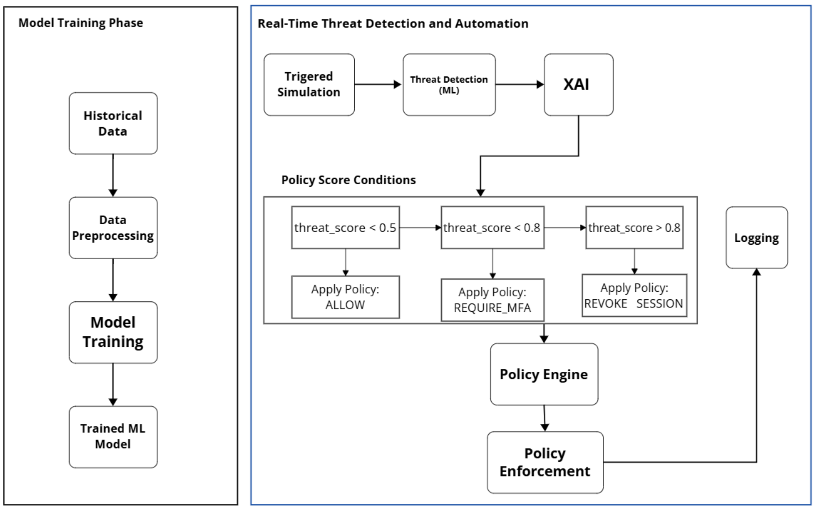


Figure 2: Block Diagram

ing, and contextual subdivision into the so-called internal and external threat datasets. The prepared data is then used to train specialized models. The main design decision was to have an explainability step that involves SHAP, where not only the model decision-making process would be correct but also be comprehensible encoders and pipelines of model serialization. The end products of this plan series model pipelines and the encoders are written back into S3 to be deployed.

To host and execute the machine learning models, the AWS SageMaker is used to train and run the models, and the final pipeline activation was based on AWS Lambda functions. Both an internal and external threat simulation is executed by the Lambda functions using an execution endpoint, which enables on-demand stateless process in the absence of provisions of the server. It will allow a full serverless architecture where scalability and low overhead requirements are not a challenge. The simulation script calls the pre-trained model deployed in SageMaker and makes predictions to provide simulation data to lambda based on which actions are taken.

The Lambda functions upon inference assess the predicted threat score of the model and apply security policies in real-time. Depending on predefined sets, the actions like allow, require MFA, revoke session are selected dynamically. This correspondence to the severity of the threats simulated by these actions in line with a Zero Trust policy response. The full action logs helps to trace of decisions and actions as entirely recorded in AWS CloudWatch Logs. This is transparent and helps the operations meet the requirements because of the real time reactivity. The architecture diagram 3 showcases the complete design of the system using AWS services along with contributions from each service.

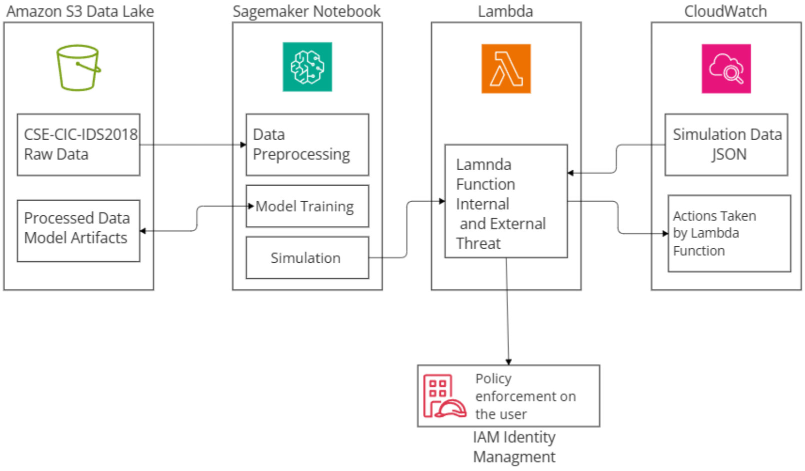


Figure 3: Architecture diagram

# 5 Implementation

The process of implementing the Self-Adaptive Zero Trust ML framework, which comprises the design described in Section 4, completed with a fully functional cloud-native environment in the form of a number of artifacts. These outputs are described in this section and they represent the physical application of the research goals. The data science and application logic were built in Python as the language for this implementation. Amazon Web Services (AWS) infrastructure stuck.The intial dataset used is cleaned up during data processing phase. The initial set of preprocessing installed on an Amazon SageMaker JupyterLab enchanted the raw data of CSE-CIC-IDS2018 into two different and context-referred files available as CSV files: internalthreats.csv and externalthreats.csv. These files are stored in amazon S3 bucket and this bucket is the source of data with regards to model training.

The most critical part of the framework is the machine learning models. The models Random Forest and Support Vector Machine (SVM) classifier is used for external threat analysis. Logistic Regression classifier is trained for internal threat analysis. The final trained versions of these models are not stored as raw code but as serialized python objects. The pipeline is created using Scikit-learn and joblib libraries in each model. This ensures that the exact state of the trained model is preserved for reference. Explainability artifacts are generated containing values for SHAP. It provides crucial of insights into the predictions.

The simulated threat system starts by including the incorporation of a pre-trained machine learning (ML) model pipeline, which is further utilized to simulate the inner and outer security threats. The system makes use of SHAP (Shapley Additive Explanations) that ensures transparency and interpretability because the values of the features behind each decision are recorded. The simulation of threat system starts with utilizing the incorporation of a pre-trained machine learning (ML) model pipeline to simulate the inner and outer security threats.The system makes use of SHAP (Shapley Additive Explanations) that ensures transparency and interpretability because the values of the features behind each decision are logged.A portion of the data is sampled and a prediction is made with the help of the ML model. Higher threats are simulated to showcase the action recommended response.

Two independent AWS Lambda functions are created Internal-test and External-test for both type of simulations. The information threat level, user id, action recommended is sent to these functions by the simulation script. After accessing the data, the Lambda function will establish whether there are any security policies already known to work on the user. In case some policies are discovered, the role determines the discrepancy of the new suggestion with the existing one. In case policy should be updated, old policy is eliminated and the new policy is used depending upon the level of threat. Specifically, when the threat score is less than 0.5 then the action is to allow, when the score is between 0.5 and 0.8 then the action is to require MFA and when it is above 0.8 then the action is to revoke session or to deny access. For the simulation purposes user with user id - user123 is created with AllowAccess policy in place. Upon threat level for internal or external simulation being high the allow access will be removed and the recommended policy will be applied.

Finally, all the decisions taken by the AI model interpreted by SHAP are logged in cloudwatch along the with the data received by lambda. The actions taken by lambda script is also logged in the same log group. This provides ability for the admins and analysts to evaluate and audit the decisions made by the system.

# 6 Evaluation

This section includes an extended analysis of Self-Adaptive Zero Trust ML framework that has been implemented. The assessment is rigorous to the research questions and objectives that have been stated earlier so that performance of this system can be determined. The analysis will be distributed into three segments which target three different components. The accuracy of the exterior threat identification model, accuracy of the interior threat differentiation model, and effectiveness of the automated policy enforcement loop in real-time. This showcases a clear and objective evaluation of the capabilities and limitations of the system carried out by standard statistical metrics and the qualitative analysis.

## 6.1 External Threat Model Performance

The goal of this experiment is to determine success or failure of the machine learning models when it comes to detecting the external threats in network parameter. Random Forest classier and SVM (Support Vector Machine) are trained for this purpose. Both models are fitted using multi-class dataset 'enternal-threats.csv' to answer the Research Question 2 that is how the two models can be used to perform accurate classification of different categories of attacks to external networks.

The evaluation of the trained pipelines was done using a fixed 30 percent test set which the models never used during training. Random Forest classifier evaluation resulted in a total accuracy of 91.22 percent which indicates that it can with distinction separate benign traffic and malicious activity of different types. Arguably, the SVM model was ranked equally high, in comparison but not strictly high, recording overall accuracy of 89.91 percent. Nevertheless, accuracy is still a misleading measure whenever used in multi-class classification tasks. Thus, further finer-grain analysis was made by providing the extended results of classification and confusion matrices in the form of detailed reports of both models.

The test set for fixed 30 percent is used to evaluate the trained pipelines which was not used in training. Random Forest got the accuracy of 91.22 percent which indicates that it can with distinction separate benign traffic and malicious activity of different types. On the other hand SVM scored the accuracy of 89.91 percent.

Further analysis is done with metrics such as Recall, F1 score and precision which is showcased in table 1

## 6.2 Internal Threat Model Performance

For internal threat analysis Logistic Regression model is used on the dataset 'internal-threats.csv' where Bot and infiltration traffic is consisted. The idea was to see whether such specialized model can be found that would be more accurate with these particular, more frequent types of threats. Logistic Regression model has given excellent results with accuracy of 99.32 percent. This major enhancement of the this model in classifying these threats justifies the design decision of applying different special models to special threat environments.

Figure 3 indicates that the classification report is very close to a perfect performance. The near perfect model performance showcased in table 1 evidences the fact that when the model is limited in its scope to distinguishing between these two kinds of malicious activity. It could have the high certainty level of confidence.

## 6.3 Comparison of the ML models with Benchmarks

As each model has different strengths and weaknesses the model performance has varied yet all the models have given cosistently high performance.

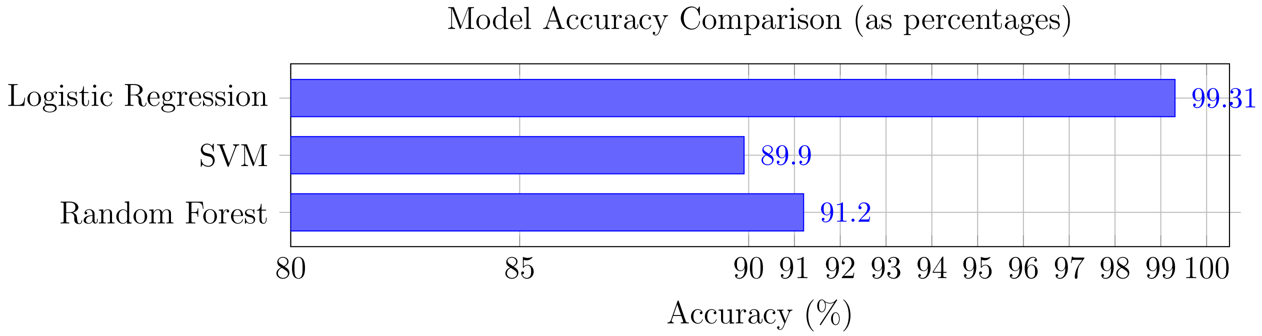


Figure 4: Model Accuracy Comparison as percentages

To assess the performance of the models, the results from this study are compared with those from prior research considered as benchmarks. Points to be taken under consideration are the datasets used in benchmark studies are different than used in current study. This comparison is purely on for referencing of how well the model can perform.

The analysis demonstrates the difference in performance of SVM, Random Forest (RF) and Logistic Regression (LR) models at different datasets and research studies. According to (Alshamrani et al.; 2024), SVM was highly effective on Dataset I: the accuracy was 90.7 and the F1-score 96.2.Whereas on Dataset II the accuracy decreased dramatically down to 34 and also F1-score decreased down to 37%. Compared to it, the SVM model in the current study demonstrated balanced outcomes, that is, an accuracy of 89.9%

Table 1: Comparison of Model Performance Across Studies

| Model | Source | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| SVM | Dataset I Alshamrani et al. (2024) | 90.7% | 82% | 90.7% | 96.2% |
| SVM | Dataset II Alshamrani et al. (2024) | 34% | 34% | 39% | 37% |
| SVM | Current Study | 89.9% | 89.7% | 90.1% | 89.9% |
| RF | Dataset I Alshamrani et al. (2024) | 99.7% | 99.7% | 99.7% | 99.6% |
| RF | Dataset II Alshamrani et al. (2024) | 36% | 36% | 39% | 37% |
| RF | Current Study | 91.2% | 91.0% | 91.4% | 91.2% |
| LR | Chalichalamala et al. (2023) | 99.99% | 99.99% | 99.99% | 98.99 |
| LR | Current Study | 99.31% | 99.91% | 99.00% | 99.99% |

and an F1-score of 89.9%. Equal to the above situation with RF showed a very high performance on Dataset I accuracy of 99.7% and F1-score of 99.6%, but when transferred to Dataset II there was a significant drop in performance with the accuracy is only 36%. Nevertheless, in this research, RF showed a constant performance with an accuracy level of 91.2% and F1-score. Logistic Regression, seemingly an easier model, performed best with the team methodology paper presented by (Chalichalamala et al.; 2023) at 99.99 accuracy. The LR model that was created as part of the presented study showed the competitive performance with 99.31 accuracy rate, stable precision, recall, and F1-score of 0.99.

Based on the overall findings, it can be assumed that models trained on better tuned and preprocessed data with the current study can be better generalized in terms of the threat categories than just benchmark-based trained models.

Another point to be taken under consideration for the model is computational utilization and time taken to train the model.Based on the graph it can be seen that Random Forest takes the least time and utilities the least memory followed by Logistic Regression and SVM models.

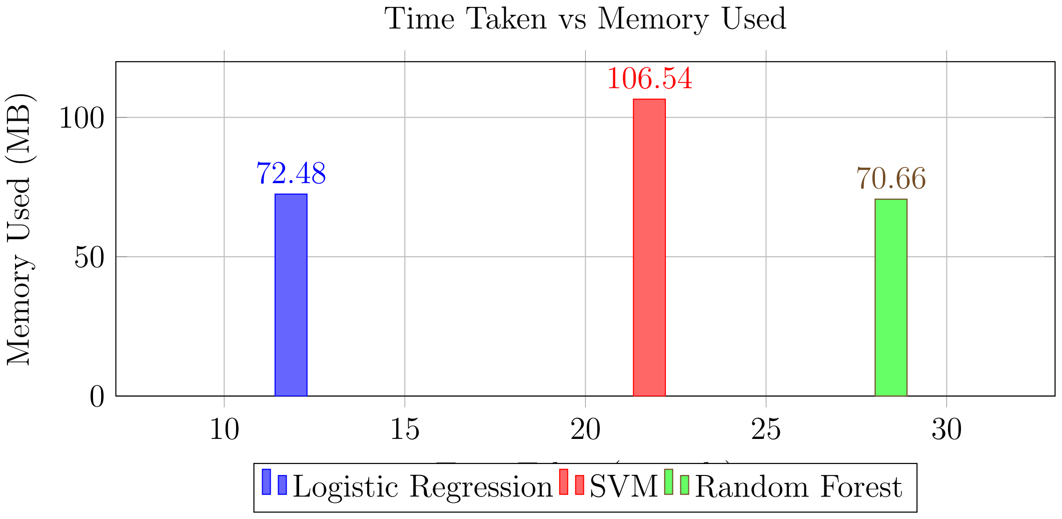


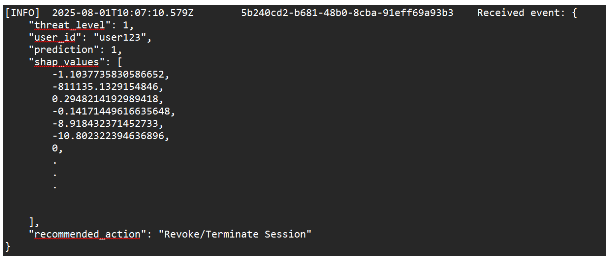
Figure 5: Bar Graph: Time Taken vs Memory Used (Internal Threat Analysis)

Initially CNN (Convolutional Neural Network) was considered for internal threat analysis. However this model performed poorly giving accuracy of only 14 percent and with high computation time hence the model was replaced with Logistic Regression Model. For final recommendations based on model performance Random Forest is preferred for external threat analysis and Logistic Regression is recommended for internal threat analysis.

## 6.4 Simulation

The experiment testes how the whole end to end framework is running. The aim is to ensure that the system would effectively translate a simulated security event into an automated policy action thus addressing RQ3. This is not measured using a quantitative measurement, but by a qualitative reading of the audit trail system in AWS CloudWatch Logs. It is also confirmed on the CloudWatch logs that simulation script was triggering their respective Lambda functions (Internal-test, External-test). The logs were able to give a complete trace the appearance of the function step-by-step e.g. loading of the model to per making deductions and working the policy rationality.

Lambda function is triggered upon receiving the data with details referred in figure 6. The sample shows is for internal threat. Threat levels are from 0.1 to 1.0. Where 0.1 is lowest and 1.0 is highest.

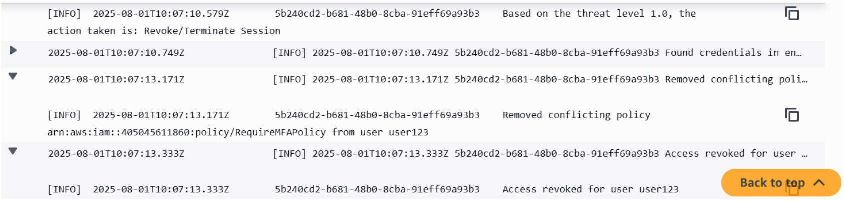


```
[INFO]  2025-08-01T10:07:10.579Z        5b240cd2-b681-48b0-8cba-91eff69a93b3    Received event: {
    "threat_level": 1,
    "user_id": "user123",
    "prediction": 1,
    "shap_values": [
        -1.1037735830586652,
        -811135.1329154846,
        0.2948214192989418,
        -0.14171449616635648,
        -8.918432371452733,
        -10.802322394636896,
        0,
        .
        .
        .


    ],
    "recommended_action": "Revoke/Terminate Session"
}
```

Figure 6: Sample JSON data recivied as an event

Based on the received data the policy enforcement is taken place as shown in figure 7.



```
    [INFO]  2025-08-01T10:07:10.579Z        5b240cd2-b681-48b0-8cba-91eff69a93b3    Based on the threat level 1.0, the
    action taken is: Revoke/Terminate Session
▶   2025-08-01T10:07:10.749Z        [INFO] 2025-08-01T10:07:10.749Z 5b240cd2-b681-48b0-8cba-91eff69a93b3 Found credentials in en…
▼   2025-08-01T10:07:13.171Z        [INFO] 2025-08-01T10:07:13.171Z 5b240cd2-b681-48b0-8cba-91eff69a93b3 Removed conflicting poli…

    [INFO]  2025-08-01T10:07:13.171Z        5b240cd2-b681-48b0-8cba-91eff69a93b3    Removed conflicting policy
    arn:aws:iam::405045611860:policy/RequireMFAPolicy from user user123
▼   2025-08-01T10:07:13.333Z        [INFO] 2025-08-01T10:07:13.333Z 5b240cd2-b681-48b0-8cba-91eff69a93b3 Access revoked for user …

    [INFO]  2025-08-01T10:07:13.333Z        5b240cd2-b681-48b0-8cba-91eff69a93b3    Access revoked for user user123
```

Figure 7: Policy Enforcement Logs

## 6.5 Discussion

This research sets out to address the question of how a functional self adaptive zero trust framework could be implemented on the cloud platform to enforce security policies based on real time machine learning predictions. The primary objectives were to design and implement a functional pipeline to train and evaluate effective threat detection models and convert their predictions into automated policy actions.

The results also showcase that a self adaptive security framework that leverages machine learning can effectively and automatically enforce Zero Trust policies in a simulated environment in real time proving the primary hypothesis. The high accuracy of the models indicate that the core ML component is robust. This algin with the findings of (Almehdhar et al.; 2024) which showcases ML is a powerful tool for IDS. This thesis achieved these objectives by constructing a complete serverless architecture on AWS that demonstrates the lifecycle from data ingestion to automated enforcement. It bridges a significant gap in context of previous research while many studies focus on model focus on performance in ML model in lab setting (Shah et al.; 2021) and others show high level architectural survey (Bashaa et al.; 2025) , (Coronado et al.; 2022) . This thesis provides a complete reproducible design that includes how to operationalize an intelligent ZT framework on a public cloud. The successful end to end test confirm that the integration of MLOps, serverless computing and ML is practical and effective strategy in building self adaptive security systems. The use of SHAP for explainability is important, it provides the critical need for transparency in AI driven security as highlighted by (Wang et al.; 2024) . It provides confidence that the model learning is meaningful patterns rather than misleading correlations.

This study has a number of limitations that need to be mentioned. Firstly, the models have been tested on the static datasets and are not necessarily comprehensive enough to represent the dynamic cyber threats present in reality, potentially shifting toward online or ongoing learning to enhance adaptability. Secondly, although the system proved to be highly accurate, the experiments were done in a simulation setup and the outcome of the system applied in reality has not been tested in any way. For explainability SHAP is used but more global methods of interpretation like LIME or counterfactual analysis constraining complexity of decision-making transparency are not explored. The AWS audit logs were used to validate that the real-time policy enforcement did not receive high-load or adversarial stress validation. Finally, the threat types were fixed, narrowed and this can inhibit the generalization of new or zero-day attack patterns that can be executed by the system.

Despite of these, limitations the key contributions of this study are threefold. First one is time taken to apply the required policy after the threat is detected during simulation is less than a minute in all the ML models which reduces the response time significantly compared to time taken for manual application after the threat alert is received. Second is the integration of SHAP for model explainability. It provides transparency to the model into model's decision making process and addressing the black box issue in regards to the AI adoption in security. Third is the demonstration of end to end test confirmed that it is feasible to create a closed loop even driven system where ML predictions can automatically trigger the policy enforcement actions providing a tangible proof of concept.

# 7 Conclusion and Future Work

## 7.1 Conclusion

The study introduced the design, development and testing of a machine learning-based Zero Trust Security (ZTS) system that could identify both internal and external threats and active restrictions to the system as well as authorizations in real-time via serverless architecture. This method combines lightweight and scalable models, simulations of real-world data and enforcement policies, such as automated actions to overcome the main shortcomings of the current literature.

It tested three different pieces separately: threat classification with an external threat based on Random Forest and SVM. The identification of the internal threat through the Logistic Regression and direct automated enforcement in real-time with the help of AWS lambdas. All of them were strictly evaluated on the basis of precision, recall, F1-score and accuracy along with the resource measures. These findings showcased a high classification accuracy on every model with Logistic Regression performing best with 99.31 percent on the internal threat and Random Forest performing best on the external threat with 91.2 percent accuracy which confirmed the applicability of the selection of the models in their specific environments.

Results in comparative analysis of previous studies showed that the models that had been developed in the present thesis were superior than their benchmark models in general terms and consistency across datasets. Unlike the existing studies with their performance decrements on new data the pipelines of this work remained stable and still delivered more accurate predictions. Mostly because of better preprocessing, threat-specific modeling and strong validation measures.

The feasibility of the system was validated in terms of real-time simulation of real security incidents and the action of policy enforcement measures in AWS Lambda. It serves as a vital operational aspect to the current research as research effort frequently presents issues of end-to-end implementation and explainability to Zero Trust systems.

In summary, this piece adds to the reproducible and efficient Zero Trust framework that suits modern cloud architecture. It brings theoretical foundations and the practical implementation together. Whcih shows that explainable and lightweight ML models can be not only powerful but also resource-effective in terms of network security in modern networks.

## 7.2 Future Work

This experiment is well designed to test the key components of the system in isolation and then the entire system is tested all together. However there are certain limitations.

- Meaningful future work could extend this framework in several critical directions. A more sophisticated policy engine which uses reinforcement learning to dynamically learn the optimal security response based on the threat context and a predefined risk appetite moving beyond the current static and rule based approach.
- Use of dynamic dataset. Simulation or working model with live monitoring.
- In 'real-time' aspect the simulation is triggered manually. A true production system would need to be integrated with a real time event stream such as AWS Kinesis or direct network logs which would need to address further challenges related to data velocity and feature extraction at scale.
- Exploring more global tools for explainability such as LIME or counterfactual analysis .
- Performance of the model under high traffic and heavy load.

Implementation of the future work could create a superior self driven AI powered security system on the basis of currently developed scalable cloud based design.

**GitHub Repository:** `https://github.com/MaitreyeeM-student/Research`$_{project}$

# References

Abid, N. (2020). Advancements and Best Practices in Data Loss Prevention: A Comprehensive Review, *Global Journal of Universal Studies* **1**(1): 190–225. Cited by 19, [Accessed 13 July. 2025].

Ahmadi, S. (2024). Zero Trust Architecture in Cloud Networks: Application, Challenges and Future Opportunities. Cited by 44, HAL Open Science, [Accessed 16 June. 2025].

Almehdhar, M., Albaseer, A., Khan, M. A., Abdallah, M., Menouar, H., Al-Kuwari, S. and Al-Fuqaha, A. (2024). Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks, *IEEE Open Journal of Vehicular Technology* **5**: 869–906. Cited by 58, [Accessed 15 June. 2025].

Alshamrani, A., Ahmed, M. and Sukhatme, G. S. (2024). Threat Detection in Internet of Things Networks Using Deep Learning: A Comparative Analysis, *IEEE Internet of Things Journal* .

Awotunde, J. B., Imoize, A. L., Jimoh, R. G., Adeniyi, E. A., Abdulraheem, M., Oladipo, I. D. and Falola, P. B. (2023). AIoMT enabling real-time monitoring of healthcare systems: security and privacy considerations, *Handbook of Security and Privacy of AI-enabled Healthcare Systems and Internet of Medical Things*, pp. 97–133. Cited by 28, [Accessed 06 June. 2025].

Bashaa, M. H., Bhaya, W. S. and Al-aaraji, N. H. K. (2025). Integration of Zero Trust Architecture and Machine Learning for Improving the Security of Software Defined Networking: A Review, *Journal of Intelligent Informatics, Networking, and Cybersecurity* **1**(1): 1. Cited by 1, [Accessed 09 June. 2025].

Chalichalamala, S., Govindan, N. and Kasarapu, R. (2023). Logistic Regression Ensemble Classifier for Intrusion Detection System in Internet of Things, *Sensors* . doi - 10.3390/s23239583, Cited by 26, [Accessed 12 June. 2025].

Coronado, E., Behravesh, R., Subramanya, T., Fernandez-Fernandez, A., Siddiqui, M. S., Costa-Pérez, X. and Riggio, R. (2022). Zero touch management: A survey of network automation solutions for 5G and 6G networks, *IEEE Communications Surveys & Tutorials* **24**(4): 2535–2578. Cited by 94, [Accessed 29 May. 2025].

DeCusatis, C. and Zhang, L. (2016). Implementing Zero Trust Cloud Networks with Transport Access Control and First Packet Authentication, *IEEE Transactions on Cloud Computing* **8**. Cited by 100, CORE A, [Accessed 13 June. 2025].

Dhanushkodi, K. and Thejas, S. (2024). AI Enabled Threat Detection: Leveraging Artificial Intelligence for Advanced Security and Cyber Threat Mitigation, *IEEE Access* . Cited by 12, CORE A, [Accessed 24 June. 2025].

Dritsas, E. and Trigka, M. (2025). Machine Learning in Intelligent Networks: Architectures, Techniques, and Use Cases, *IEEE Access* . [Accessed 25 June. 2025].

Duchesne, L., Karangelos, E. and Wehenkel, L. (2020). Recent developments in machine learning for energy systems reliability management, *Proceedings of the IEEE* **108**(9): 1656–1676. [Accessed 13 July. 2025].

Grace, A. (2025). Intelligent Workflow Resilience in Multi-Cloud Architectures. [Accessed 19 June. 2025].

Jeyalakshmi, V., Balan, V. and Benitha, V. S. (2025). Security Issues in IoT: Perspective Review, *Computer Networks and Communications*, pp. 101–122. Cited by 200, [Accessed 18 June. 2025].

Khurshid, K., Khurshid, K., Hadi, M. U., Al Bataineh, M. and Saeed, N. (2025). Securing AIoT Surveillance: Techniques, Challenges, and Solutions, *IEEE Open Journal of the Communications Society* . [Accessed 03 July. 2025].

Kumar, P. (n.d.). Next-generation secure authentication and access control architectures: advanced techniques for securing distributed systems in modern enterprises. [Accessed 21 May. 2025].

Pujol, V. C., Donta, P. K., Morichetta, A., Murturi, I. and Dustdar, S. (2023). Edge intelligence—research opportunities for distributed computing continuum systems, *IEEE Internet Computing* **27**(4): 53–74. Cited by 54, [Accessed 20 June. 2025].

Rodigari, A., Liu, S. and Zhang, X. (2021). Zero Trust Performance in Multi-cloud Setups: Service Mesh Architectures like Istio, *International Journal of Network Security* **15**. Cited by 45, CORE B, [Accessed 12 June. 2025].

Shah, N., Bhagat, N. and Shah, M. (2021). Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention, *Visual Computing for Industry, Biomedicine, and Art* **4**(1): 9. Cited by 203, [Accessed 24 July. 2025].

Shaik, M. and Gudala, L. (2021). Towards Autonomous Security: Leveraging Artificial Intelligence for Dynamic Policy Formulation and Continuous Compliance Enforcement in Zero Trust Security Architectures, *African Journal of Artificial Intelligence and Sustainable Development* **1**(2). Cited by 2, African Science Group, [Accessed 22 May. 2025].

Tariq, S., Baruwal Chhetri, M., Nepal, S. and Paris, C. (2025). Alert fatigue in security operations centres: Research challenges and opportunities, *ACM Computing Surveys* **57**(9): 1–38. Cited by 11, [Accessed 15 July. 2025].

Vlachos, I. and Reddy, P. G. (2025). Machine learning in supply chain management: systematic literature review and future research agenda, *International Journal of Production Research* pp. 1–30. Cited by 4, [Accessed 31 May. 2025].

Walia, G. K., Kumar, M. and Gill, S. S. (2023). AI-empowered fog/edge resource management for IoT applications: A comprehensive review, research challenges, and future perspectives, *IEEE Communications Surveys & Tutorials* **26**(1): 619–669. Cited by 110, [Accessed 02 July. 2025].

Wang, S., Qureshi, M. A., Miralles-Pechuán, L., Huynh-The, T., Gadekallu, T. R. and Liyanage, M. (2024). Explainable AI for 6G use cases: Technical aspects and research challenges, *IEEE Open Journal of the Communications Society* **5**: 2490–2540. Cited by 46, [Accessed 13 June. 2025].