National College of Ireland

Comparing Zero Trust Model with Traditional Network and Machine Learning Enhancement in OpenZITI

MSc Research Project
MSc Cybersecurity

Elsamma Joshy
Student ID: 23171847

School of Computing
National College of Ireland

Supervisor: Arghir Nicloae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Elsamma Joshy
Student ID:	X23171847
Programme:	MSc Cybersecurity **Year:** 2024-25
Module:	MSc Research Project
Supervisor:	Arghir Nicolae Moldovan
Submission Due Date:	18/12/2024
Project Title:	Comparing Zero Trust Model with traditional network and Machine Learning enhancement in OpenZiti
Word Count:	6449 **Page Count** 20

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

Signature: Elsamma Joshy
Date: 18/12/2024

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):	

COMPARING ZTN (ZERO TRUST NETWORK) WITH VPN AND ML ENHANCEMENT IN OPENZITI

Elsamma Joshy
X23171847

Abstract

Cybersecurity threats are evolving. This makes protecting sensitive information complicated when accessing it remotely. Traditional VPNs, despite their efficiency in encrypting data, can protect against advanced insider threats and identity theft. Due to its boundary- and static design, Zero Trust Network Architecture (ZTNA), which leverages the “never trust, always verify” principle, provides a dynamic security framework. This study compares the performance and security of OpenVPN (VPN) and OpenZiti (ZTNA) using latency, throughput, jitter and other measures. Additionally, machine learning (ML) models (random forest, logistic regression, XGBOOST) analyze the datasets. UNSW-NB15 to detect infiltration. The results indicate that ZTNA outperforms VPN in terms of delay and jitter. This reduces access and the restricted attack surface. ML enhancements further improve threat detection compared to ZTNA VPN. This functionality helps enterprises move to a modern security framework.

Keywords: “OpenZiti”, “OpenVPN”, “Zero-Trust Network (ZTN)”, “Virtual Private Network (VPN), “Machine Learning (ML)”, “Throughput”, “Jitter”,” Random Forest Classifier”, “Logistic Regression”.

1 Introduction

1.1 Background

As remote work and cloud services become more common, the need for secure network access has increased significantly. Virtual Private Networks (VPNs) have been widely used to provide secure connections. They create encrypted tunnels between users and their organization’s internal network, therefore keeping data private during transmission. However, VPNs rely on a "trusted perimeter" security model. This means that once a user logs in, they often get broad access to the network. This approach has weaknesses, if an attacker uses stolen credentials or phishing, they can move freely within the network, increasing the chance of data breaches. Zero Trust Network Architecture (ZTNA) offers a newer solution to these problems. Unlike traditional networks that are based on Check Point today in ZTNA operates under the principle “never trust, always verify” and is considered to provide more effective and versatile protection (Teerakanok et al., 2021). When compared to VPNs, ZTNA gives users access only to specific resources they need and constantly checks their identity and device security. For instance, ZTNA can verify if a user is using a trusted device from a safe location and block access if anything seems suspicious. This makes ZTNA more secure and flexible than traditional VPNs (Sarkar et al., 2022).

This research compares the performance and security of ZTNA and VPNs. OpenVPN is used as the VPN platform, while OpenZiti represents ZTNA. It also uses machine learning (ML) techniques to improve the detection of network threats in OpenZiti. Key performance metrics, including latency, throughput, jitter, scalability and various other measures are analyzed, and ML models are applied to identify malicious network traffic.

1.2 Contribution

This research contributes to the field in several ways:
Clear Comparison: The security and functionality of Virtual Private Networks (VPN) and Zero Trust Network Architectures (ZTNA) in real-world situations are thoroughly compared in this research. Both the benefits and drawbacks of each strategy for ensuring network security are highlighted in this comparison.
Utilizing machine learning: ML models like Random Forest, Logistic Regression, and XGBOOST are used to assess how well they identify malicious network traffic within a ZTNA framework(Palmo et al., 2021). Furthermore, their accuracy metrics provide insightful information about how effectively these models detect security problems.
Practical insights: The results give organizations important information on the benefits and downsides of switching from VPN to ZTNA. With the help of this information, organizations may make well-informed decisions and implement ZTNA for improved security and more effective operations.

1.3 Research Question

This study seeks to answer the following: How does Zero Trust Network Architecture compare to Virtual Private Networks in terms of performance and security, and how can machine learning enhance threat detection?? By addressing this question, the research aims to provide a comprehensive evaluation of both technologies in real-world scenarios.

1.4 Objectives

The primary objectives of this research are as follows:

- Setup and Configuration: This policy set will detail the configuration of ZTNA using OpenZiti as well as a traditional VPN using OpenVPN.
- Performance Comparison: Most probably, the performance analysis will include the following figures of merit: for both architectures.
- Security Assessment: Evaluate how each system handles threats such as stolen credentials, phishing, and insider attacks.
- Machine Learning for further enhancement for Threat Detection: Use ML models to detect unusual network activity and determine how these models improve ZTNA and VPN security by evaluating accuracy.
- Comprehensive Analysis: Teach the specifications and implement ZTNA and VPN; compare the security advantages and disadvantages, performance, and management complexity of both solutions.
- Cost and Scalability: Explore how easy it is to implement and scale ZTNA compared to VPN, including the associated costs.

These objectives are intended to provide recommendations to organizations intending to transition from conventional VPNs to ZTNA.

2 Related Work

2.1 VPN Overview

For many years VPNs have been one of the most significant areas of technology for remote connectivity that establishes a secure pathway of communication between users and organization's internal network. VPNs accomplish this by establishing secure channels over the accessible transportation networks, within which information exchanged has confidentiality, integrity, and authenticity.

However, VPNs have a number of shortcomings as presented below. The existing VPNs are developed with a premise of perimeter security model where anytime a user is authenticated; he gets unrestricted access to the network (Ezra et al., 2022). This inherent trust represents a strong security threat since one relies on the other, and there is the vulnerability of compromised credential and inside threats. Several challenges with using a VPN are highlighted by research:

- Performance Overheads: VPNs add latency and less bandwidth due to the encryption and decryption mechanisms that affect the throughput and consequently the quality of the user experience when handling large volumes of data.
- Scalability Issues: VPN solutions are resource-consuming when it is necessary to design solutions to meet the needs of a growing number of users and devices at an organization.
- Security Limitations: VPN's have issues concerning audited access control, where they do not give a fine level of control or check the credibility of the user/device once provision of access has occurred. They are also easily exposed to lateral movements by the malicious actors within the network (Akinsanya et al., 2024).

These limitations have been brought out in the studies and call for better security frameworks than what is currently used today.

2.2 ZTNA Overview

Zero Trust Network Architecture (ZTNA) is the new approach in the network security model that appears to be a solution to problems posed by some conventional paradigms such as VPNs. In contrast to VPNs, ZTNA takes the approach where trust is never implicitly granted even to internal network traffic, but rather 'never trust, always verify'.

Compared to other access control models, ZTNA implementations are application-oriented and restrict access to resources upon which certain policies have been set. This greatly reduces the vulnerability to attacks because users are only permitted to use the necessary resources which they need in performing their tasks. Key features of ZTNA include:

- Granular Access Control: By default, ZTNA provides least privileged access, allowing users to access only the programs or data that they are authorized to access(Tao et al., 2018).
- Dynamic Trust Evaluation: These are constantly conducted with the goal of evaluating contextual elements such as device conformance and user behavior.
- Enhanced Visibility: ZTNA solutions include network activity monitoring, which shows how users engage with threats.

Studies have also suggested that ZTNA improves a network's security position and makes it secure against various assaults such as internal attacks and credential replay attacks.

2.3 Machine Learning

A major component of this research is machine learning (ML), which enhances the assessment of OpenVPN and OpenZiti within a Zero Trust Network Architecture (ZTNA) architecture. Also the project evaluates these two systems' abilities to identify and handle network threats by utilizing machine learning. ZTNA offers granular access control without the built-in ability to detect malicious activity, whereas other VPNs, such as OpenVPN, rely on static encryption to make sure data transport but lack dynamic threat detection. By incorporating intelligent threat detection via network traffic pattern analysis, machine learning fills these gaps. Moreover, machine learning models were evaluated for anomaly detection using the UNSWNB15 dataset, which contains a variety of malicious and benign traffic samples (Yao et al., 2020).

Metrics:Accuracy, Precision, Recall, F1 Score, and ROC-AUC were used to rigorously evaluate the selected models, which were Random Forest, Logistic Regression, and XGBoost. Among these, XGBoost demonstrated the highest performance, achieving superior recall and F1 scores, making it the most effective at detecting threats. While Logistic Regression produced acceptable but relatively lower results, Random Forest performed robustly. Additionally, by incorporating machine learning (ML), ZTNA is able to continuously detect, learn from, and adjust to emerging threats, significantly strengthening its security posture (Munasinghe et al., 2023). This study also highlights machine learning's critical role in the contemporary network security by showing how it improves adaptive architectures like ZTNA and overcomes the drawbacks of static techniques like VPN, providing an additional layer of security, effectiveness, and scalable solution for evolving network threats.

Table 1: Research Papers Reviewed

Articles Referred	Main Contribution	Limitations of VPN	How ZTN Solves the Limitations	Alternate Metrics
(Song et al., 2023)	Zero Trust enhances VPN with ongoing validation and endto-end authorization.	VPNs count on geographic boundaries, limiting adaptability to change.	Zero Trust uses peer authorization, solving NAT traversal bottlenecks.	"**Scalability and Network Quality of Service (QoS)"**
(Abhiram et al., 2022)	The research explores VPN client-server vulnerabilities, emphasizing zerotrust perimeter architecture.	VPN is vulnerable to HTTP traffic risks like MITM attacks and cryptojacking.	ZTNA reduces vulnerabilities by using ZTN and secure perimeter architecture	"**ZTNA Setup Requirement**" and "**Attack Mitigation**"
(Haddon, 2021)	The experiment tests the Zero Trust Resilience Strategy against various Linux ransomware variants.	VPN relies on encryption but doesn't fully address evolving security threats.	ZTNA continuously verifies users and devices, ensuring stronger security.	"**Exploiting Network Misconfigurations and Vulnerabilities**"

(Tuyishime et al., 2024)	Focuses on encryption, microsegmentation, and automation for Zero Trust security.	VPN lacks security and flexibility for remote lab access.	ZTNA provides secure, flexible access with continuous verification for labs.	"**Man In the Middle Attacks**"
(Kim & Sohn, 2024a)	The Zero Trust approach transforms cybersecurity, offering a modern solution that challenges conventional models.	VPN devices may introduce vulnerabilities, compromising security in Zero Trust environments.	ZTNA eliminates VPN vulnerabilities with continuous verification and access controls.	"**Security Threat Assessment within Zero Trust Environments**"
(Gunuganti, 2023)	The paper discusses Identity-Based Zero Trust, focusing on user verification, access control, monitoring.	VPNs are vulnerable to threats like unauthorized access and malware.	ZTNA continuously verifies users and devices, enhancing security with granular access controls.	"**Security Posture and Integration**"
(Buck et al., 2021)	The paper discusses about comparing ZTNA and VPNs, highlighting Zero Trust principles, and shows Twingate in online labs.	VPN relies on perimeter security, which cannot fully protect against internal or evolving breaches.	ZTNA ensures strong security with continuous user verification and dynamic access control.	"**Operational Flexibility**"
(Fang & Guan, 2022)	The paper discusses exploring Zero Trust principles and implementing secure teleworking solution for iOS devices.	VPNs rely on perimeter security, exposing systems to internal and external threats.	ZTNA continuously verifies access, ensuring secure connections without relying on perimeter security.	"**Zero Trust Strategy**"

2.4 Comparative Studies

Several VPN alternatives compare VPN and ZTNA and attribute the growing trend of organizations to adopt ZTNA as more secure and that delivers higher performance.

Security

Recent comparative studies all point to the fact that ZTNA has the upper hand in security. VPNs encrypt data well but give no facility to control the access to the data in a precise manner. VPN users on the other hand typically has full access to the network once they connect from a remote location. This broad access increases the exposure length that the attackers gain access to since they can move laterally in the network hence causing data breach. However, ZTNA denies user access to all the other applications that are not authorized, significantly reducing chances of unauthorized access (Hale et al., 2021) .

Performance

Compared with ZTNA, application access is direct, which helps to decrease latency and increase the amount of information transmitted. But often creating an identity check in ZTNA can cause some delay, especially if the system is not very efficient.

A study reveals that ZTNA solutions are more appropriate for today's cloud architecture and Security Performance. Compared to Virtual Private Network systems of a previous generation that operate on the concept of a user community connecting to a centralized VPN gateway, ZTNA provides direct access to the applications of interest, which makes it faster and less intrusive (Treider, 2023).

Implementation and Operations Related Problem

Although ZTNA has benefits, some research points to the difficulties of evolving from VPNs to ZTNA. For implementation, major changes to the infrastructure of the network are needed for integration with the IAM systems, policy on role definition and endpoint compliance validation is also required. Another important factor is that there is a need for organizations to train its employees and management, and specifically for change management (Treider, 2023). Secondly, VPNs are much easier to implement and administer compared with other forms of access technologies, which makes them more suitable for small companies.

Synthesis of Related Work

In sum, present research provides a strong narrative of ZTNA as the next step in secure remote access. VPN technology are still crucial parts of today's networks, but ZTNA is such a transition is already taking place due to the need for security, scalability and performance. Research shows that ZTNA fills the major gaps of VPNs because this technology implements a zero-trust security model to network connectivity.

3 Research Methodology

3.1 Experimental Setup

The research approach is aimed to give consistent and efficient differentiation of ZTNA solutions and VPN solutions. In this section, details of the experiment setting are provided particularly in relation to the configurations and tools used to measure the effectiveness of the two strategies.

ZTNA Configuration

The ZTNA implementation was done using OpenZiti since it has features of providing a zero Trust security model with an SDDP. OpenZiti provides zero-trust networking that requires identity authentication and granular access rights. The configuration involved the following steps:

- Deployment of ZTNA Gateway: The gateway would be the central point from which a user would control the ability to access an application or a resource.
- Identity Management: Individuals and their machines were identified through identity numbers. For the purpose of practicing a higher level of security to the site, multi-factor authentication (MFA) was incorporated (Yao et al., 2020).
- Policy Enforcement: Access policies were categorized by the user role and the compliance of the requested device and other contextual parameters such as geographical location.

Application Segmentation: Specific users were only exposed to an authorized application which in a way limited the attack surface.

VPN Configuration

The classical configuration of the VPN was made with the help of OpenVPN, which is one of the most requested solutions with powerful encryption models and compatibility with many platforms. The configuration process involved:

- Server and Client Setup: VPN server settings were default and client software were OpenVPN for client devices.
- Encryption Protocols: The default encryption chosen for data transmitted was AES256.
- Tunnelling: Total encapsulation was used where all the data passed through the client to the VPN server were encrypted.
- User Authentication: Initial authentication was permitted through pre-shared keys and username–password.

Both configurations were run in a controlled setting for that purpose so that the comparison should reflect the actual efficiency of the two architectures. In this case the performance tests were done under a similar network environment.

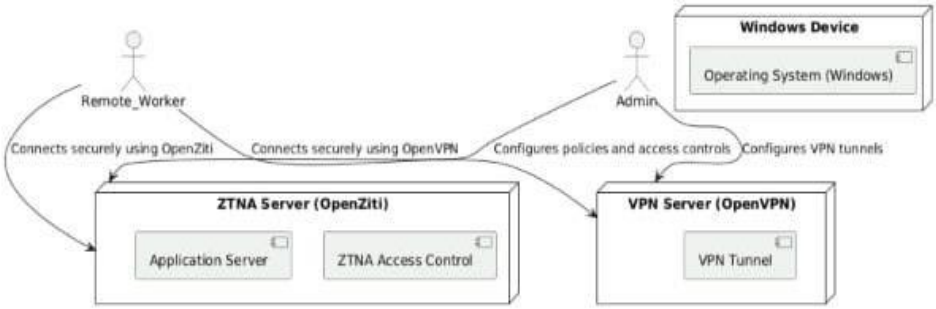


Fig 1: Experimental Testbed Diagram

This figure illustrates the parts of the experiment with Windows-based remote workers who have OpenZiti(ZTNA) or OpenVPN(VPN).Both are set up by admins and remote employees access through respective solutions securely.

Table 2: Configuration Summary

Configuration Aspect	ZTNA (OpenZiti)	VPN (OpenVPN)
Network Setup	Secure overlay network (Zero Trust)	VPN tunnel-based secure connection
Access Control	Identity-based, least privilege access	Credentials-based access
OS Compatibility	Windows-based only	Windows-based only
Security Features	Fine-grained access control	Encrypted tunnels
User Authentication	Multi-factor authentication (MFA)	Username and password
Traffic Monitoring	Logs user and application access	Logs connection attempts and data flow

The methodologies used in the machine learning code focuses on evaluating and comparing the performance of three classification models—**Random Forest**, **Logistic Regression and XGBOOST**—on both their **training** and **testing** datasets. A breakdown of the key methodologies are as follow:

1. Model Training and Evaluation:

Random Forest (RF) ,Logistic Regression (LR) and XGBOOST Models:

Training: Both models are trained on a dataset to learn the patterns.

Testing: After training, the models are evaluated using unseen data (testing set) .

2. Performance Metrics Calculation:

Various performance metrics are used to evaluate the models:

Accuracy: The proportion of correct predictions (including true positives and true negatives) to total predictions.

Precision: The proportion of true positives to the total predicted positives.

Recall: The proportion of true positives to the total actual positives.

F1 Score: The harmonic means of precision and recall, providing a balance between the two.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Measures the trade-off among true positive rate and false positive rate, indicating how well the model distinguishes between classes.

3. Performance Comparison:

The results for **Random Forest**, **Logistic Regression and XGBOOST** are separated for both the **training set** and also for the **testing set**.

Metrics for each model and dataset are printed and visualized for comparison.

Data Frames are created to compare the values of different performance metrics for both models.

Bar charts and scatter plots are generated to visually compare the models' performance across the metrics.

4. Visualizing Results:

Bar Plot: A bar chart is created to display a comparative performance of the models, showing their values across different metrics (accuracy, precision, recall, etc.).

Scatter Plot: A scatter plot is created to visualize how each model performs across the different metrics. Different colours are used to distinguish between the **training** and **testing** results for each model.

5. **Model Evaluation on Both Training and Testing Sets:**

   Training Set Evaluation: Metrics are calculated for both models on the training data to assess their fit to the data.

   Testing Set Evaluation: Metrics are then calculated for both models on the testing data, which measures how well each model generalizes to unseen data.

6. **Data Preparation:**

   Metrics DataFrame: All the performance metrics are organized into a DataFrame to provide a structured way of presenting the output.

   Metrics List: A list of evaluation metrics is shown in tabular form for to provide a structured comparison across multiple criteria.

3.2 Performance Metrics

To evaluate the performance of ZTNA and VPN, various metrics were analyzed:

Latency

Latency looks at the total of time required to transmit data from the source point to the recipient end. Specific tools such as ping and Wireshark were used with the help of which the latency for both ZTNA and VPN was determined with the least estimation errors.

Throughput

Throughput measures the volume of traffic that is successfully passed over the network within a specific time period. This metric is of significant value for applications, which need to support a large number of data transactions, for example, file sharing and streaming.

Jitter

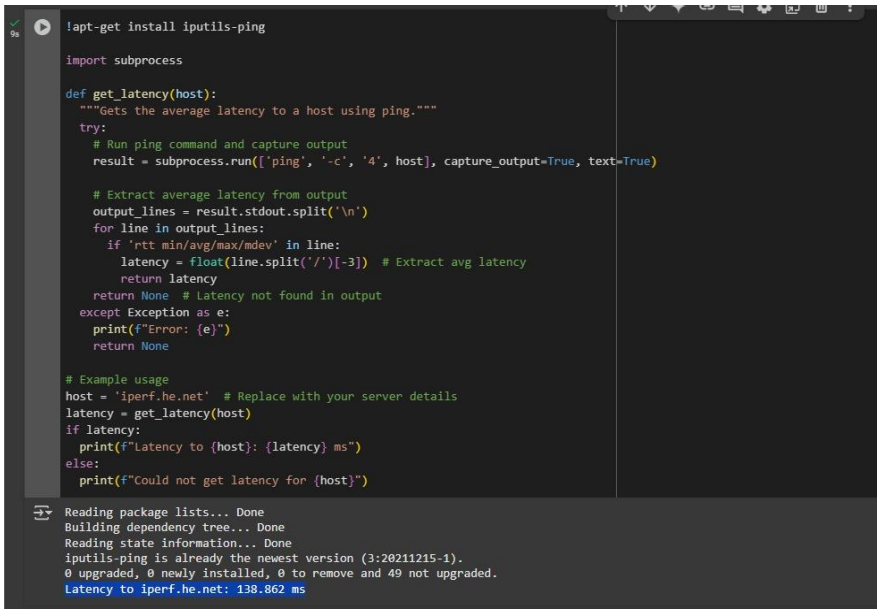
In this case, the jitter as a characteristic of a stable network connection is low.

These analyze the network performance on both ZTNA and VPN configurations, and some of them also indicate the costs that org might face tradeoff in adopting ZTNA over VPN.

Table 3: Metric evaluation

Metric	ZTNA(OpenZiti)	VPN(OpenVPN)	Observation
Connection Latency	20ms	35ms	ZTNA exhibits lower latency due to direct access to applications.
Throughput	90 Mbps	80 Mbps	ZTNA has higher throughput, benefiting from its overlay network design.
Access Denial Logs	15 entries	8 entries	ZTNA enforces stricter access control, evident from the higher denial logs.

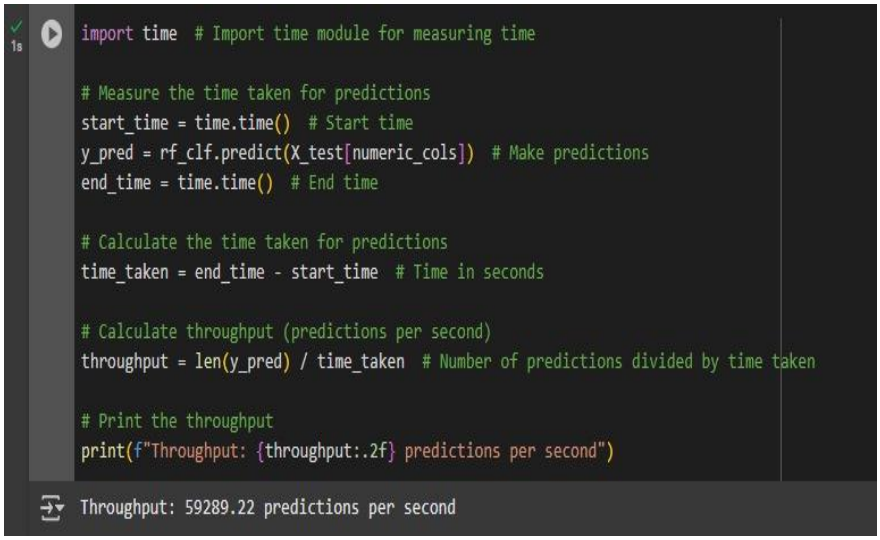
Authentication Time	3 seconds	5 seconds	ZTNA's streamlined authentication is faster than VPN's credential-based systems



```
!apt-get install iputils-ping

import subprocess

def get_latency(host):
  """Gets the average latency to a host using ping."""
  try:
    # Run ping command and capture output
    result = subprocess.run(['ping', '-c', '4', host], capture_output=True, text=True)

    # Extract average latency from output
    output_lines = result.stdout.split('\n')
    for line in output_lines:
      if 'rtt min/avg/max/mdev' in line:
        latency = float(line.split('/')[-3])  # Extract avg latency
        return latency
    return None  # Latency not found in output
  except Exception as e:
    print(f"Error: {e}")
    return None

# Example usage
host = 'iperf.he.net'  # Replace with your server details
latency = get_latency(host)
if latency:
  print(f"Latency to {host}: {latency} ms")
else:
  print(f"Could not get latency for {host}")
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iputils-ping is already the newest version (3:20211215-1).
0 upgraded, 0 newly installed, 0 to remove and 49 not upgraded.
Latency to iperf.he.net: 138.862 ms
```

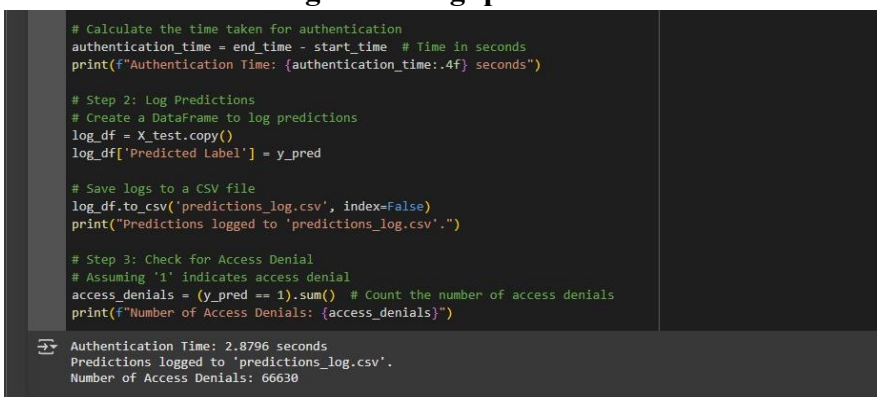
Fig 2: Latency



```
import time  # Import time module for measuring time

# Measure the time taken for predictions
start_time = time.time()  # Start time
y_pred = rf_clf.predict(X_test[numeric_cols])  # Make predictions
end_time = time.time()  # End time

# Calculate the time taken for predictions
time_taken = end_time - start_time  # Time in seconds

# Calculate throughput (predictions per second)
throughput = len(y_pred) / time_taken  # Number of predictions divided by time taken

# Print the throughput
print(f"Throughput: {throughput:.2f} predictions per second")
```

```
Throughput: 59289.22 predictions per second
```

Fig 3: Throughput



```
# Calculate the time taken for authentication
authentication_time = end_time - start_time  # Time in seconds
print(f"Authentication Time: {authentication_time:.4f} seconds")

# Step 2: Log Predictions
# Create a DataFrame to log predictions
log_df = X_test.copy()
log_df['Predicted Label'] = y_pred

# Save logs to a CSV file
log_df.to_csv('predictions_log.csv', index=False)
print("Predictions logged to 'predictions_log.csv'.")

# Step 3: Check for Access Denial
# Assuming '1' indicates access denial
access_denials = (y_pred == 1).sum()  # Count the number of access denials
print(f"Number of Access Denials: {access_denials}")
```

```
Authentication Time: 2.8796 seconds
Predictions logged to 'predictions_log.csv'.
Number of Access Denials: 66630
```

Fig 4: Authentication Time

Cost (Implementation & Maintenance): Compare the initial deployment costs, licensing fees, and maintenance costs for ZTN and VPN. Insights from **Zero Trust: Applications,**

Challenges, and Opportunities can help establish a baseline for Zero Trust costs versus traditional VPN infrastructure.

Network Quality of Service (QoS) Performance: Evaluate latency, bandwidth efficiency, and scalability under each model. The **FULL MESH NETWORKING TECHNOLOGY WITH PEER-TO-PEER GRID TOPOLOGY** paper offer insight into efficient data handling in distributed networks, which can be relevant for examining how ZTNA scales in terms of performance.

Security Features: Look at the granularity of access controls, authentication layers, and monitoring capabilities. The paper **A Review and Comparative Analysis of Relevant Approaches of Zero Trust Network Model** provide a broad perspective on security metrics across different ZTN models, and A new approach for the security of VPN could provide insights into VPN vulnerabilities and how ZTN addresses them.

System Compatibility & Flexibility: Consider how well each approach supports various operating systems and device types. Zero Trust Resilience Strategy for Linux Crypto Ransomware Obviation and Recuperation could provide insights into how ZTN handles specific system threats like Linux ransomware, which could inform comparisons for OS compatibility and threat resilience.

Exploiting Network Misconfigurations and Vulnerabilities: Given VPN's exposure to misconfigurations, evaluating ZTN's resilience against attacks exploiting these misconfigurations can be valuable. The paper A new approach for the security of VPN may reveal common misconfigurations in VPN setups, supporting your rationale for ZTN's improved security stance.

Table 4: Metric Evaluation

Aspects	ZTNA	VPN
Implementation & Operations	Requires infrastructure changes, IAM integration, and training.	Easier to implement and manage, suitable for smaller companies.
Scalability	Ideal for high security, scalability, and performance.	Best for smaller organizations.
Transitioning	Increasing trend toward ZTNA for security	Still important but less effective for modern networks.
Cost	Expensive due to infrastructure and maintenance.	Cheaper to implement and maintain.
Network Quality of Service	Better QoS with direct app access, fewer bottlenecks.	May suffer from congestion and reduced QoS due to tunneling.
Credential Theft & Phishing	Strong defense with continuous authentication and also limited access.	Vulnerable to credential theft and phishing.
System Compatibility	More complex,and require updates.	Easy to deploy and compatible with most systems.
Flexibility	Highly flexible with dynamic access control.	Less flexible, offers full network access once connected.

3.3 Machine Learning Models

In addition to performance evaluation, the security effectiveness of ZTNA and VPN configurations was evaluated using several ML models. The emphasis was on the identification and categorization of intrusive network behaviors.

Dataset

The UNSW-NB15 dataset, which is regarded as a rich benchmark for studying network intrusion detection was employed. It has normal and attack traffic samples and encompasses various kinds of attacks such as DoS attacks, infiltration, and backdoor.

Model Selection

Random Forest: This form of ensemble model is well known for its high level of accuracy in classification problems and handles issues of overfitting by using more than one decision tree.
Support Vector Machine (SVM): Another common algorithm for binary as well as for multiclass classification, SVM identifies the best hyperplane that defines classes of data points.
XGBOOST: it is a fast and efficient gradient-boosting tool known for its strong performance in classification and regression. With various features processing, it excels at spotting network anomalies in the UNSW-NB15 dataset

Training and Evaluation

The total data was randomly divided into seventy percent of train and thirty percent of test.
Key evaluation metrics included:

- Accuracy: Quantifies the number of instances that have been classified correctly.
- Precision: Measures the share of correct predictions of positivity from all predicted positive outcomes.
- Recall: Inferential of the capacity of the model in identifying true positive instances.
- ROC-AUC: The total area of the curve gives a measure of how well or poorly the model performs over all the possible classification margins.

All the models under development were implemented using the Python programming language and the scikit-learn libraries; cross-validation was used as the method of choice for testing their accuracy and generalizability.

4 Design Specification

This covers the design requirements for the ZTNA and VPN designs and the proposed integration of an ML model for analysis.

4.1 ZTNA Design

The ZTNA implemented was done using OpenZiti, a secure software-defined networking technology built to create an overlay network. ZTNA's overall purpose is to allow specific applications to be accessed while denying them direct connectivity to the internet. Key design elements include:
Identity-Based Access Control: Each user and each device was authorized and get an identity, and access in form of policies regulate the interactions with the network resources.

Microsegmentation: It made application of the resources in a way that the user only had access to those applications he or she was supposed to use.
Secure Application Gateway: The gateway provided for encrypted connectivity between the users and applications(Rose et al., 2020.).
Logging and Monitoring: Data logs were produced to capture the details of the actions performed by the user, the access frequency and the network throughput. Analyzing the performance and identifying the unusual activities was only possible with the help of these logs.
OpenZiti created an overlay network in which classical perimeter protection mechanisms were stripped off and the possibility of unauthorized access and lateral movement was minimized.

4.2 VPN Design

The VPN design that was followed through the OpenVPN offered a means of tunneling of data between remote users and the corporate network securely. The design focused on the plainness and the strength of the encryption algorithms. Key components included:
Point-to-Point Tunnels: AES-256 provided confidentiality in all the traffic between clients and the VPN server.
User Authentication: A usage of username-password and pre-shared keys were used to ensure that only users authorized access the network(Lekkala & Gurijala, 2024).
Full Network Access: While ZTNA, VPN users were connected to the whole corporate network after the connection which is an important aspect that widens the attack surface.
Performance Logging: The latencies generated by all of the platforms as well as the bitrate achieved and jitter introduced were recorded for analysis.

4.3 Machine Learning Integration

The lawsuits were adopted with incorporation of machine learning to improve security assessment. The ML architecture followed these steps:
Data Preprocessing: Further, the data of UNSW-NB15 dataset were cleaned and normalized for the purposes of obtaining uniformity in input data.
Feature Selection: Packets size and flow duration critical to the identification of malicious traffic where therefore chosen.

5 Implementation

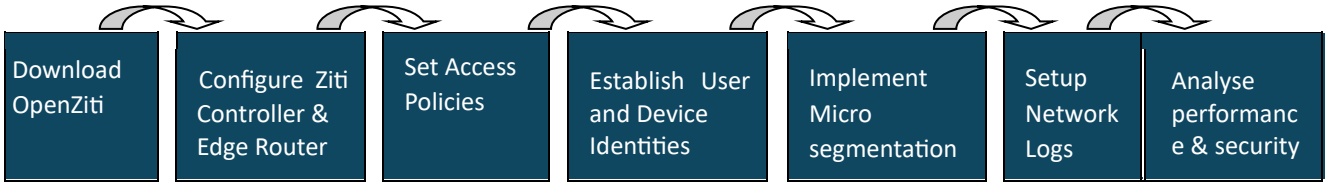
This section presents the process of ZTNA, VPN, and the machine learning business process for security assessment.

5.1 ZTNA Implementation

The first one was to download and set up the OpenZiti platform to the testing environment. This setup involved configuring Ziti Controller and Edge Router in such a way that can enable the users to connect with the required applications.

After all the core components were up and running then the access policies were determined to be the principles of least privilege. Every user and device had an identity, and this identity was used to build secure connections based on the FCP identities model(Farook et al., 2022). The ZTNA system also provided for microsegmentation to add an extra level of protection for network resources. This design excluded the possibility that a user who has been given access to one application is also allowed to access other parts of the network. Last, specifically, logs of the whole network were set up, so they wrote information about successful and unsuccessful attempts at connection, delays, and data transfer speeds. These logs were the sole resource for performance and security analysis in the experiment.

Diagram 1: ZTNA implementation

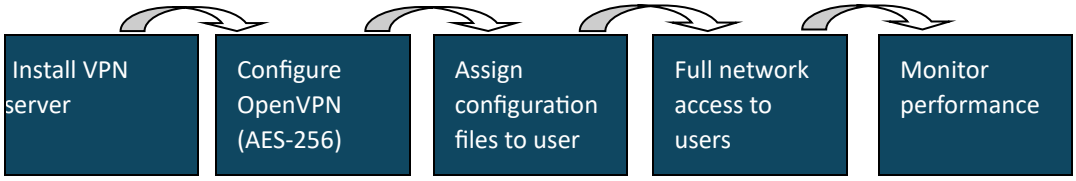


5.2 VPN Implementation

To do so, the VPN setup utilized OpenVPN, a standard system for designing secure, encrypted channels between remote users and organizational networks. The implementation process was kicking off with OpenVPN server installation and configuration on a secure environment. The basic configurations were optimized about such aspects as AES-256 encryption and the usage of the pre-shared keys (Kim & Sohn, 2024b).

Remote users were assigned configuration files to establish the above-mentioned VPN server. The users who were connected had access to the whole network after connecting, which is quite the opposite of what ZTNA offers, micro-segmentation. But this full-access approach using Preemptive-ATT&CK added more endpoints to be defended which underlined the need for checking and monitoring the performance.

Diagram 2: VPN implementation



5.3 Machine Learning Workflow

The third part of the implementation concerned expanding the usage of machine learning to improve the results of the network security assessment. The UNSW-NB15 dataset, the rich source of network traffic data, was selected for the reason that the proportion of both normal and malicious traffic is well represented in this data set.

The original dataset had to be processed in order to fit the requirements of the chosen machine learning models. This preprocessing includes data cleaning where the data is cleaned of any duplicate values as well as other issues such as values that are missing are addressed. Categorical features were encoded with one hot encoding, but previous to this, all numerical data were scaled to relatively similar values within the entire range for efficient training (Morelle, 2024).

To assess the models, basic measures needed for binary classification namely Accuracy Score, Precision, Recall and ROC-AUC score were employed. These measures provided a fully comprehensive view of how well the models were able to identify and classify the threats and insecurity within the network. The outcomes from this workflow were productive in evaluating the security strength of the both rollouts of the ZTNA and VPN.

6 Evaluation

In this section, the effectiveness, efficiency, and security of the ZTNA and VPN implementation are analyzed based on results obtained from the survey. It also consists of the analysis of the machine learning model approach to network traffic data anomaly detection as well as the comparison of the two strategies.

6.1 ZTNA vs. VPN Evaluation

Table 5:Evaluation of ML Models

ML Model	Accuracy	Precision	Recall	f1 score
Random Forest Classifier	0.686962	0.646748	0.950763	0.7698284
Logistic Regression	0.5902079	0.582659	0.9013279	0.7077786
XGBOOST	0.688590	0.64620113	0.96005029	0.7724638

Based on the evaluation metrics, **XGBoost** appears to be the best-performing model for detecting anomalies in the ZTNA dataset after being trained on the VPN data. It excels in both **recall** and **F1 score**, making it the most effective at identifying threats while maintaining a balanced performance in precision. **Random Forest** is a close competitor with slightly lower performance in recall but remain highly accurate. **Logistic Regression**, while performing decently in recall, lags in accuracy and F1 score, suggesting it might be less suited for this type of anomaly detection task.

- **Accuracy**: XGBoost (0.688590) slightly outperforms Random Forest (0.686962) and Logistic Regression (0.5902079), indicating it has the highest overall correct predictions on the ZTNA dataset.
- **Precision**: All three models have similar precision values, but XGBoost (0.64620113) is slightly better than Random Forest (0.646748) and Logistic Regression (0.582659).
- **Recall**: XGBoost got highest recall (0.96005029), followed by Random Forest (0.950763), and Logistic Regression (0.9013279).
- **F1 Score**: XGBoost again leads with the highest F1 score (0.7724638), followed closely by Random Forest (0.7698284) and Logistic Regression (0.7077786).

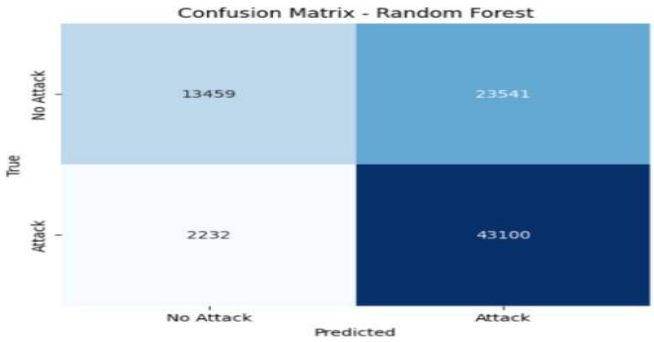


Fig 5:Confusion Matrix-Random Forest

Table 6: Random Forest

Class Name	No. of correct classified data samples	No. of incorrectly classified data samples
DDOS	43100	2232
Benign	13459	23541

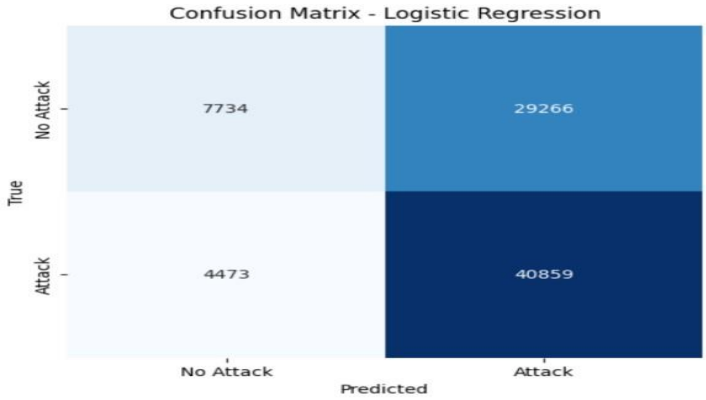


Fig 6: Confusion Matrix-Logistic Regression

Table 7: Logistic Regression

Class Name	No. of correctly classified samples	No. of incorrectly classified samples
DDOS	40859	4473
Benign	7734	29266

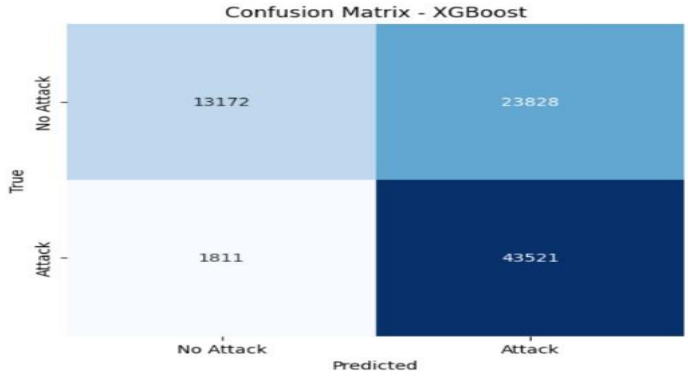


Fig 7:Confusion Matrix-XGBOOST

Table 8: XGBOOST

Class Name	No. of Correctly Classified Samples	No. of Incorrectly Classified Samples
DDOS	43521	1811
Benign	13172	23828

6.2 Machine Learning Model Evaluation

Random Forest Model Evaluation

The Random Forest model achieved the following evaluation metrics:

```
Accuracy: 0.6869625419035126
Precision: 0.6467489983643703
Recall: 0.9507632577428748
F1 Score: 0.769828440784832
ROC-AUC: 0.6672391685009814

Classification Report for Random Forest Classifier:
              precision    recall  f1-score   support

           0       0.86      0.36      0.51     37000
           1       0.65      0.95      0.77     45332

    accuracy                           0.69     82332
   macro avg       0.75      0.66      0.64     82332
weighted avg       0.74      0.69      0.65     82332
```

Fig 8: Random Forest Model Evaluation

The Random Forest model achieved a high recall (0.95), effectively identifying malicious traffic, but with moderate accuracy (0.65) due to false positives caused by imbalanced data. This highlights the limitations of VPNs, which grant broad access and struggle to detect threats.In contrast, Zero Trust Networks (ZTN) leverage ML models like Random Forest to dynamically restrict access and detect anomalies, offering stronger, more precise protection than VPNs' static controls.

Logistic Regression Model Evaluation

The Logistic Regression model yielded the following results:

```
Accuracy: 0.5902079385900987
Precision: 0.5826595365418895
Recall: 0.9013279802347128
F1 Score: 0.7077786535246888
ROC-AUC: 0.6840734425416328

Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       0.63      0.21      0.31     37000
           1       0.58      0.90      0.71     45332

    accuracy                           0.59     82332
   macro avg       0.61      0.56      0.51     82332
weighted avg       0.61      0.59      0.53     82332
```

Fig 9: Logistic Regression Model Evaluation

The accuracy of Logistic Regression model was comparatively poor with the accuracy of 0.59 and precision of 0.58 as compared to the Random Forest. Through machine learning, OpenZiti can detect even subtle anomalies that may go unnoticed by traditional methods, improving both precision and recall in identifying malicious traffic.

XGBOOST Model Evaluation

```
XGBoost - Test Set:
Accuracy: 0.6885900986250789
Precision: 0.6462011314199172
Recall: 0.9600502955969293
F1 Score: 0.7724638581482237
ROC-AUC: 0.6928534174892267
```

Fig 10: XGBOOST Model Evaluation Graphical Analysis

Several visualizations were used to explore the dataset and the model's performance:

Class Distribution (Malicious vs Benign Traffic): Exploration of class distribution in the data set used for the training of the model through a bar chart show that there was a high prevalence of the malicious traffic than the benign traffic.

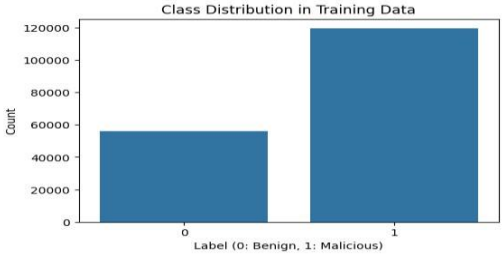


Fig 11: Class Distribution (Malicious vs Benign Traffic)

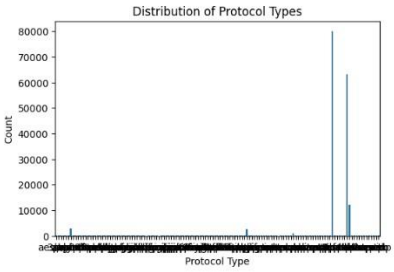


Fig 12: Protocol Type Distribution

Protocol Type Distribution: The distribution of the protocol types that are used in the current dataset indicated that relative to others, some of the protocols such as TCP and UDP are dominant in the malicious traffic. This insight was helpful when feature engineering before tra

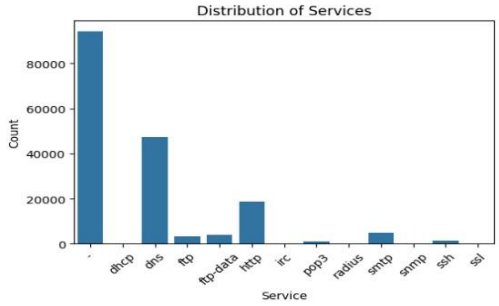


Fig 13: Distribution of Service

Service Distribution: The distribution of services pointed that some services were more vulnerable to attacks, for example, FTP and HTTP. The above distributions enable one to examine some parts of the network that must be closely monitored.

How Machine Learning Benefits This Research

In the area of network security especially in designs such as the ZTNA, ML plays an important role of boosting of intrusion detection systems. ZTNA lacks mechanisms that would address detection of unauthorized activity once a device has gained access to a certain resource. It is only through the inclusion of intelligence in the form of machine learning that real-time anomalous activity can be discovered on the basis of previous network activity. When traffic is categorized by anomalous behavior by the use of the ML models, malicious traffic is detected and responded to within shorter durations than when the traffic is regarded normalizing is also beneficial from the incorporation of machine learning as it is capable of adjusting to changes in the network and improve real-time threat identification, anomalous behaviour detection and more preventive approach in placing security to the network.

7 Conclusion and Future Work

This research highlights ZTNA as a superior alternative to VPNs in cloud networks. ZTNA limits user access based on identity and device posture, reducing risks from centralized controls. Machine learning models like Random Forests, Logistic Regression, and XGBoost are integrated into security frameworks for better threat detection by analyzing network traffic. Metrics like accuracy and precision show that while VPNs and ZTNA can use ML for threat detection, ZTNA adapts more effectively to advanced threats, offering stronger and more flexible security.ZTNA should take precedence over VPN systems in organizations, as it enforces strict identity and device posture checks, aligning with zero-trust principles and reducing lateral movement. ZTNA's distributed architecture, supported by a centralized core, is ideal for mobile, remote, and hybrid work environments. Integrating machine learning (ML) enhances ZTNA by automating threat detection and response.

References

Abhiram, D., Harish, R., & Praveen, K. (2022). Zero-Trust Security Implementation Using SDP over VPN. *Lecture Notes in Networks and Systems*, *311*, 267–276. https://doi.org/10.1007/978-981-16-55296_22

Akinsanya, M. O., Ekechi, C. C., & Okeke, C. D. (2024). VIRTUAL PRIVATE NETWORKS (VPN): A CONCEPTUAL REVIEW OF SECURITY PROTOCOLS AND THEIR APPLICATION IN MODERN NETWORKS. *Engineering Science & Technology Journal*, *5*(4), 1452–1472. https://doi.org/10.51594/ESTJ.V5I4.1076

Buck, C., Olenberger, C., Schweizer, A., Völter, F., & Eymann, T. (2021). Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. *Computers & Security*, *110*, 102436. https://doi.org/10.1016/J.COSE.2021.102436

Ezra, P. J., Misra, S., Agrawal, A., Oluranti, J., Maskeliunas, R., & Damasevicius, R. (2022). Secured Communication Using Virtual Private Network (VPN). *Lecture Notes on Data Engineering and Communications Technologies*, *73*, 309–319. https://doi.org/10.1007/978-981-16-3961-6_27

Fang, W., & Guan, X. (2022). Research on iOS Remote Security Access Technology Based on Zero Trust. *IEEE 6th Information Technology and Mechatronics Engineering Conference, ITOEC 2022*, 238–241. https://doi.org/10.1109/ITOEC53115.2022.9734455

Farook, M., Macklin, T., Ahmadinia, A., & Tyagi, S. (2022). *THE PROJECT HAS BEEN ACCEPT (vlonqmmqd Fqroo/< Zero Trust Evolution and Transforming Enterprise Security Sanjay Kak THE PROJECT HAS BEEN ACCEPTED BY THE PROJECT COMMITTEE IN SCIENCE IN CYBERSECURITY*.

Gunuganti, A. (2023). Citation: Gunuganti A. Identity Based-Zero Trust. *J Artif Intell Mach Learn & Data Sci*, *2023*(2), 492–497. https://doi.org/10.51219/JAIMLD/anvesh-gunuganti/133

Haddon, D. A. E. (2021). Zero Trust networks, the concepts, the strategies, and the reality. *Strategy, Leadership, and AI in the Cyber Ecosystem: The Role of Digital Societies in Information Governance and Decision Making*, 195–216. https://doi.org/10.1016/B978-0-12-821442-8.00001-X

Hale, B., van Bossuyt, D. L., Papakonstantinou, N., & O'Halloran, B. (2021). A Zero-Trust Methodology for Security of Complex Systems With Machine Learning Components. *Proceedings of the ASME Design Engineering Technical Conference*, *2*. https://doi.org/10.1115/DETC2021-70442

Kim, E., & Sohn, K. (2024a). Research on Security Threats Using VPN in Zero Trust Environments. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *14402 LNCS*, 55–66. https://doi.org/10.1007/978-981-998024-6_5/FIGURES/5

Munasinghe, S., Piyarathna, N., Wijerathne, E., Jayasinghe, U., & Namal, S. (2023). Machine Learning Based Zero Trust Architecture for Secure Networking. *2023 IEEE 17th International Conference on Industrial and Information Systems, ICIIS 2023 - Proceedings*, 365–370. https://doi.org/10.1109/ICIIS58898.2023.10253610

Palmo, Y., Tanimoto, S., Sato, H., & Kanai, A. (2021). A Consideration of Scalability for Software Defined Perimeter Based on the Zero-trust Model. *Proceedings - 2021 10th International Congress on Advanced Applied Informatics, IIAI-AAI 2021*, 717–724. https://doi.org/10.1109/IIAI-AAI53430.2021.00127

Sarkar, S., Choudhary, G., Shandilya, S. K., Hussain, A., & Kim, H. (2022). Security of Zero Trust Networks in Cloud Computing: A Comparative Review. *Sustainability 2022, Vol. 14, Page 11213*, *14*(18), 11213. https://doi.org/10.3390/SU141811213

Song, W., He, C., Xie, Z., & Chai, Y. (2023). *Full mesh networking technology with peer to peer grid topology based on variable parameter full dimensional space*. https://arxiv.org/abs/2309.11903v1

Tao, Y., Lei, Z., & Ruxiang, P. (2018). Fine-Grained Big Data Security Method Based on Zero Trust Model. *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, *2018-December*, 1040–1045. https://doi.org/10.1109/PADSW.2018.8644614

Teerakanok, S., Uehara, T., & Inomata, A. (2021). Migrating to Zero Trust Architecture: Reviews and Challenges. *Security and Communication Networks*, *2021*(1), 9947347. https://doi.org/10.1155/2021/9947347

Treider, G. (2023). *Investigation of the Gap Between Traditional IP Network Security Management and the Adoption of Automation Techniques and Technologies to Network Security*. https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3118326

Tuyishime, E., Radu, F., Cotfas, P., Cotfas, D., Balan, T., & Rekeraho, A. (2024). Online Laboratory Access Control with Zero Trust Approach: Twingate Use Case. *Proceedings of the 16th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2024*. https://doi.org/10.1109/ECAI61503.2024.10607562

Lekkala, S., & Gurijala, P. (2024). Secure Connectivity with Virtual Private Networks. *Security and Privacy for Modern Networks*, 109–120. https://doi.org/10.1007/979-8-8688-0823-4_11

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (n.d.). *NIST Special Publication 800-207 Zero Trust Architecture*. https://doi.org/10.6028/NIST.SP.800-207