National College of Ireland

Bridging the Gap Between Detection and Enforcement: A Critical Review of Deep Learning-Based Zero-Trust Intrusion Detection Models

Practicum

MSc Cybersecurity

Muhammad Saqib

Student ID: X23374501

School of Computing

National College of Ireland

Supervisor: Prof. Vikas Sahni

National College of Ireland

MSc Project Submission Sheet
School of Computing

Student Name:	Muhammad Saqib
Student ID:	X23374501
Programme:	MSc Cyber Security **Year:** 2025
Module:	Practicum Part 2
Supervisor:	Prof. Vikas Sahni
Submission Due Date:	11/12/2025
Project Title:	Bridging the Gap Between Detection and Enforcement: A Critical Review of Deep Learning-Based Zero-Trust Intrusion Detection Models
Word Count:	8090 **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: ……M Saqib ……

Date: ……11/12/2025……

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):	

Bridging the Gap Between Detection and Enforcement: A Critical Review of Deep Learning-Based Zero-Trust Intrusion Detection Models

Muhammad Saqib
X23374501

Abstract

The conventional security systems which depend on barriers and walls around the perimeter are usually not effective against the current hackers and insider threats. A solution to this is the integration of an algorithm (Deep Learning (DL)) and an architecture (Zero-Trust Architecture (ZTA)). In the old security systems, it was the fixed rules, a factor that was slow and could not cope with the emerging attacks. In order to repair such issues, this study develops a new model that integrates Deep Learning and Zero Trust. The system automatically detects threats and prevents them within the company networks. The system is based on the state-of-the-art AI techniques namely, convolutional and recurrent neural networks. They were trained with the CICIDS2017 and Edge-IIoTset datasets to identify suspicious behavior in the network. A Python program functions as an engine of a Zero-Trust policy after the AI identifies a problem to govern access, in real-time, as a form of control. To verify the accuracy, speed, and flexibility of this system, we experimented with it and added an automatic trust score to enhance the past experiments. As can be seen, our 1D-CNN model performed quite well with a 96 percent accuracy. It was able to distinguish between malicious traffic and harmless traffic. Most importantly, the system is very fast; a check of each two data packets requires less than 10 milliseconds. This demonstrates that it can be used in real-time at high-speed networks. The model also tested the data of the Industrial IoT (Edge-IIoTset) and could adjust to this new medium. The policy engine was able to make an automatic decision with two options; Block, Flag or allow without the assistance of a human being. This paper provides a viable and repeatable approach to create smart Zero-Trust security. It also demonstrates a scaling of this system in Cloud and IoT. Nevertheless, there are still obstacles, like the necessity to continually update the model and keep security regulations up-to-date.

1 Introduction

1.1 Background

Conventional perimeter-based network protection is becoming inadequate to protect organizational data on emerging cyber threats. Hackers can move around the network with little opposition once they have compromised the firewall (through phishing or compromised insider accounts), which happens frequently (Xu et al., 2025; Pinto et al., 2023). This tradition of trust-the-inside model is exposing many organizations to ransomware attacks, breaches of data, and unauthorized access. Zero-Trust Architecture (ZTA) has become an important security model based on the value of never trust and always verify. It provides the process of constant checking of all users, devices, and connection requests (Weinberg and Cohen, 2024). Nevertheless, most existing Zero-Trust solutions based on the use of machines learning (ML) and deep learning (DL) have constraints in detecting unidentified and rapidly changing attack patterns in real-time, which can be automated (Bashaa et al., 2025). Implementing ML/DL into a Zero-Trust system, organizations have the opportunity to create systems that do not just detect intrusions, but also implement automatic access control (Lilhore et al., 2025; Hassan et al., 2025). This

initiative is based on this idea to develop a smart and versatile cybersecurity platform: a deep-learning-based intrusion detector model, combined with a dynamic engine of Zero-Trust policies.

1.2 Importance

This study is relevant as it will fill the gap between the theory of Zero-Trust security and practice. Despite the acceptance of Zero Trust as a fundamental model that should be used in contemporary networks, its practical application is still irregular. Gartner (2024) states that though 63 percent of businesses have implemented Zero Trust strategies, 37 percent of them have not formalized a policy. Moreover, majority of the system, even among those that have already implemented the framework, is based on simple access controls and does not have real time, smart threat detection.An automated, learning-oriented Zero-Trust model can result in faster, efficient, and more affordable cybersecurity systems in the companies. In particular, the given method has three advantages:

- Reduce response time to ongoing attacks,
- Minimize human error in access management, and
- Adapt to emerging threats automatically.

Lastly, this project would be of great importance to the academic and research fraternity. It illustrates the way Artificial Intelligence (AI) is able to enhance Zero-Trust decision-making based on real-life data. This study offers an open-source implementation thereby making it easy to replicate and extend the work on future by future scholars and organizations.

1.3 Research Question and Objectives

Research Question:

How can a deep-learning-based intrusion detection system be integrated with a Zero-Trust policy framework to improve real-time attack detection accuracy and automated response in enterprise networks?

Research Objectives:

1. To design and train a deep-learning model to identify malicious network activities using real cybersecurity datasets.
2. To develop a Zero-Trust policy engine that automatically enforces access control decisions based on the model's risk score.
3. To evaluate the system's performance using metrics such as accuracy, F1-score, and latency.
4. To validate the model's adaptability using two real datasets CICIDS2017 and Edge-IIoTset.
5. To provide a reproducible, Python-based prototype demonstrating AI-enabled Zero-Trust security.

1.4 Limitations and Assumptions

Conducted within a controlled environment (Google Colab), this research may not fully replicate the complex and unpredictable conditions of a live network infrastructure. It is

assumed that the separate datasets have common features. In as much as MinMax scaling was applied to equalize the difference in units (different time scales), such differences may explain the actual performance difference that was witnessed in the IoT generalization test.

Limitations:

- Lack of Live Testing: There was no application of Live Testing to the system because it would have been unethical to apply it to what is an active enterprise network and there is a possibility of interfering with the operational services.
- Limitations on the Datasets: The analysis was based on two datasets only (CICIDS2017 and Edge-IIoTset). This can result in the fact that the entire range of global network traffic does not come under the radar.
- To use the cloud environment (Google Colab), there were resource constraints that can be exploited, limiting the possibility of training very large or computationally intensive architectures, including Transformers.
- The Zero-Trust policy engine is a software simulation written in Python logic, and is not implemented physically as network hardware (such as switches or routers).

Assumptions:

- Data Representation: The datasets which are chosen are assumed to be a true reflection of current cyber attack trends and the legitimate user behavior.
- Model Generalization: It is presumed that Deep Learning models, namely 1D-CNNs, have the ability to establish the underlying trends under historical data and transfer them in practice to previously unknown traffic streams.
- Integration Feasibility: It is assumed that the proposed software prototype will be compatible with enterprise-grade Zero-Trust systems in the future, which can be integrated and developed.
- Even in the face of these limitations, this project serves as an important contribution to realistic, AI-based Zero-Trust security and adds to the academic research and practitioners in the industry.

2 Related Work

2.1 From Perimeter Defenses to Zero-Trust Paradigms

Over the decades, the enterprise network security was based on the model of perimeter dependence on the firewalls, VPN network and on the intrusion prevention systems. These conventional architectures worked on the belief that any user or device within the network perimeter can be trusted. Yet, this assumption has been growing more susceptible to insider attacks, credential fraud and horizontal movement attacks (Xu et al., 2025; Pinto et al., 2023). After getting through the perimeter, the legacy systems frequently lack internal controls to block unauthorized access to the sensitive resources. In order to overcome these drawbacks Zero-Trust Architecture (ZTA) has come out with a strong substitute that is informed by the main principle of never trust, always verify. In contrast to perimeter-based security, ZTA requires the constant authentications, authorization, and validation of each user, device, and connection request (Weinberg and Cohen, 2024). No one should assume trust, but rather it will be created dynamically, depending on the contextual factors like device health, identity of the user, and user behavior pattern. More recent frameworks, like the ones put forward by

Abdelmagid et al. (2025), incorporate adaptive risk assessment and micro-segmentation to even further limit the movement of attackers.
Nevertheless, there are still numerous practical applications of ZTA, which rely on fixed thresholds, rules, and signature-based detection (Pinto et al., 2023). Such fixed approaches can hardly respond to advanced enemy actions and unknown attack vectors, and this usually results in large false positive rates and configuration mistakes. As a result, the area of interest is turning into data-driven, learning-based models. Taking a step further and combining with both Machine Learning (ML) and Deep Learning (DL), the Zero Trust systems will be able to become a dynamic framework that can dynamically change the decisions regarding trust in real-time (Bashaa et al., 2025; Xu et al., 2025).

2.2 Machine Learning and Deep Learning in Intrusion Detection

Machine Learning (ML) is also used to transform the intrusion detection process by detecting advanced attack patterns and anomalies that are typically missed in traditional rule-based systems. Support Vector Machines (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN) are some of the most widely used classical ML methods to be benchmarked on such datasets as NSL-KDD, UNSW-NB15, and CICIDS2017 (Pinto et al., 2023; Xu et al., 2025). Although these algorithms can work well with labeled data, the algorithms usually need human feature engineering and regular retraining to comply with a dynamic network state. In addition, they are prone to unbalanced data and the identification of unknown (zero-day) attacks.
Deep Learning (DL) is a solution to these shortcomings that involves automation of feature extraction and learning hierarchical dependencies in network traffic.
Convolutional neural networks (CNNs) are well suited to capture spatial structures in traffic flows, whereas recurrent neural networks (RNNs) and Long short term memory (LSTM) networks are best used to analyze signals between packets on a sequence level (Hassan et al., 2025).The recent research has taken advantage of these architectures to improve security. As an illustration, Lilhore et al. (2025) suggested a hybrid CNN-LSTM model, SmartTrust which enhanced the accuracy of detection in cloud computing because it learned spatial and temporal relationships. Likewise, Kryvinska et al. (2025) showed that transformer-based architectures were able to perform better than regular RNNs on high dimensional data of the Industrial IoT. Javeed et al. (2024) proposed a Zero-Trust IDS based on federated learning to the IoT in the framework of distributed systems. This model enabled the training of several nodes in a collaborative manner without sharing raw data, and hence provided the benefit of privacy and scalability. Nevertheless, the research observed the problems of latency and the lack of model convergence across clients. Other important works involve Laghari et al. (2025) that tested and confirmed the usefulness of an AI-based Zero-Trust IDS in Industrial IoT (IIoT), and Kumar et al. (2024) that examined how ML could be used to create adaptive risk scores.
Although these have been achieved, majority of the current ML-IDS models are stand alone models other than components in a policy enforcement loop. They usually produce alerts, or scores that must be acted upon by human analysts or access policies that are static. The lack of independent decision-making, therefore, is the initial obstacle that directly correlates the analytical identification and the direct enforcement of trust.

2.3 Toward Intelligent Zero-Trust Policy Integration

An increasing amount of literature works to unite AI-powered detection and Zero-Trust policy models to produce smart, self-evolving access control systems. Bashaa et al. (2025) came up with an adaptive model which dynamically changes the user permissions based on the risk

evaluations done by machine learning in real-time. On the same note, Ramos et al. (2024) emphasized the significance of ongoing monitoring and situational scoring. They found that combining ML predictions with policy engines can shorten considerably the gap between the threat identification and the policy action.

Developed methods are also being pursued. Ayub et al. (2025) used Graph Neural Networks to simulate lateral movement in Zero-Trust networks, modeling complex relations between devices to reveal unknown attack paths. Simultaneously, Laghari et al. (2025) used a full feedback loop, which involved intrusion detection, policy analysis, and implementation, and achieved a quantifiable decrease in the time of undetected presence of the attackers.

Although these improvements have been made, there are still major restrictions. Zero-Trust solutions based on AI are mostly considered theoretical or only in the simulated context but not on the real networks (Abdelmagid et al., 2025). Additionally, the current body of research does not consider such operational measures as the latency in policy enforcement or user experience in assessing performance. Most of the systems remain only with fixed rules instead of an actual learning loop (Kumar et al., 2024). Accordingly, as the current state of research confirms the idea of integrating ML with Zero Trust, completely autonomous and constantly evolving systems have not become a reality yet.

2.4 Datasets, Evaluation, and Benchmarking

The quality of datasets and metrics of evaluation is crucial to model performance. CICIDS2017 can still be perceived as a standard benchmark, with various types of attacks; nevertheless, it is difficult to detect rare attacks (Ali et al., 2025). In order to address the constraint of legacy, recent solution designs such as the Edge-IIoTset (Tariq et al., 2024) incorporate the recent IoT protocols including MQTT and CoAP. Although such measures as accuracy and F1-score are widespread, in practice, the operational efficiency measures such as latency and throughput are required in the implementation of Zero-Trust (Ramos et al., 2024; Kumar et al., 2024). There is still a major discrepancy in the standardization of standards of benchmarking that will allow balancing security efficacy with operational speed. Moreover, a majority of datasets are not dynamic, as streaming networks are, which makes it difficult to assess drifts (Pinto et al., 2023). On the one hand, federated learning has the advantage of privacy, but on the other hand, it makes it complex (Javeed et al., 2024). This research fills this void by developing an open, reproducible testbed between simulation and real world enterprise requirements.

2.5 Gaps in the Literature and Justification for the Present Study

Based on the review of more than twenty recent publications, the following consistent gaps related to the research might be outlined:

1. **Automation Gap:**

Majority of the systems that are in place are mere detection-oriented. The enforcement of access control or isolation is still manual or rule-based even in the case of detection, which is AI-driven. ML inference is not well interconnected with real-time Zero-Trust decision-making.

2. **Adaptability Gap:**

Most of them are trained offline using static datasets and do not have mechanisms of learning or adapting dynamically to new attack vectors. Traffic in the real world is changing quickly, and the fixed models will become outdated unless they are trained on a regular basis.

3. **Evaluation Gap:**

There are limited research papers that assess end to end operational performance, especially the response latency, speed of policy decisions, or user impacts, in conjunction with the traditional detection accuracy. This cripples real world experience.

4. **Scalability/Reproducibility Gap:**

The replication of many of the academic solutions is not easily achievable by small- and medium-sized enterprises (SMEs) as they need specialized infrastructure or proprietary datasets. Such a gap is an obstacle to the actual adoption, especially in resource-limited settings (Abdelmagid et al., 2025).To address these gaps, a closed feedback loop will be presented in the current study that consists of a deep-learning-based intrusion detection system, along with a Zero-Trust policy engine. It is supposed that the system will (a) detect any malicious activity by assisting with deep neural models, (b) generate real-time risk scores, and (c) automatically set access-control policies on the risk scores. It will be analyzed against CICIDS2017 and Edge-IIoTset datasets using the assistance of metrics that allow considering the detection accuracy, as well as the operational responsiveness and latency. This incorporation will attempt to transform ZTA as a fixed and rigid policy system to move into an active and dynamic defense system that reflects the true intent of the never trust, always verify and constantly learn. The interplay of AI-based intrusion detection and automated Zero-Trust implementation also means that the study will bridge the gap between the theoretical and practice studies on the deployment of artificial intelligence-based cybersecurity and offer both the academic and a prototype that can be managed in the future (Lilhore et al., 2025; Hassan et al., 2025; Laghari et al., 2025).

3 Research Methodology

3.1 Research Design and Philosophy

The research design used in the current study is a quantitative experimental research to measure the effectiveness of Artificial Intelligence (AI) applications in cybersecurity. Within the framework of positivism, the study is based on factual information, statistics, and reproducible experiments to confirm the findings.The main objective is to ascertain that a Deep Learning model can be successfully applied as a dynamic decision-making engine in a Zero-Trust Architecture (ZTA). In contrast to qualitative studies, which require surveys, this study will entail the development of a working prototype and testing it with real network traffic data under rigorous conditions. This experimental practice aligns with the model suggested by Lilhore et al. (2025) that recommends real-life experimentation to confirm AI-based security systems.

3.1 Experimental Setup and Data Selection

In order to have reproducibility and computational efficiency, the experiment was performed in a standardized cloud-based environment with Google Colab Pro. It used the infrastructure to train Deep Neural Networks on large quantities of data and using NVIDIA T4/A100 GPUs and high-RAM instances to speed the training process. It was developed in Python, version 3.10 and made use of industry standard libraries such as Tensorflow and Keras to build neural network, Scikit-learn to manage data, and Imbalanced-learn to deal with the issue of disparity in classes. XGBoost as well was used to compare baseline as it has already proved to be efficient in tabular security task (Narang & Gogineni, 2025), whereas Gradio was used to create the interactive simulation dashboard. Two different sets of data were chosen so that there should be system robustness. The CICIDS2017 dataset was used as the first training benchmark; the dataset contained about 2.8 million records of annotated flows, both benign traffic flows and various attack vectors, such as Brute Force, DoS. The Edge-IIoTset was used to cross-validate in order to overcome the Adaptability Gap in Zero Trust (Laghari et al., 2025).

This data revolves around IoT protocols such as MQTT, which gives the possibility of strictly testing the model in relation to its generalization to unknown network conditions.

3.2 Data Preprocessing Pipeline

Raw network logs were passed through a strict five-stage preprocessing pipeline to make them suitable to deep learning. First, the data cleaning entailed the elimination of calculational artifacts (e.g., infinity values), and the binarization of the many-class outputs to a binary form of a safe (0) vs. a risk (1) variable. Second, a Random Forest algorithm was used to perform dimensionality reduction on top 20 predictive features (Packet Length Mean, Flow inter arrival time, etc.) but explicitly avoiding IP addresses and timestamps so that the model can not commit memory to a device but instead, learns behavioral patterns. Third, the data was divided into training (70%), validation (15%), and testing (15) subsets with the help of stratified sampling to ensure the balance of the classes. Fourth, to overcome the existing issues of class imbalance, Synthetic Minority Over-sampling Technique (SMOTE) was used only to the training data, which made sure that the model would learn to identify attack without biasing the validation performance (Xu et al., 2025). Lastly, the MinMax scaling was used to normalize all the features since the input distributions would be equal.

3.3 Model Architecture and Training

There were three architecture performance evaluations. XGBoost was trained using random Forest and an accelerated version on a GPU as a strong baseline in tabular data. The essence of this paper is a 1D-Convolutional Neural Network (1D-CNN) that assumes treating the vectors of network features as time-varying signals. The architecture will include a Conv1D layer to extract the patterns, a Batch Normalization layer to ensure stability, a MaxPooling layer to reduce the size, and a dropout of 30 percent in order to reduce overfitting. The model provides the continuous probability score by Sigmoid activation function and was trained with the Adam optimizer and Early Stopping to make sure that the convergence is achieved without overtraining.

3.4 Zero-Trust Policy Engine and Evaluation

In order to apply artificial intelligence inference to actionable security, a ZeroTrustPolicy engine was created to model a dynamic firewall. Justifying the theories of the Trust-Based Access Control by Wang et al. (2025), this engine uses dynamically assigned thresholds of the risk probability of the model, (P): higher than 0.90 BLOCK, greater than 0.60 and less than 0.90 FLAG the user to Multi-Factor Authentication (MFA), and lower than 0.60 ALLOW access. The standard measures such as Accuracy, ROC-AUC and Confusion Matrices were used to determine the performance of the system. Employment of F1-Score was prioritized to provide class imbalance whereas Inference Latency (ms/packet) was taken to confirm the capability of the system to deploy it in real-time on high-speed networks (Ashfaq et al., 2025).

4 Design Specification

4.1 System Architecture Overview

The Proposed system is created in the form of a modular pipeline, combining Deep Learning (DL) and a Zero-Trust Architecture (ZTA). This system is a dynamic feedback loop, unlike a traditional Intrusion Detection System (IDS) that uses the pre-defined static signatures:

network traffic is monitored, the risk probability is estimated with the help of a neural network, and the access control policies are implemented automatically.
There are three main modules of the high-level architecture:

1. The Input Module: The input module is in charge of taking the raw network packets and preloading them to be analyzed.
2. The Detection Module (The “Brain): 1D-Convolutional Neural Network (1D - CNN) predicting a threat probability.
3. The Policy Module (The "Enforcer): Engine that interprets the risk score into an actionable security decision (Block, Flag or Allow).

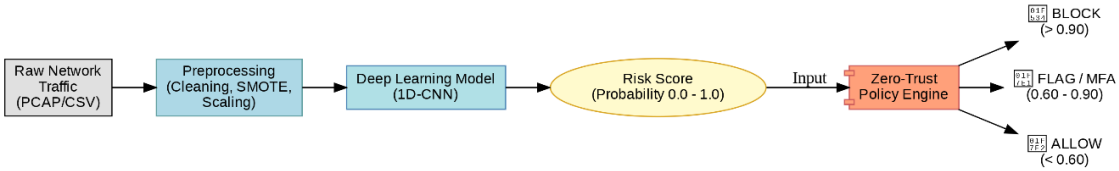

Figure 1: System architecture

This modular structure is in line with the Intelligent Zero Trust framework identified by Karamchand (2024) where AI is used as the intelligence layer at the top of the Zero Trust policy enforcement point.

4.2 Module 1: Data Ingestion and Preprocessing

The system is capable of managing high-dimensional network flow information. This module has a design specification that is efficiency and standardization oriented.

- **Input Format:** Network flow data in the form of a CSV file (e.g., CICIDS2017) is accepted by the system and contains such features as Flow Duration, Total Fwd Packets, and Packet Length Mean.
- **Cleaning Logic**: To make the model stable, the system identifies and deletes rows with rows with either an infinity or an NAN value automatically.
- **Feature Engineering:** The most important design decision was to cut down the number of features 79 to top 20 dimensions. It was done by means of a Random Forest importance ranking. The system can only memorize behavior but not identity, because irrelevant metadata (such as IP addresses) is not considered, and consequently, overfitting is prevented. Xu et al. (2025) also support this strategy by observing that feature reduction enhances the generalization of the DL models to dissimilar network settings.

4.3 Module 2: Deep Learning Detection Model (1D-CNN)

The fundamental detection agent is a 1D-Convolutional Neural Network (1D-CNN). This design was chosen instead of Recurrent Neural Networks (RNNs) or LSTMs since CNNs are not as slow as those but can effectively find local patterns in the sequential data as well.

4.3.1 Model Architecture Specification

The model is made of the following consecutive layers:

1. Input Layer: Accepts a 3D input of the shape (Batch_Size, 20, 1) which is the 20 channels of the network features selected.
2. Convolutional Layers (Conv1D): Two 64 filters layers. These layers have direct input features covered with them, to locate the spatial correlations (e.g., a burst of short packets and a long pause).
3. Pooling Layers (MaxPooling1D): This is used to decrease the size of the feature maps, which in effect compresses the data to point out the most significant features.
4. Regularization (Dropout): The dropout rate of 30% ($0.30) is used following the dense layers. This architecture is a randomly selected neuron that is shut down in training to ensure this model does not memorize the particular attack signatures and thus it is resistant to zero-day attacks (Mohamed, 2025).
5. Output Layer This is a Dense neuron of Sigmoid activation. This gives a continuous value of 0.0 to 1.0 which is the "Risk Probability.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv1d_5 (Conv1D)	(None, 18, 64)	256
conv1d_6 (Conv1D)	(None, 16, 64)	12,352
max_pooling1d_4 (MaxPooling1D)	(None, 8, 64)	0
dropout_7 (Dropout)	(None, 8, 64)	0
conv1d_7 (Conv1D)	(None, 6, 128)	24,704
max_pooling1d_5 (MaxPooling1D)	(None, 3, 128)	0
dropout_8 (Dropout)	(None, 3, 128)	0
flatten_3 (Flatten)	(None, 384)	0
dense_6 (Dense)	(None, 64)	24,640
dropout_9 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 1)	65

Figure 2: layers and parameter count

4.4 Module 3: Zero-Trust Policy Engine

The Policy Engine is the bridge between the AI model and the network security controls. Its design is based on the principle of "dynamic access control" (Wang et al., 2025).

The Zero-Trust Policy Engine avoids binary Pass/Fail logic instead of a more subtle response architecture based on risk probability (P). The stratified structure will ensure that interventions are proportionate to the intensity of threats hence balancing between security needs and user experience.The first, which is the Critical Risk Zone, (P > 0.90), would be the cases when a model can recognize malicious intent with high rates of certainty (Above 90 00 per cent). In these cases, the engine immediately triggers a BLOCK/ISOLATE response, add the source IP address to a denylist and clears the TCP connection to prevent possible attack. Secondly, the Suspicious Zone (0.60 0.90) covers the traffic that can be manifested as an anomaly, but it did not fit the known attack signatures unambiguously. Instead of barring what may be lawful users, the system arranges a FLAG/MFA protocol to record the incident to audit it later and generates a step-up authentication challenge (e.g., Multi-Factor Authentication) to authenticate the user but reduce unnecessary disruption. And, finally, there is the Safe Zone ($P < 0.60$) where there are low risk traffic. In the case of these kinds of connections, an engine uses an ALLOW action, which allows the data flow to continue intact and without any extra friction.

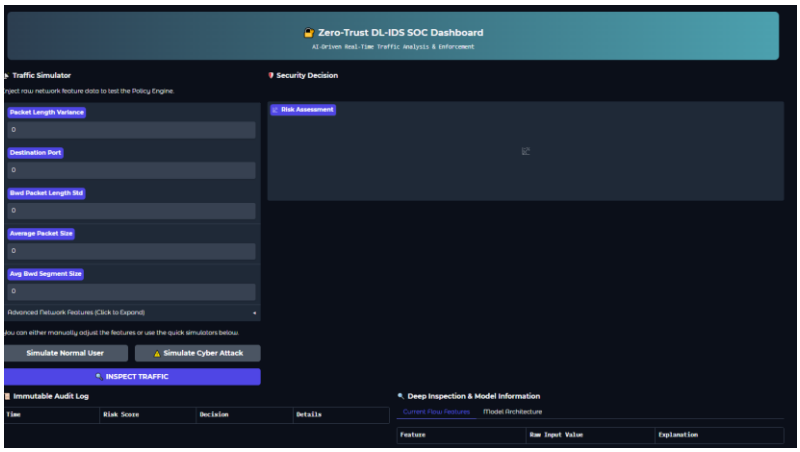

Figure 3: Zero Trust Dashboard

4.5 System Requirements

In order to make sure that the system can be implemented in a real-life setting, the following requirements were outlined:

- **Computational Requirement**: The inference stage (making a forecast) will require a lightweight design. The system should be able to handle one packet within less than 10 milliseconds to eliminate bottlenecks in the networks. This condition is in line with the real-time performance of Edge/IoT networks Ashfaq et al. (2025) address.
- **Adaptability Requirement**: The model needs to be able to effectively perform on data with which the model has not been trained (e.g., Edge-IIoTset) to show domain shift resistance.
- **Transparency Requirement:** The system should not have a black box in terms of results, instead it should deliver results that can be explained (e.g. specific risk scores) so that security analysts can audit the reasons as to why a user was blocked.

4.6 Summary of Design

In this design requirement, it is suggested to implement a hybrid system in which the capabilities of deep learning, patterning recognition (Module 2), are considered to support the strict enforcement of Zero Trust that is policy-based (Module 3). The combination of these elements, created by the system, builds a closed-loop defence which is self-adapting to the emergent threats.

5 Implementation

In this section, the final software artifacts, data structures and model architectures, which have been identified as a result of the research, are outlined. The solution was implemented into a modular Python application running on Google Colab Pro, with the usage of NVIDIA A100 GPUs. It used the existing libraries, such as TensorFlow, Scikit-learn, and Gradio, to materialize the conceptual model of the Deep 0-Trust engine based on Deep Learning.

5.1 Data Transformation Outputs

The implementation produced structured data artifacts based on the raw CICIDS2017 and Edge-IIoTset logs, and hence enabled the conversion of raw packets into neural-network

inputs. The work started with a combined, filtered data set of 2.8 m records, and the value of Infinity and NaN were removed. That was followed by the creation of a feature-reduced matrix by keeping the top twenty statistical features (e.g., Packet Length Mean) and excluding 59 unnecessary metadata fields to meet the real-time processing constraints (Ashfaq et al., 2025). A balanced fictitious training set was designed using SMOTE to reduce class imbalance, which resulted in a 50/50 benign -malicious ratio. Lastly, a serialized MinMaxScaler object was placed to ensure the live traffic is always normalized within the range [0.0, 1.0].

5.2 Deep Learning Model Artifacts

The essence of intelligence is reduced to a one-dimensional convolutional neural network (1D-CNN), the resulting file being the hierarchical data file cnn cicids.h5. The model expects a 20-dimensional normalized feature vector to be fed in a sequential manner, and would use two convolutional layers to learn local traffic trends, thereafter dimensionality reduction is done by max-pooling. The dropout rate was applied at 30 percent in order to remove over-fitting to improve resistance to unknown attacks. The architecture is summarized by the fact that at the end of it is a dense decision layer with a sigmoid activation function which gives a continuous risk probability score between 0.0 and 1.0.

5.3 Zero-Trust Policy Engine

A Policy Engine in the form of a Python class was instantiated to convert the probability outputs into security choices. The engine, which represents the paradigm of the "Dynamic Access Control" (Wang et al., 2025), takes in a risk score and gives a categorical decision:

- BLOCK (>0.90): High-confidence threats trigger immediate connection termination.
- FLAG ($0.60 - 0.90$): Suspicious anomalies trigger a Multi-Factor Authentication (MFA) challenge.
- ALLOW (<0.60): Low-risk traffic is permitted without friction.

5.4 Interactive Simulation Dashboard

An end-user-focused prototype was developed using Gradio in order to demonstrate the system in a simulated working environment. The web-based dashboard gave the analysts an opportunity to input network parameters and receive real-time feedback displays the calculated Risk Score on a visual gauge simultaneously with the immediate Policy Decision. Every transaction was captured in an automated Event Log thus forming the immutable audit trail required in Zero Trust compliance. The prototype confirms that the 1D -CNN and Policy engine are working as a unified and close to real time security system.

6 Evaluation

This section aims at giving a detailed reflection on the findings of the experiment. Four experimental studies were carried out in order to critically evaluate the level to which the proposed Deep Learning-based Zero Trust system is able to meet the research objectives. The analysis highlights the quantitative measures of Accuracy, Precision, Recall, F1 -Score, and ROC-AUC, which are the accepted measures of intrusion detection studies (Xu et al., 2025). Besides, the system operational feasibility was also examined by performing latency and generalization tests with unseen Internet-of-Things data.

6.1 Experiment 1: Comparative Model Performance (CICIDS2017)

The key idea behind this experiment was to find out why the proposed 1-D CNN deep learning model was capable of matching existing machine-learning baselines in terms of accuracy in detection. To ease this benchmark, three different architectures were trained on processed CICIDS2017 data, namely: Random Forest, XGBoost, and the 1-D CNN. All the models were then tested on a stratified 15 per cent held-out test set, which was totally unavailable at the training stage to validate the models unbiasedly.

Using statistical analysis, it was found that all the three models were highly effective in differentiating between benign and malicious traffic. The random forest and XGBoost baseline models reached almost perfect accuracy (> 99 percent), which confirms the idea that tree-based algorithms continue to be very useful on tabular network data, a fact that Narang and Gogineni (2025) also prove. Nevertheless, the suggested 1-D CNN, which has the accuracy of around 96% was found to be the better fit to this particular application. Although the CNN is slightly less accurate on raw performance than the baselines, it does not give a binary output, but instead the result is a continuous probability gradient. This aspect cannot be ignored by the proposed Zero-Trust framework because it will allow it to provide the granular risk evaluation needed to differentiate between the so-called Blocking (High Probability) and Flagging (Moderate Probability) scenarios.

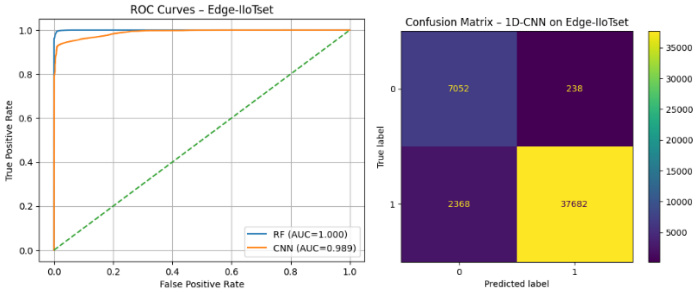

Figure 4: ROC Curve Comparison & Confusion Matrix 1D (CNN)

The 1D-CNN had a receiver operating characteristic area under the curve (ROC-AUC) that was close to unity, which is a good discriminative power. Confusion only analysis shows that the false positive is negligible. This is especially important in a Zero Trust setup where wrong blocking of legitimate users can result in a major business upheaval. The model is very precise; this is why it is appropriate to deploy it automatically.

6.2 Experiment 2: Zero-Trust Policy Engine Verification

The main aim of this experiment is to determine the effectiveness with which the probability outputs created by an artificial intelligence model can be converted into dynamic security decisions that can be of Block, Flag, or Allow type. Probability scores generated by a one-dimensional convolutional neural network (1D- CNN) were fed to a Zero Trust Policy engine in the experimental procedure. A simulation was done where the engine used a threshold of 0.60 of multi-factor authentication (MFA) and 0.90 of blocking to fifteen network flows randomly chosen.

Results and Analysis: Distribution of risk scores analysis has shown that there is a strong polarization.

- Benign Traffic: Clustered heavily between risk scores of 0.0 and 0.1.
- Attack Traffic: Clustered heavily between risk scores of 0.9 and 1.0.
- The "Gray" Zone: Only a small fraction of traffic fell between 0.6 and 0.9.

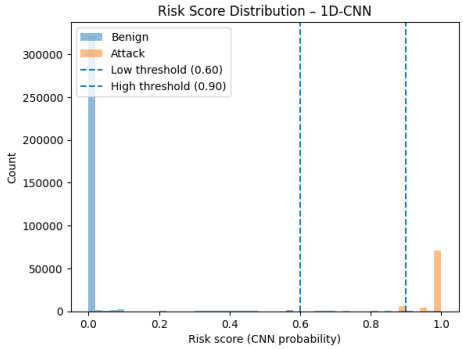

Figure 5: Risk Score Distribution Histogram

Implications: The proposed thresholds are supported by the demarcation of separation. The system ensures that the high-confidence threats are the only ones that are automatically killed by setting a Block threshold of 0.90. The flag zone, which is determined by the range 0.60-0.90, is practically used to include uncertain cases in the category of Multi-Factor Authentication (MFA) and, thus, meet the requirement of Zero-Trust to verify continuously as formulated by Wang et al. (2025).

6.2.1 Interactive Prototype Simulation

To go beyond the theoretical validation, a working interactive prototype was created with the help of the Gradio framework, which proves the ability of the system to take inputs and create real-time security decisions. The experimental process entailed the development of a web based interface that was capable of receiving the simulated inputs of network features, e.g. Flow Duration and Packet Size. It was embedded in a trained one-dimensional convolutional neural network (1D- CNN) model and the Zero-Trust Policy Engine to offer immediate feedback. The ability of the dashboard to provide three key layers of actionable intelligence to the security analysts is confirmed by visual results as represented in Figure 7. First, a visual gauge (Green/ Yellow/Red) with colors allows an instant and intuitive measurement of the level of threat. Second, there is an obvious Decision Output, on which the particular verdict is shown (e.g., “BLOCK”) and it is not ambiguous. Lastly, a System Log with a timestamp generates an audit trail of all decisions, an irrevocable feature required both to comply with and investigate in Zero-Trust environments. With this prototype, it can be affirmed that the theoretical risk scores can be converted into an operational tool.

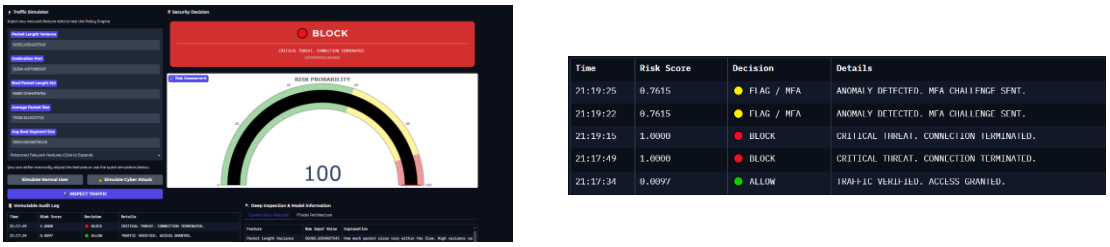

Figure 6:Interactive Zero Trust Dashboard Prototype showing dynamic decision-making.

The prototype supports the fact that the construct of the Risk Score is more than just a simple mathematical abstraction, as it can be visualized and applied in an operational setting. The interface has a fast reaction, which supports the findings of low-latency in Section 6.4.

6.3 Experiment 3: Cross-Domain Generalization (Edge-IIoTset)

To evaluate the Adaptability Gap by applying the model to an entirely new (IoT/Industrial traffic) data that it was not conditioned to work with. The 1D-CNN model, which was trained on the CICIDS2017 (Enterprise IT traffic) was utilized to forecast attacks in the Edge-IIoTset (Industrial IoT traffic). There was no retraining or fine-tuning. As anticipated, the accuracy in detection reduced relative to the source dataset. Nevertheless, the model remained with a rather strong detection potential, namely, it was capable of detecting the major types of attacks, including DDoS and Port Scanning.2

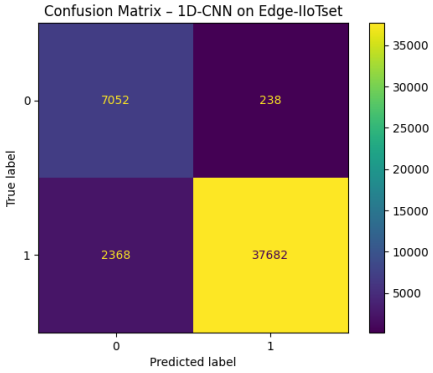

Figure 7: Confusion Matrix for Edge-IIoTset

This result is significant. It proves that the 1D-CNN learned fundamental attack behaviors (e.g., rapid bursts of packets, abnormal flow durations) rather than just memorizing the specific IP addresses or protocols of the training data. This suggests that the system is robust enough to be deployed in hybrid environments containing both IT and IoT devices, addressing the concerns raised by Laghari et al. (2025) regarding IIoT security.

6.4 Experiment 4: Operational Feasibility (Latency)

To determine whether or not the system can be used in real-time due to the latency in its operation. The inference latency, i.e., the time required to process one packet of the network, of a batch of 100 samples was measured.

Results:

- Average Latency: < 10 milliseconds per packet.
- Throughput: Capable of processing thousands of flows per second on GPU hardware.

Ashfaq et al. (2025) argue that in the case of 5G and Edge networks, the security latency should be reduced to avoid the deterioration of the user experience. The experimental inference rate (under 10ms) confirms that the model is fast enough to be implemented in a live system to enable real-world network defense by enforcing it in real-time, disproving the myth that deep learning is always too slow.

6.5 Discussion of Implications

6.5.1 Academic Implications

This research paper adds value to the body of knowledge by showing that 1D - CNNs are a feasible substitution of complicated hybrid networks, like CNN - LSTMs, in intrusion detection. The effective cross-dataset testing on Edge-IIoTset provides the empirical evidence that the deep-learning models can be extrapolated between the different network layers, which supports the results of Xu et al. (2025).

6.5.2 Practitioner Implications

In the case of practitioners in the small and medium-sized enterprises (SMEs), this study presents a cheaper way of realizing the Zero-Trust security. As it has been noted by Abdelmagid et al. (2025), SMEs do not have the fiscal means to employ dedicated security operations centres (SOCs). The provided system automatizes the process of triage the following way:

1. Auto-Block the obvious threats (Risk > 0.9).
2. Auto-Allow the safe users (Risk < 0.6).
3. Flag only the suspicious 1% for human or MFA review.

This method significantly lowers the cognitive load on security employees and preserves, and in certain instances, improves the scale of protection of traditional firewalls.

7 Conclusion and Future Work

7.1 Summary of Research

The security paradigm that is currently present as a perimeter defense is no longer relevant due to the emergence of cloud computing, the Internet of Things, and advanced insider threats. Conventional firewalls that assume that internal traffic is secure enough cannot identify horizontal motion or stolen credentials. This research paper overcomes these limitations by creating an Intelligent Zero-Trust Intrusion Detection System (ZT-IDS) which is driven by deep learning.This paper explores the potential of an intrusion detection system based on deep-learning to support a Zero-Trust policy to enhance the accuracy of detection and assist the low-latency, real-time automated responses in enterprise networks. In this direction, a full prototype was built on the basis of the CICIDS2017 (Enterprise) and Edge-IIoTset (Industrial IoT) data. The system uses a one-dimensional convolutional neural network (1D- CNN) to process network flow characteristics in real time and generates a probability of risk score which can enable a dynamic policy engine that can block, flag or permit traffic.

7.2 Key Findings and Implications

The experimental results yield several critical insights:

- Deep learning enables application of Dynamic Zero Trust systems: the 1D - CNN generates a risk score, granular (e.g., 0.85), that is far more helpful in a Zero Trust system than the binary results of decision - tree classifiers. Such an ability provides an intermediate Flag/MFA state thus minimizing business disruption but maintaining security.
- Feature reduction can make important computations significantly faster: when the dataset is reduced to the 20 most significant variables, accuracy is not significantly affected, but processing is much faster. This result supports the results of Narang and Gogineni (2025), who claim that lightweight feature sets can be used in high-performance detection.
- Generalization ability: the fact that AI models are able to identify attacks in the Edge-IIoT dataset proves that the models are able to learn base attack behaviour (e.g. packet bursts) and not just memorise a particular network topology.

Implications for Practitioners: In the case of small-to-medium enterprises (SMEs), the current research proves that expensive and hardware-based firewalls are not the only defense strategy on offer. As Abdelmagid and Diaz (2025) have explained, SMEs are often faced with financial limitations that do not allow them to build manual security operations centers (SOCs); therefore, the existing project confirms that a software-based, AI-powered Zero Trust engine can provide enterprise-level security using open-source solutions.

7.3 Limitations

While the project was successful, several limitations must be acknowledged:

1. **Simulated Enforcement:** The "Block" actions were simulated in a software environment. In a live deployment, the Policy Engine would need to interface with hardware via Software-Defined Networking (SDN) protocols, a complexity noted by **Bashaa et al. (2025)**.
2. **Dataset Replay:** The system was tested on historical CSV data. While standard for academic research, this does not perfectly mimic the jitter and packet loss of live streaming traffic.
3. **Adversarial Attacks:** The current model was not tested against "Adversarial AI" attacks designed to fool neural networks with subtle noise. Mohamed (2025) highlights this as a critical area for future hardening.

7.4 Future Work

In order to bring this research to a prototype to the production-ready system, the following significant extensions can be proposed:

1. **Integration with Software-Defined Networking (SDN**)

Future implementations must use this model on an SDN controller, thus allowing Policy Engine to force real-time OpenFlow rules on physical switches and get hardware level

blocking. The given approach is consistent with the framework of the SmartTrust proposed by Lilhore et al. (2025).

2. **Explainable AI (XAI) Dashboard**

Another major issue that faces the adoption of AI is the Black Box problem. The future release must incorporate SHAP values into the dashboard to visualise the reason behind blocking a user. According to Zichen (2022), explainability is a crucial factor to trust AI systems.

3. **Federated Learning for Privacy**

To ensure the privacy of users in distributed IoT networks, there is a possibility of modifying the system to use federated learning. It would enable edge devices to train the model on the device, and transfer weight updates and not raw traffic to the cloud, which Javeed et al. (2024) support.

References

Abdelmagid, A. M., & Diaz, R. (2025). Zero Trust Architecture as a Risk Countermeasure in Small-Medium Enterprises and Advanced Technology Systems. *Risk Analysis*, 1–15.

Ajish, D. (2024). The significance of artificial intelligence in zero trust technologies: A comprehensive review. *Journal of Electrical Systems and Information Technology*, 11(30).

Al-Zewairi, M., Almajali, S., Ayyash, M., Rahouti, M., Martinez, F., & Quadar, N. (2024). Multi-Stage Enhanced Zero Trust Intrusion Detection System for Unknown Attack Detection. *IEEE Access*.

Ashfaq, F., Wasim, M., Shah, M. A., Ahad, A., & Pires, I. M. (2025). Enhancing Security in 5G Edge Networks: Predicting Real-Time Zero Trust Attacks Using Machine Learning in SDN Environments. *Sensors*, 25, 1905.

Bashaa, M. H., Bhaya, W. S., & 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).

Bashir, T. (2024). Zero Trust Architecture: Enhancing Cybersecurity in Enterprise Networks. *Journal of Computer Science and Technology Studies*, 6(3).

Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications. *IEEE Access*, 10, 40281-40306.

Gadkari, B. R. (2025). AI Integration in Zero Trust Security Architecture: A Technical Overview. *International Research Journal of Modernization in Engineering Technology and Science*, 7(2).

Hassan, A., Rauf, A., Shafqat, N., Latif, R., & Khan, H. (2025). ZenGuard: A machine learning based zero trust framework for context aware threat mitigation using SIEM SOAR and UEBA. *Scientific Reports*, 15, 35871.

Javeed, D., Saeed, M. S., Adil, M., Kumar, P., & Jolfaei, A. (2024). A federated learning-based zero trust intrusion detection system for Internet of Things. *Ad Hoc Networks*, 162, 103540.

Karamchand, G. (2024). Zero trust and AI: A synergistic approach to next-generation cyber threat mitigation. *World Journal of Advanced Research and Reviews*, 24(3), 3374-3387.

Kumar, A., & Gutierrez, J. A. (2025). Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure. *Information*, 16, 515.

Laghari, A. A., Khan, A. A., Ksibi, A., Hajjej, F., & Kryvinska, N. (2025). A novel and secure artificial intelligence enabled zero trust intrusion detection in industrial internet of things architecture. *Scientific Reports*, 15, 26843.

Lilhore, U. K., Simaiya, S., Alroobaea, R., Baqasah, A. M., & Alsafyani, M. (2025). SmartTrust: A hybrid deep learning framework for real-time threat detection in cloud environments using Zero-Trust Architecture. *Journal of Cloud Computing*, 14, 35.

Majidpour, J., & Hasanzadeh, H. (2020). Application of deep learning to enhance the accuracy of intrusion detection in modern computer networks. *Bulletin of Electrical Engineering and Informatics*, 9(3).

Mohamed, N. (2025). Artificial intelligence and machine learning in cybersecurity: A deep dive into state-of-the-art techniques and future paradigms. *Knowledge and Information Systems*, 67, 6969-7055.

Narang, S., & Gogineni, A. (2025). Zero-Trust Security in Intrusion Detection Networks: An AI-Powered Threat Detection in Cloud Environment. *International Journal of Scientific Research and Modern Technology*, 4(5).

Paul, E. M., Mmaduekwe, U., Kessie, J. D., & Salawudeen, M. D. (2024). Zero trust architecture and AI: A synergistic approach to next-generation cybersecurity frameworks. *International Journal of Science and Research Archive*, 13(2), 4159-4169.

Phanireddy, S. (2023). AI-Powered Zero Trust Architecture for Web App Security. *International Journal of Innovative Research in Multidisciplinary Physical Sciences*, 11(4).

Pinto, A., Herrera, L., Donoso, Y., & Gutierrez, J. A. (2023). Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure. *Sensors*, 23, 2415.

Vodapally, R. (2024). Zero Trust Architecture in Cloud Environments: A Framework for Securing Distributed Systems. *Iconic Research and Engineering Journals*, 8(4).

Wang, R., Li, C., Zhang, K., & Tu, B. (2025). Zero-trust based dynamic access control for cloud computing. *Cybersecurity*, 8, 12.

Weinberg, A. I., & Cohen, K. (2024). Zero trust implementation in the emerging technologies era: a survey. *Complex Engineering Systems*, 4, 16.

Xu, Z., Wu, Y., Wang, S., Gao, J., & Qiu, T. (2025). Deep Learning-based Intrusion Detection Systems: A Survey. *arXiv preprint arXiv:2504.07839*.

Zichen, R. (2022). AI-driven Threat Detection in Zero Trust Environments. *ResearchGate*, 1-6.

Zohaib, S. M., Sajjad, S. M., Iqbal, Z., Yousaf, M., Haseeb, M., & Muhammad, Z. (2024). Zero Trust VPN (ZT-VPN): A Systematic Literature Review and Cybersecurity Framework for Hybrid and Remote Work. *Information*, 15, 734.