Arab American University
Faculty of Graduate Studies
Department of Natural, Engineering and Technology Sciences
Master Program in Cyber Security

Using Machine Learning to Detect Network Client Health Security in Zero Trust Architecture.

Montaser I.M. Tanina

202216399

Supervision Committee:
Dr. Huthaifa Ashqar

Dr. Mohammad Hamarsheh

Dr. Nael Abuhalaweh

This Thesis Was Submitted in Partial Fulfilment of the Requirements for the Master Degree in Cyber Security

Palestine, Feb / 2026

Arab American University
Faculty of Graduate Studies
Department of Natural, Engineering and Technology Sciences
Master Program in Cyber Security

Thesis Approval

Using Machine Learning to Detect Network Client Health Security in ZTA

Montaser I.M. Tanina

202216399

This thesis was defended successfully on 14.2.2026 and approved by:

Thesis Committee Members:

Name	Title	Signature
1. Dr. Huthaifa Ashqar	Main Supervisor	.....
2. Dr. Mohammad Hamarsheh	Member of Supervision Committee	.....
3. Dr. Nael Abuhalaweh	Member of Supervision Committee	.....

Palestine, Feb/ 2026

Declaration

I declare that, except where explicit reference is made to the contribution of others, this thesis is substantially my own work and has not been submitted for any other degree at the Arab American University or any other institution.

Student Name: Montaser I.M. Tanina

Student ID: 202216399

Signature: 

Date of Submitting the Final Version of the Thesis: 24.3.2026

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Huthaifa Ashqar, for his unwavering guidance, insightful feedback, and constant encouragement throughout this research. I am also sincerely thankful to Dr. Mohammad Hamarsheh and Dr. Nael Abuhalaweh, who served on my thesis committee. Their expert advice, constructive critiques, and generous support were invaluable in shaping the direction and quality of this work.

I extend my appreciation to the faculty and colleagues at the Arab American University, especially in the Department of Natural, Engineering and Technology Sciences, for providing a supportive academic environment and the resources necessary to conduct this study. I am grateful to the professionals and domain experts who offered their time and expertise during the development of this research – their practical insights helped bridge the gap between theory and application. Finally, I thank the Arab American University's Master Program in Cyber Security for imparting the knowledge and skills that underpinned this thesis. This work would not have been possible without the support and encouragement of all those mentioned above.

Using Machine Learning to Detect Network Client Health Security in

Zero Trust Architecture

Montaser I.M. Tanina

Dr. Huthaifa Ashqar

Dr. Mohammad Hamarsheh

Dr. Nael Abuhalaweh

Abstract

Modern organizations face increasing cybersecurity challenges as cyberattacks expand due to remote work, cloud services, and Bring Your Own Device (BYOD) policies. Zero Trust Architecture (ZTA) has emerged to address these challenges by applying a "never trust, always verify" model to every user and device. This thesis targets a critical vulnerability in ZTA: the real-time assessment of endpoint security health. We propose a machine learning-based framework for continuously assessing device security health and integrating this information into ZTA decision-making processes.

A comprehensive dataset was created by collecting data from multiple sources (such as update status, antivirus presence, vulnerabilities, and system behavior indicators) from different environments. To overcome the limitations of real-world data, synthetic data augmentation techniques (including a GPT-based) were applied, expanding the dataset while maintaining realistic distributions. Each device was assessed using a Device Risk Measure (DRM) that combines factors such as compromise likelihood and potential impact, enabling the training of supervised learning models with clear accept/deny labels.

Several machine learning algorithms (such as support vector machines, decision trees, and ensemble methods) were trained and evaluated based on their ability to classify devices as "healthy" (acceptable) or "unhealthy" (should be denied from network access). The models achieved high accuracy in distinguishing device trust levels, with the best-performing model exceeding 99% classification accuracy. The integration of feature extraction highlighted the most critical security features contributing to device risk.

The results demonstrate the potential for effectively integrating data-driven adaptive device health checks into a zero-trust (ZTA) model. This approach enables dynamic policy implementation, allowing the policy decision point to trust or quarantine devices based on their current risk level. This helps reduce the attack surface and prevents compromised or non-compliant devices from compromising the network. The research has significant implications for cybersecurity practices, providing a blueprint for enhancing ZTA implementations using machine learning, ultimately improving automated threat prevention and organizational resilience.

Keywords: device health check, Zero trust Architecture, Machine Learning, Risk Assessment, Network Access control

Table of Contents

Contents

List of Tables

List of Figures

List of Definitions of Abbreviations

Abbreviations	Title
BYOD	Bring Your Own Device
ZTA	Zero Trust Architecture
ML	Machine Learning
SOC	Security Operations Center
NIST	National Institute of Standards and Technology
FAIR	Factor Analysis of Information Risk
NAC	Network Access Control
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PAP	Policy Administration Point
DRM	Device Risk Metric
AI	Artificial Intelligence
IAM	Identity and Access Management
IoT	Internet of Things
IDS	Intrusion Detection System
SIEM	Security Information and Event Management
UEBA	User and Entity Behavior Analytics
ZTNA	Zero Trust Network Access
MFA	Multi-Factor Authentication
VPN	Virtual Private Network
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
VA	Vulnerability Assessment
MITRE	MITRE Corporation
APT	Advanced Persistent Threat
DNN	Deep Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
KNN	K-Nearest Neighbors
RF	Random Forest
XGBoost	Extreme Gradient Boosting
SMOTE	Synthetic Minority Over-sampling Technique
GPT	Generative Pre-trained Transformer
VM	Virtual Machine
OS	Operating System
CPU	Central Processing Unit
GPU	Graphics Processing Unit
FTP	File Transfer Protocol
RPC	Remote Procedure Call
RDP	Remote Desktop Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol

ISP	Internet Service Provider
C2	Command and Control
UAC	User Account Control
EDR	Endpoint Detection and Response
XDR	Extended Detection and Response
WSUS	Windows Server Update Services
AD	Active Directory
VLAN	Virtual Local Area Network
IDS	Intrusion Detection System
IoC	Indicator of Compromise
PDP	Policy Decision Point
RIB	Rule-Based Industrial Baseline
SOC	Security Operations Center
DoS	Denial of Service
KPI	Key Performance Indicator
CSV	Comma-Separated Values
OSINT	Open-Source Intelligence
TOR	The Onion Router
FPR	False Positive Rate

Chapter 1 Introduction to the Study

1.1 Introduction

Cybersecurity threats are rapidly evolving, with attackers increasingly targeting endpoints as the weakest link in an organization's defenses (Waterson, 2020). Traditional perimeter-based security models are no longer sufficient in a world dominated by remote work, cloud adoption, and bring-your-own-device (BYOD) practices (Ojha et al., 2025). To address these challenges, Zero Trust Architecture (ZTA) has emerged as a leading security model, built on the principle of "never trust, always verify." (Stafford, 2020).

While ZTA provides robust frameworks for access control, a critical research gap remains: how to dynamically and in real-time assesses device security health to ensure only compliant and trusted/healthy devices access sensitive networks. Existing studies often focus on user-level verification, policy enforcement, and anomaly detection in traffic patterns, but lack a systematic approach to device security health checks integrated into ZTA policy engines (Gudala et al., 2021) (Yunanto et al., 2022). However, user-level and network-level analysis alone cannot guarantee complete confidence because a compromised or unpatched device can still serve as an entry point for attackers even if user credentials and network policies appear secure.

This study seeks to bridge this gap by leveraging machine learning (ML) to dynamically assess device security health. By analyzing security posture features such as patch status (Morris et al., 2020), antivirus protection (Moe et al., 2022), system behavior (Wang et al., 2018), vulnerabilities assessment (NIST SP 800-115; Walkowski et al., 2021), and exposure to external threats (Bermudez et al., 2018), the proposed approach aims to build predictive models capable of classifying and determining devices as safe (“accept”) or unsafe (“Deny”). This integration of machine learning and ZTA addresses the urgent need for automated, data-driven, and real-time device verification mechanisms, contributing to the advancement of both theory and practice in the field of cybersecurity. Introducing adaptive risk metrics for dynamic trust assessment

1.2 Significance of the Study

This study advances cybersecurity research and practice by incorporating machine learning-based device security health assessment into a zero-trust architecture (ZTA), elevating device dimension status to a first-order trust level, along with user and network factors (Yunanto et al., 2022). We design and test a comprehensive machine learning pipeline for endpoint classification and introduce risk-focused metrics that operationalize device status for policy engines, expanding the scope of data-driven adaptive access control models (Wazid et al., 2022).

In practice, the framework enables continuous, real-time verification of endpoints, so that compromised, outdated, or misconfigured devices are identified and blocked before they pose a risk. This results in: (a) a reduced attack surface, improved threat management and limited lateral movement (Kaur et al., 2025), (b) improved compliance through automated and continuous checks, and (c) operational efficiency by reducing manual workload and making access decisions repeatable and auditable. The findings provide a concise roadmap for enterprises and security operations centers (SOCs) to enhance trust decisions, and for vendors/developers to integrate device security health intelligence into ZTA platforms.

The research is relevant to multiple stakeholders includes, Enterprises by enhancing resilience against breaches and reducing exposure to advanced threats. Security professionals by providing a structured, data-driven framework and next-generation cybersecurity tools. Finally this study contributes a strategic vision and practical roadmap, bridging the gap between theoretical innovation and practical implementation of Zero Trust security.

1.3 Research Problem

Despite the growing adoption of zero trust architecture (ZTA), a fundamental challenge remains unsolved: How can device-level security health be continuously assessed and incorporated and integrated into ZTA policies to ensure robust and adaptive access decisions?. Current ZTA implementations primarily focus on user authentication and network-level verification. However, these implementations often lack effective mechanisms for real-time endpoint security health assessment, exposing organizations to

the risk of insecure devices that may contain unpatched vulnerabilities, outdated configurations, or compromised security.

In environments characterized by remote work, global teams, hybrid cloud infrastructures, uncontrolled devices and Bring Your Own Device (BYOD) practices, this gap becomes particularly critical. Even a single infected or misconfigured device can enable attackers to penetrate internal networks, expand privileges, and cause breaches (Anisetti et al., 2020).

This problem is of scientific importance because it reveals the limitations of implementing static policies and emphasizes the need for data-driven, adaptive approaches capable of continuously verifying the reliability of devices security health. The lack of appropriate datasets and practical experiments further complicates the problem.

There is currently no standardized or adaptive methodology for device security validation in dynamic environments, such as the rule-based validations implemented in systems (Onwuegbuzie et al., 2025), assess devices based on static conditions (such as antivirus installation or patch application). While useful, these rule-based mechanisms remain static and fail to capture the dynamic and evolving nature of modern threats. Focusing on isolated indicator integrating them into a standardized, constantly updated model of device health. This increases the risk of misclassifying devices, leaving networks vulnerable. Machine learning, enables the analysis of multi-source data and the detection of complex patterns for more accurate and adaptive classification, although challenges such as dataset availability, feature engineering, and balancing strategies remain unaddressed.

Another challenge is developing effective risk metrics that can accurately assess device security health and translate raw technical data into actionable insights, such as a comprehensive security posture assessment, risk level and risk rank. Without such metrics, Zero Trust Analytics (ZTA) applications will remain limited to static, fragmented scans, unable to detect the dynamic risks posed by modern enterprise environments.

1.4 Research Objectives

The primary objective of this study is to design and evaluate a machine learning-based framework that enables the detection and assessment of device security posture

integrity in Zero Trust Architecture (ZTA) environments. By focusing on the device-level security dimension, the research aims to improve access control decisions making and develop dynamic policies to ensure that only trusted devices are granted and gain access to the network.

This study will develop a quantitative device risk metric, aligned with the National Institute of Standards and Technology (NIST) and Factor Analysis of Information Risk (FAIR) models, to capture both likelihood and impact at the endpoint level. The metric will combine measurable status signals into a single risk score that serves two purposes: (a) data labeling for supervised machine learning (secure/accepted vs. insecure/denied), and (b) policy enforcement in ZTA. Trigger thresholds are adjustable and will be empirically validated during model development. risk-aware approach.

1.5 Research Questions

To achieve objectives, this study is guided by the following research questions: How can the proposed machine learning model be integrated into (ZTA) policies to improve real-time access control decisions? And how can device-level features, such as patch level, protection status, and discovered vulnerabilities, be systematically collected and engineered to provide an accurate reflection of endpoint security health.

In this regard, the study also asks: How can ML-derived Device Risk Metric be architecturally integrated into a ZTA Policy Decision Point (PDP), where It is consumed by Network Access Control (NAC)/Policy Enforcement Points (PEPs) to drive continuous and adaptive endpoint admission and session re-evaluation within the policy entry workflow.

1.6 Research hypothesis

This study is built on several hypotheses that guide its experimental design and expected contributions. Main Hypothesis (H): Incorporating machine learning-based device health checks into ZTA polices improves real-time access control decisions and device classification accuracy compared to static polices.

1.7 Study Limitations

This study has some limitations that must be considered. At a conceptual level, the scope of the study was intentionally limited to device integrity and security health checks

within a zero-trust architecture (ZTA). Other vital security dimensions such as user behavior analytics, and network traffic analysis were excluded from the study. While this narrow focus enhances the study's contribution to enhancing endpoint-based trust, it also limits its comprehensiveness within the broader zero-trust architecture.

Data privacy and security concerns, collecting detailed device security metrics, especially in a real-world enterprise and corporate environment, can raise data privacy concerns. Organizations may or may not accept to provide access to sensitive data about device configurations, user behavior, or security incidents due to confidentiality and compliance issues. Can mitigate and address this issue by ensuring that data collection follows strict anonymization and encryption protocols.

The collected dataset in this study consists of approximately 305 observations collected from simulated environments, including virtual machines, enterprise endpoints, and security operations center (SOC) training datasets. To address class imbalances and size of dataset, Synthetic data generation techniques were applied. While this approach improves data diversity, it may not fully simulate the complexities of large-scale, real-world environments. In terms of temporal limitations, the dataset was collected over a specific period. Therefore, this data may not fully reflect long-term changes in threats, vulnerabilities, and device behavior patterns, which may limit the generalizability of the results over long periods of time.

Due to spatial limitations, the experimental setup was limited to virtual lab and enterprise-level data collection related to device security health. Therefore, the results cannot be directly generalized to global or heterogeneous networks, which may exhibit greater variation in device types, security configurations, and the landscapes of threat.

Finally, at the methodological level, some commercial vulnerability assessment tools (such as Nessus) have been excluded due to licensing restrictions, which limits the scope of feature collection. To address this problem in this study, we programmed and customized a vulnerability assessment tool, as this is difficult and comparability with industry-standard applications may be limited.

1.8 Conceptual and Procedural Definitions

Zero Trust Architecture (ZTA): A security framework that enforces the "never trust, always verify" principle, ensuring continuous verification of all access requests.

Device Security Health: The overall resilience of a device to threats, measured by patch status, protections, vulnerabilities, and behaviors.

Network Access Control (NAC): Mechanisms that govern endpoint admission and ongoing network connectivity based on identity and device posture.

Policy Administration Point (PAP): The component that defines, manages, and distributes access policies to decision/enforcement components.

Policy Decision Point (PDP): The engine that evaluates policies against attributes (user, device, context, risk) and issues a decision (e.g., allow/deny/quarantine).

Policy Enforcement Point (PEP): The control that applies the policy enforcement point decision (for example, switch, VPN gateway, reverse proxy, endpoint agent).

Device Risk Metric (DRM): A quantitative score derived from mode signals (e.g., patch level, CVSS severity, control status, exposure) to express the liklihood/impact of an endpoint, used for data labeling and policy thresholds.

Machine Learning (ML): A subset of AI that enables systems to learn patterns from data and make predictions or classifications.

Feature Engineering: Identify and transform raw telemetry (e.g., patch level, CVSS, control states) into model-ready features.

Data labeling (for ML): Assign base labels from the DRM or baseline to the rule (e.g., low risk → accept, medium/high → deny) for supervised learning.

Synthetic data augmentation: Generate additional samples from empirical distributions to mitigate small sample size and class imbalance, while observing accuracy limits.

Class imbalance: A skewed label distribution (e.g., more accepts than deny) can lead to learning bias.

Rule-based industrial baseline (RIB): A transparent, standards-inspired baseline that uses deterministic checks (e.g., patch level, CVSS thresholds, control states) to produce accept/deny decisions.

Policy threshold: The cut-off point in the risk score or probability of a model that corresponds to accept versus deny, set on validation data and fixed for testing.

Privacy Anonymization: Techniques (such as hashing, tokenization, aggregation) applied to telemetry to protect sensitive identifiers while enabling analysis.

Chapter 2 Literature Review

2.1 Theoretical Background

2.1.1 Zero Trust Architecture (ZTA)

Zero trust architecture (ZTA) is a modern cybersecurity paradigm that assumes no implicit trust is granted to any user, device, or network component, even if they are within the traditional network perimeter (Gambo and Almulhem, 2025). In contrast to perimeter-based security (which previously trusted internal traffic by default), ZTA continuously verifies every access request under the principle "never trust, always verify," treating every user or device as vulnerable until proven otherwise (Lilhore et al., 2025). This paradigm shifts from static defenses to dynamic, context-aware controls emerged in response to evolving threats and distributed networks (such as cloud services, telework, and BYOD) that render old trust assumptions obsolete (NIST SP 800-207, 2020). The ultimate goal of ZTA is to protect data and resources through continuous authentication and conditional authorization for every interaction, rather than a one-time gateway scan. By verifying the identity, device status, and context of each session, and granting access only to the least privileged, ZTA reduces the likelihood of accounts or devices being compromised laterally moving across the network. This proactive approach, based on the assumption of compromise, significantly strengthens an organization's security posture, reducing the impact of insider threats and compromised devices (Gambo and Almulhem, 2025).

In practice, ZTA is implemented through a set of logical components that enforce ZTA principles across the network. The infrastructure components include a policy decision point (PDP) sometimes called a policy engine or controller, and one or more policy enforcement points (PEPs) (Wang et al., 2025). The PDP is responsible for evaluating incoming access requests against security policies and contextual trust data to determine whether the request should be allowed. It takes into account a comprehensive set of attributes, the user's identity and credentials (with multi-factor authentication being common practice), the device's state (such as operating system version, security patches, and compliance status), the sensitivity of the resource being accessed, and environmental factors (time, geographic location, observed anomalies, etc.) (Wang et al., 2025; NIST SP 800-207, 2020). Based on these factors, the PDP calculates a dynamic trust score or

judgment for the request. On the other hand, the PEP protocol resides in the data path (at network gateways, application front-ends, etc.) and enforces the (PDP)'s decision to allow, restrict, or block traffic. For example, a PEP at an application API gateway only establishes a client session after receiving OK from the PDP, and can terminate the session if the PDP subsequently revokes consent due to a security change. This separation of decision and enforcement is fundamental to ZTA's resilience, policies can be centrally managed and settled in the PDP, while enforcement is distributed closer to the assets. Modern ZTA implementations often include continuous risk scoring, where the trust levels of the user, device, and session are recalculated immediately. If user behavior becomes anomalous or the device fails to comply with standards, ongoing sessions can be reevaluated or terminated immediately (Gambo and Almulhem, 2025). This dynamic access control contrasts with older models such as static network access control lists or one-time login grants, significantly limiting the opportunities for attackers to escalate privileges or remain unnoticed.

ZTA's emphasis on dynamic, context-based policies makes it particularly relevant to adaptive access control and device security. Traditional identity and access management (IAM) systems (such as role- or attribute-based access control) have often been static and perimeter-centric, making it difficult to address the dynamic and distributed nature of cloud and mobile environments (Wang et al., 2025). With zero trust, devices are treated as an integrated security entity: the "health" of each device is continuously evaluated before and during its network connections (NIST SP 800-207, 2020). For example, if a enterprise laptop is missing critical patches or shows signs of compromise, a ZTA policy might automatically restrict its network access or route it to remediation VLANs until it’s secured, in other words, "never trust" the device until it proves healthy. This concept, sometimes called device trust or device posture, is critical to thwarting threats from insecure endpoints. By assessing device status (antivirus software, operating system version, disk encryption, etc.) as part of access decisions, ZTA ensures that even authenticated users cannot connect to unsecured devices (NIST SP 800-207, 2020). Essentially, verifying the user's identity alone is not enough, the device and context must also be verified. This significantly reduces risks, such as untrusted devices or attackers exploiting stolen credentials on unmanaged systems (Gambo and Almulhem, 2025). The comprehensive scanning and continuous

monitoring provided by ZTA help limit the scope of attacks, even if one component is compromised, automatic zero-trust and micro-segmentation prevent attackers from freely moving to other systems.

Since around 2020, Zero Trust has rapidly gained traction in both industry and academia. Analysts report that the majority of companies worldwide have at least partially implemented Zero Trust, reflecting its perceived effectiveness in the era of cloud and remote work (Gambo and Almulhem, 2025). Gartner predicts that the global Zero Trust market will grow significantly (e.g., exceeding $130 billion within a decade) (Gambo and Almulhem, 2025). In research, there has been an explosion of studies, frameworks, and use cases focused on ZTA. For example, the NIST Standard ZTA publication (SP 800-207) in 2020 established core principles and design patterns that have influenced both corporate strategies and government mandates (NIST SP 800-207, 2020). A systematic literature review in 2023 highlighted that numerous studies have analyzed ZTA principles, proposed improvements (such as integrating blockchain or advanced trust algorithms), and explored its application in fields ranging from cloud computing to the Internet of Things (IoT). These recent works highlight the versatility of zero trust, for example, applying continuous authentication to IoT devices in the healthcare sector, or implementing “zero trust” access authorization using smart contracts. In summary, zero trust architecture has evolved from a buzzword to a fundamental pillar of modern cybersecurity. Its “always verify” approach, fine-grained controls, and adaptability to changing conditions make it well-suited for today’s dynamic access needs and advanced threat landscape (Jambo and Almholm, 2025).

2.1.2 Machine Learning in Cybersecurity

Machine learning (ML) is a branch of artificial intelligence that enables systems to learn patterns from data and improve decision-making over time without the need for explicit programming. In the context of cybersecurity, machine learning techniques have become invaluable for detecting and responding to complex or novel threats that defy simple rule-based detection (Mohamed, 2025). Traditional security tools (such as signature-based antivirus or static firewall rules) struggle to keep up with complex attacks such as zero-day exploits, advanced malware, or insider information abuse, which may not match any known signatures or pre-defined patterns (Mohamed, 2025). Machine learning addresses this gap by adapting to evolving threats: it can analyze massive amounts of

security data (network traffic, system logs, user behavior, etc.) and learn what normal behavior looks like versus malicious behavior, often in real time (Mohamed, 2025). This adaptive, data-driven approach has transformed cybersecurity from a reactive practice (reliant on known indicators of compromise) to a more proactive one. For example, a machine learning-based system might flag a subtle deviation in network login habits or process activity as suspicious, enabling early detection of an attack that would bypass traditional defenses. Indeed, one of the biggest advantages of machine learning in security is its ability to perform anomaly detection, identifying “outliers” or abnormal patterns that may indicate previously unseen threats. Studies have shown that AI/ML models can detect hidden zero-day attacks or advanced persistent threats (APTs) by recognizing statistical anomalies (such as unusual network flows or user actions) that humans or signature scanners might miss. By continuously learning from new data, these models evolve alongside attackers’ tactics, bridging the gap where static defenses become blind. Additionally, machine learning automates and speeds up many tasks, such as analyzing millions of log events or quickly classifying malware samples, significantly improving detection and response speed and reducing the burden on security analysts (Mohamed, 2025). In summary, machine learning provides a dynamic, scalable, and intelligent layer of cybersecurity that complements and enhances traditional controls.

In practice, a wide range of machine learning techniques and algorithms are used in cybersecurity tasks, particularly for intrusion detection, malware analysis, and user behavior modeling. These techniques can be broadly classified into supervised learning (using labeled attack examples against normal data), unsupervised learning (finding anomalies or clusters without labels), and hybrid or reinforcement learning methods. Common classification algorithms used in security include support vector machines (SVMs), decision trees, random forests, and neural networks (Mohamed, 2025). For example, in network intrusion detection systems (IDSs), a supervised machine learning model may be trained on historical network flows labeled as "benign" or "malicious." Using a decision tree or SVM, the model learns to classify new traffic by considering features such as packet rates, protocol patterns, or payload content. These models have achieved high accuracy in detecting known threats and can be updated with new training data as attacks evolve. Neural networks, especially deep learning models, are becoming

increasingly popular in cybersecurity due to their ability to capture complex, nonlinear patterns. Deep learning (such as the use of deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks/long-term neural networks (LSTMs)) has proven effective in analyzing multidimensional and temporal data in security events. For example, LSTMs can learn sequences of system calls or network events to detect the slow and subtle behaviors of advanced threats that unfold over time. These models can even be generalized to detect polymorphic malware (malware that constantly changes its code) by learning underlying behavioral patterns rather than specific byte signatures. On the other hand, unsupervised machine learning techniques are widely used for anomaly detection a critical capability for discovering insider threats or new attack variants for which no prior examples exist. Techniques such as clustering (e.g., K-Means and DBSCAN) and density-based anomaly detection (e.g., Isolation Forest and Local Outlier Factor) establish a baseline of normal activity and then identify outliers that deviate significantly. Similarly, self-encoding neural networks can compress typical behavior patterns. and noticing when an input (such as a user's action sequence or a device's network traffic profile) does not conform to the learned standard. These anomaly-based methods are particularly effective at detecting previously unknown threats, for example, a sudden increase in data from an accounting computer at 3 a.m. could be an anomaly indicating a data leak, even if the malware or user misuse had not previously been detected. Their drawbacks include the potential for more false positives (because not every anomaly is malicious), but they significantly enhance the visibility of unusual events. In practice, many security solutions now use a hybrid approach, combining supervised classification of known threats with unsupervised anomaly detection of unknown threats (Mohammed, 2025). This results in a more robust defense that can recognize known attack patterns and adapt to new behaviors.

Machine learning is a natural fit and force multiplier for zero-trust architectures. Zero-trust environments generate vast amounts of telemetry continuous authentication logs, network micro-segmentation logs, and, in our study, real-time device posture assessments, etc. Which can be too large or complex to analyze manually. By incorporating machine learning-driven analytics, a zero-trust system can intelligently interpret this data to make or suggest access decisions in an automated manner (Laghari et al., 2025; DISA & NSA,

2022). For example, machine learning models can assess the risk of each access request by examining dozens of attributes (user role, historical behavior, device vulnerability status, geographic location, time of day, and anomalies) much faster than a human administrator can. The U.S. Department of Defense's Zero Trust Architecture Reference explicitly states that machine learning algorithms are used to identify baseline normal patterns and provide data inputs for zero-trust policy enforcement (DISA & NSA, 2022). In other words, machine learning helps determine what is "normal" for a given user or device and continuously adjusts the trust level. If machine learning detects an anomaly (e.g., a user downloading a large amount of data or a device connecting to a rare host), the zero trust policy engine can automatically request additional verification or cut off access (DISA & NSA, 2022; Portnox, n.d.). This type of risk-adaptive access control is a hallmark of advanced ZTA deployments: the system leverages artificial intelligence/machine learning (AI/ML) to determine the amount of trust granted in real time. Recent research and solutions often refer to "AI-driven zero trust," where techniques such as behavioral analytics (aided by machine learning) feed into the zero trust decision loop (Laghari et al., 2025). Concretely, machine learning models integrated into a Zero Trust network might analyze network flows to detect any compromised IoT device exhibiting anomalous traffic and signal the controller to quarantine it, or analyze user keystroke dynamics to verify that the logged-in user is indeed the legitimate user (session hijacking detection). By automating threat detection and context analysis, machine learning enables Zero Trust systems to be dynamic and scalable, applying the principle of "always verify" to every access without burdening human administrators (Laghari et al., 2025; DISA & NSA, 2022). In summary, machine learning has become a cornerstone of modern cybersecurity. Its ability to perform intelligent anomaly detection, threat classification, and adaptive policy control makes it essential to achieving the full vision of a Zero Trust architecture, where decisions must be made continuously and correctly amidst a sea of data. As threats continue to evolve in complexity, combining the analytical power of machine learning with the rigorous security model of Zero Trust provides a promising path to staying ahead of attackers (Laghari et al., 2025; Mohamed, 2025).

2.1.3 Network Access Control (NAC)

Network Access Control (NAC) is a security mechanism that manages and determines who (and which devices) can connect to an enterprise network, and under what conditions. Essentially, NAC acts as a gatekeeper, it grants or denies device access to the network based on predefined security policies and real-time assessments of device compliance and identity (Cisco, n.d.). This is critical in enterprise networks where a wide range of devices (corporate PCs, BYOD mobile devices, IoT endpoints, guest laptops, etc.) are attempting to connect, each device must be checked for security requirements before it can connect to the network. A typical NAC solution, such as Cisco ISE or similar, authenticates the user/device upon connection (via methods such as 802.1X, web portal login, or device certificates) and simultaneously assesses the security posture of the device. If a device is incompatible, for example, missing patches, outdated antivirus software, or an unrecognized device, NAC can deny it access entirely, or place it in an isolated VLAN or restricted area where its capabilities are limited (for example, only able to access processing servers). This prevents insecure or rogue devices from infiltrating the main network and potentially spreading malware or stealing data. Only devices that successfully authenticate and pass policy checks are allowed normal access to network resources. NAC thus provides robust network visibility and control, giving administrators the ability to control who and what is on their network at all times (Cisco, n.d.). In today's environment of explosive mobile and IoT growth, NAC has become essential, it is not possible to implicitly trust every device within the network by default, so NAC helps enforce trust on a per-device basis at the entry point.

Modern NAC solutions offer a range of capabilities for securing enterprise networks. common NAC functions include (Cisco, n.d.):

• Device identification and profiling: NAC systems can automatically identify and profile devices before they fully join the network. Using techniques such as DHCP fingerprinting, MAC address lookup, or proxy-based scanning, NAC can identify device attributes (OS type, device type, etc.) and apply appropriate policies. For example, a laptop managed by IT may be allowed access after a posture check, while an unrecognized personal device may be classified as a guest and granted exclusive access to the internet.

Early profiling helps prevent malicious or unknown devices from accessing the network by enforcing differentiated access.

• Security posture checking and compliance enforcement: A key feature of NAC is assessing device compliance with security policies (based on user role, device type, etc.). The Network Control Center (NAC) can check whether the required security software (firewall, antimalware) is enabled on the device, whether the operating system is up to date, whether any blocked applications are running, etc. If the device fails any check, it can mitigate the threat by automatically blocking or quarantining it. For example, a contractor's laptop that doesn't meet the company's patch level may only be able to access a restricted update portal. This automated implementation helps keep the overall network clean without manual intervention.

• Incident Response and Quarantine: The Network Control Center (NAC) can integrate with threat detection systems to respond to compromised devices in real time. If a device is flagged by an intrusion detection or EDR (Endpoint Detection and Response) system as infected, the NAC solution can immediately revoke its access to the network or move it to an isolated VLAN, containing the threat's propagation. This rapid, policy-based response (often called a "dynamic NAC ") is an effective way to stop active attacks. The NAC's ability to block, quarantine, or repair devices immediately, without waiting for human intervention, significantly reduces incident response time and damage.

In addition, NACs often manage guest access (by providing temporary internet access to visitors via a captive portal), integrate BYOD devices (securely enrolling personal devices), and integrate with directory services and identity providers (such as Active Directory, Azure AD, and Okta) to link network access decisions to user identities and roles (CloudNuro, 2025).

Machine learning can complement NAC systems by making them more intelligent and adaptive. Traditional NAC policies are often rule-based (such as allowing known MAC

addresses and denying outdated operating system versions), which may not identify all risky scenarios, especially with the rise of sophisticated or insider threats. By incorporating machine learning-based analytics, NAC solutions can analyze the behavior of devices and users over time, rather than just one-time scans (CloudNuro, 2025). For example, a machine learning-enhanced NAC system might learn the normal network behavior of a corporate printer or IoT sensor. If that device suddenly starts sending large amounts of data to an unfamiliar server, the NAC system could detect the anomaly and shut down the device or alert security teams. Recent research proposals illustrate this trend. In 2022 study presented a dynamic NAC framework (called SADAC) that continuously evaluates the "security profile" of mobile devices in a Wi-Fi-based network environment and users over time and adjusts their access privileges accordingly (García, 2022). This approach utilizes attributes from the mobile device's configuration and operation (such as recent vulnerabilities, running processes, Wi-Fi and Bluetooth activity, and battery level) and implements a dynamic supervision loop, If a device's security posture deteriorates or behaves erratically, the system automatically tightens its network access. Such adaptive NAC can diagnose the root causes of noncompliance (such as identifying a broken security setting on the device) and guide the device/user to remediation (García, 2022). Under the hood, techniques such as anomaly detection or pattern recognition (classic machine learning tasks) can drive these decisions, essentially enabling risk-adaptive network access control. Furthermore, NAC vendors are beginning to incorporate AI features, many support agentless device fingerprinting and behavioral analytics to address IoT and unmanaged devices. Instead of relying solely on lists of known devices, NACs can monitor device behavior on the network (which protocols they use, when they connect, and which resources they access), and if behavior falls outside the profile, they are classified as high-risk and quarantined (CloudNuro, 2025). This is particularly useful for detecting compromised devices (for example, an IP camera embedded in a botnet might start scanning the network, a NAC with machine learning can notice this anomaly and isolate it). In summary, machine learning enhances NAC by providing continuous learning and anomaly detection, making network access decisions more contextually aware and threat-sensitive than static rules alone.

In a Zero Trust architecture, NAC plays a pivotal role as one of the network-level enforcement mechanisms. The Zero Trust principle of trusting no device by default aligns perfectly with NAC's mandate to verify the trustworthiness of each device before allowing access. In fact, NAC can be thought of as an implementation of Zero Trust at the network edge, it ensures that only authenticated and compliant devices (which can also mean authenticated users on those devices) pass through the "front door," and even then only to specific authorized network segments. Implementing NAC is often one of the first steps organizations take when pursuing a Zero Trust strategy, as it addresses a fundamental question: Can I trust this device on my network right now? In Zero Trust, the answer should always be "not until it's verified," and NAC provides this verification through device authentication and status checks. It also provides device-level authentication that can feed into the broader Zero Trust policy engine. For example, a Zero Trust system might incorporate NAC status information into its access decisions: If the NAC reports a device as non-compliant or unknown, the Zero Trust policy will not grant it access to sensitive applications. This tightens the overall security posture. Notably, Gartner's concept of Zero Trust Network Access (ZTNA) expands the idea of NAC (which traditionally secures internal LANs) to include application-level access, particularly for remote users, effectively replacing VPNs with a model where each application's access is individually authenticated and authorized. While classic NAC focuses on LAN/WLAN access, the philosophies overlap and complement each other within the full Zero Trust framework. NAC also generates valuable telemetry data (logs of who connected, from where, device status, etc.) that can be used for continuous monitoring and anomaly detection, in line with Zero Trust's need for continuous visibility. At its core, NAC implements the principle of "trust no device, always verify" at the network entry point, a key component of Zero Trust defense in depth. Industry experts emphasize the importance of NAC in Zero Trust deployments, as it provides "fundamental device-level control and enforcement of security posture, which is critical for verifying trust before allowing access." Without NAC or a similar control system, a malicious or infected device could freely join an internal network and undermine Zero Trust efforts. Therefore, NAC remains a core component of any machine learning-enhanced Zero Trust approach, working in conjunction with identity management and analytics to secure the network at the device level. Recent best practices recommend

combining NAC with other Zero Trust components (such as identity providers, multi-factor authentication (MFA), and security analytics) to achieve a coherent and adaptive security posture (CloudNuro, 2025).

2.1.4 Risk Management in Cybersecurity and Zero Trust Architecture

In cybersecurity, risk management is a continuous process that includes identifying risks, analyzing or assessing their potential impact, implementing mitigation measures, and ongoing monitoring. These steps ensure that organizations systematically address threats to their assets and operations. Risk identification involves identifying potential threats and vulnerabilities that could impact critical systems or data. Risk analysis (assessment) then evaluates the likelihood and impact of these threats, prioritizing risks based on severity. Following the assessment, organizations plan and implement mitigation or response strategies, by deploying controls or changes to reduce the risk to an acceptable level (e.g., strengthening access controls or patching vulnerabilities). Finally, continuous risk monitoring and reporting is conducted to track the status and effectiveness of risks over time. Frameworks such as the National Institute of Standards and Technology (NIST) Risk Management Framework and ISO 27001 emphasize this lifecycle approach, emphasizing that risk management is not a one-time project, but rather an iterative practice. Effective risk management aligns cybersecurity investments with the most critical risks, thus maximizing the security return on investment for the organization (Microsoft, 2022).

Aligning Risk Management with Zero Trust Architecture (ZTA): Zero trust architecture is inherently a risk-based security model. As one industry expert noted, "Compliance is about managing and mitigating risk, and zero trust is the same." Traditional perimeter-based security systems have assumed that internal network traffic can be trusted, but ZTA assumes no implicit trust "never trust, always verify", and thus directly addresses the risk of insider threats or network breaches (Microsoft, 2022; TrustCloud, 2025). Instead of static, network-defined trust, zero trust enforces identity and risk-based access controls. Each access request is authenticated and continuously evaluated based on contextual risk factors, such as user identity, device posture, geolocation, and behavioral anomalies, before access is granted. This applies the principle of least privilege, meaning that users or devices receive only the minimum access necessary, a proactive risk mitigation to limit potential damage in the event of an account compromise. For example, robust identity verification

(such as multi-factor authentication) reduces the risk of unauthorized access, and micro-segmenting networks contains the "blast zone" of breaches by preventing lateral movement. At its core, ZTA's core principles (explicit verification, enforce least privilege, and assume breach) align with risk management objectives, continuously assessing risks before and during each session, and mitigating them through tighter access controls. ZTA approaches also integrate with the risk management lifecycle. During the identification phase, Zero Trust encourages a holistic view of assets across identities, endpoints, applications, networks, and data, ensuring that no asset or vulnerability is ignored. To assess risk, ZTA offers continuous risk assessment, instead of periodic reviews, each access request triggers a real-time risk assessment based on policy rules and contextual signals. In terms of response, zero trust systems can automatically enforce adaptive policies, for example, blocking, restricting, or requiring additional authentication for a suspicious session, in real-time, aligning with threat response strategies to mitigate or avoid threats. Finally, ZTA improves monitoring by providing granular monitoring and logging at all levels (user, device, transaction), enhancing an organization's ability to detect incidents and continuously report on its security posture. This granular monitoring means security teams can track risk indicators (such as abnormal data downloads or malware alerts) and aggregate them into an enterprise risk view, balancing operational security with governance and compliance needs (Microsoft, 2022; TrustCloud, 2025). In summary, Zero Trust is a modern architecture designed specifically for risk: it operationalizes risk management principles by incorporating continuous verification and applying least privileges into every access decision, reducing cybersecurity risks in a dynamic threat environment. In fact, recent research suggests that adopting ZTA can significantly enhance an organization's security posture and reduce cyber risks, provided that organizations also manage the new risks and complexities that ZTA itself may introduce (Abdul Majeed and Diaz, 2025).

Improving Risk Management with Machine Learning in ZTA: Machine learning (ML) techniques enhance risk management in the context of Zero Trust by enabling dynamic risk scoring, behavioral analytics, and adaptive policy enforcement. Traditional rule-based or static risk models may not keep pace with rapidly evolving threats, whereas machine learning systems can learn patterns and adjust risk assessments instantly. For example, machine learning-powered User and Entity Behavior Analytics (UEBA)

establishes baselines of normal user behavior and continuously detects any anomalies that may indicate a risk. If an authenticated user suddenly performs unusual actions (such as downloading an unusually large amount of sensitive data or accessing systems at unusual times), a machine learning-enhanced Zero Trust system can flag this with a heightened risk assessment and trigger an adaptive response, such as requesting enhanced authentication or denying the action in real time (TrustCloud, 2025). This dynamic risk assessment evaluates signals from across the IT environment (login characteristics, device health, geographic location, recent alerts, etc.) and calculates a risk level for each request or entity. Recent studies have demonstrated the effectiveness of dynamic risk assessment. For example, Koli et al. (2025) propose an AI-powered internal user risk management system that integrates behavioral analytics with dynamic risk assessment to enable real-time policy enforcement and threat mitigation. In their approach, a machine learning model analyzes user activity logs to continuously update risk estimates, allowing the system to automatically enforce security policies (such as blocking access or triggering an incident response) when risks exceed a certain threshold (Koli, 2025). This type of risk-based adaptive access control is a natural extension of zero-trust principles often called risk-adaptive access or continuous adaptive trust. This means that access decisions are not one-time binary checks, but are flexible and responsive to the latest risk context. Furthermore, machine learning improves the accuracy of threat detection by reducing false positives and detecting subtle malicious behaviors that humans might miss. Techniques such as anomaly detection, clustering, and classification can sift through large-scale security data (network logs, endpoint data, etc.) to uncover patterns indicative of attacks, thus enriching the risk management process. Machine learning also benefits risk mitigation automation, advanced systems can automatically contain or remediate threats (e.g., isolate a device behaving suspiciously) without requiring human intervention, reducing response times and minimizing potential damage. Overall, integrating machine learning into risk management enables organizations to continuously measure and respond to risks in a more accurate and proactive manner. This promotes zero-trust deployments by ensuring that security policies are always informed of up-to-date risk assessments, ultimately enhancing the organization's ability to prevent breaches and adapt to new attack techniques (Koli, 2025; TrustCloud, 2025).

Modern cybersecurity frameworks recognize that effective risk management is essential to defending against advanced threats. Zero trust architecture enables risk management by requiring continuous verification and least privilege access, making security decisions informed and risk-based. The addition of machine learning brings a dynamic edge to this process, enabling real-time risk assessment, intelligent behavioral insights, and automated execution. Together, these approaches create a resilient security posture where access to resources is consistently governed by the principle of risk reduction, aligning technology, policies, and controls with the ever-changing threat landscape. Studies published since 2020 strongly support that organizations adopting zero trust, supported by machine learning-based risk management, can significantly reduce their exposure to cybersecurity risks while maintaining the resilience of their defenses (Abdulmajid and Diaz, 2025; Koli et al., 2025).

2.1.5 Research Gap and Motivation

The primary research question of this study is: How can machine learning be integrated into Zero Trust Architecture (ZTA) to dynamically assess device security health and improve access control policies in real time? While zero trust and machine learning frameworks have been individually explored in the cybersecurity domain, limited research has addressed the integration of machine learning to assess device security health in dynamic policy engines in zero-trust environments. Additionally, there is a poor of relevant datasets on this topic. This chapter demonstrates how existing studies focus on adjacent domains but fail to explore the dynamic machine learning-based approach to assess device security health in the context of zero trust.

Several Zero Trust Architecture (ZTA) Studies have focused on ZTA as an effective framework for cybersecurity, emphasizing its principles of “never trust, always verify”, least privilege access, and “assume breach”. However, most of these works primarily address policy enforcement and access controls leaving a significant gap in addressing delving deeply into how real-time device health assessments enhance security in device security health checks.

This study aims to fill a significant gap by integrating machine learning into ZTA for real-time device security health checks and assessments within a dynamic policy engine.

While existing research has lacked a of combination ZTA with machine learning to device security health check in policy engine management, to create a dynamic, machine learning-driven system for device security assessment in ZTA environments. For example, previous studies focus on network-level threat detection but do not integrate these findings into ZTA frameworks. This research expands the body of knowledge by developing a new methodology that combines device health assessments and real-time access control, creating a more secure and adaptable network environment.

This research contributes to the existing literature by presenting a novel approach to integrating machine learning into zero-trust architectures, with a particular focus on device security health validity in dynamic policy engines. It bridges the gap between machine learning-based threat detection and zero-trust policy enforcement, providing a dynamic solution that enhances real-time decision-making. This study will establish a framework that not only enhances security in BYOD environments but also lays the foundation for future research in data-driven intelligent access control systems.

2.2 Study

The authors in (Gudala et al., 2021), Integrated machine learning with Zero trust Architecture (ZTA), the ML models have been effectively applied to detect anomalies in user behavior, network traffic, and system configurations, allowing for automated response and mitigation of threats, such as account lockout and device isolation. However, it does not focus on real-time device security health assessments to take proactive security measures.

In (Ramezanpour et al., 2022), the authors designed an intelligent ZTA (i-ZTA) to secure the dynamic and untrusted environments of 5G/6G networks, the research uses reinforcement learning to dynamically approve access requests and assess security risks in real-time in IoTs devices. This study mitigates key vulnerabilities such as lateral movement, denial of service (DoS) attacks, and man-in-the-middle exploitation, but it lacks focus on device security health indicators as a determinant of access.

In healthcare, (Edo et al., 2024) uses ZTA to counter insider threats and close healthcare vulnerabilities such as data breaches, and patient records sold on the dark web.

Using guidance from the National Institute of Standards and Technology (NIST), they primarily focus on user-level controls (e.g., separate role-based profiles, unique identifiers to enhance privacy, activity logs for monitoring, and threat intelligence mechanisms to understand user behavior, and anomalies and prevent unauthorized lateral movements within the system) rather than device-level security health checks.

For IoT security, (Outchakoucht et al., 2017), proposes a dynamic and distributed access control policy using blockchain and machine learning, specifically reinforcement learning, to enhance security. Traditional security solutions are not suitable for IoT due to device limitations and centralized and static access control, the research uses blockchain to achieve decentralization and reinforcement learning for dynamic and adaptable policy. There remains a gap in real-time assessments of IoT device security health as a ZTA-enhancing factor.

The authors in (Detken, et al., 2017), proposed CLEARER project that integrates security information and event management (SIEM) features into network access control (NAC), NAC systems control access to networks through authentication, authorization, and role-based controls, it prevents unauthorized or infected devices from accessing networks but lacks the real-time security analysis that SIEM systems cover. SIEM provides real-time monitoring, log auditing, and incident response. This approach does not fully utilize device security health checks with machine learning to preemptively secure network access.

Research by (Li et al., 2024) highlights the importance of a zero-trust security approach for protecting the rapidly expanding Internet of Things (IoT), particularly in 5G network environments. This research proposes a new framework, BasIoT, for securing device and data authentication using blockchain technology, adhering to the principle of "assume breach and continuously verify." While this approach effectively addresses scalability and trust issues in IoT systems, relying on blockchain and continuous verification and monitoring can introduce latency and complexity, potentially hindering the need for fast and efficient verification in IoT security environments. In contrast, our research focuses on data-driven machine learning models instead of blockchain-based trust management, aiming to provide a more flexible and efficient assessment of device security status.

The authors in the study (Wang et al., 2024) proposed a deep learning-based approach for anomaly detection and network log analysis, focusing on improving the accuracy and efficiency of detecting abnormal behaviors through automated analysis and feature extraction from network logs. This approach demonstrates the capability of deep learning models to process large and complex datasets, particularly in enhancing network intrusion detection systems. However, while their research focuses on detecting general anomalies in computer network traffic, it does not directly address client trustworthiness assessment in zero-trust architecture (ZTA). Our research addresses the more specific issue of evaluating device security levels and making decisions based on that assessment. The limitation of the previous study lies in its lack of focus on device security health level.

2.3 Feature justification and security concepts

The features selected in this study are deeply rooted in the security domain and supported by previous research and best practices in cybersecurity. These feature such as failed login attempts, the presence of unsigned drivers, pending updates, User Account Control (UAC) level, real-time protection status, tamper protection, antivirus status, memory and CPU usage, open ports, and session activity were not chosen randomly. Each feature reflects an important indicator of the security health of a device or system from a security perspective. For example, multiple failed login attempts may indicate brute force attacks, while unsigned drivers are often exploited to execute kernel-level malware. Similarly, pending system updates and the disabling of real-time protection could indicate patch mismanagement or deliberate security reductions caused by malware.

These features have been confirmed in related work, such as studies using behavior-based detection techniques, endpoint monitoring, and zero-trust assessments. By focusing on features of proven importance in the field of cybersecurity, this research ensures that the model is consistent with realistic threat scenarios and provides valuable insights for proactive defense mechanisms. (Mazhar et al., 2021), mention that Device identification and profiling helps reduce the likelihood of IoT attacks and attack surface by verifying the health of devices on the network, just like user authentication.

Resource Utilization, including Memory, CPU, Disk, and Network utilization, that refers to and Indicates the amount of computing resources being consumed in the system.

(Wang et al., 2018), To enhance security, help protect cell phone users from eavesdroppers while sharing resources efficiently. Also (Jamshidi et al., 2025), They analyze cyber threats and their impact on power consumption, CPU usage, and load. The system ensures robust security with minimal power consumption and CPU overhead.

And (Vokorokos et al., 2015), propose an effective resource management mechanism within the Apache web server to optimize resource utilization and deal with security threats such as denial-of-service attacks. The study (Hoque et al., 2024), focuses on resource consumption aspects, such as energy, computing power, memory, latency, bandwidth, and human resources, all of which are essential factors for enhancing the efficiency, reliability, and sustainability of network security solutions.

According to Trend Micro's Deep Security documentation, insufficient memory may prevent packet processing due to exhausted system resources (Trend Micro, Intrusion Prevention Reference). Additionally, Attackers often use injection techniques that directly manipulate memory, such as DLL injection or reflective loading, which increases memory usage (MITRE ATT&CK – T1055.015).

(Kavalanekar et al., 2008) they suggest that Collecting large sets of comprehensive traces is difficult because the high disk workload can raise security and system performance concerns. And in the study (Thummapudi et al., 2023) ensured that ransomware typically accesses files from the hard disk and uses the processor intensively to encrypt data, resulting in intensive activity that consumes system resources, a useful indicator for malware detection. Similarly, when data is leaked to the dark web, abnormal spikes in network usage may indicate a potential security breach.

The File Transfer Protocol (FTP) includes several mechanisms that may pose security risks (Allman et al., 1999), such as allowing a client to direct the server to transfer files to a third-party device, potentially compromising the network. An attacker could exploit a vulnerability in the server service's remote procedure call (RPC) handling to execute arbitrary code with system-wide privileges, potentially leading to a complete system compromise and enabling lateral network movement (Ullah, 2016). Telnet and RDP, widely used for remote access to devices, are prime targets for attackers. They are often

exploited through connection attempts and brute force attacks (Başer et al., 2021) (Mohta et al., 2024).

While (FTP) is commonly used by servers for file transfer services, but users' devices also establish FTP connections to upload or download files. Monitoring FTP ports helps identify insecure file transfers and potential exploitation attempts.

Also Telnet is a remote access protocol historically used to manage network devices and servers. However, clients can also initiate Telnet sessions. Because it transmits credentials as plain text, it is often a target for brute-force attacks.

According to MITRE T1068 Exploitation for Privilege Escalation Documentation, Attackers use vulnerable or unsigned drivers to escalate privileges or execute code in the kernel. (MITRE ATT&CK – T1068).

A running process is an instance of a program running on a computer. It has its own memory space, system resources, and execution context. MITRE ATT&CK – T1055 Describes how attackers use running processes to inject malicious code, and monitoring the running process is very important to detect the malicious process (MITRE ATT&CK – T1055).

External IP address (also called a public IP address) is the address assigned to a device by an Internet Service Provider (ISP) to communicate with the outside world, according to (Vigna et al., 1998), NetSTAT is a new approach to detecting network intrusions. Using a formal model of both the network and the attacks, NetSTAT can identify which network events to monitor and their locations. Attackers often exhilarate data or establish command and control (C2) through connections to external IP addresses (MITRE ATT&CK – T1071).

Normal user sessions (from the query user) - local users or RDP console users logged into the system, and remote Telnet sessions, are identified by checking TCP port 23 in the established state. Weak session control allows unauthorized access and session hijacking. Attackers often exploit weak session management mechanisms to impersonate legitimate users (OWASP – Session Management Cheat Sheet

Failed login (or failed authentication attempt Event ID 4625) occurs when a user or process provides invalid credentials (username, password, token, etc.) while trying to access a system or service. In study (SITAPURA, 2022), Different types of failed login attempts, common usernames attempted, recurring attack sources over time, and the geographic location of the attackers. Attackers attempt various methods to breach the system, using techniques such as brute force attacks.

Uptime is total time the system has been running since its last reboot or Boot time, the exact date and time the system was last started. In thesis study (Moe et al., 2022) indicate that operating systems may contain new bugs and threats that need to be addressed. Scalability and performance must also be ensured to maintain optimal operation. And keep the uptime high important.

Pending updates refer to software or system updates (for example, Windows security patches, driver updates, and application fixes) that have been downloaded or selected but have not yet been installed or applied. In study (Morris et al., 2020) mention Installing a pending update or experiencing a certain "Stop" state, tending to check for updates, checking for updates to fix security issues.

UAC (User Account Control) is a security feature in Windows that helps prevent unauthorized changes to the operating system by requiring user or administrative approval before performing sensitive actions. In study (Moe et al., 2022) also mention If a user's device is infected with malware, it is essential to limit its ability to spread. User Account Control (UAC) plays a key role in mitigating the impact of these infections. By applying the "minimal privileges" principle, UAC ensures that users operate with only the permissions necessary for their current tasks. This reduces the chances of malware or attackers gaining administrative privileges to access sensitive data or make unauthorized changes to the system.

Antivirus software is security software designed to detect, block, and remove malware, such as viruses, worms, Trojan horses, spyware, and ransomware. It uses signature-based, heuristic, and behavioral detection methods. Real-time protection is a

feature of antivirus or endpoint protection systems that constantly monitors the device and scans for running process, File changes and executable code, etc.

Tamper Protection prevents attackers or unauthorized users (even with administrator rights) from modifying security settings, especially those related to Microsoft Defender Antivirus. In study (Moe et al., 2022) also mention that Real-time protection is one of the most important and advanced features of Microsoft Defender. It is designed to protect the system 24/7, constantly monitoring for threats.

This feature is enabled by default in Windows 10 and 11, but can be disabled if desired. Real-time protection scans all downloaded files and applications against a comprehensive threat database, immediately blocking and removing anything deemed suspicious. This process, known as "first-sight blocking," relies on machine learning and cloud-based threat intelligence. Additionally, this feature performs regular background checks to ensure the system is free of malware and other security risks. For most users, Windows Defender will be a standard security function already present on the system, and should not be tampered with. Windows Defender should continue to be used due to its broad range of built-in security functions.

Dark Web mentions include any reference to the organization, IP addresses, user credentials, domains, or sensitive data found on Dark Web forums, marketplaces, or breach dumps. In study (Bermudez et al., 2018), mention It's important to note that credentials are leaked through dark web outlets, most often accessed via the Tor network. In this study's analysis, dark web statistics indicate that exposed accounts pose a risk to the organizations and individuals involved.

A vulnerability assessment (VA) is the process of identifying, evaluating, and prioritizing security vulnerabilities in systems, networks, or applications. This assessment helps organizations understand their vulnerabilities and how quickly they can respond. The assessment typically includes scanning systems, analyzing common vulnerabilities (CVEs), and assigning risk levels to each vulnerability. Severity is often based on Common Vulnerability Assessment System (CVSS) scores or security tool policies. According

(NIST SP 800-115) vulnerability assessment provides insight into exploitable security weaknesses, supporting risk management and remediation efforts.

In study (Walkowski et al., 2021), Prioritizing vulnerabilities is an essential part of data communications network security management. It helps organizations focus on addressing the most critical vulnerabilities in a timely manner to avoid financial losses and reputational damage. This is often a challenging task, but using the Common Vulnerability Rating System (CVSS) enables security analysts to assess vulnerabilities based on environmental context, improving decision-making. By adopting CVSS-based vulnerability management strategies, organizations can reduce their exposure to threats by ensuring that 90% of critical vulnerabilities are addressed within two weeks of discovery, underscoring the importance of rapid, metrics-based responses.

Chapter 3 Research Methodology

In this chapter, we present the methodology used to design, collect, and analyze data for assessing device security in a Zero Trust Architecture (ZTA) environment using machine learning. The goal is to develop a robust and interpretable model that can distinguish between secure and insecure (healthy and unhealthy) devices based on their observed features and configuration. The proposed approach follows a systematic path that begins with multi-source data collection and feature extraction, followed by risk-based feature engineering and labeling, dataset augmentation, feature selection, and finally, model training and evaluation. This path combines domain-led insights (such as using security expertise to identify risk factors) with data-driven processing to ensure data quality, accuracy, and generalizability. The comprehensive approach is designed for high predictive performance also for helping practical deployment in cybersecurity monitoring and ZTA policy enforcement, ensuring that only healthy and secure devices are granted network access.

This study adopts a quantitative approach to collect, analyze, and classify device security health data. Machine learning models are used to classify devices based on their security health status and measure the risk levels to predict potential security risks. This approach is designed to meet the study objectives and provides measurable data that is analyzed statistically, thus helping to identify patterns and correlations between various factors and features and categorize devices based on security health indicators such as antivirus status, patch level, vulnerabilities, CPU usage etc. This is in line with the study's goal of enhancing device-level security through a zero-trust architecture (ZTA) principle of "never trust, always verify" and predictive patterns, which is essential for proactive threat management.

To better illustrate the components of the methodology, Figure 1 shows the full pipeline of the proposed method. The pipeline begins with collecting raw data from multiple sources and extracting features, followed by risk scoring and data labeling based on a quantitative risk formula. Next data preprocessed and cleaned, then the data is expanded and balanced to address class imbalances. The final stages involve training machine learning models and evaluating their performance in classifying endpoint devices

as secure (“accept”) or insecure (“deny”). This pipeline aligns with the objectives of this study, ensuring that only devices meeting the security criteria are allowed to access the network.

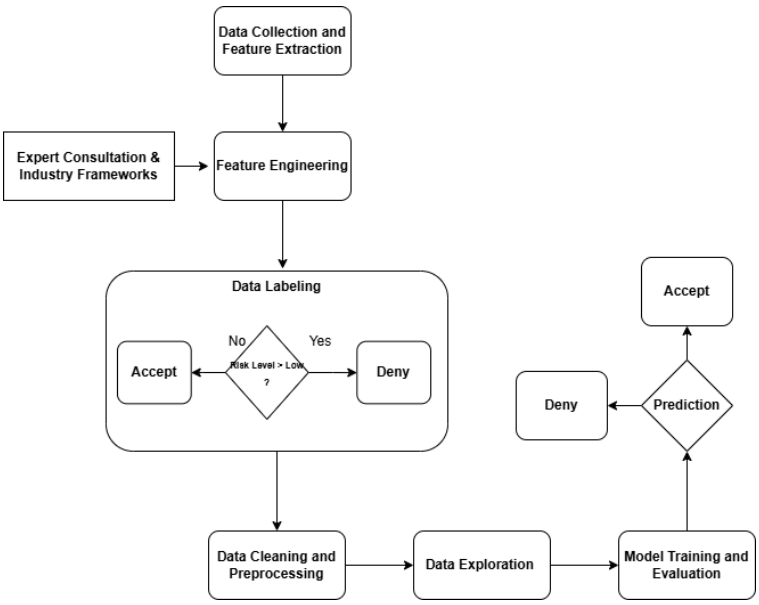


Figure 3.1 Overall Methodology Pipeline for Device Security Health Detection in ZTA

3.1 Data Collection and Feature Extraction

The methodology began by collecting quantitative data from multiple sources to capture a wide range of endpoint and actual security states. To build a reliable machine learning model for detecting the security status of network clients in a ZTA environment, we used a multilayered systematic data collection strategy. The goal was to obtain a comprehensive and representative dataset of endpoints and security statuses. Data were collected from various environments, including real enterprise endpoints, personal/friend computers, and controlled virtual machines (VMs) managed on hypervisors (Hyper-V and VMware) running Windows 8.1 Pro, Windows 10 Pro, and Windows 11 Pro. By including both production and simulated environments, we ensured diversity in device configurations and conditions.

Feature data were extracted using a combination of automated scripts and manual methods to collect measurable indicators reflecting the security of each device. The selected features represent distinct behavioral, methodological, and security configuration aspects of each endpoint. A standardized collection approach was used to ensure coverage

of all relevant security dimensions. Where possible, automated tools collected the data, in some cases, enterprise security tools and reports were used, and some metrics required manual retrieval. The following subsections describe the data sources and collection techniques used in detail.

3.1.1 Multi-source data collection strategy

Data were gathered from multiple sources and tools, informed by both industry framework materials and practical enterprise resources such as EC-council, Wazuh, threat intelligence platforms (TIP), Trend Micro, and security log systems. Based on this material, the next step was to identify endpoint features and device-level signals that are consistently emphasized across Security Operations Center (SOC) practices.

In line with the insights drawn from these sources, Security Operations Center (SOC) studies highlight several key endpoint indicators for the early detection of compromised devices or security vulnerabilities. These indicators typically include repeated failed login attempts, unusual external network connections detectable by commands such as `netstat -an`, a large number of open ports, long time system uptime, and numerous pending system updates. According to SOC and incident response references, these indicators are often associated with brute-force attacks (Johnson, 2025), lateral movement activity (United States Department of Defense et al., 2022), or unpatched systems that may create exploitable vulnerabilities (EC-Council, n.d.). Based on these widely documented guidelines, the feature set in this study is designed to incorporate these recognized indicators of compromise to ensure that the dataset captures security-related endpoint features supported by current SOC best practices.

While these endpoint indicators provide evidence of risk Security Information and Event Management (SIEM) platforms play a crucial role in identifying system vulnerabilities and assessing security risks. In this study, Wazuh, an open-source SIEM solution, was used as a reference technology due to its proven ability to detect vulnerabilities and correlate them with Common Vulnerability Assessment System (CVSS) metrics. According to Wazuh's official documentation, the platform automatically analyzes installed applications and running services, correlates them with known vulnerabilities, and determines the corresponding CVSS scores (Wazuh, n.d.). These standardized metrics form

a fundamental basis for measuring the severity and potential impact of detected vulnerabilities. Therefore, the vulnerability-related features in the dataset were aligned with the risk indicator types defined in SIEM practices, ensuring that the "vulnerability risk" component reflects established security assessment methodologies.

To enhance the measurement of external exposure risk, an External Threat Indicator was added based on threat intelligence feeds. Manual queries were performed on dark web breach credential databases, such as Have I Been Pwned, DeHashed, and IntelX. If the username/email associated with the device was found in known data breaches or dark web data dumps, this was recorded as an exposure feature (Dark Web Mention). This feature is an External Threat Indicator, where exposed credentials indicate the possibility that the user/device has been targeted or compromised via credential stuffing or other attacks (Bermudez et al., 2018).

The Trend Micro Vision One platform, a commercial endpoint security and XDR (Extended Detection and Response) platform, was used to gain insights into endpoints and vulnerabilities. This tool provided information about known software vulnerabilities present on each device by retrieving Common Vulnerabilities and Exposures (CVE) data and risk scores for installed applications and services. The number of detected vulnerabilities at different severity levels (low, medium, high, critical) was listed in our dataset as a feature (Trend Micro, n.d.). The Trend Micro Admin Console was also used to view indicators of device security status (such as antivirus and security software status) and served as a benchmark for important security status metrics (such as whether antivirus software was enabled, real time status, and patch compliance through update status). In cases where direct data export was not available, vulnerability and security status details were manually retrieved from Trend Micro dashboards and reports.

Automated Feature Extraction using PowerShell and Python: a customized tool has developed to collect a wide range of system and security metrics from devices:

1.Custom PowerShell Scripts: A comprehensive PowerShell scripting framework was deployed on endpoints to extract key security indicators and system status information,

exporting the results to a CSV file for each device and observation. This script combines several groups of features:

For group one, Device Identity, this group contains basic identifiers for the record. These were used to distinguish records and link to user context if needed, but were anonymized and not used as model features (to avoid any bias or privacy issues). The collected identifiers included the device name representing the name of the host machine, the user name referring to the currently logged user, and an email generated by combining the username with a fixed company domain.

For group two, System Performance, this group includes CPU Utilization, Memory Utilization, Disk Utilization, GPU Utilization, Network Utilization throughput (KBPS). Resource usage metrics indicate workload and performance, they were chosen because abnormal usage patterns often indicate a harmful activity, such as malware presence. High CPU or disk activity may reveal encryption procedures from ransomware or DLL injection (Thummapudi et al., 2023; Miter ATT&CK T1055, n.d.). In addition, sustained abnormal resource consumption is a red flag for compromise in both IoT and user endpoints (Jamshidi et al., 2025).

For group three, Network exposure, this group includes open ports count, the total of open TCP ports using listening state TCP get connections, and open UDP ports (since UDP is connectionless, we look for bindings using Get-NetUDPEndpoint), and ports detected using PowerShell network commands. In addition to open ports, specific ports recorded individually, such as FTP (port 21) is used for file transfers, RPC (port 135, dynamic) allows remote function calls between systems, Telnet (port 23) provides remote command line access, and RDP (port 3389) enables remote desktop access for Windows computers. These increase the attack surface or indicate policy violations. Furthermore, the external IPs capture the remote IP address connected to the device, filtered to exclude local subnets. The ‘Netstat –an’ command was used. The included features related to the network to detect lateral movement, command-and-control behavior, or unauthorized access (MITRE ATT&CK T1071, n.d.; Vigna & Kemmerer, 1998).

For Group four, Device Protection and Security Configurations, this group records features that reflect the security status and defense of each endpoint. The number of unsigned drivers identifies untrusted or potentially malicious kernel-mode drivers that could expose the system to low-level compromise. The User Account Control (UAC) level, collected from the Windows Registry, indicates protection against privilege escalation attacks, a value of 0 indicates that UAC is completely disabled. Additional protection indicators include whether antivirus software is enabled, whether real-time protection is enabled, and whether tamper protection is applied, all collected from system security settings. The group also includes uptime in days, used to determine whether a device requires a restart to apply a patch, and the number of pending updates, which shows patch compatibility. These configuration and protection features assess a device's ability to resist or detect attacks, as missing or misconfigured defenses are often exploited to elevate privileges or disable security tools (MITRE ATT&CK, T1068; Moe & Nerhagen, 2022). Pending updates and vulnerability indicators are integrated to represent the integrity of patches and exposure to known CVEs, in line with established vulnerability management practices (NIST SP 800-115, 2008; Walkowski et al., 2021).

For group five, access audit logs, this group focuses on authentication-related events that help identify unauthorized access attempts. The primary logged feature is the number of failed logins, extracted from the Windows Security Event Viewer logs, specifically event ID 4625. The frequency of this event may indicate brute-force attempts, password guessing, or credential stuffing attacks. All common techniques used by attackers to gain unauthorized access to user accounts or endpoints (Sitapura, 2022). Integrating this audit log feature enhances the dataset's ability to detect any anomalies associated with early access and aligns with established security monitoring practices.

2. Vulnerability Assessment Using Python: A Python script was used alongside PowerShell scripts to organize data collection across devices and to perform additional data collection tasks beyond the scope of PowerShell. One of the core Python modules was a custom vulnerability assessment script. This script enumerated installed applications, running services, and related software versions on each device and then queried the National Vulnerability Database (NVD) API for known vulnerabilities (CVEs) associated

with these software components. The script then counted the vulnerabilities by severity (e.g., the number of critical, high, medium, and low vulnerabilities on the system) using the NVD's CVSS version 4 assessment system (NIST, n.d.). From this, aggregated properties such as the total vulnerabilities and the number of vulnerabilities per severity were obtained. This customized approach was necessary because many commercial vulnerability scanning programs (such as Nessus and Rapid7) could not be used in our research due to licensing and environmental limitations. Using custom Python script for vulnerability scanning allowed for automation across multiple endpoints and ensured consistency in output format and severity metrics.

To ensure that the collected features align with real-world enterprise practices, there are several Commercial Solution Benchmarks reviewed tools such as Cisco ISE, Trend Micro Vision One, and WSUS, to determine key metrics and baseline criteria such as antivirus protection, tamper protection, and system update status (patch level). Additionally, Trend Micro vulnerability results were manually incorporated when available.

While many feature values were automatically collected via PowerShell scripts and Python script, others required manual extraction due to variations in the tools used and limitations of automated collection methods. Some metrics, such as disk usage in some cases, could not be reliably captured by automated scripts because the data export occurred on the disk itself, potentially inaccurate real-time measurements. To obtain accurate values, disk usage was manually monitored using system monitoring tools. Additional features, such as pending WSUS updates, Trend Micro vulnerability results, and the number of failed login distributions from active directory (AD) security logs, were also manually collected. These values were retrieved from organizational WSUS reports, Trend Micro Vision One dashboards, and domain controller logs, because they either lacked export APIs, used proprietary formats, or included administrative portals. These manual methods, while more time-consuming, ensured data completeness and allowed for the inclusion of critical security indicators that would otherwise have been missed, also increased the dataset sample from different data resources.

3.1.2 Dataset Diversity and Augmentation

To ensure the representativeness and robustness of the dataset, data were collected under both normal and compromised endpoint operating conditions. This approach introduced controlled variation into the dataset, which is crucial for training an effective model. Specifically, we simulated attack scenarios on some test machines (especially virtual machines) to generate data that accurately reflects the endpoint's behavior during a security incident. For example, we simulated brute-force login attacks on some machines, resulting in a high number of failed login attempts. We intentionally installed vulnerable software versions (such as an outdated WinRAR or an older Apache server) on some systems to increase their vulnerability count. We also intentionally configured certain faulty security settings (such as temporarily disabling User Account Control or antivirus software) on some machines, ran compression programs (to increase CPU, memory, or network usage) to simulate malware activity, and opened insecure ports such as telnet, services like RDP, and added an FTP site on Internet Information Service (IIS). All of these actions were performed in isolated environments to avoid any unintended harm. These steps ensured that our data covered both healthy (well-configured and uncompromised devices) and harmful (compromised or misconfigured devices) states, providing a diverse sample for the model. By observing devices in a compromised state, the model can learn to recognize patterns associated with security breaches, fulfilling a key research objective of dynamically distinguishing between healthy and unhealthy devices.

After collecting and labeling the initial data (as described in the next sections), we observed an imbalance in the class distribution in the dataset, with a greater number of "Accept" (secure) devices than "Deny" (insecure) devices. Specifically, of the 1181 observations collected, approximately 59% were labeled "accept" and 41% "deny" (since a much larger number of observations were in a low-risk state compared to a high-risk state). This class imbalance is examined in this study, as it can affect a machine learning model’s performance by bias toward the majority class and ignore important patterns from the minority class. To mitigate this imbalance and enhance the model's learning ability, we significantly augmented the dataset with synthetic data. Instead of using traditional oversampling methods (such as randomly repeating minority examples or applying SMOTE interpolation), a GPT-based data generation approach was employed to realistically expand

the dataset. Using OpenAI's pre-trained generator transform (GPT), additional device records were generated for both classes, focusing on creating more diverse "deny" states (which were underrepresented) as well as new "accept" states to achieve a large and balanced dataset. The GPT model was guided by existing dataset distribution rules and cybersecurity domain rules to produce reasonable sets of feature values that mimic real-world device. This approach allowed us to introduce new variations not present in the original dataset.

Through this synthetic augmentation, the dataset was expanded to a total of 6,000 records, consisting of 3,000 accept and 3,000 deny cases (a 50:50 balance). Each synthetic record maintained logical consistency between features (e.g., an excessively large number of vulnerabilities would be paired with a high risk label, a device with its antivirus software disabled might also have other weaknesses, etc.), as ensured by the generative model and prompt constraints. The augmented data improved the model's tolerance to variance and prevents overfitting of the limited original samples. By generating entirely new samples, we aimed to maintain realistic relationships between features, which is difficult to achieve using basic oversampling. All generated records were carefully reviewed for plausibility before inclusion. This step addressed the dataset availability challenge identified in Chapter One by providing a sufficiently large and balanced dataset for training machine learning models.

Prompt for Synthetic Data Generation

You are a data scientist tasked with expanding the size of a tabular dataset used for evaluating the security health of endpoint devices within a Zero Trust Architecture (ZTA). The original dataset contains 1181 records. Your objective is to study its structure and statistical distributions, then generate synthetic data that preserves the original dataset's schema, logical relationships, and statistical characteristics.

1. Dataset Context

The dataset represents the security posture of endpoint devices in a ZTA environment. It includes the following primary feature categories:

A. Identity Features: Device Name, Username, Email, Operating System

B. Resource Utilization Features: CPU Utilization, GPU Utilization, Disk Utilization, Memory Utilization, Network Utilization

C. Specific Security Ports & Related Features: FTP Status, Telnet Status, RPC Status, RDP Status

D. Protection & Security Features: Antivirus Enabled, Real Time Protection, Tamper Protection, Active Sessions, Failed Login Attempts, Unsigned Drivers, UAC Level

E. Network Features: External IP Count, Open Ports, Dark Web Mention

F. System Features: Installed Apps Count, Running Processes Count, Boot Time, Uptime Days, Pending Updates, Admin User Count, Guest Account Enabled, Admin Account Enabled

G. Vulnerability Features: Vulnerabilities Critical, Vulnerabilities High, Vulnerabilities Medium, Vulnerabilities Low, Vulnerabilities Total

2. Derived Features

The dataset also includes derived attributes that contribute to risk classification: Specific Security Ports, Vulnerability Severity, Likelihood (L), Threat (T), Risk Score, Risk Level, Final Label (Accept/Deny)

Each derived feature must be computed according to the formulas defined below.

2.1 Specific Security Ports: Counts the number of open security-related ports (FTP, Telnet, RPC, RDP):if 1 open port → 1, if 2 open ports → 2, if 3 or 4 open ports → 3

2.2 Vulnerability Severity: Determined by the highest present vulnerability level: if Critical > 0 → 3, Else if High > 0 → 2, Else → 1

2.3 Likelihood (L): Calculated as the rounded-up average of multiple weighted components (e.g., open ports, unsigned drivers, pending updates, uptime, UAC level, protections enabled, vulnerability severity).

2.4 Threat (T): Defined as the maximum value derived from thresholds applied to: Failed Login Attempts, Active Sessions, Dark Web Mention, CPU/Memory/Disk/Network utilization

2.5 Risk Score and Risk Level:

Risk Score = Likelihood × Threat × 3

Risk Level: if risk score ≤ 6 → Low, Else if ≤ 14 → Medium, Else if > 14 → High

2.6 Final Label:

Label = “Accept” if Risk Level = “Low”; otherwise “Deny”.

3. Synthetic Data Requirements

After analyzing the original dataset, generate new synthetic records that:

- Maintain the same schema and column types.
- Preserve the statistical distributions of numerical and categorical features.
- Maintain logical dependencies and respect all formulas.
- Reflect realistic value ranges, such as: Failed Login Attempts: 0 to large integer values, Unsigned, Drivers: 0–100, Open Ports: 40–400, Active Sessions: 1–10, Total Vulnerabilities: 1–350 (distributed logically across severity levels)

4. Output Requirements

The generated synthetic dataset should: Be larger than the original dataset, Maintain structural and statistical fidelity. Be internally consistent based on the risk model. Be provided in tabular form (CSV or similar).

This prompt ensures that synthetic data generation aligns with the structure, logic, and risk scoring methodology used in the original dataset, enabling realistic expansion for experimentation, modeling, or evaluation purposes.

3.1.3 Privacy Considerations

To maintain confidentiality, personally identifiable information (PII) such as usernames, device names, and email addresses were anonymized or masked during dataset storage and model training. That only abstracted features. This ensures that the methodology adheres to privacy best practices and that the resulting model can be used or shared without exposing sensitive organizational information.

Table 3.1 below summarizes the key features extracted for model training, categorized by functional category. Each feature is accompanied by a brief description of its importance to endpoint security and a supporting reference from academic or technical sources.

Table 3.1 Feature Mapping

Feature Name	Category	Purpose / Description	Reference
CPU Utilization	System performance	Detects high usage caused by malware or encryption	Thummapudi et al., 2023

		routines by ransomware attack.	
Memory Utilization	System performance	High memory usage can indicate reflective DLL injection	MITRE ATT&CK – T1055.015
Disk Utilization	System performance	Disk spikes may relate to ransomware behavior	Thummapudi et al., 2023
Network Utilization	System performance	Detects abnormal traffic caused by C2 or data exfiltration	Bermudez et al., 2018
External IPs	Network exposure	Connections to public IPs may indicate C2 traffic	MITRE ATT&CK – T1071
Open Ports Count	Network exposure	High port counts increase attack surface	Ullah, 2016
FTP, Telnet, RPC, RDP	Network exposure	Commonly targeted services for lateral movement	Mohta et al., 2024; Allman, 1999
Failed Login Count	Access Protection	Indicator of brute-force or credential attacks	Sitapura, 2022
Unsigned Drivers Count	Device Protection	Unverified drivers may enable privilege escalation	MITRE ATT&CK – T1068
Antivirus Enabled	Security configurations	Detect if the system has AV protection	Moe & Nerhagen, 2022
Real Time Protection	Security configurations	Measures live scanning capability	Moe & Nerhagen, 2022
Tamper Protection	Security configurations	Ensures AV settings aren't bypassed	Moe & Nerhagen, 2022
UAC Level	Device Protection	Measures privilege enforcement level	Moe & Nerhagen, 2022
Pending Updates Count	Device Protection	Reflects patch compliance	Morris et al., 2020
Uptime Days	Device Behavior	Long uptime can suggest delay in patching	Moe & Nerhagen, 2022
Vulnerabilities Low–Critical	Vulnerabilities	Quantifies known risks based on CVSS severity	NIST SP 800-115
Dark Web Mention	External Threat	Indicates leaked credentials on dark web	Bermudez et al., 2018

After collecting raw data representing device security indicators, the next step was to transform this raw data into meaningful risk metrics. This was achieved through a structured feature engineering process, where helper features such as likelihood, threats, and risk score were extracted to support interpretability and data labeling.

3.2 Feature Engineering

After extracting the raw features, a set of derived features was designed to help construct and classify the final dataset. These designed several features derived to enter training, features help measure the overall risk exposure and were used to determine the risk rank of each device to support analysis and labeling.

3.2.1 Features support the model training

Firstly, uptime in days was calculated based on device boot time. Secondly, the total vulnerabilities were computed by summing all detected vulnerabilities (Low, Medium, High, and Critical). Thirdly, Vulnerability severity, if a device has only low/medium-severity vulnerabilities and no high/critical vulnerabilities, it is considered low severity (1), if it has any high-severity vulnerabilities (but not critical), it is considered medium severity (2), and if it has any critical vulnerabilities, it is considered high severity (3). This reflects the fact that critical, unpatched vulnerabilities increase the likelihood of a device being compromised. Finally, specific security ports, the number of high-risk services running (FTP, Telnet, RPC, and RDP) were considered. If none of these service ports were open, or only one was open, this indicates low exposure so 1, if two ports were open, this indicates medium exposure so 2, if three or four categories were open, this indicates high exposure so 3. (This counts service categories, not just raw port numbers, to emphasize the diversity of risky services.)

3.2.2 Derived Risk Components:

we adopted a simplified risk formula inspired by standard risk assessment frameworks such as NIST and FAIR. According to NIST SP 800-30 (Revision 1, 2012), “risk is a function of the likelihood of a threat event’s occurrence and potential adverse impact should the event occur.”. so calculated Risk =Impact × Likelihood.

According to the FAIR (Factor Analysis of Information Risk) model, risk is defined as the likely frequency and likely magnitude of future loss, i.e., risk = loss event frequency × loss magnitude. Where Loss Event Frequency (LEF) is the number of times a loss event is expected to occur, LEF = Threat Event Frequency × Vulnerability, and Loss Magnitude (LM) = the expected size/impact of loss if the event occurs, the formula will be Risk = (Threat Event Frequency × Vulnerability) × Loss Magnitude (FAIR, 2013). In this study, we adapted this formula into a simplified form expressed as follows:

$$\text{Risk Score} = L \times T \times I \text{ with } L, T, I \in \{1,2,3\}$$

While L (Likelihood), represents exposure/weakness/vulnerabilities at access time. T (Threat) represents active indicators of compromise (IOCs). I (Impact) represents business harm (use I=3 now, all endpoints high priority). The scoring scale follows three levels, Low=1, Medium=2, High=3.

To ensure the validity and practical relevance of the assessment methodology, consulted six cybersecurity experts. Their insights were used to review the designed features, refine the risk score formula and the sub-rules applied to likelihood and threat scores, and ensure the proposed model aligned with operational realities in modern security environments. This feedback strengthened the assessment framework's reliability and enhanced its applicability to real-world endpoint risk assessment.

1. Calculating Likelihood (L): Several “posture exposure” drivers were identified that contribute to how vulnerable or exposed a device is. These include factors like the number of open ports and services, the state of security controls, patch levels, and vulnerabilities present. Each contributing feature was evaluated against a threshold to assign it a sub-score of 1 (Low risk), 2 (Medium), or 3 (High). For example:

For open services/ports, the analysis considered the number of high-risk services running (FTP, Telnet, RPC, RDP). If none of these ports were open, or only one was open, it was classified as low exposure; if two were open, it was classified as medium exposure; if three or more were open, it was classified as high exposure. (This is calculated within service categories, not just the raw port numbers, to emphasize the diversity of high-risk services.)

In addition, the number of open ports (All Ports), If a device has, for example, fewer than 100 open ports, it was classified as low exposure; if it has 100-125 open ports, it was classified as medium exposure; if it has more than 125 open ports, it was classified as high exposure. (A very large number of open ports indicates a large attack surface.)

Regarding pending updates, If the device has fewer than 4 pending updates, we consider it to be within the low exposure level (relatively up-to-date), if the device has 4-10 pending updates, this is classified as a medium exposure level, If a device has 11 or more pending updates, this is classified as high (significant delays in patches, indicating a high probability of unpatched vulnerabilities)

For uptime in days, if a device has been running (without restarting) for an extended period, it may not have received updates that require a restart. The analysis established thresholds where uptime of less than approximately 32 days is low, around 33-55 is medium, and more than approximately 55 days is high risk (indicating that the device may be behind schedule for a maintenance cycle).

Initial threshold values were determined in consultation with cybersecurity experts and then validated through empirical analysis of the original dataset. The distributions of Pending Updates Count, Open Ports Count, and Uptime Days were examined using descriptive statistics and percentage analysis, demonstrating a strong correlation between expert predictions and observed data behavior. Specifically, the majority of observations were concentrated below the 75th percentile, representing typical operational cases, while values between the 75th and 95th percentiles reflected elevated but still common conditions and were therefore classified within the medium range. Values at the top of the distributions, approximately above the 95th percentile, were rare and associated with abnormal operational behavior, thus, they were classified within the high range. This integrated approach, based on both expertise and data, ensures that the chosen thresholds are both security-relevant and statistically supported. The Table below show dataset distributions on these three features percentile and basis stats.

Table 3.2 percentile and basic stats for (open ports, pending updates and uptime in days)

Feature	Count	Mean	Std	Min	1%	5%	10%	25%	50%	75%	90%	95%	99%	Max
Open Ports Count	1181	82.5	37.7	35.0	36.0	36.0	36.0	51.0	81.0	111.0	118.0	124.0	192.2	362.0
Pending Updates Count	1181	2.7	3.4	0.0	0.0	0.0	0.0	1.0	2.0	3.0	4.0	10.0	16.2	26.0
Uptime Days	1181	14.5	18.4	0.0	0.0	0.0	0.0	0.0	4.0	32.0	33.0	55.0	63.0	113.0

For example, the figure 3.2 below shows the distribution of uptime days, that most devices have less than 32 uptime days, indicating a low level of risk exposure and reflecting recent reboots or maintenance. Devices with uptime days between 33 and 55 days, corresponding to approximately the 75th-95th percentiles, exhibit moderate exposure associated with longer uptime periods. Conversely, devices with uptime days exceeding 55 days, located at the top of the distribution (≥ the 95th percentile), represent a high level of exposure due to their extended operating periods and the increased likelihood of delayed updates or maintenance.

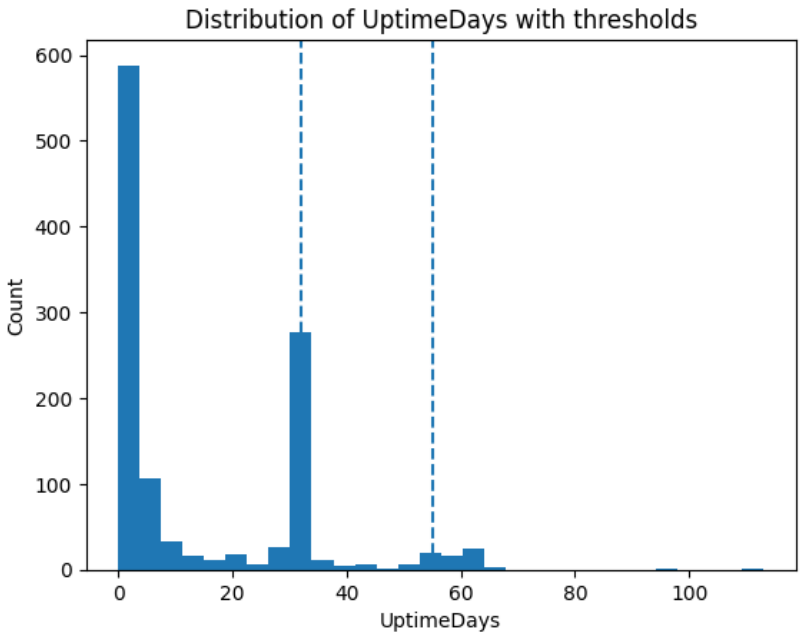


Figure 3.2 Histogram of Uptime Days with exposure threshold boundaries

Another factor considered was the security settings, features such as User Account Control level (UAC Level) and antivirus status were directly correlated, for example, enabling UAC Level (Level = 1) was low risk, while disabling UAC Level (0) was high risk for this factor. Similarly, enabling antivirus/real-time protection/tamper protection is good (low risk for these factors), while disabling any of these protections is high risk. To increase the risk.

Finally, presence of vulnerabilities, the analysis condensed the vulnerability information into an index, if a device had only low/medium- severity vulnerabilities and no high/critical vulnerabilities, it was considered low (1), if it had any high-score vulnerabilities (but no critical ones), it was considered medium (2), if it had any critical vulnerabilities, it was considered high (3). This reflects that unpatched critical issues make a device more vulnerable to compromise.

Once each driver was scored, this study combined them to derive an overall likelihood score (L) for the device. An approximate averaging approach was used; these sub-scores were averaged (with some drivers weighted equally) and then rounded to the nearest integer from 1 to 3. This provided a single L value that captures the overall level of exposure for the device. For example, a device with many open ports, missing updates, and disabled protection will end up with a high risk of compromise. L = 3 (high), while a fully updated device with minimal services will get L = 1 (low).

Rate each driver/features using the thresholds in Table 3.3 below, then calculate L = approximation (the average of all sub-scores).

Table 3.3 Thresholds for System Security Drivers and Approximation Level Likelihood (L) Calculation

Driver	Low = 1	Medium = 2	High = 3	Notes
Specific security ports (FTP, Telnet, RDP, count how many of these 4 categories are open	exactly 1	exactly 2	3 or 4	Count categories, not raw port total
Unsigned Drivers	0	N/A	≥ 1	1 or more unsigned = High
Open Ports Count (all ports)	< 100	100–125	> 125	Attack surface
Pending Updates Count	< 3	4–10	≥ 11	Patch level
Uptime Days	< 32	33–55	> 55	Long uptime can imply patch lag
UAC Level	1	N/A	0	1=on (good) → Low; 0=off (bad) → High
Antivirus Enabled	true	N/A.	false	Boolean to risk
Real Time Protection	true	N/A.	false	
Tamper Protection Enabled	true	N/A.	false	
Vulnerability severity	only Low/Medium → 1	any High (no Critical) → 2	any Critical → 3	Vulnerability Index

2. Calculating Threat (T): For the threat component, the study focused on real-time indicators of system compromise. The analysis examined features that might indicate an ongoing attack or breach. The key threat indicators considered were: a high number of failed logins, multiple active user sessions (which might indicate suspicious system use), dark web exposure, and abnormal resource usage (CPU, memory, hard drive, and network spikes). A sub-score of 1/2/3 was assigned to each based on its severity. For example:

For failed login attempts, if the number of recent failed login attempts for the device was very low (<15), the score was low, medium (15-50), or very high (>50), the score indicated a high threat level (possibly a brute-force attack in progress).

Regarding active sessions, a user endpoint normally has one active session; two sessions may be medium, three or more concurrent sessions may indicate a high threat level (possibly unauthorized users or malware opening remote sessions).

For dark web mention, If the device user's credentials or email were found in a security breach (Dark Web Mention = True), this was treated as a high threat factor (Score 3) because it likely means the account is known to attackers (False = Low, due to no known exposure).

Regarding resource usage, high limits were defined for CPU, memory, hard drive, and network usage. Exceeding these limits indicates a potential sustained attack (e.g., CPU usage above 70% for an extended period is unusual, and is assigned a high score of 3, a medium score of 2 between 35% and 70%, and a low score of 1 for the CPU sub-score). Similar limits were applied to memory, hard drive, and network throughput (e.g., network usage above a certain KB/s threshold indicates a high probability of data leakage) as shown in the table below.

Initial threshold values were determined in consultation with cybersecurity experts and then validated through empirical analysis of the original dataset. The distributions of system utilizations were examined using descriptive statistics and percentage analysis, demonstrating a strong correlation between expert predictions and observed data behavior. Specifically, the majority of observations were concentrated below the 75th percentile, representing typical operational cases, while values between the 75th and 95th percentiles reflected elevated but still common conditions and were therefore classified within the medium range. Values at the top of the distributions, approximately above the 95th percentile, were rare and associated with abnormal operational behavior, thus, they were classified within the high range.

For the figure 3.3 below, the histogram of Memory Utilization indicates that most devices operate within the 50%–70% range, representing low exposure and normal memory usage. Values falling between the dashed thresholds reflect moderately elevated utilization between approximately 70% and 90%, corresponding to medium exposure caused by increased memory pressure. The sparse upper tail beyond the higher threshold represents high exposure, where memory utilization approaches saturation and may negatively impact system stability or security mechanisms.

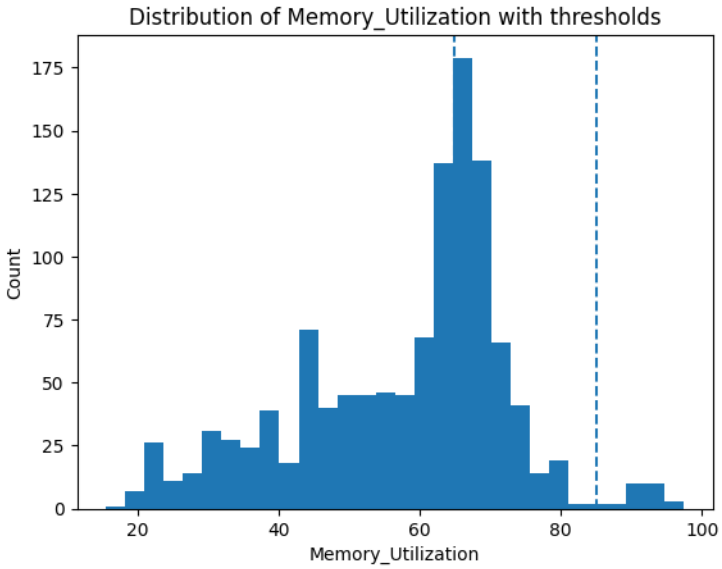


Figure 3.3 Histogram of Memory Utilization with exposure threshold boundaries

Considered that any single strong indicator might be sufficient to indicate a breach. Therefore, Consequently, sub-scores were calculated for each IOC and then the highest value among them was taken as the overall threat score (T). In other words, if any of the threat indicators were rated high (3), then set T = 3 for that device, since a single critical sign of an attack was sufficient to treat the situation as a high threat. If none were high but one or more were medium, T = 2, only if all indicators were low (no significant signs of a problem) would T = 1. For example, if a device was moderately using CPU but its outgoing network traffic was very high, we would set T = 3 (due to the high network anomaly). This "maximum score" approach aligns with a conservative security philosophy, assumes the worst threat level if a critical indicator exists.

Calculate the sub-score for each IOC using the thresholds in the table 3.4 below, then set T = max (Failed Logins, Active Sessions, Dark Web Mention, CPU, Mem, Disk, Net). That means, if only one usage metric is high (3), set T to 3.

Table 3.4 IOC Thresholds and Sub-Score Mapping for Threat Level (T) Determination

IOC → sub-score (1/2/3)	**Low = 1**	**Medium = 2**	**High = 3**
Failed Login Count	< 15	15–50	> 51
Active Sessions	1	2	≥ 3
Dark Web Mention	false	N/A.	true
CPU Utilization %	< 35%	35–70%	≥ 70%
Memory Utilization %	< 70%	70–90%	≥ 90%
Disk Utilization %	< 70%	70–75%	≥ 75%

Network Utilization (KBPS)	< 35 KBPS	35–600 KBPS	≥ 600 KBPS

3. Impact (I): Current policy, all endpoints have the same priority, so priority is high which means all endpoints have the same consequences on the network if they are composed, set I = 3.

Under the current security policy, all endpoints are given the same business priority; therefore, the impact level (I) is set at high (I = 3) for all devices. This decision is justified by the standardization of access privileges granted to endpoints, the similar sensitivity of the data they process, and the risk of any compromised endpoint being used as an entry point for lateral movement, privilege escalation, or network-wide attacks. Because modern threats often exploit endpoints as initial entry points rather than final targets, the business impact is determined based on the potential consequences for the entire network, not on the specific role of the compromised device. This approach also aligns with the principles of "zero trust" and ensures consistency and objectivity in the risk assessment model in the absence of a formal asset-based impact classification.

3.2.3 Final Risk Score and Risk Level

All of the above components are combined into a risk score that determines the overall risk score of the device. With L, T, and I determined, then calculated the overall Risk Score for each device is calculated as:

Risk Score = L × T × 3, Range 3-27, so Ranges (example): Low 3–6, Medium 7–14, High 15–27.

Since I = 3 for all, the actual risk score = 3 * L * T, which ranges from a minimum of 3 (if L = 1 and T = 1, i.e., low likelihood * low threat) to a maximum of 27 (if L = 3 and T = 3, i.e., high likelihood * high threat). The risk levels are defined on this 3-27 scale as follows (for reference and interpretability): Low risk: 3-6 (This corresponds to cases where L or T is low and the other is at most medium; these devices have a strong security posture and no active threats have been detected). Medium risk: 7-14 (These typically involve cases where at least one L or T is medium or high, but not both are at the highest level; this represents a moderate concern). High risk: 15-27 (These indicate both high likelihood and high threat, essentially worst-case scenario devices, or any scenario that pushes the risk product into the higher range).

To define risk levels as shown in the table 3.5 below risk map. Assuming Impact (I) equals 3, all endpoints have the same priority.

Table 3.5 Risk Map to determine risk level

L \ T	1	2	3
1	3 (Low)	6 (Low)	9 (Low/Med)
2	6 (Low)	12 (Med)	18 (High)
3	9 (Low/Med)	18 (High)	27 (High)

These derived features were essential not only for labeling but also for interpreting model decisions and understanding the real-world effects of device health metrics. This risk formulation provides a concise measure that integrates various aspects of device security. It aligns with established risk assessment frameworks, essentially the concept of "Risk = Likelihood × Impact" (NIST) with an additional emphasis on immediate threats (from the FAIR threat event frequency concept). The use of discrete levels (1, 2, 3) ensures and maintains interpretability and ease of calculation. The risk score serves a dual purpose in methodology: (a) as an interpretable indicator for security teams to understand the device's status (the reason for classifying a device as insecure can be broken down into L and T factors), and (b) as a basis for labeling data in the Supervised Machine Learning.

3.3 Data Labeling

In order to train a supervised machine learning model, each device instance required a class label indicating whether it should be allowed or denied network access under a zero-trust policy. Consistent with the ZTA assume breach principle, devices are not trusted by default and must continuously demonstrate an acceptable security posture to gain access. Accordingly, this method uses "accept" for a device considered sufficiently secure to be granted access, and "deny" for a device considered too risky to be allowed network access. These ratings were determined based on a calculated risk score and risk level for each device.

There are two Classes (labels)

• Accept: means devices grant access to the network with a low risk level and a risk score is 3-6.

• Deny: means devices reject access to the network with a medium/high risk level and a risk score above 6.

Using the risk model described above, a threshold rating policy for labeling was established: Devices with a low risk score (between 3 and 6) were labeled "Accept." These devices have a strong level of security and do not pose any serious threats, indicating that they meet the trust criteria in the context of ZTA. Devices with risk scores above 6 were labeled "Deny." In this study, that means any device falling within the medium or high risk category (score ≥ 7) is treated as insecure and unacceptable for access.

This labeling scheme was applied to all 1181 original records, and then to the synthetically generated records (the synthetic data was generated in a way that inherently respected these risk-label rules). The result was a binary category classification, used as the target for machine learning models, 0/1 or accept/deny. Initially, applying this rule to the original dataset resulted in approximately 687 Accept versus 494 Deny (as mentioned earlier, roughly 60/40). After increasing the number of records to 6,000 using GPT-based (where that included equal classes), we obtained 3,000 records labeled "accept" and 3,000 records labeled "deny," which were used to train the model.

3.4 Problem Formulation

Although quantitative risk score and risk level assessment provide a structured and interpretable method for evaluating device security, they rely on fixed thresholds and predefined rules that cannot fully capture the complex and nonlinear relationships between security features. These assessment mechanisms assume stable conditions and independent indicators, which simplifies decision-making but limits flexibility. In practice, several weak signals combined may indicate high risk even if each individual indicator remains below its threshold, a scenario that fixed risk assessment may fail to detect.

In real-world "zero trust" environments, device behavior is shaped by a range of indicators rather than individual metrics, and security conditions are constantly evolving due to software updates, configuration changes, and emerging attack techniques. Therefore, machine learning is used to extract these subtle and dynamic patterns directly from the data. By modeling feature interactions and adapting to observed behavior, machine learning-based classifiers provide decision-making boundaries that are more responsive to real-world operating conditions than static, rule-based systems.

Another important limitation of risk score methods alone is their reliance on the complete and reliable availability of all features. In operational environments, some security

measures may be unavailable due to agent failures, limited access privileges, varying endpoint configurations, or temporary data collection interruptions. When such gaps occur, accurately calculating risks becomes difficult or unreliable. However, machine learning models can remain effective even with partial feature availability, as they are trained to infer decisions from available information and can tolerate missing or degraded inputs after appropriate pre-processing.

Finally, integrating machine learning with risk-based classification enables the scalability and long-term sustainability of the "zero trust" principle. While the risk index provides domain-based interpretation and consistent classification aligned with established frameworks, machine learning enables data-driven, adaptive decisions that improve over time as new feedback becomes available. This hybrid approach ensures that access control decisions remain interpretable, flexible, and scalable, supporting the "zero trust" principle "never trust, always verify" in complex and evolving enterprise environments.

Using (GPT) models to augment synthetic data raises several challenges that must be carefully considered, including reproducibility, the plausibility of the generated samples, and the potential for a high duplication rate. To mitigate these concerns, strict prompt rules were implemented to restrict the generation process within realistic and domain-consistent limits, ensuring that the synthetic records adhere to defined plausibility criteria. Reproducibility was further enhanced by fixing generation parameters and documenting model settings. Additionally, statistical validation was applied to compare the distribution of the synthetic data with the original dataset. To address duplication, a dedicated pre-processing and cleaning step was introduced, identifying and removing identical or duplicated records before model training. These measures ensure that the augmentation of synthetic data enhances data diversity without inflating the dataset or introducing bias, thus maintaining the integrity and reliability of the learning process.

3.5 Data Cleaning and Preprocessing

To ensure the integrity of the dataset and make it suitable for training a machine learning model, a comprehensive data cleaning and preprocessing workflow was implemented. This step is critical to ensuring the consistency, reliability, and interpretability of the input characteristics.

3.5.1 Data Cleaning:

A comprehensive check was conducted to identify and resolve any data quality issues. All features were checked for missing values and null. The missing or unavailable values in numerical features were handled using mean imputation, where each missing value was replaced by the mean of its corresponding feature calculated from the available data. For example, in the Pending Updates Count feature, 76 records contained not available values. The reason may be endpoint misconfiguration, or transient telemetry collection failures. These values were first converted to missing values (NaN). The mean of the feature was then calculated using only the valid (non-NaN) observations. Finally, the missing values were replaced with the mean value, ensuring that the overall distribution of the feature was maintained while avoiding data loss. The dataset was confirmed to be complete, with no missing values detected. Each column was also checked for data type consistency to ensure it contained the appropriate data type (e.g., numeric, logical). All feature types were verified and found to be valid for processing.

Duplicate rows were then checked. While there were no duplicates in the original dataset but the synthetic dataset contained 2260 duplicate rows, which were removed to maintain the uniqueness of each device observation. These checks ensured that the dataset was accurate, consistent, and reliable before any conversions.

3.5.2 Data Preprocessing:

To prepare the dataset for training classic machine learning models such as logistic regression (LR), k-nearest neighbors (KNN), and support vector classifier (SVC), several preprocessing techniques were applied. Numerical features were scaled to a standardized range using Min-Max Normalization or Standard Scaling, depending on the model requirements. This ensured that features contributed proportionately to model training and prevented features with larger numeric ranges from dominating. The target label class (Accept or Deny) was encoded and converted to a binary format using label encoding: Accept $\rightarrow$ 0, Deny $\rightarrow$ 1. Boolean features such as real-time protection, antivirus enabled, external threat, and specific open ports were also encoded (True $\rightarrow$ 1, False $\rightarrow$ 0). Although the dataset consists primarily of numeric features, any textual identifiers (e.g., device names, usernames, Email, and operating system used for logging purposes) have been removed or masked to avoid introducing noise or bias into the models.

Feature refinement involved dropping columns with the same value across all rows and low-variance columns, such as GPU Utilization, Admin Account Enabled, and Gust Account Enabled. Timestamp-related columns (e.g., Boot Time) were also dropped, while the derived uptime in days column was retained because the duration is more important and makes the model more relevant and informative for analysis. Additionally, the risk component, such as Risk Score, Risk Level, likelihood and threat features excluded from the training because they were used only as helper fields for determining the class labels and are no longer needed for model training. Also, based on avoiding Data Leakage, to exclude any derived features that were directly related to the label.

3.6 Data Exploration

To better understand the structure and quality of the collected dataset, an in-depth exploratory data analysis (EDA) was conducted. The original dataset consists of 1181 records and observations (rows) and 44 features (columns) representing the security health features of endpoint devices. The goal of this analysis was to verify the consistency, accuracy, and distribution of the data before proceeding with model training.

Class Distribution: The dataset was classified based on the calculated security risk, with records divided into two classes: Accept and Deny. Initially, there was an imbalance between classes. To ensure fair representation and improve the model's generalization, synthetic data was introduced to the minority class. Deny: 494 records (41.8%), Accept: 687 records (58.2%). The final class distribution is shown in Figure 3.4 below.

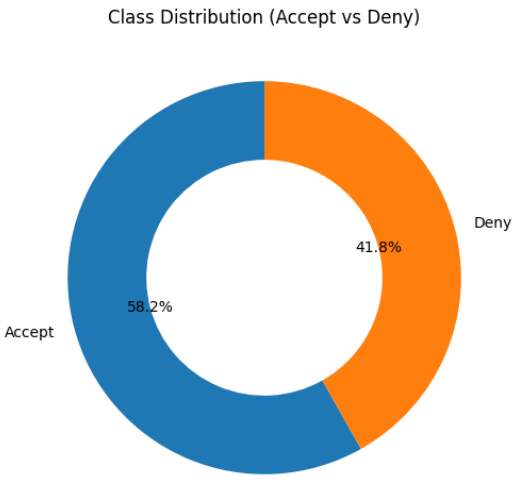


Figure 3.4 Class Distribution (Accept vs Deny)

To assess how the distribution of security-related features evolved across different dataset construction phases, five comparative box plots were created. Each plot represents the values of a key feature in the original dataset and the four expanded datasets resulting from incremental synthetic augmentation. These visualizations allow for a direct comparison of how feature ranges, median values, variability, and the presence of outliers change as the dataset size increases. The five selected features:

The first graph illustrates the distribution of CPU utilization across different dataset phases, including the original dataset and progressively expanded synthetic datasets. As synthetic records are added, the median and interquartile range gradually increase, indicating a broader representation of CPU utilization patterns. The presence of consistent upper extreme values across all stages suggests the maintenance of peak utilization values, supporting the realism and diversity of the expanded datasets. As shown in figure 3.5 below.

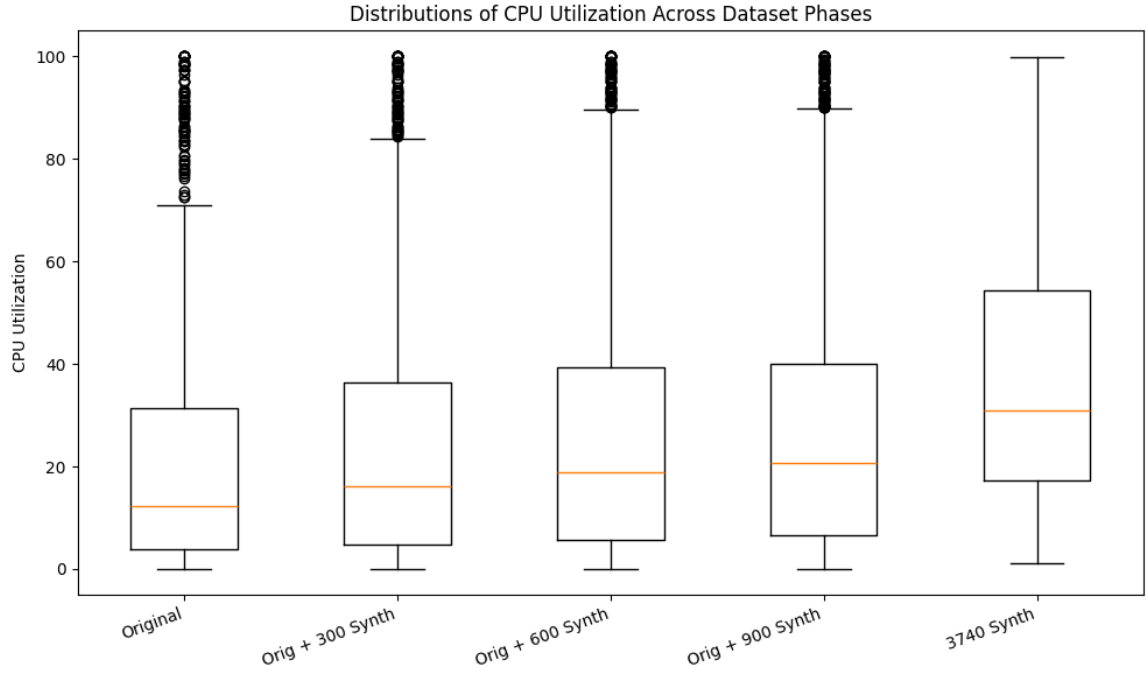


Figure 3.5 Distributions of CPU utilization across dataset phases

The second graph shows that unsigned drivers are almost absent in the original dataset, with values concentrated at zero. As synthetic data is added, the distribution widens sharply, introducing higher counts and many outliers. The full synthetic dataset shows the greatest variability, reflecting and more diverse.

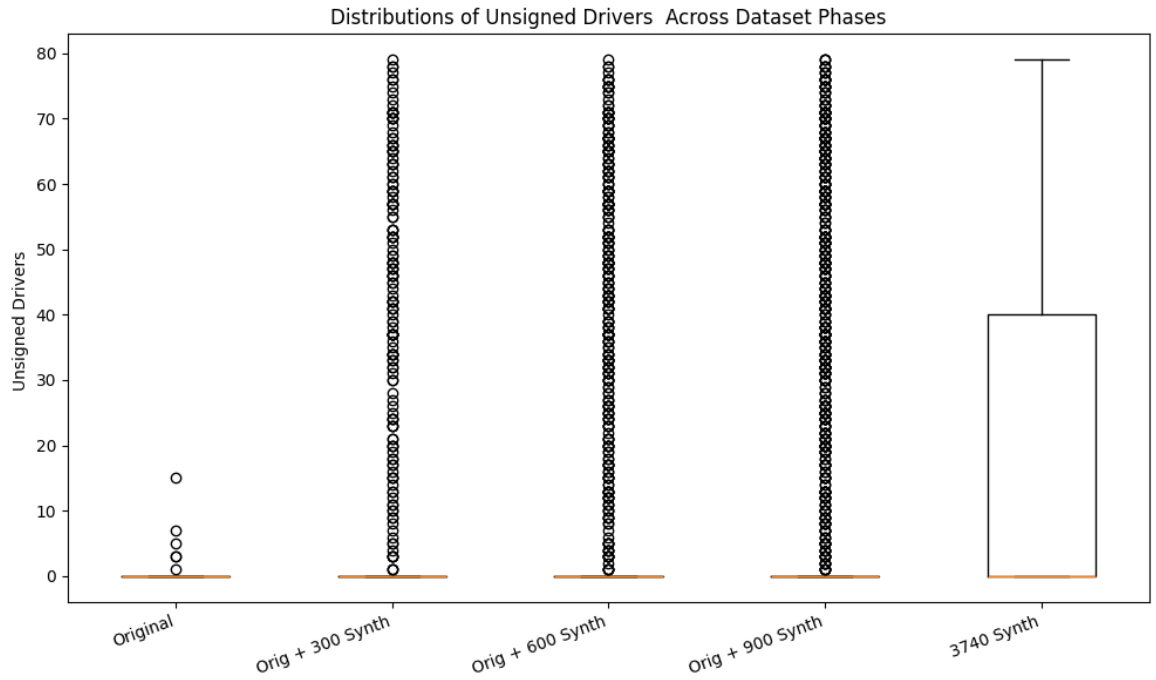


Figure 3.6 Distributions of Unsigned Drivers across dataset phases

The third graph shows that active sessions remain almost constant at one session in the original and early synthetic datasets. As more synthetic data is added, the distribution broadens, showing devices with two to seven concurrent sessions. The full synthetic dataset displays the widest distribution, reflecting more varied and potentially suspicious session activity.

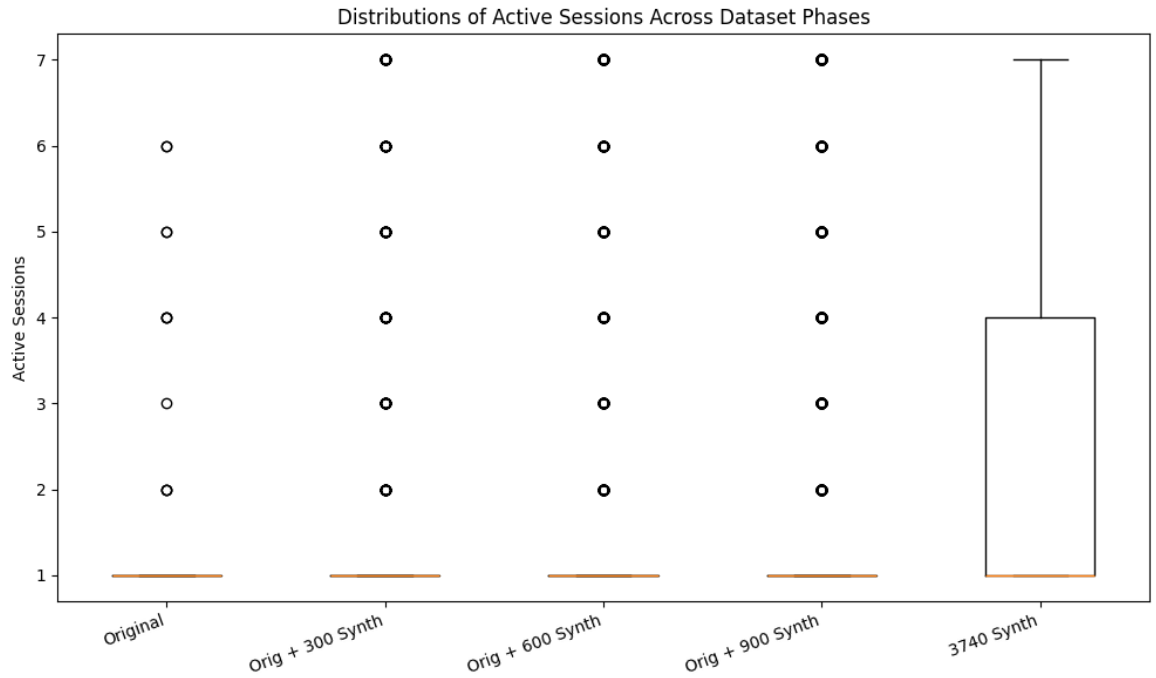


Figure 3.7 Distributions of active sessions across dataset phases

The fourth graph shows the distribution of critical vulnerabilities across the original dataset and progressively expanded synthetic datasets. As synthetic data is added, the overall prevalence and the presence of higher outliers increase, indicating greater variability in the number of critical vulnerabilities. However, the median remains low at all phases, suggesting that most devices still exhibit few or no critical vulnerabilities, with rare extreme cases are preserved.

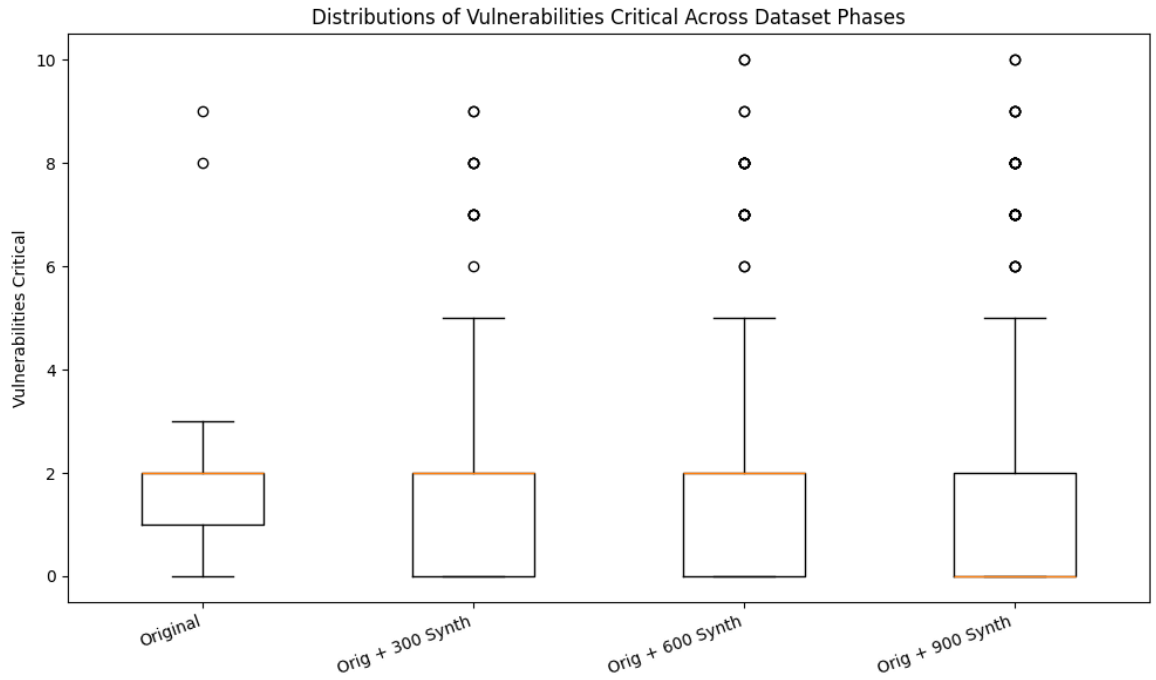


Figure 3.8 Distributions of vulnerabilities critical across dataset phases

The fifth graph shows a clear upward shift in pending updates as synthetic data is added. The median increases steadily from the original dataset through all synthetic phases, and the interquartile range widens. The full synthetic dataset shows the highest medians and largest variability, indicating more diverse patching states across devices.

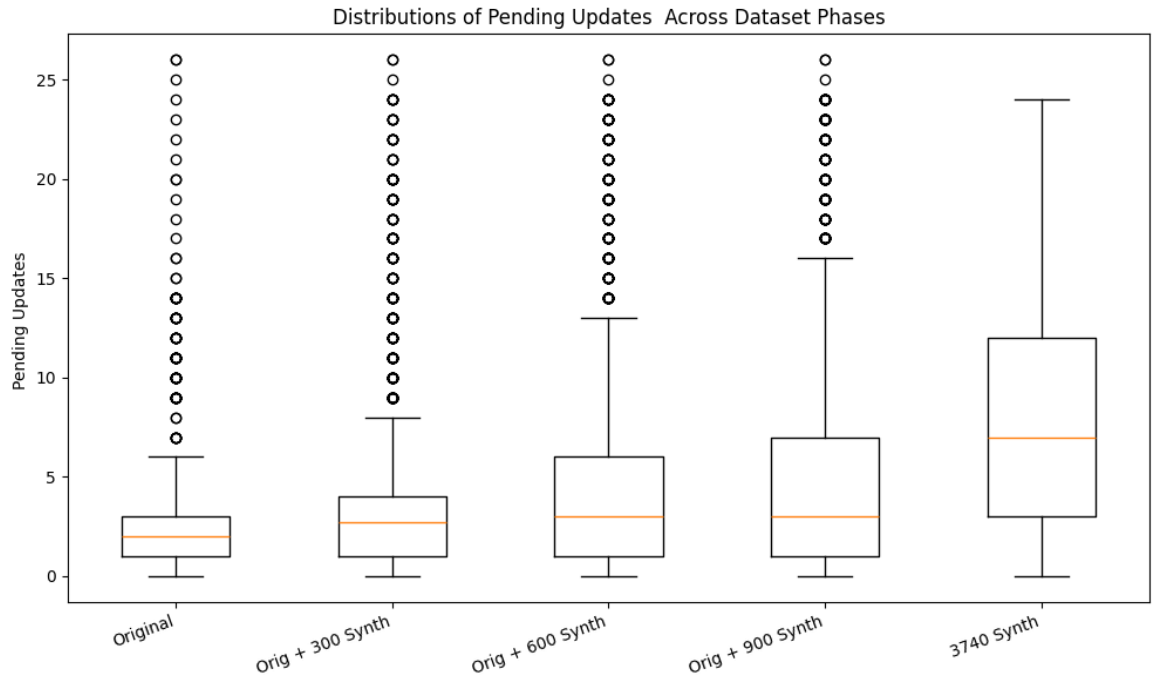


Figure 3.9 Distributions of pending updates across dataset phases

3.7 Feature Selection

The selection of features in this study was guided by the goal of preserving the most effective security indicators for assessing the device's status within a zero-trust framework. After initial cleaning and processing steps, which removed identity-related fields, timestamp variables, low-variance features, and any features that could cause data leakage, the remaining features constituted the core feature set.

To ensure the selected features accurately represent real-world security conditions, the resulting feature set was reviewed and validated through consultation with six cybersecurity experts. Their evaluation focused on ensuring that each retained feature reflected a significant aspect of device vulnerability, system performance, or potential compromise. Based on this expert validation, the final selected features included: CPU usage, FTP status, RPC status, Telnet status, RDP status, unsigned drivers, active sessions, number of failed logins, uptime, number of pending updates, critical vulnerabilities, high vulnerabilities, User Account Control (UAC) level, antivirus activation, real-time protection, and dark web mentions. Collectively, these features monitor service exposure, misconfiguration risks, authentication anomalies, patch status, vulnerability severity, and external threat indicators.

To evaluate the effectiveness of this reduced set of features, the models were trained and tested using only the features approved by experts. The performance results showed a significant decrease in accuracy, precision, recall, and F1 score compared to models trained on the full set of features remaining after preprocessing. This result indicated that some of the removed features still contribute additional contextual value that supports model performance.

Therefore, the most reliable and effective approach was to retain all remaining features after the cleaning and preprocessing phase. This ensured that no potential informational signals were lost, and enabled classification models to access the full range of relevant device characteristics when making zero-trust access decisions.

3.8 Model Training and Evaluation

To evaluate the effectiveness of machine learning in detecting the health of network devices, several models were trained and evaluated on the prepared dataset. The goal was to identify the most accurate and interpretable model for predicting whether a device would be accepted or denied based on its security posture.

3.8.1 Model Selection, data splitting and scaling:

A diverse set of machine learning algorithms was selected to evaluate various learning approaches, including linear models, tree-based methods, and ensemble techniques: (Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbors (KNN), Random Forest, Gradient Boosting, XGBoost) for more details in Table below.

To mitigate the risk of biased evaluation and ensure reliable performance assessment, the dataset was split into two subsets: a training set and a test set, using a stratified random approach. 70% of the data was allocated to training and 30% to testing. Stratification was applied based on class label to preserve the original class distribution in both subsets. Although the split was performed at the record level rather than the device level, each record represents an independent snapshot of the endpoint's security state. Therefore, there is no temporal or correlational dependency between records that could lead to device data leakage. The use of a fixed random seed value (random_state = 42) ensures that the split can be fully reproduced. This strategy balances sufficient data availability for model learning with a statistically unbiased and representative test set, while avoiding systematic overfitting.

Table 3.6 Justifications for choose the ML algorithm

Algorithm	**Justification for use in this study**	**Reference**
Logistic Regression (LR)	It is widely used for binary classification tasks (e.g, distinguishing malicious vs. benign) due to its good detection performance and interpretability	Gu j, 2020
Support Vector Machine (SVM)	Known for its strong classification performance, especially on high-dimensional data, the SVM finds optimal super levels for class separation, which is useful for accurately distinguishing between normal and malicious cases. Kernel tricks enable it to handle the nonlinear boundaries common in cybersecurity data.	Hesham et al., 2024
Decision Tree	It produces clear, human-understandable models (mimicking and simulating decision logic) that can identify attack patterns by branching on feature values. This interpretability helps in intrusion detection and reduces false positives.	Hesham et al., 2024
Random Forest	An ensemble of decision trees that improves accuracy and controls overfitting by averaging multiple trees. Its robustness to noise and ability to capture non-linear feature interactions make it ideal for detecting complex threats.	Hesham et al., 2024
k-Nearest Neighbors (KNN)	KNN is a simple instance-based classifier that classifies by similarity. Effective for anomaly detection since it flags points that are distant from “normal” instances. it quick to deploy for threat detection with smaller datasets.	Hesham et al., 2024
XGBoost (Extreme Gradient Boosting)	A powerful enhancement algorithm known for its speed and high accuracy. It can handle large datasets and record complex interactions between features, helping to identify precise attack patterns.	Hitachi Vantara Federal, 2024

3.8.2 Approaches:

To evaluate how the size and composition of the dataset affected model performance, several training strategies were implemented. The first approach relied exclusively on the original dataset of 1181 real device records and observations. This baseline allowed the models to learn directly from high-quality, documented observations without the need for any artificially generated samples. This baseline served as the benchmark against which all subsequent augmentation approaches were compared.

Based on the baseline, the study employed a incremental synthetic augmentation approach. The cleaned synthetic dataset was used to extract three balanced subsets, each containing approximately 300 records. These subsets were sequentially added to the original dataset, resulting in three progressively larger training sets, df_original_with_300_synthetic, df_original_with_600_synthetic, and df_original_with_900_synthetic. This design enabled a controlled examination of how progressive increases in synthetic data affected the model's generalization and stability, while maintaining class balance and diversity.

In addition to these incremental combinations, the models were trained using a final approach with a fully cleaned synthetic dataset, consisting of 3,740 records after the removal of duplicates and inconsistencies. This complete synthetic dataset allowed the models to learn from a much larger sample space and provided a deeper understanding of whether synthetic data could produce reliable and comparable results with only mixed or real datasets. Taken together, these approaches provided a structured framework for evaluating the effects of data augmentation, dataset composition, and training volume on the overall model performance.

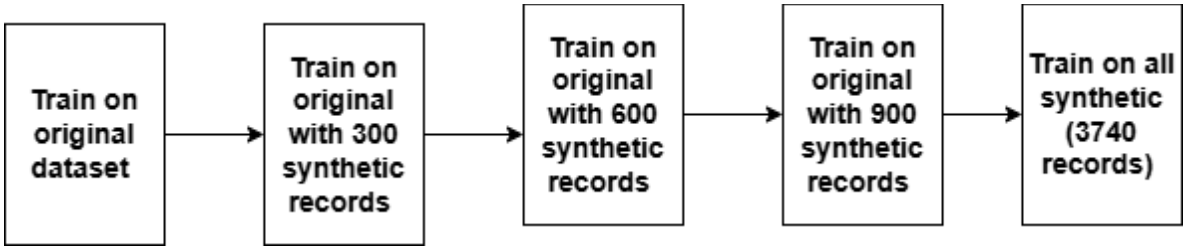


Figure 3.10 Machine learning Training Approaches

3.8.3 Evaluation Metrics:

The performance of all models was evaluated using a set of standard assessment metrics to ensure prediction accuracy and robustness. The table 3.7 below summarizes the metrics used in this study, including accuracy-based metrics, error-based metrics, and computational efficiency indices.

Table 3.7 Evaluation metrics applied to measure predictive accuracy, robustness, and computational performance of the machine learning models.

Metric	**Description**
Accuracy	Measures overall prediction correctness
Cross-Validation Accuracy	Evaluates model stability across multiple training/testing splits, using a 5-fold.
Precision	Proportion of correctly predicted positive cases out of all predicted positives.
Recall	Ability to identify actual positive cases.
F1 Score	Harmonic mean of precision and recall
Training Time (seconds)	Time required for the model to complete training.
Inference Time (milliseconds)	Time required for the model to generate a prediction for a new instance.
Confusion Matrix (TP, FP, TN, FN)	Detailed breakdown of classification outcomes.

All experiments were conducted in Google Colab cloud runtime environment, which provided a virtualized CPU environment based on an AMD EPYC 7B12 processor. The machine used in this study offered two virtual CPU cores, built on a 64-bit x86_64 architecture. The processor included a cache hierarchy consisting of 32 KB L1 data cache, 32 KB L1 instruction cache, 512 KB L2 cache, and a 16 MB L3 cache. The approximate CPU frequency of this instance is around 2.25 GHz, consistent with typical AMD EPYC 7B12 configurations. These hardware specifications were verified using standard Linux system commands (lscpu).

Chapter 4 Results

This chapter presents and analyzes experimental results obtained from applying different machine learning models to assess the security of network devices within a zero-trust architecture. The results are organized according to the study objectives and research questions identified in Chapter One. The main aim of this chapter is to provide an objective report on the effectiveness of the selected machine learning models in classifying devices into "Accept" or "Deny" classes based on their security features.

Several machine learning models were trained and tested using the prepared dataset to evaluate their effectiveness in predicting device access decisions. These models included linear, tree-based, and ensemble methods, all implemented under identical experimental conditions. The dataset was divided into training and testing subsets to ensure a fair evaluation. The performance of each model reflects its ability to learn patterns from multiple security indicators and produce accurate classification results in a zero-trust environment.

To evaluate the performance of machine learning models, standard classification metrics were used, including accuracy, cross-validation accuracy, precision, recall, F1 score, training and inference time. These metrics provide a quantitative measure of how accurately each model classifies secure and risky devices. The use of these metrics aims to identify the most reliable and effective model for assessing the security of devices and to demonstrate the viability of using machine learning as a data-driven decision-making mechanism for access control within a zero-trust architecture.

The experiments were conducted using several dataset phase approaches; the original real-world dataset was used as a baseline, followed by multiple extended datasets created by gradually adding synthetic records. In successive phases, 300, 600, and 900 synthetic records, finally all synthetic records were approximately 3740. Each expanded dataset was treated as an independent experimental stage, and the same set of machine learning models was applied to ensure consistent and comparable results across all stages.

4.1 Evaluating the performance of machine learning models across incremental expansions in datasets

The results are displayed based on each approach

For approach one, Baseline Evaluation Using the Original Dataset, in this initial approach, the dataset includes only a real-world dataset, approximately 1200 records. The applied machine learning models achieved accuracy values ranging from 87.32% to 98.87%, with Gradient Boosting recording the highest performance (Accuracy: 98.87%, F1-score: 99.02%), followed by Decision Tree and XG Boost models.

Table 4.1 Performance of Machine Learning Models on the Original Dataset

Model	**Accuracy**	**CV**	**Precision**	**Recall**	**F1 Score**	**PR-AUC**	**False positive rate**	**Training Time**	**Inference (ms/sample)**
Logistic Regression	87.32	77.32	87.5	91.3	89.36	93.86	18.24	0.0392	0.013
SVM	87.32	68.68	85.84	93.72	89.61	96.39	21.62	0.0328	0.0374
KNN	87.61	75.89	85.28	95.17	89.95	92.55	22.97	0.0096	0.154
Decision Tree	97.46	89.1	98.06	97.58	97.82	97.56	2.70	0.0143	0.0074
Random Forest	95.21	88.42	93.98	98.07	95.98	99.49	6.08	0.2838	0.0476
Gradient Boosting	98.87	94.93	99.67	98.07	99.02	99.85	0.42	0.6166	0.0119
XG Boost	96.9	89.27	97.12	97.58	97.35	99.85	4.05	0.1215	0.0224

In the table 4.1 above, a low cross-validation score for an SVM indicates its reduced robustness and generalizability across different data partitions. This behavior is primarily due to the SVM's sensitivity to hyperparameter selection, the variability in support vector selection across folds, and the potential for mismatch between the selected kernel and the underlying data distribution. Consequently, while the SVM achieved competitive accuracy in single-split, its cross-validation performance indicates weaker stability compared to ensemble-based models.

In the figure 4.1 below, tree-based and ensemble models outperform linear models in the original dataset, achieving higher accuracy, higher F1 scores, and more stable cross-validation performance as shown in the figure below.

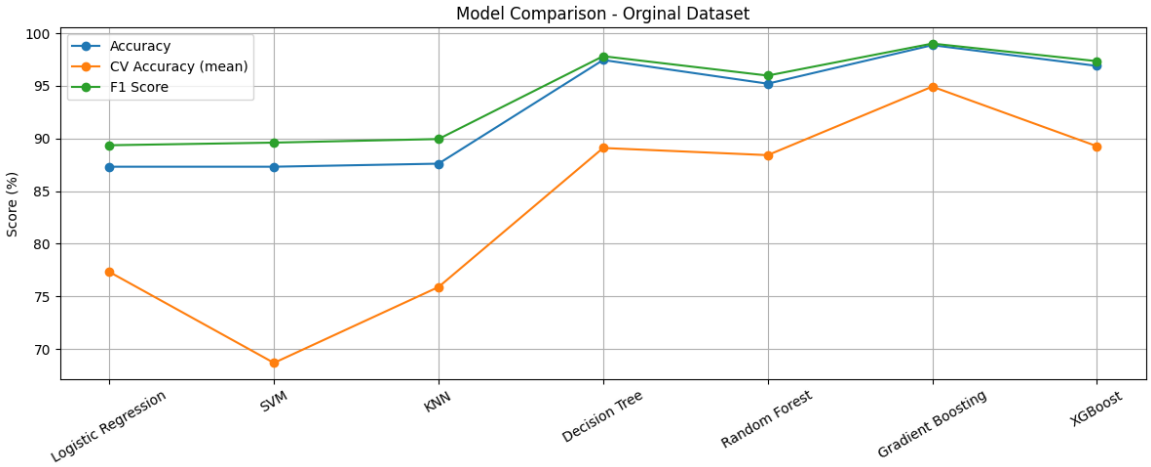
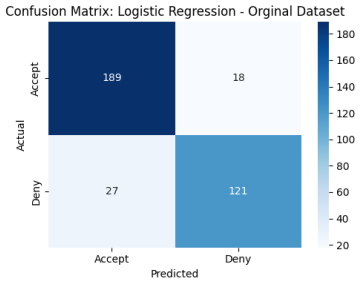
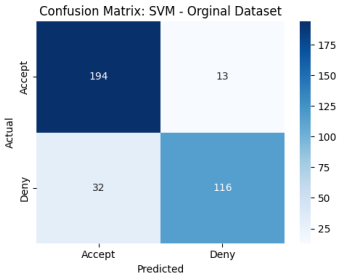


Figure 4.1 Comparison of Accuracy, Cross-Validation Accuracy, and F1-Score Across Machine Learning Models on the Original Dataset

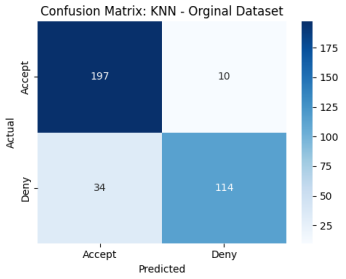
Confusion matrices show that all models achieved high correct classification rates for both the accept and deny classes on the original dataset. Tree-based and ensemble-based models, particularly decision trees, random forests, gradient boosting, and XG Boost, produced fewer misclassifications compared to linear and distance-based models. Overall, the results indicate consistent and balanced classification performance among the evaluated models as shown in the figure 4.2 below.



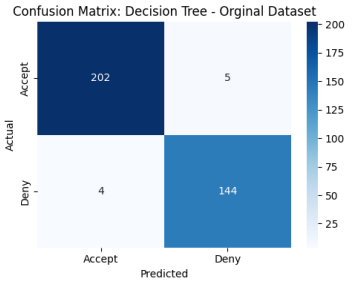
(a) LR



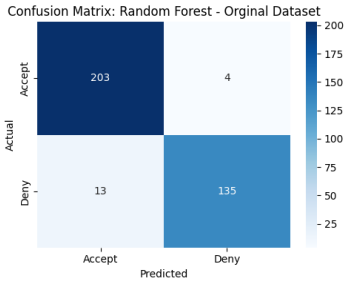
(b) SVM



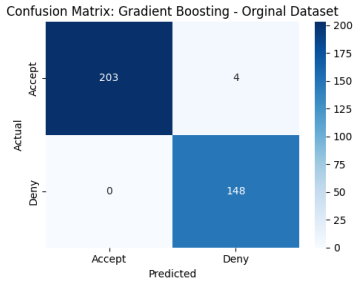
(c) KNN



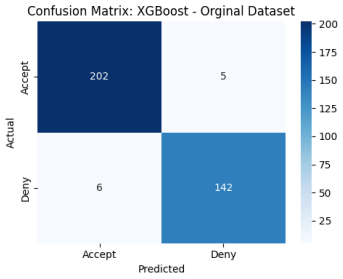
(d) DT



(e) RF



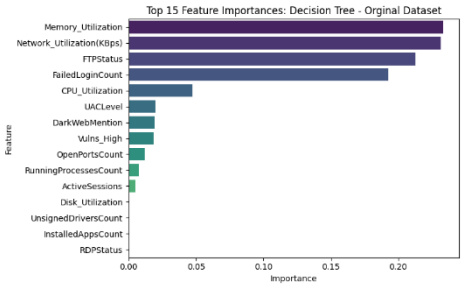
(f) GB



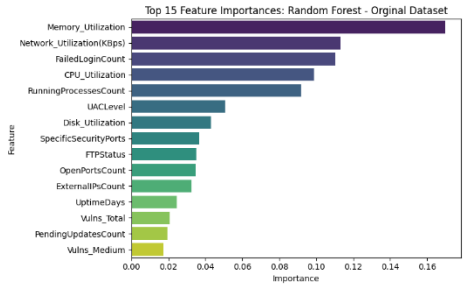
(g) XGB

Figure 4.2 Confusion Matrices of Machine Learning Models for Accept and Deny Classification on the Original Dataset

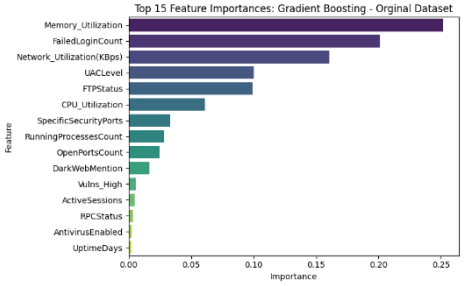
Figures 4.3 below show the rankings of the top 15 features generated by the Decision Tree, Random Forest, Gradient Boosting, and XG Boost models using the original dataset. The figures illustrate the relative contribution of each feature to the ranking decision, as directly calculated by the trained models. Across all models, System Usage, Network Activity, Authentication-Related, and Security Configuration Indicators consistently rank among the top-ranked features.



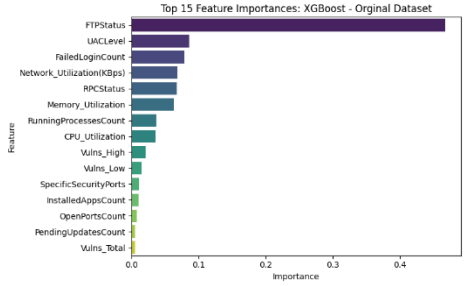
(a) DT



(b) RF



(c) GB



(d) XGB

Figure 4.3 Top 15 Feature Importance Rankings Produced by Tree-Based and Ensemble Models on the Original Dataset

Next, the same machine learning models were applied using a reduced feature set consisting of the 16 most important security-related features, which were selected based on expert consultation in the field of cybersecurity. The results show an overall decrease in performance across all models compared to the full-featured baseline, with accuracy values

ranging from 81.41% to 89.58% and lower F1 scores. Because of this decrease in performance, this feature reduction approach was not considered in subsequent experimental phases.

Table 4.2 Performance of Machine Learning Models Using the Top 16 Security-Relevant Features

Model	Accuracy	CV	Precision	Recall	F1	Training Time	Inference (ms/sample)
Logistic Regression	81.97	74.27	81.5	89.37	85.25	0.0304	0.0114
SVM	81.41	75.29	88.11	78.74	83.16	0.0488	0.0561
KNN	82.25	74.45	83.03	87.44	85.18	0.0147	0.0382
Decision Tree	84.79	74.86	86.96	86.96	86.96	0.0089	0.0077
Random Forest	86.48	79.17	86.64	90.82	88.68	0.2568	0.04
Gradient Boosting	89.58	84.68	90.48	91.79	91.13	0.2564	0.0085
XG Boost	86.76	81.54	87.38	90.34	88.84	0.0758	0.0158

For approach two, using the original dataset plus 300 synthetic records, all machine learning models showed an improvement in classification performance compared to the baseline dataset. Accuracy values ranged from 88.09% to 99.10%, with tree-based and ensemble models achieving the highest scores. High accuracy, recall, and F1 values were consistently observed in most models, indicating stable performance across this dataset expansion.

Table 4.3 Performance of Machine Learning Models Using the Original Dataset Augmented with 300 Synthetic Records

Model	Accuracy	CV	Precision	Recall	F1	Training Time	Inference (ms/sample)
Logistic Regression	88.99	77.06	85.82	96.41	90.81	0.1985	0.0172
SVM	88.09	77.26	83.22	98.8	90.35	0.0668	0.0794
KNN	88.99	76.58	85.31	97.21	90.88	0.0169	0.0412
Decision Tree	99.1	91.11	99.6	98.8	99.2	0.0132	0.0062
Random Forest	98.65	88.2	98.8	98.8	98.8	0.3304	0.0308
Gradient Boosting	98.43	90.5	98.41	98.8	98.61	0.5976	0.007
XG Boost	98.2	90.23	98.41	98.41	98.41	0.1001	0.0166

Tree-based and ensemble models achieve the highest accuracy and F1-scores after adding 300 synthetic records, with improved cross-validation stability as shown in the figure 4.4 below.

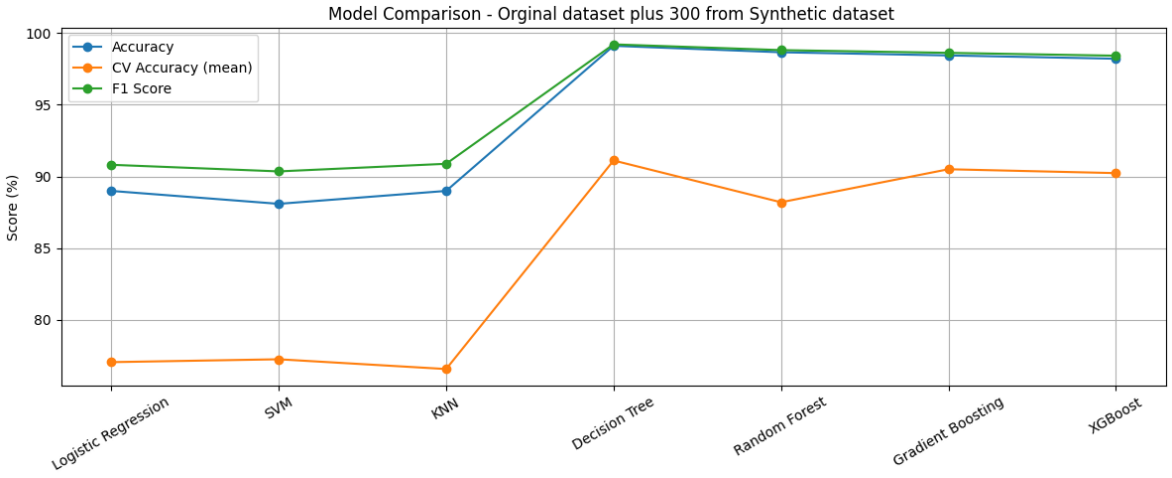
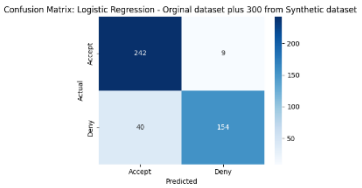
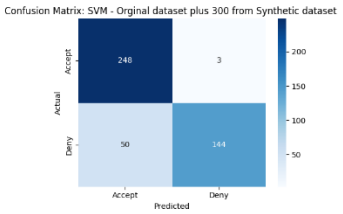


Figure 4.4 Comparison of Accuracy, Cross-Validation Accuracy, and F1-Score Across Machine Learning Models Using the Original Dataset Augmented with 300 Synthetic Records

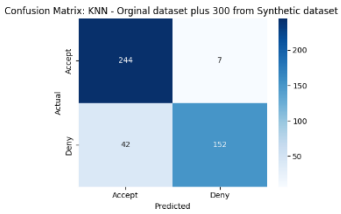
Figures 4.5 below show the confusion matrices for all machine learning models evaluated using the original dataset augmented with 300 synthetic records. The matrices indicate large numbers of correctly classified accepts and denies across all models, with relatively low false classification rates. Both tree-based and ensemble models exhibit strong classification consistency for both categories under this dataset configuration.



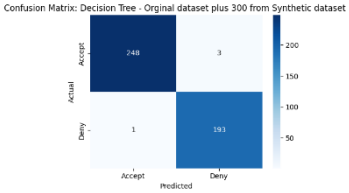
(a) LR



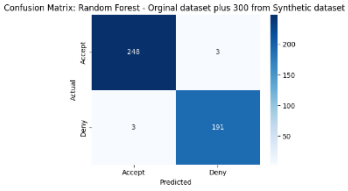
(b) SVM



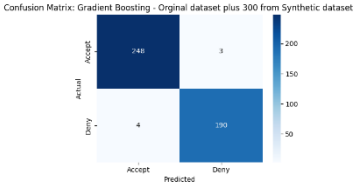
(c) KNN



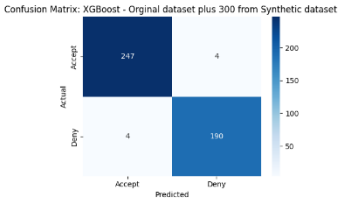
(d) DT



(e) RF



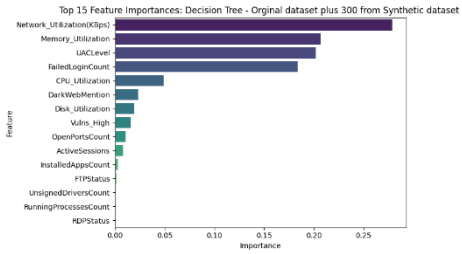
(f) GB



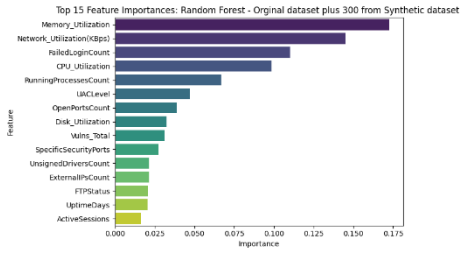
(g) XGB

Figure 4.5 Confusion Matrices of Machine Learning Models Using the Original Dataset Augmented with 300 Synthetic Records

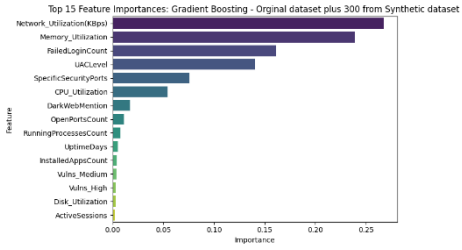
Figures 4.6 below show the rankings of the top 15 features generated by the Decision Tree, Random Forest, Gradient Boosting, and XG Boost models using the original dataset augmented with 300 synthetic records. The figures illustrate the relative contribution of each feature to the ranking as calculated by the trained models. In all models, System Usage, Network Activity, Authentication Indicators, and Security Configuration features appear among the top-ranked features.



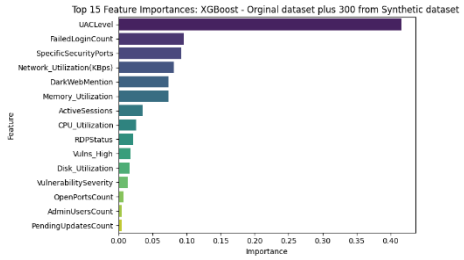
(a) DT



(b) RF



(c) GB



(d) XGB

Figure 4.6 Top 15 Feature Importance Rankings Produced by Tree-Based and Ensemble Models on the Original Dataset Augmented with 300 Synthetic Records

For approach three, using the original dataset augmented with 600 synthetic records, the evaluated machine learning models achieved high classification performance across all metrics. Accuracy values ranged from 90.84% to 98.88%, with tree-based and ensemble models maintaining the highest levels of accuracy, precision, recall, and F1 scores. Cross-validation results indicate stable performance across all folds, while training and inference times remained within acceptable limits.

Table 4.4 Performance of Machine Learning Models Using the Original Dataset Augmented with 600 Synthetic Records

Model	**Accuracy**	**CV**	**Precision**	**Recall**	**F1**	**Training Time**	**Inference (ms/sample)**
Logistic Regression	91.78	84.52	89.38	96.62	92.86	0.0634	0.0141
SVM	91.59	79.69	87.24	99.32	92.89	0.0647	0.0384

KNN	90.84	81.65	88	96.62	92.11	0.0109	0.0314
Decision Tree	98.88	91.65	98.99	98.99	98.99	0.0174	0.0062
Random Forest	98.5	91.87	98.32	98.99	98.65	0.3613	0.0262
Gradient Boosting	98.88	92.1	98.99	98.99	98.99	0.7257	0.0073
XG Boost	98.13	90.13	97.99	98.65	98.32	0.1206	0.0145

Tree-based and ensemble models maintain superior accuracy and F1-scores with improved cross-validation stability after adding 600 synthetic records as shown in the figure 4.7 below.

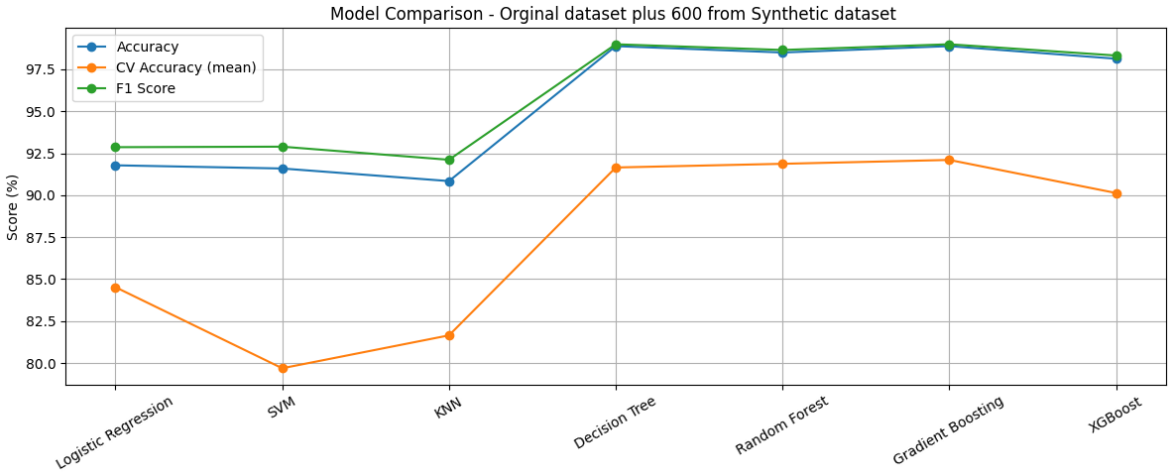
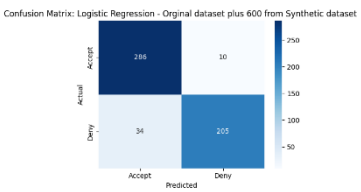
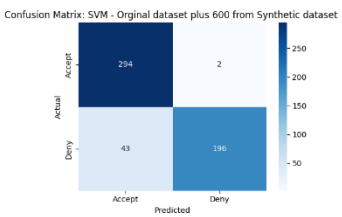


Figure 4.7 Comparison of Accuracy, Cross-Validation Accuracy, and F1-Score Across Machine Learning Models Using the Original Dataset Augmented with 300 Synthetic Records

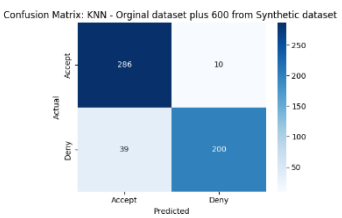
Figures 4.8 below present the confusion matrices of the evaluated machine learning models using the original dataset augmented with 600 synthetic records. The results show a high number of correctly classified Accept and Deny instances across all models, with minimal misclassification. Tree-based and ensemble models exhibit particularly balanced classification performance for both classes under this dataset configuration.



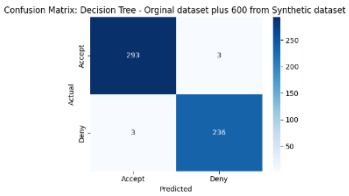
(a) LR



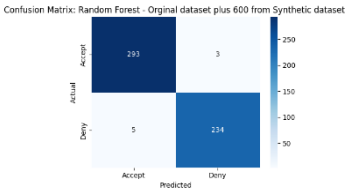
(b) SVM



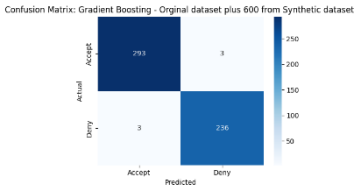
(c) KNN



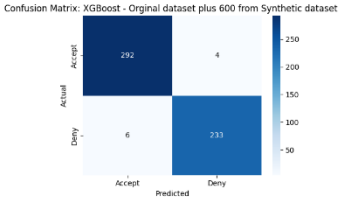
(d) DT



(e) RF



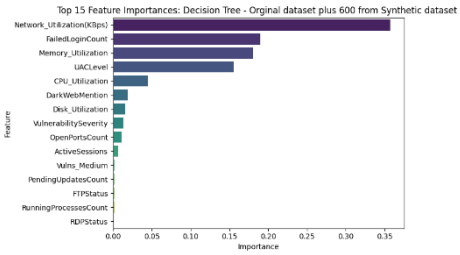
(f) GB



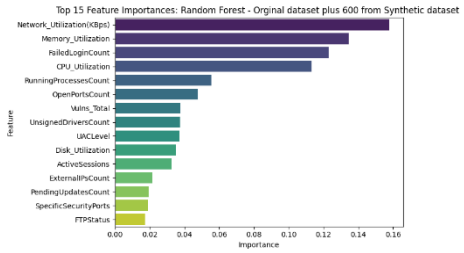
(g) XGB

Figure 4.8 Confusion Matrices of Machine Learning Models Using the Original Dataset Augmented with 600 Synthetic Records

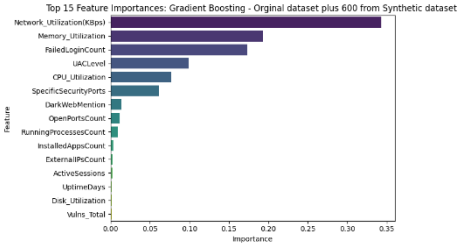
Figures below 4.9 show the rankings of the top 15 features, obtained using decision tree, random forest, gradient boost, and XG Boost models, with the original dataset augmented by 600 synthetic records. The figures show the relative contribution of each feature to the ranking decision, as calculated by the trained models. In all models, network usage, memory usage, authentication indicators, and system activity consistently appear among the top-ranked features.



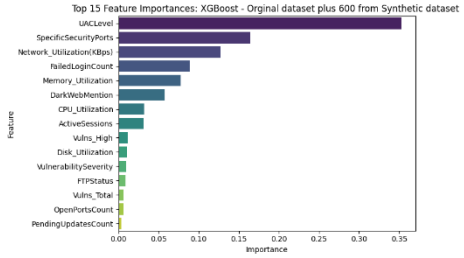
(a) DT



(b) RF



(c) GB



(d) XGB

Figure 4.9 Top 15 Feature Importance Rankings Produced by Tree-Based and Ensemble Models on the Original Dataset Augmented with 600 Synthetic Records

For approach four, using the original dataset augmented with 900 synthetic records, all machine learning models continued to demonstrate high classification performance

across the evaluation metrics. Accuracy values ranged from 90.56% to 98.56%, with tree-based and ensemble models achieving the highest accuracy, precision, recall, and F1 scores. Cross-validation results indicate stable model performance, while training and inference times remained within acceptable ranges.

Table 4.5 Performance of Machine Learning Models Using the Original Dataset Augmented with 900 Synthetic Records

Model	Accuracy	CV	Precision	Recall	F1 Score	Training Time (	Inference (ms/sample)
Logistic Regression	91.84	88.62	89.84	95.89	92.77	0.1901	0.017
SVM	91.52	87.52	87.11	99.12	92.73	0.0781	0.076
KNN	90.56	86.22	87.5	96.48	91.77	0.0111	0.0304
Decision Tree	98.56	93.72	98.82	98.53	98.68	0.0184	0.0045
Random Forest	97.76	93.19	97.12	98.83	97.97	0.4089	0.0216
Gradient Boosting	97.76	93.23	96.85	99.12	97.97	0.832	0.0056
XG Boost	97.92	92.23	97.4	98.83	98.11	0.3603	0.0288

Tree-based and ensemble models maintain high accuracy and F1-scores with stable cross-validation performance after adding 900 synthetic records, as shown in the figure 4.10 below.

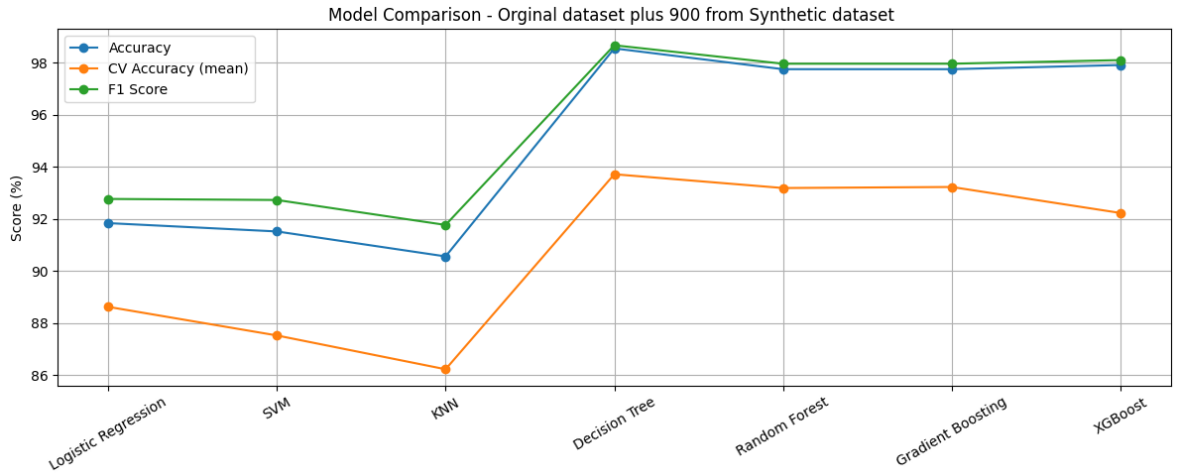
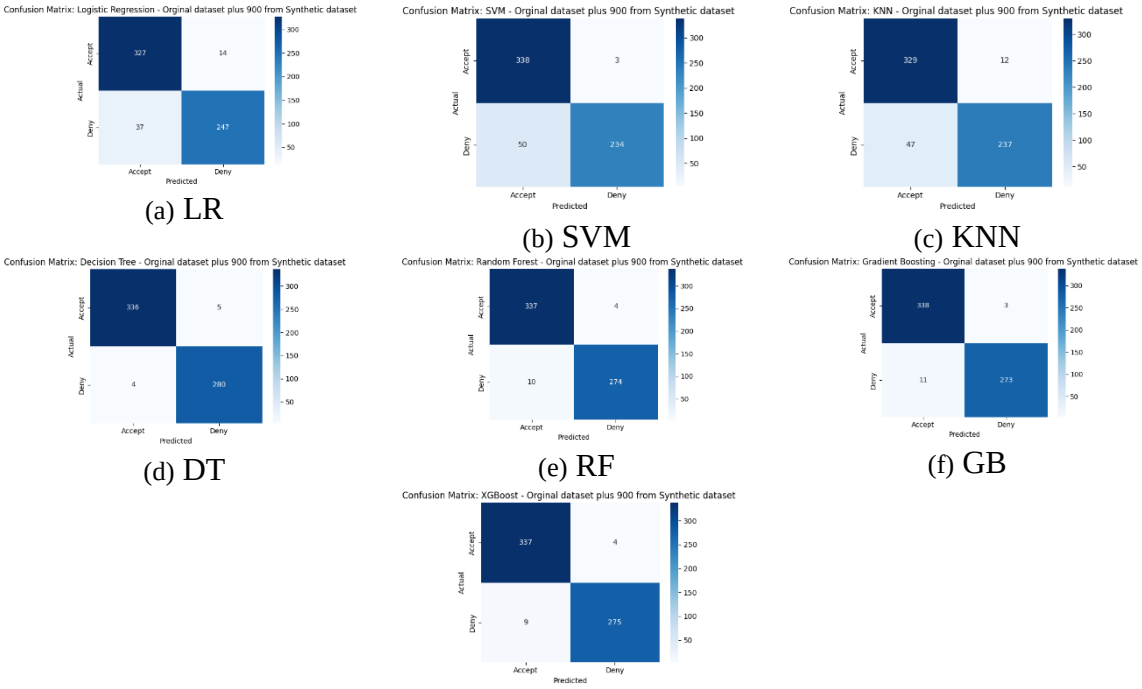


Figure 4.10 Comparison of Accuracy, Cross-Validation Accuracy, and F1-Score Across Machine Learning Models Using the Original Dataset Augmented with 300 Synthetic Records

Figures 4.11 below show the confusion matrices for all machine learning models evaluated using the original dataset augmented with 900 synthetic records. The results indicate a high number of correctly classified accepts and denies across all models, with

low misclassification rates. Both tree-based and ensemble models maintain particularly balanced classification performance for both classes under this dataset configuration.



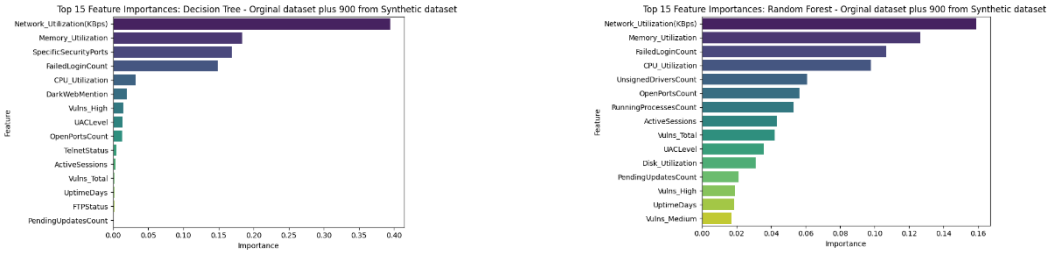
(a) LR (b) SVM (c) KNN

(d) DT (e) RF (f) GB

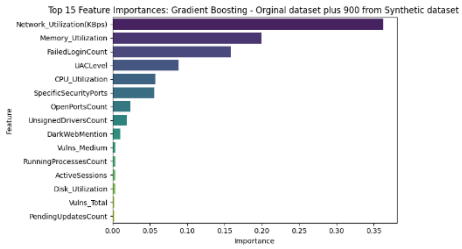
(g) XGB

Figure 4.11 Confusion Matrices of Machine Learning Models Using the Original Dataset Augmented with 900 Synthetic Records

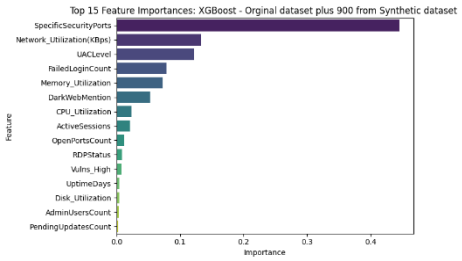
Figures 4.12 below show the rankings of the top 15 features, generated by the Decision Tree, Random Forest, Gradient Boost, and XG Boost models, using the original dataset augmented with 900 synthetic records. The figures illustrate the relative contribution of each feature to the ranking, as calculated by each model. In all evaluated models, network usage, memory usage, authentication indicators, and security settings features consistently ranked among the top features.



(a) DT (b) RF



(c) GB



(d) XGB

Figure 4.12 Top 15 Feature Importance Rankings Produced by Tree-Based and Ensemble Models on the Original Dataset Augmented with 900 Synthetic Records

For approach five, full dataset evaluation using the original dataset (approximately 1200 records) augmented with the complete synthetic dataset (approximately 3,740 records) the sum approximately 5000 records, all machine learning models achieved very high classification performance across all evaluation metrics. Accuracy values ranged from 96.82% to 99.66%, with tree-based and ensemble models achieving the highest accuracy, precision, recall, and F1-scores. Cross-validation results indicate strong performance stability, while training and inference times remained within acceptable ranges.

Table 4.6 Performance of Machine Learning Models Using Original Dataset with Full Synthetic Data Augmentation

Model	**Accuracy**	**CV**	**Precision**	**Recall**	**F1**	**False Positive Rate**	**Training Time**	**Inference (ms/sample)**
Logistic Regression	96.82	92.89	95.69	98.31	96.98	3.59	0.1487	0.0081
SVM	97.7	90.5	97.05	98.57	97.8	2.82	0.1636	0.0521
KNN	97.22	92.81	96.42	98.31	97.36	3.95	0.0136	0.0521
Decision Tree	99.39	96.24	99.48	99.35	99.41	0.28	0.044	0.002
Random Forest	99.66	92.71	99.74	99.61	99.67	0.42	0.9116	0.0168
Gradient Boosting	99.59	96.37	99.87	99.35	99.61	0.14	1.7967	0.0034
XG Boost	99.32	96.04	99.35	99.35	99.35	0.71	0.2534	0.0222

Using the full augmented dataset, all models achieve very high accuracy and F1 scores, while maintaining superior performance for tree-based and ensemble models.as shown in the figure 4.13 below.

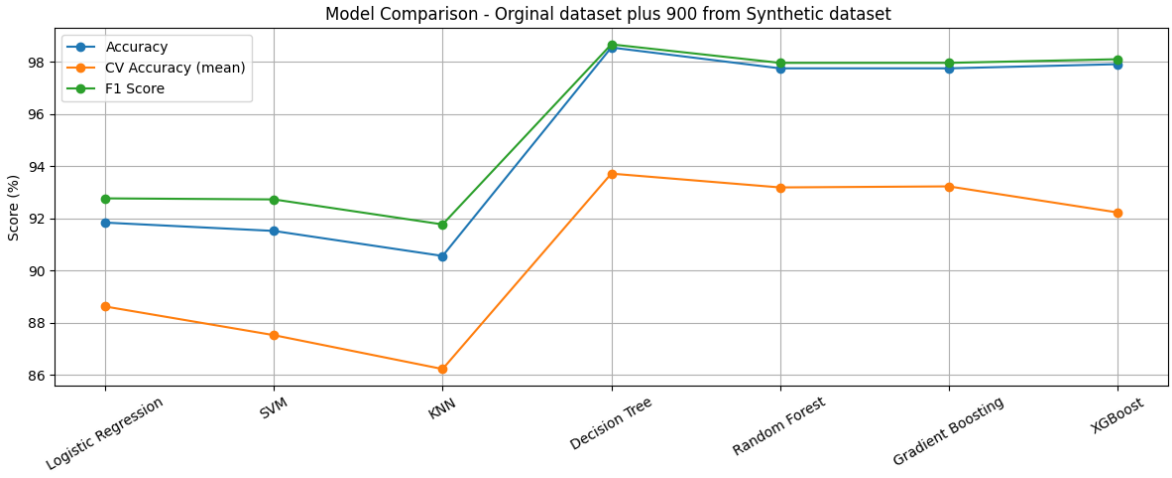
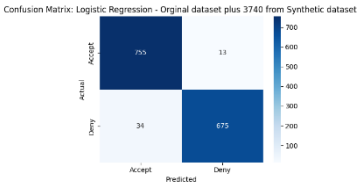
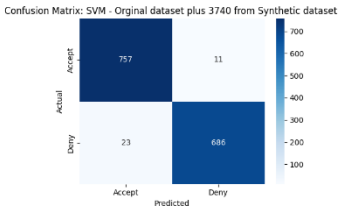


Figure 4.13 Comparison of Accuracy, Cross-Validation Accuracy, and F1-Score Across Machine Learning Models Using the Original Dataset Augmented with full Synthetic Records

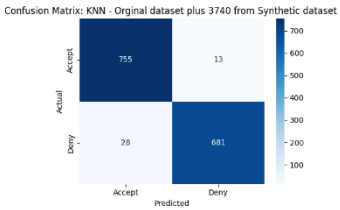
Figures 4.14 below show the confusion matrices for all machine learning models evaluated using the original dataset augmented by the full synthetic dataset (approximately 3740 records). The results indicate a very high number of correctly classified accepts and denies across all models, with a minimal number of misclassifications. Both tree-based and ensemble models consistently maintain balanced classification performance for both classes under this dataset configuration.



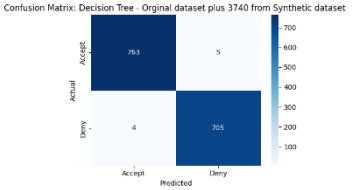
(a) LR



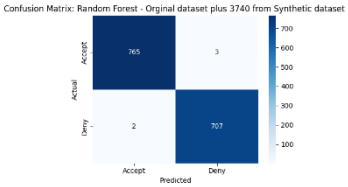
(b) SVM



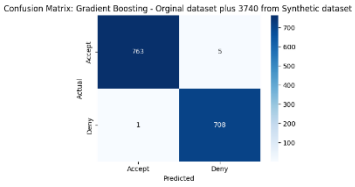
(c) KNN



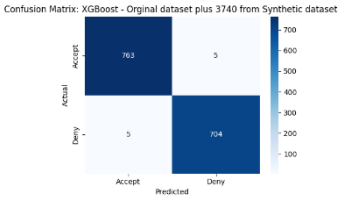
(d) DT



(e) RF



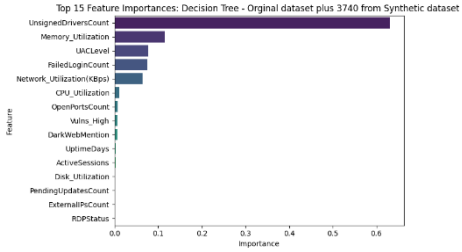
(f) GB



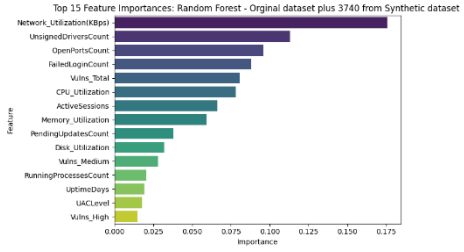
(g) XGB

Figure 4.14 Confusion Matrices of Machine Learning Models Using the Original Dataset Augmented with full Synthetic Records

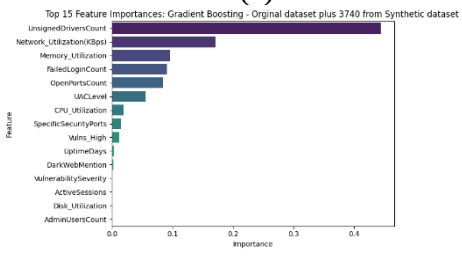
Figures 4.15 below show the rankings of the top 15 features, generated by the Decision Tree, Random Forest, Gradient Boost, and XG Boost models, using the original dataset augmented with full synthetic records. The figures illustrate the relative contribution of each feature to the ranking, as calculated by each model. In all evaluated models, network usage, memory usage, authentication indicators, and security settings features consistently ranked among the top features.



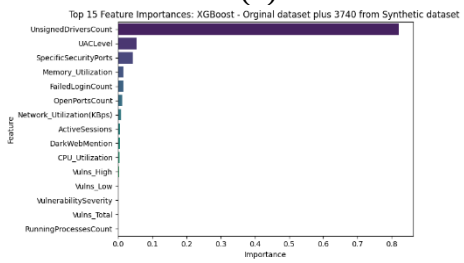
(a) DT



(b) RF



(c) GB



(d) XGB

Figure 4.15 Top 15 Feature Importance Rankings Produced by Tree-Based and Ensemble Models on the Original Dataset Augmented with full Synthetic Records

4.2 Comparative Analysis of Top-Performing Models Across Dataset Phases

This section provides a comparative analysis of all experimental datasets to identify the best-performing machine learning algorithms and measure the impact of dataset expansion on their classification performance. The comparison is conducted using consistent evaluation metrics, including accuracy, precision, recall, F1 score, and cross-validation accuracy, across all five dataset configurations, the original dataset and four incrementally expanded datasets with synthetic records.

Based on the aggregated results from all experimental phases, the Random Forest and Gradient Boosting algorithms consistently achieved the highest performance among all

evaluated models. These two algorithms demonstrated superior accuracy and high F1 scores across all dataset sizes, along with stable cross-validation results, indicating strong generalization performance. While the Decision Tree and XG Boost algorithms also achieved high scores, the Random Forest and Gradient Boosting algorithms maintained more consistent performance trends as the dataset size increased.

To assess the impact of data expansion, the performance of the random forest and gradient boosting models was tracked across successive dataset expansions. As the dataset size increased from approximately 1,200 real-world records to a fully augmented dataset of approximately 3,740 records, both models showed a clear and measurable improvement in classification performance. The accuracy of the random forest model increased from 95.21% on the original dataset to 99.66% on the fully augmented dataset, while its F1 score improved from 95.98% to 99.67%. Similarly, the accuracy of the gradient boosting model increased from 98.87% to 99.59%, with its F1 value score from 99.02% to 99.61%.

In addition to improving accuracy metrics, expanding the dataset enhanced the cross-validation stability of both models. Cross-validation accuracy increased, and variance across folds decreased with the addition of more synthetic data, indicating reduced sensitivity to data segmentation and improved learning of underlying security patterns. Training time increased with dataset size, particularly for the gradient boosting model, while the inference time per sample remained low for both models, maintaining their suitability for real-time or near-real-time access control decisions.

Overall, this comparative analysis demonstrates that increasing the size of the dataset through increased controlled synthetic data has a direct and positive impact on model performance. The results confirm that the random forest and gradient boosting algorithms are the most effective in this study, and that their improved performance is closely linked to the availability of larger and more diverse training datasets, as shown in the table 4.7 below.

Table 4.7 Comparative performance of the two best-performing machine learning models across increasing dataset sizes

Dataset Configuration	Dataset Size (Approx.)	Random Forest Accuracy (%)	Random Forest F1-score (%)	Gradient Boosting Accuracy (%)	Gradient Boosting F1-score (%)
Original dataset (baseline)	~1,200	95.21	95.98	98.87	99.02
Original + 300 synthetic	~1,500	98.65	98.80	98.43	98.61
Original + 600 synthetic	~1,800	98.50	98.65	98.88	98.99
Original + 900 synthetic	~2,100	97.76	97.97	97.76	97.97
Original + full synthetic	~5,000	99.66	99.67	99.59	99.61

Also, the figure 4.16 below shows the Impact of dataset expansion on the classification performance of Random Forest and Gradient Boosting models across all experimental phases, measured using accuracy and F1-score.

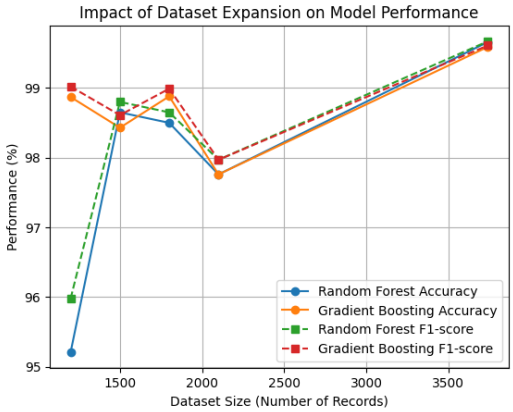


Figure 4.16 Impact of Dataset Expansion on Model Performance

4.3 Global Summary and Comparative Synthesis of Experimental Results

The figure 4.17 below illustrates the accuracy of seven machine learning algorithms across increasingly large datasets, starting with the original dataset and extending to increasing amounts of synthetic data. ensemble-based models, particularly decision trees, gradient boosting, random forests, and XG Boost, consistently maintain high accuracy (around 97–99%) across most dataset sizes, demonstrating strong robustness and stable generalization. In contrast, KNN, logistic regression, and SVM algorithms exhibit lower accuracy on the original dataset but steadily improve with the addition of synthetic data, indicating greater sensitivity to the size of the training data. Overall, the results confirm that

increasing the amount of synthetic data improves classification performance, with the most significant gains observed in non-ensemble models, while ensemble methods remain the most accurate and stable.

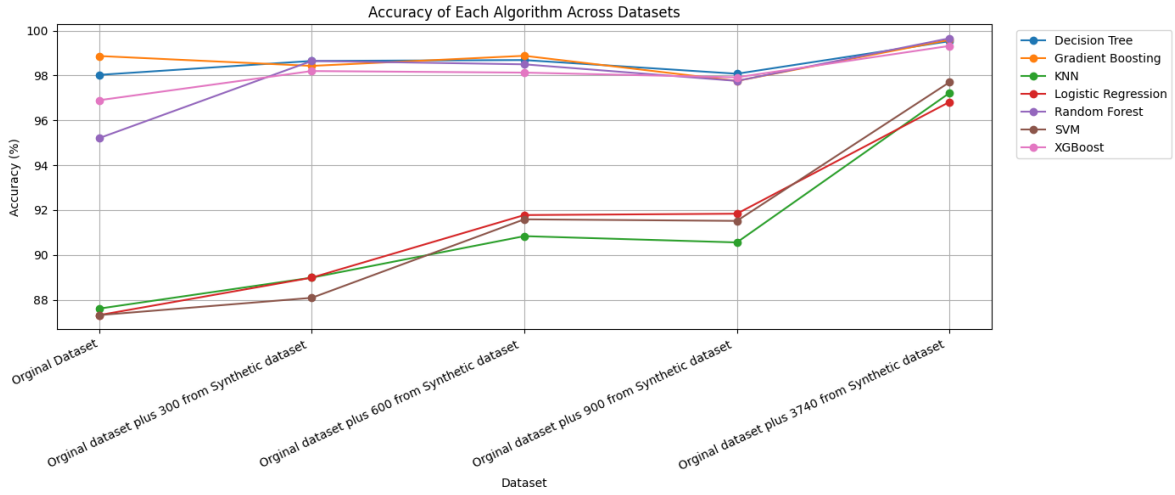


Figure 4.17 Accuracy comparison of machine learning algorithms across the datasets

The figure 4.18 below illustrates the F1 performance of seven machine learning algorithms across increasingly large datasets, starting with the original dataset and extending to increasing amounts of synthetic data. ensemble models, including decision tree, gradient boosting, random forest, and XG Boost, consistently achieve high F1 scores (around 98–99%) across most dataset configurations, indicating strong and balanced classification performance. In contrast, KNN, logistic regression, and SVM algorithms initially show lower F1 scores on the original dataset but steadily improve with the addition of synthetic data, with the largest dataset achieving the highest F1 scores of all models. Overall, the results demonstrate that increasing the synthetic data improves balanced prediction performance, particularly for models more sensitive to data size, while ensemble methods remain consistently stable and superior.

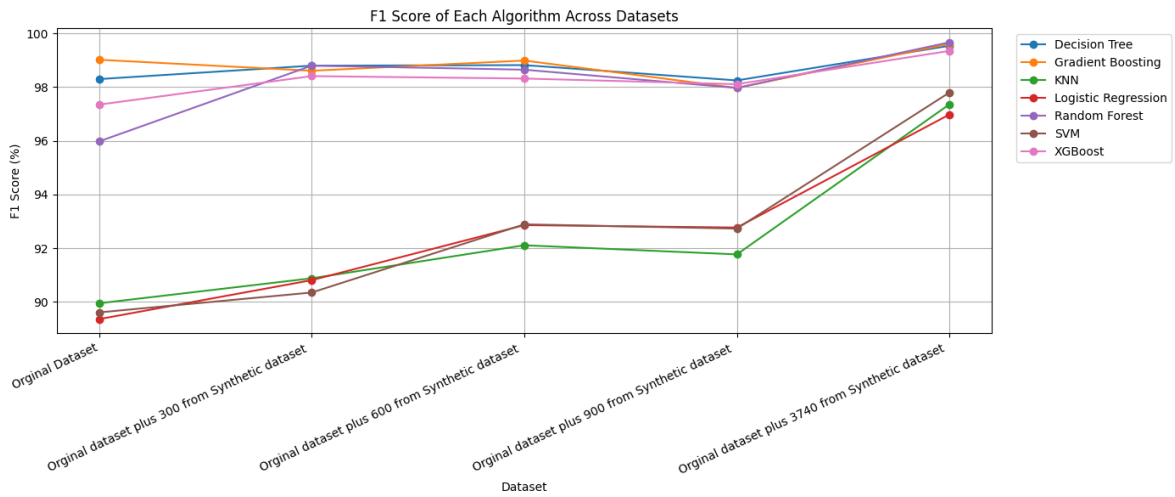


Figure 4.18 F1-score comparison of machine learning algorithms across datasets

The figure 4.19 below illustrates the cross-validation accuracy of seven machine learning algorithms across datasets with increasing amounts of synthetic data. Ensemble-based models, particularly decision trees, gradient boosting, random forests, and XG Boost, consistently achieve higher cross-validation accuracy and show steady improvement as the dataset size increases, demonstrating strong generalization and stability. In contrast, KNN, logistic regression, and SVM algorithms start with relatively lower cross-validation accuracy on the original dataset but show significant gains with the addition of synthetic data, with the most notable improvement observed in SVM. Overall, the results suggest that increasing the dataset with synthetic samples improves model generalization, reduces interfold variance, and leads to more reliable performance estimates, especially for models that are more sensitive to limited training data.

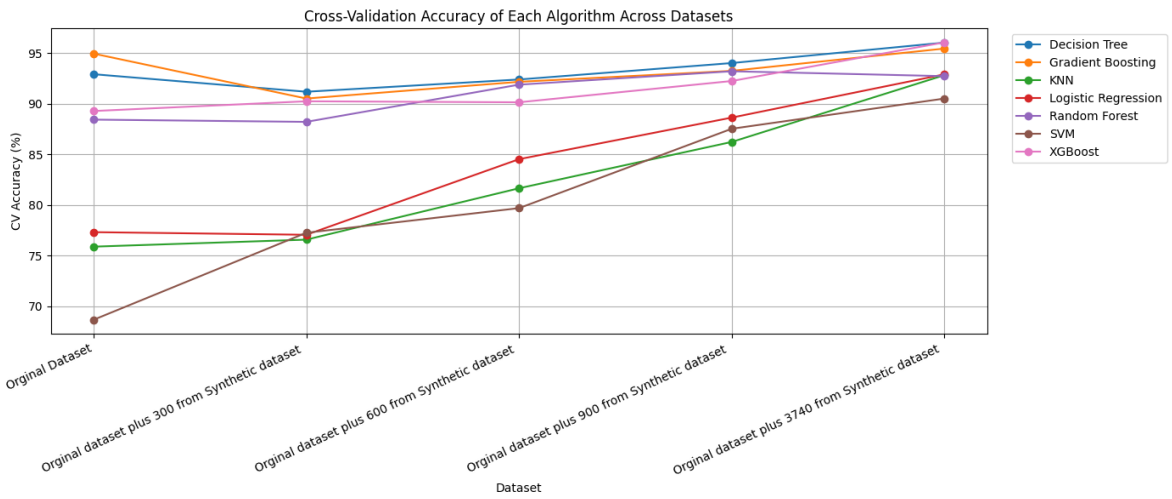


Figure 4.19 Cross-validation accuracy of machine learning algorithms across datasets

The table 4.8 summarizes the best-performing model at each phase of dataset expansion based on classification accuracy, showing cross-validation accuracy, F1 score, and inference time. For the original dataset, the Gradient Boosting model achieved the highest accuracy (98.87%) with strong cross-validation performance and a high F1 score, indicating balanced and reliable predictions. With the addition of 300 synthetic samples, the Decision Tree model became the top performer, maintaining high accuracy (98.65%) while providing the fastest inference time, highlighting its efficiency on medium-sized datasets. With the addition of 600 synthetic samples, the Gradient Boosting model again delivered the best accuracy (98.88%), demonstrating its robustness and consistent performance as the data size increased. In the data expansion stage to 900 samples, the Decision Tree model regained its superiority, achieving competitive accuracy (98.08%) with improved cross-validation stability and a lower inference cost. Finally, for the largest dataset of 3740 synthetic samples, the random forest model emerged as the best, achieving the highest accuracy (99.66%) and the highest F1 value (99.67%), indicating superior generalizability to large-scale datasets at the expense of increased inference time. Overall, the table shows that the optimal model varies with the size of the dataset, and that ensemble methods tend to become more dominant as the dataset grows larger and more diverse.

Table 4.8 Best-performing machine learning model at each dataset expansion phase based on accuracy

Dataset	Model	Accuracy	CV Accuracy	F1 Score	Inference (ms/sample)
Original Dataset	Gradient Boosting	98.87	94.93	99.02	0.0068
Original dataset plus 300 from Synthetic dataset	Decision Tree	98.65	91.17	98.8	0.0041
Original dataset plus 600 from Synthetic dataset	Gradient Boosting	98.88	92.15	98.99	0.0045
Original dataset plus 900 from Synthetic dataset	Decision Tree	98.08	94.0	98.25	0.0035
Original dataset plus 3740 from Synthetic dataset	Random Forest	99.66	92.71	99.67	0.0138

The table 4.9 below provides a comparative summary of all the machine learning algorithms evaluated in the dataset phase, showing each algorithm's highest accuracy, which in all cases corresponds to the largest dataset combining the original data with 3740 synthetic samples. The results show that ensemble-based models dominate performance in this range, with the random forest algorithm achieving the highest accuracy (99.66%) and the highest F1 value (99.67%), closely followed by gradient boosting and decision tree algorithms. Although XG Boost achieves slightly lower accuracy, it demonstrates strong cross-validation stability. In contrast, SVM, KNN, and logistic regression algorithms achieve relatively lower accuracy and F1 values, indicating higher sensitivity to model assumptions even with large datasets. In terms of efficiency, decision tree and gradient boosting exhibit the lowest inference times, while SVM and KNN incur the highest computational cost per sample. Overall, the table shows that increasing the size of the dataset increases performance across all algorithms, as ensemble methods provide the best balance between accuracy and robustness, and simpler models provide faster inference.

Table 4.9 Best accuracy achieved by each machine learning algorithm and the corresponding dataset phase

Model	Dataset	Accuracy	CV Accuracy	F1 Score	Inference Time (ms/sample)
Random Forest	Original dataset plus 3740 from Synthetic dataset	99.66	92.71	99.67	0.0138
Gradient Boosting	Original dataset plus 3740 from Synthetic dataset	99.59	95.43	99.61	0.0024

Model	Dataset	Accuracy	CV Accuracy	F1 Score	Inference Time (ms/sample)
Decision Tree	Original dataset plus 3740 from Synthetic dataset	99.53	96.02	99.54	0.0014
XGBoost	Original dataset plus 3740 from Synthetic dataset	99.32	96.04	99.35	0.0131
SVM	Original dataset plus 3740 from Synthetic dataset	97.70	90.50	97.80	0.0308
KNN	Original dataset plus 3740 from Synthetic dataset	97.22	92.81	97.36	0.0290
Logistic Regression	Original dataset plus 3740 from Synthetic dataset	96.82	92.89	96.98	0.0027

4.4 Impact of Synthetic Data Augmentation on Model Performance

To investigate the impact of synthetic data on model performance, two controlled experimental environments were evaluated: a baseline scenario in which models were trained and tested exclusively on the real (original) dataset of approximately 1200 records, and a synthetic data-augmentation scenario in which models were trained on a combination of the real (original 1200 records) and full synthetic datasets (3740 records) approximately 5000 records, while evaluation and testing was performed on the real data only.

To ensure validity and integrity, strict procedures were applied to prevent data leakage and duplication, including the removal of duplicate records within the real and synthetic datasets independently, deduplication between the two datasets before merging them to ensure no duplication of any real record in any synthetic sample, and the creation of the test set exclusively from the real data. This experimental design ensures that any observed performance improvements are attributable to the synthetic data-augmentation learning and not to memorization or data leakage.

The task involves a binary classification using "accept" and "deny" labels, where "accept" represents the positive class and this is our target to find healthy to allow and gain access to the network. In the context of "zero trust," "accept" refers to compliant and secure clients, and all evaluation metrics are calculated accordingly to prioritize the accurate identification of legitimate devices.

The first experiment assesses the model's performance when it is trained and tested only on real data. As shown in the first results table 1, most models perform strongly, with

ensemble-based methods such as random forest, gradient boosting, and XG Boost showing particularly high accuracy and F1 scores. However, despite these strong baseline results, the limited size of the real dataset constrains the models' ability to generalize, especially for more complex resolution boundaries, thus encouraging the introduction of synthetic data as a controlled form of data augmentation.

In this experiment, the training data was augmented with 3740 synthetic samples while the test set remained entirely real (only from the original dataset). The results table 4.10 below shows that the incorporation of synthetic data leads to consistent or improved performance across most models. Tree-based and ensemble methods showed improved accuracy, F1 scores indicating better generalization, and more stable cross-validation accuracy indicating reduced variance and increased robustness during training. Only slight degradation occurred for simpler models such as logistic regression, meaning that the synthetic data does not introduce harmful bias. Overall, these results confirm that properly generated and purified synthetic data can effectively enrich the training distribution without contaminating the evaluation process.

Table 4.10 Performance comparison of machine learning models trained on combined real and synthetic data and evaluated on real-only test data.

Model	Accuracy	CV Accuracy (mean)	Precision	Recall	F1 Score	PR-AUC (AP %)	False Positive Rate	Training Time (s)	Inference Time (ms/sample)	Test Set Source
Logistic Regression	86.48	86.44	84.12	94.69	89.09	91.87	25.00	0.0430	0.0131	real-only
SVM	86.20	88.74	84.65	93.24	88.74	94.93	23.65	0.1198	0.0342	real-only
KNN	88.17	88.14	85.41	96.14	90.45	92.87	22.27	0.0110	0.0424	real-only
Decision Tree	98.87	98.18	99.96	98.07	99.02	99.19	0.42	0.0482	0.0059	real-only
Random Forest	96.62	97.70	95.77	98.55	97.14	99.82	6.08	0.7347	0.0455	real-only
Gradient Boosting	97.46	98.91	97.60	98.07	97.83	99.90	3.38	1.7770	0.0065	real-only
XG Boost	96.90	97.82	97.12	97.58	97.35	99.44	4.05	0.2026	0.0344	real-only

In addition to accuracy-based metrics, the Precision-Recall Area Under the Curve (PR-AUC) was used to provide a more informative assessment, particularly under the possibility of class imbalances. It measures the trade-off between precision and recall across different decision thresholds with an explicit focus on the positive (accept) class, making it particularly suitable for security-related decision systems where false acceptance

of dangerous clients can have serious consequences. In both experimental settings, the ensemble models showed high PR-AUC values, indicating strong discrimination ability. In the scenario augmented with synthetic data, these values were generally maintained or improved, demonstrating that synthetic data enhances the models' ability to identify acceptable clients without increasing false positives. Thus, it confirms that synthetic data not only improves threshold-based metrics but also improves ranking-based performance.

In the figure 4.20 below, the ensemble models outperform others after synthetic augmentation, achieving higher accuracy, F1, CV stability, and PR-AUC compared to simpler classifiers in general.

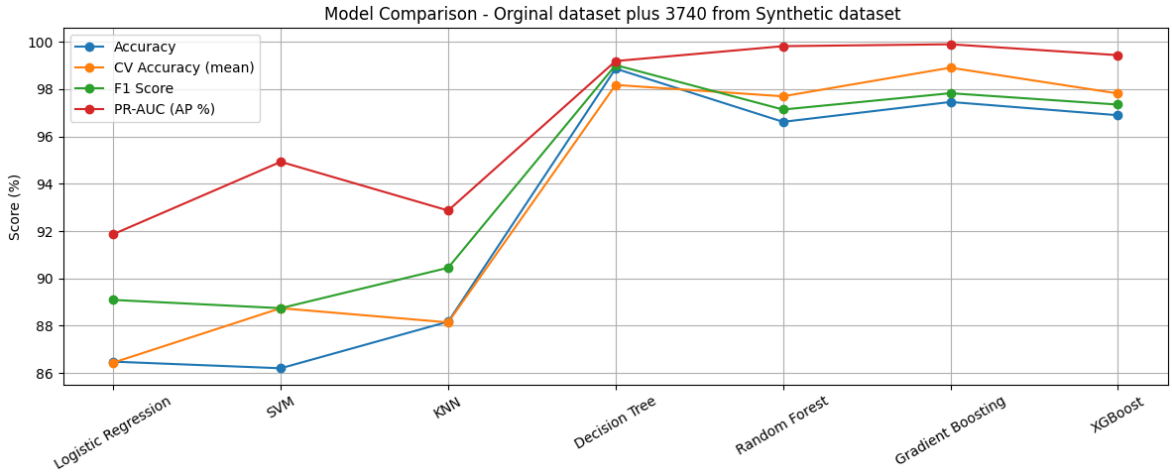


Figure 4.20 Performance comparison of machine learning models trained on real data augmented with 3,740 synthetic samples across multiple evaluation metrics, and the test was only on the real dataset

The following charts present a comparative analysis of machine learning model performance under two experimental approaches. The first approach trains and tests models using the original dataset, while the second approach trains models on a combined dataset of original and synthetic data, then evaluates them on real-world data. The charts illustrate differences in accuracy, cross-validation accuracy, F1 metric, and false positive rate.

In figure 4.21 the accuracy comparison shows similar performance in both approaches, with slight improvements for Decision Tree and Random Forest.

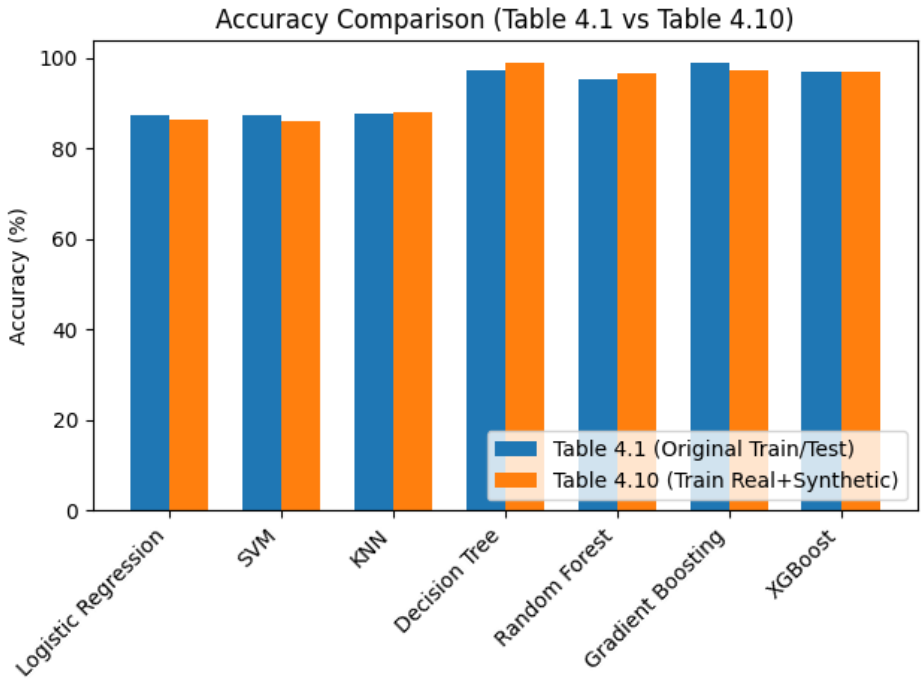


Figure 4.21 Accuracy comparison of machine learning models between the original dataset approach and the combined original–synthetic training approach.

In figure 4.22 the cross-validation accuracy improves significantly when training with combined original and synthetic datasets, indicating better model generalization.

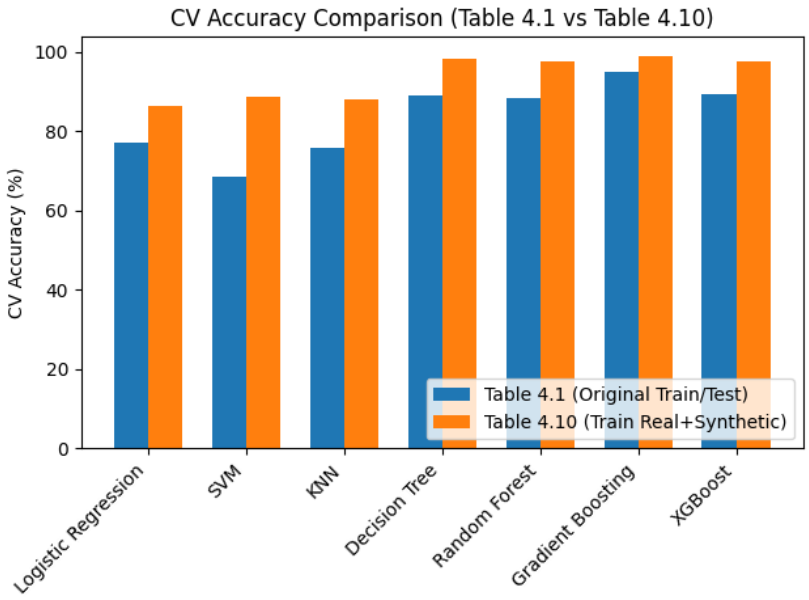


Figure 4.22 Cross-validation accuracy comparison of machine learning models using the original dataset and the combined original–synthetic training approach.

In figure 4.23 F1 score slightly improves for several models with synthetic training data, showing balanced precision and recall.

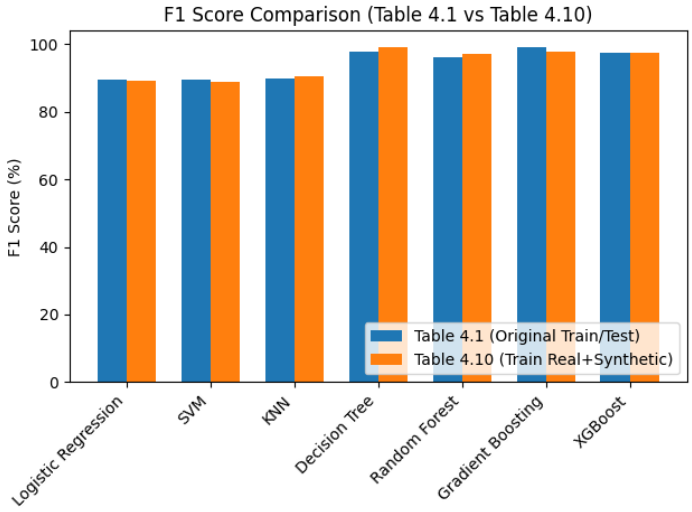


Figure 4.23 F1 score comparison of machine learning models between original dataset training and combined original–synthetic training approaches.

False positive rate varies across models; tree-based methods show lower rates, while some models increase when trained with synthetic data.

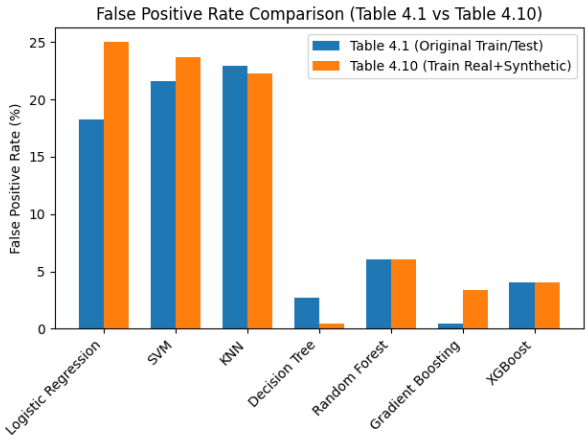


Figure 4.24 False Positive Rate (FPR) comparison of machine learning models between the original dataset approach and the combined original–synthetic training approach.

Chapter 5 Discussion

This chapter interprets the results of evaluating multiple machine learning models for device security classification in a Zero Trust Architecture (ZTA) environment, and within the context of the research objectives. The model families were compared (linear vs. tree vs. ensemble) in terms of classification performance, stability, and efficiency, and examined feature significance patterns. Then relate the results to the principles of zero trust and the research questions, identifying which models best support enforcement decisions, whether synthetic data improved predictions, and which device features are most indicative of risk. Finally, we consider the practical implications of deploying these models within the ZTA framework and address the limitations of the study with suggestions for future work. Throughout the discussion, technical precision and clarity are maintained, focusing on how these results contribute to secure access control in practice.

5.1 Addressing Research Objectives and Questions

The objective of this study was to identify the most suitable machine learning model(s) for classifying uncontrolled devices in a way that facilitates the implementation of a zero-trust policy (i.e., accurately determining whether to grant or deny access). Based on this study evaluation, ensemble tree-based models, such as Random Forest, Gradient Boosting, and XG Boost, are best suited to meet the needs of implementing the zero-trust policy, with a slight advantage for Gradient Boosting in the initial experiments and for Random Forest in later experiments using fully augmented data. These models consistently achieved the highest accuracy and highest F1 scores at all stages.

Importantly, these models produced the lowest number of false positives, as evidenced by their confusion matrices, indicating a strong ability to correctly deny access to dangerous devices in most augmented synthetic data scenarios, which is crucial for the zero-trust policy (to detect and block almost all unsafe devices). For example, Random Forest ultimately achieved approximately 99.6% accuracy with an approximately 99.61% recall score and a low false positive rate of approximately 0.42% in the final dataset phase. This means that it rarely misses a dangerous device attempting to connect, a crucial property for enforcement.

Random Forest achieved very strong performance across most evaluation metrics; it "rarely misses a dangerous device". In this study, the positive class is defined as Accept, and the False Positive Rate (FPR) is a more appropriate metric, as it measures the proportion of unsafe devices incorrectly classified as Accept. As shown in Table 4.10, the evaluated models demonstrate a consistently low FPR of 6.08% when tested exclusively on real (original) data, indicating that the likelihood of allowing an insecure device is minimal. This result supports the practical reliability of the model in Zero Trust enforcement scenarios, where false accept represents the critical risk.

Gradient boosting demonstrated similar strength, with a recall score of approximately 99.35%, higher precision (around 99.87%) and a lower false positive rate (FPR) of approximately 0.14%, indicating that it identifies dangerous devices with very few false alarms. In contrast, while logistic regression became highly accurate with increasing dataset size, its recall score, although high (98–99%), was slightly lower, and its overall decision threshold was less flexible in the face of complex circumstances (which could be a drawback as new threat patterns emerge). Therefore, to answer the question, our results suggest that ensemble models (particularly random forest and gradient boosting) are the most reliable options for classifying "zero trust" devices, combining excellent detection performance with robust behavior across data variations. These models will best support the zero-trust enforcement engine by providing reliable device risk assessments. However, all the tested models ultimately achieved high performance. Therefore, an organization with constraints might find a simpler model acceptable if properly configured, but the optimal choice for maximum security is to use an ensemble model. Furthermore, the stability of ensemble models (low variance and consistently high cross-validation scores) means they will remain effective even when using slightly variable data (which is crucial in a dynamic enterprise environment).

In addition, the experimental results provide a clear answer; enhancing the training set with synthetic data significantly improved the performance of our device security classification models. We observed consistent increases in accuracy, recall, and F1 score at every step of the synthetic augmentation process, compared to the baseline based only on the original data. Regarding the detection of the minority class (risky devices), the synthetic

data helped the models generalize better false positive rate improved with the addition of synthetic records, indicating that the classifiers detected a higher proportion of risky devices than before. For example, the logistic regression false positive rate jumped down from approximately 18.24% without synthetic augmentation to about 14.64% with the addition of 600 synthetic records, and then to about 3.95% with full augmentation (similarly, better false positive rates were observed for the SVM and KNN models).

Furthermore, cross-validation results demonstrated that synthetic data made performance more stable and reliable by mitigating over-allocation on the limited real-world dataset. There were no indications of performance degradation as a result of using synthetic data, a crucial point, as some might fear that synthetic data could introduce noise or unrealistic patterns. Instead, our carefully generated synthetic records (presumably designed to simulate potential hardware security scenarios) provided an additional signal for learning. It is also worth noting that we experimented with gradually increasing the synthetic data rather than adopting an all-in-one approach, allowing us to determine whether yield degradation had begun. We observed that after a certain point (around 600–900 synthetic samples), the improvements in some metrics for the most powerful models were less pronounced, suggesting that the models were approaching their maximum performance on this classification task. However, even in these instances, we did not see any decline, but rather a stabilization of performance, and the recent large data addition significantly improved the performance of the simpler models. This suggests that synthetic augmentation can be confidently used to boost the training of security classification models, especially when real-world data is limited or unbalanced. Our findings are consistent with other research in fields such as healthcare, where the generation and augmentation of synthetic data have improved the accuracy and robustness of models in classification tasks (Kannan, 2025).

In addition to above metric, this study focuses on the precision-recall Area Under the Curve (PR-AUC) as a key evaluation measure. PR-AUC is particularly relevant to this problem because the dataset is moderately unbalanced and the positive class is defined as "Accept" (healthy devices). PR-AUC assesses the trade-off between precision and recall across all thresholds, directly reflecting how well the model identifies safe devices without

allowing unsafe ones to appear. As shown in the results tested on real-world data only (Table 4.10), simpler models, such as logistic regression, achieved a PR-AUC of 91.87%, while SVM and KNN reached 94.93% and 92.87%, respectively. In contrast, tree-based and ensemble models demonstrated significantly stronger performance, with Decision Tree achieving 99.19%, Random Forest 99.82%, Gradient Boosting 99.90%, and XG Boost 99.44% on the PR-AUC. These results indicate that ensemble models maintain very high precision even at high recall, meaning that most accepted devices are indeed safe, while false acceptances remain minimal. From a zero-trust architecture perspective, this is crucial, as a high PR-AUC confirms the model's ability to reliably grant access to safe devices while maintaining strict control over potentially risky endpoints. Therefore, the PR-AUC provides a more informative and security-appropriate assessment than accuracy alone for evaluating device reliability ratings.

By analyzing feature importance, we identified four main categories of features that consistently predict device-level risk: (a) authentication-related features, (b) system usage patterns, (c) network activity, and (d) security configuration settings. These are the features that models primarily relied on to differentiate between "Accept" and "Deny" device profiles. In other words, authentication features (such as the number of failed login attempts) often indicate suspicious behavior. For example, devices/users with numerous failed login attempts were categorized as risky, which aligns with the idea that password guessing or credential theft attempts manifest as login failures.

System usage features (CPU load, memory, activity days, etc.) helped detect anomalies, such as a device performing tasks unusual for its role (which could indicate a breach). Network features (number of connections, external IP addresses) were crucial for identifying devices that might be leaking data or scanning networks, as a spike in unusual network activity is a serious indicator in monitoring for "zero trust." Finally, security configuration features (open ports, missing updates, disabled security tools) were strong indicators of poorly configured devices (e.g., numerous open ports exposing services, which are known to increase the attack surface).

The consistent ranking of these types of features as highly important across multiple ML algorithms confirms that they represent the most useful dimensions for assessing

device risk within the proposed model. Devices were more likely to be classified as "Deny" when multiple risk indicators appeared across these domains, reflecting a less secure situation rather than relying on a single, isolated metric. This finding directly supports the study's objective of identifying the device dimension features essential for accurate and transparent risk assessment. The results provide both empirical evidence and practical guidance, suggesting that organizations aiming to implement a data-driven "zero trust" principle should prioritize collecting and linking features from these four areas to achieve robust and defensible device risk assessments.

Data Leakage Prevention Measures: Stringent data leakage prevention measures were implemented at all stages of the dataset to ensure the integrity of model evaluation. All duplicate records were identified and removed at every stage of data preparation. The original (real) dataset was verified to be free of duplicates, while the synthetic dataset initially contained approximately 2,260 duplicate records, which were removed to ensure the uniqueness of each sample. Duplicate removal was also performed between the datasets before merging: any synthetic record identical to a real record was excluded to avoid overlap. Furthermore, the test set in some experiments was created exclusively from the original real data, without any synthetic samples, to ensure a clean evaluation partition. These precautions (duplicate removal and strict isolation of test data) ensured that the models were never evaluated on data they had already seen during training. In this way, any performance improvements observed with enhanced data can be attributed to genuine learning benefits, not to data leakage or unintentional memorization.

5.2 Implications for Real-World Zero Trust Implementation

The results point to several important practical implications of implementing a zero-trust architecture (ZTA), particularly regarding how machine learning models can be used to support access control decisions, and how these models are selected and deployed in an operational environment.

Model selection trade-offs (performance vs. speed vs. complexity): Since ensemble models have demonstrated the best predictive performance, it might be concluded that they are the optimal choice for a zero-trust system. In many cases, this is true when security is a top priority; the small gains in detection and recall accuracy resulting from using a robust

model like gradient boosting are worth the added complexity. However, there are trade-offs to consider:

Performance vs. Speed: In this study experiment, the inference speed of all models was within acceptable limits (milliseconds or less per device evaluation), meaning that even the ensemble model can make real-time access decisions. However, in environments with extremely low latency requirements (such as high-frequency trading networks or critical infrastructure control systems), the simplicity of a linear model may be attractive due to its absolute predictability and minimal overhead. Nevertheless, since random forest and XG Boost models also demonstrated inference times sub-millisecond, their deployment in most enterprise networks should not cause any noticeable delays in access checks. Training speed is also generally not a problem for deployment (models can be trained offline), but if the models require frequent retraining (such as daily updates with new data), the ensemble model will consume more computing resources. Organizations must ensure they have the infrastructure (or cloud computing resources) to retrain and service these models efficiently.

Performance vs. Interpretability: As mentioned earlier, linear and single-tree models offer greater Interpretability. In a real-world application of a zero-trust model, this can impact management trust and compliance. For example, if a system denies access to a device, the security team may need to explain why. In a linear model, a specific threshold for the feature can be indicated (e.g., "The device has logged too many failed login attempts, raising its risk score above the acceptable limit"). In a complex ensemble model, however, the rationale for the decision is more nuanced. This lack of transparency can hinder adoption, as security administrators may be hesitant to allow a black box model to make access decisions without understanding the underlying reasoning. Mitigating this requires the integration of explainable AI tools with the model, If interpretability is a priority, a middle ground approach can be taken, such as using a simpler model or a two-stage model (a set of models for initial evaluation, followed by a set of understandable rules for final decision-making). Alternatively, the model's complexity can be reduced (e.g., a random forest with fewer trees or less depth) at the cost of accuracy.

Integration and Complexity: Integrating ensemble models, especially those requiring more memory or specific libraries (such as XG Boost), into existing security infrastructure can be more challenging compared to, for example, a logistic regression model that can be programmed as an SQL query or simple script. However, modern security coordination platforms and AI services often support the deployment of complex models. The key is ensuring the model can interact with the policy engine. In a ZTA deployment, you typically have a policy decision point (PDP) that needs to consume risk signals from various sources. Our model will act as one of these sources, providing a device risk score or rating. Integration might involve feeding the model real-time device telemetry data (features) and then using the model's output (risk rating) as input to the policy decision point. For example, the National Institute of Standards and Technology (NIST) ZTA model envisions continuous device reliability verification; in our case, the model could provide a continuously updated reliability score for each device to the enforcement engine. Ensuring the output is availability at the right time (e.g., during access requests or periodically throughout a session) is crucial. Our results, which demonstrate minimal inference costs, indicate that this is feasible for every network transaction.

Integration with access control workflows: Classification models can be integrated into the "zero trust" workflow in several ways:

Pre-access risk check: When a device or user attempts to access a resource, the model can be called to assess the device's current security status. If it is rated as high risk ("Deny"), the access request can be blocked or routed for further verification. For example, if a device is rated as high risk, the policy might require additional authentication or isolate the device in a restricted part of the network. Our high-accuracy models ensure that these high-risk devices are identified most often before access is granted, supporting the "zero trust" principle of "never trust, always verify." This way can also be used for retrained ML models when the current security features are fully gathered without problems.

Continuous monitoring: The "zero trust" principle involves more than just a one-time gateway check; it includes ongoing trust assessment. Therefore, it's best to run our model periodically or when significant events occur. For example, even after a device has been granted access, if new telemetry data is received (such as an increase in network activity or

a series of failed login attempts), the model can reassess the device. If the result changes to "risk," the system can adapt by applying "zero trust" validation during the session. For example, requesting re-authentication or isolating the device. This dynamic use of model outputs is fundamental to ZTA's real-world applications and was a key driver in our model's design. The stable performance our model demonstrated in cross-validation indicates that it will behave consistently in such continuous deployments, without fluctuating significantly with minor data changes, which is crucial for avoiding oscillating trust decisions.

Orchestration and Automation: Once the model classifies a device as high-risk, automated actions (automated playbooks) can be triggered (via the Security Orchestration, Automation, and Response (SOAR) mechanisms). For example, a "Deny" classification can trigger actions such as blocking the device's IP address, initiating a software scan, or notifying an analyst. Our discussion of feature importance can also inform these actions: for instance, if open ports or an outdated operating system contribute to a high-risk device, the response might include requesting a software update or closing those ports. In summary, the model's output can feed into a broader process for enforcing a "zero trust" policy in real time.

Model deployment considerations: For real-word and practical use, where and how the model is deployed must be considered. Possible deployment architecture includes the following:

Server-centric: The model can run on a policy engine or controller that receives telemetry data from devices (endpoint detection and response systems (EDR) or security information and event management systems (SIEM), etc.). This server calculates risk scores and issues instructions to enforcement points (such as firewalls and software-defined perimeter gates) to allow or block traffic. The model's low latency, a single server can handle multiple devices in real time. This aligns with the common ZTA approach, where a central control center makes and disseminates decisions.

At the Edge: Alternatively, the model could be deployed on endpoint agents or network appliances (like NAC devices) to make decisions locally. It would be easier to

include a lightweight model like logistic regression in a small-footprint device, but an ensemble could also be embedded if resources permit. Edge deployment reduces latency further and allows scaling by distributing the computation. Our results show even a complex model could theoretically run on modern endpoint hardware (since milliseconds of CPU time are available).

Compliance with the Zero Trust Principle: It is encouraging that our model-based approach aligns with the core Zero Trust Principle. The Zero Trust Principle emphasizes the continuous verification of the device's security status.

Operational Considerations: With great power comes great responsibility, as using these models means that security operations teams rely on them. It is important to establish appropriate thresholds and monitor the model's outputs early on. For example, even with a 99% recall rate, some events may slip through, therefore, a multi-layered defense is necessary to detect them (such as anomaly detection systems or manual review of critical resources). Conversely, any false positives from the model should be measured in pilot phases to ensure that users and analysts are not misled. In our case, since the accuracy was extremely high (over 99% for the ensemble), false positives are expected to be very low, making the model's outputs readily actionable. However, it depends on the organization's risk tolerance. The advantage of having a probability or score (which our models can provide, rather than just a binary classification) is the ability to adjust the "Deny" threshold. For example, a threshold could be set so that 1% of devices are rejected, and then it could be verified whether this percentage reflects genuine problems. Over time, feedback can be used; if the model points to a problem, and it turns out to be a false alarm, this information can be used to adjust thresholds or retrain the model using the corrected classifications.

5.3 Limitations and Future Research

Although the evaluation demonstrates promising results, it is important to acknowledge the limitations of our study and explore future avenues for enhancing machine learning-based device classification in zero-trust environments.

5.3.1 Limitations:

Dataset size and diversity: Although we augmented the data with synthetic records (bringing the total to around 5,000 samples), the underlying real-world dataset was

relatively small (around 1,200 records) and derived from a specific environment. This raises questions about the generalizability of the results. The model may be over-fitted to the patterns present in the devices of that environment and the threat environment. Real-world enterprise networks can contain tens of thousands of devices with far more diverse behaviors. Furthermore, our binary classification (accept/deny) oversimplified the issue; in reality, device confidence can be multi-level or context-dependent. The limited size may also have contributed to some variability in the intermediate stages (as illustrated by the slight fluctuations in the metrics around 600–900 synthetic records). Future work should involve testing the models on larger and more diverse datasets, perhaps incorporating data from multiple organizations or using publicly available telemetry data (while respecting privacy) to ensure the model's scalability across diverse scenarios.

Realism of synthetic data: We assumed that synthetic data represents possible device behaviors. However, synthetic data may sometimes fail to capture the full complexity of real-world data relationships (Kannan, 2025). There is a risk that synthetic samples may introduce distortions or omit subtle correlations, potentially leading models to learn information that doesn't apply to real-world scenarios. For example, if synthetic records are generated by oversampling, they may closely resemble existing rare cases without adding any new information. If generated by a generative model, that model may fail to reproduce the behavior of rare anomalies. This limitation means that the performance improvements we achieved, while real in our testing, may not translate accurately to an operational environment if the synthetic data is not entirely realistic. We did not observe any performance degradation, which is encouraging, but careful verification is needed. Furthermore, attackers could theoretically exploit any biases resulting from the synthetic data (although we did not see any obvious biases). Future research could explore more advanced techniques for generating synthetic data specifically designed for security data, as well as metrics for verifying the accuracy of synthetic data, to ensure synthetic augmentation doesn’t inadvertently skew the model.

Feature Set Limitations: While feature set is rich in security indicators, it may not encompass all possible predictive factors. Other features that significantly impact device risk may not be considered (e.g., user identity features, device physical location, network

micro-segmentation information, etc.). Additionally, some features in set may be indicators of each other or open to improvement. For example, the term "network activity" is used broadly, distinguishing between benign and malicious traffic may require deeper network features (e.g., intrusion detection alerts or DNS request patterns). The heavy reliance on key features means that if these features are manipulated or misreported, the model could be fooled. For example, an attacker might try to evade detection by keeping their malware's CPU usage low or by limiting failed login attempts to remain within acceptable limits. This is an inherent limitation of using these features. In practice, data sources should be fortified. For example, ensuring that device telemetry data cannot be easily falsified, and this model should be combined with other models (such as network anomaly detectors) so that an attacker has a greater difficulty in evading all layers of security.

The assumption of constancy, model training implicitly assumes that past data (both real and synthetic) represent future conditions. In cybersecurity, this is often incorrect; threats evolve, and what is a strong indicator today may be common (and malicious) tomorrow, or vice versa. Our evaluation did not explicitly test the model against changing concepts or new types of attacks not included in the training data. For example, if a new exploit emerges that doesn't explicitly utilize existing features, the model might not detect it. This limitation is common in supervised learning for security purposes; the model's quality depends on the scenarios it has seen. Future research should consider online learning or periodic retraining as new data becomes available to keep the model up-to-date. Additionally, incorporating unsupervised anomaly detection may help identify new patterns that the supervised model misses.

Lack of field testing: Our results are based on offline experiments. We did not integrate the model into a real network to monitor real-time performance and its impact on users. Operationally, challenges we have not yet identified may arise, such as data flow issues, delays in model inference under stress, or changes in user behavior in response to the model (if users notice their devices are blocked, they may behave differently). False alarms, even if rare, may also have a human impact (blocking a working device can disrupt operations). We did not measure the "cost" of false alarms in practice. A limited pilot

deployment would be invaluable for uncovering any issues not apparent in laboratory tests. This is a recommended future step before wider rollout.

5.3.2 Future research directions:

Based on this work, several paths can be pursued:

Improving Model Sophistication: Despite the excellent performance of tree ensembles, deep learning models (such as neural networks that receive feature vectors) can be explored. Deep learning may be able to detect interactions that even an ensemble misses, especially with more data available. However, this requires fine-tuning and presents interpretive challenges. Also, increase the raw feature set to include new features related to security and feature engineering to derive new features for a long time. Moreover, test the new risk on the diversity of the dataset.

Deployment in a Real-World Environment and a Feedback Loop: Finally, deploying the model in a zero-trust production environment (even initially in a pilot setting) will provide valuable feedback. Model decisions can be logged alongside actual security results to measure the model's accuracy/recall in real-world scenarios (e.g., do the model's "Deny" decisions correspond to blocked attacks or simply false alarms?). This, in turn, contributes to iterative improvements. It would also be beneficial to examine human factors; how do administrators interact with a machine learning-based policy engine? Developing dashboards or interfaces that explain the model's logic (highlighting the key factors contributing to each decision) could be an area for development, ensuring the model's adoption by IT/security personnel. Integrating our model with existing zero-trust platforms (e.g., with identity management and SOAR workflows) in a case study will demonstrate its practicality and reveal any integration challenges.

References

Abdelmagid, A. M., & Diaz, R. (2025). Zero trust architecture as a risk countermeasure in small–medium enterprises and advanced technology systems. Risk Analysis.

Allman, M., & Ostermann, S. (1999). FTP security considerations (RFC 2577). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc2577

Anisetti, M., Ardagna, C., Cremonini, M., Damiani, E., Sessa, J., & Costa, L. (2020). Security threat landscape. White paper.

Başer, M., Güven, E. Y., & Aydın, M. A. (2021). SSH and Telnet protocols attack analysis using honeypot technique. In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 806–811). IEEE.

Bermudez Villalva, D. A., Onaolapo, J., Stringhini, G., & Musolesi, M. (2018). Under and over the surface: A comparison of the use of leaked account credentials in the dark and surface web. Crime Science, 7(1), 1–11.

Cisco. (n.d.). What is network access control (NAC)? https://www.cisco.com/site/us/en/learn/topics/security/what-is-network-access-control-nac.html

CloudNuro. (2025, May 14). Top 10 network access control (NAC) solutions for zero trust implementation. https://www.cloudnuro.ai/blog/top-10-network-access-control-nac-solutions-for-zero-trust-implementation

Couronné, R., Probst, P., & Boulesteix, A. L. (2018). Random forest versus logistic regression: A large-scale benchmark experiment. BMC Bioinformatics, 19(1), 270.

Defense Information Systems Agency, & National Security Agency. (2022). Zero trust reference architecture (Version 2.0). U.S. Department of Defense.

Detken, K. O., Jahnke, M., Kleiner, C., & Rohde, M. (2017). Combining network access control (NAC) and SIEM functionality based on open source. In IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems (pp. 300–305).

Edo, O. C., Ang, D., Billakota, P., & Ho, J. C. (2024). A zero trust architecture for health information systems. Health and Technology, 14(1), 189–199.

EC-Council. (n.d.). Certified SOC analyst (CSA). https://www.eccouncil.org/train-certify/certified-soc-analyst-csa/

Gambo, M. L., & Almulhem, A. (2025). Zero trust architecture: A systematic literature review. arXiv. https://arxiv.org/abs/2503.11659

García-Teodoro, P., Camacho, J., Maciá-Fernández, G., Gómez-Hernández, J. A., & López-Marín, V. J. (2022). A novel zero-trust network access control scheme based on the security profile of devices and users. Computer Networks, 212, 109068.

Gudala, L., Shaik, M., & Venkataramanan, S. (2021). Leveraging machine learning for enhanced threat detection and response in zero trust security frameworks. Journal of Artificial Intelligence Research, 1(2), 19–45.

Gu, J. (2020). An effective intrusion detection model based on PLS-logistic regression with feature augmentation. In China Cyber Security Annual Conference (pp. 133–140). Springer.

Hesham, M., Essam, M., Bahaa, M., Mohamed, A., Gomaa, M., Hany, M., & Elsersy, W. (2024). Evaluating predictive models in cybersecurity. In Intelligent Methods, Systems, and Applications (IMSA) (pp. 33–38). IEEE.

Hitachi Vantara Federal. (2024). Enhancing network security with XGBoost: A deep dive into intrusion detection.

Hoque, M. M., Ahmad, I., Suomalainen, J., Dini, P., & Tahir, M. (2024). On resource consumption of machine learning in communications security. Authorea Preprints.

Johnson, M. (2025). Indicators of compromise (IOCs): How to identify & manage cybersecurity threats. Guardian Digital.

Kannan, M., Umamaheswari, D., Manimekala, B., Mary, I. P. S., Savitha, P. M., & Rozario, J. (2025). An enhancement of machine learning model performance in disease prediction with synthetic data generation. Scientific Reports, 15(1), 33482.

Kaur, T., Wason, K., Aggarwal, M., Sharma, L., Duggal, P., & Gautam, S. (2025). Mitigating the risk of lateral movement within a network. In Zero-Trust Learning (pp. 271–287). Apple Academic Press.

Kavalanekar, S., Worthington, B., Zhang, Q., & Sharda, V. (2008). Characterization of storage workload traces from production Windows servers. In IEEE International Symposium on Workload Characterization (pp. 119–128).

Koli, L., Kalra, S., Thakur, R., Saifi, A., & Singh, K. (2025). AI-driven IRM: Transforming insider risk management with adaptive scoring and LLM-based threat detection. arXiv.

Laghari, A. A., Khan, A. A., Ksibi, A., Hajjej, F., Kryvinska, N., Almadhor, A., Mohamed, M. A., & Alsubai, S. (2025). AI-enabled zero trust intrusion detection in industrial IoT architecture. Scientific Reports, 15(1), 26843.

Li, S., Iqbal, M., & Saxena, N. (2024). Future industry Internet of Things with zero-trust security. Information Systems Frontiers, 26(5), 1653–1666.

Mazhar, N., Salleh, R., Zeeshan, M., & Hameed, M. M. (2021). Role of device identification and manufacturer usage description in IoT security: A survey. IEEE Access, 9, 41757–41786.

Microsoft. (2022). How to improve risk management using zero trust architecture.

MITRE. (n.d.). T1055: Process injection. https://attack.mitre.org/techniques/T1055/

MITRE. (n.d.). T1055.015: Reflective code loading. https://attack.mitre.org/techniques/T1055/015/

MITRE. (n.d.). T1068: Exploitation for privilege escalation. https://attack.mitre.org/techniques/T1068/

MITRE. (n.d.). T1071: Application layer protocol. https://attack.mitre.org/techniques/T1071/

Moe, M. R., & Nerhagen, M. (2022). Windows security baselines. Norwegian University of Science and Technology.

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

Morris, J., Becker, I., & Parkin, S. (2020). An analysis of perceptions and support for Windows 10 Home Edition update features. Journal of Cybersecurity, 6(1).

National Institute of Standards and Technology. (2020). Zero trust architecture (NIST Special Publication 800-207). https://doi.org/10.6028/NIST.SP.800-207

National Institute of Standards and Technology. (n.d.). National Vulnerability Database. https://nvd.nist.gov/

National Institute of Standards and Technology. (n.d.). CVSS scoring explained. https://nvd.nist.gov/vuln-metrics/cvss

Ojha, N., & Vaish, A. (2025). Why perimeter security is no longer enough. In Zero-Trust Learning (pp. 305–328). Apple Academic Press.

Outchakoucht, A., Hamza, E. S., & Leroy, J. P. (2017). Dynamic access control policy based on blockchain and machine learning for IoT. International Journal of Advanced Computer Science and Applications.

OWASP Foundation. (n.d.). Session management cheat sheet.

Portnox. (n.d.). Decoding the paradigm of zero trust endpoint protection.

Ramezanpour, K., & Jagannath, J. (2021). Intelligent zero trust architecture for 5G/6G networks. arXiv.

Revathi, S., & Malathi, A. (2013). Analysis on NSL-KDD dataset using machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology.

Serrao, G. J. (2010). Network access control (NAC): An open source analysis. In IEEE International Carnahan Conference on Security Technology.

Sitapura, J. (2022). Backdoor and login brute force. Rajasthan Technical University.

Thummapudi, K., Lama, P., & Boppana, R. V. (2023). Detection of ransomware attacks using processor and disk usage data. IEEE Access, 11, 51395–51407.

Trend Micro. (n.d.). Intrusion prevention event reference.

TrustCloud. (2025). Zero trust architecture: Engineering a security model.

Ullah, I. (2016). Detecting lateral movement attacks through SMB using Bro.

United States Department of Defense et al. (2022). Weak security controls and practices routinely exploited for initial access.

Vigna, G., & Kemmerer, R. A. (1998). NetSTAT: A network-based intrusion detection approach. In Annual Computer Security Applications Conference.

Vokorokos, L., Baláž, A., & Ádám, N. (2015). Secure web server system resources utilization. Acta Polytechnica Hungarica.

Walkowski, M., Oko, J., & Sujecki, S. (2021). Vulnerability management models using CVSS. Applied Sciences, 11(18), 8735.

Wang, J., Huang, Y., Jin, S., Schober, R., You, X., & Zhao, C. (2018). Resource management for device-to-device communication: A physical layer security perspective. IEEE Journal on Selected Areas in Communications, 36(4), 946–960.

Wang, R., Li, C., Zhang, K., & Tu, B. (2025). Zero-trust based dynamic access control for cloud computing. Cybersecurity, 8(1).

Wang, S., Jiang, R., Wang, Z., & Zhou, Y. (2024). Deep learning-based anomaly detection and log analysis for computer networks. arXiv.

Waterson, D. (2020). Managing endpoints, the weakest link in the security chain. Network Security, 2020(8), 9–13.

Wazid, M., Das, A. K., Chamola, V., & Park, Y. (2022). Uniting cybersecurity and machine learning: Advantages and challenges. ICT Express, 8(3), 313–321.

Wazuh. (n.d.). Wazuh documentation. https://documentation.wazuh.com/

Yunanto, W., & Pao, H. K. (2022). User behaviour risk evaluation in zero trust architecture environment. In IEEE World Forum on Internet of Things (WF-IoT).

استخدام التعلم الآلي للكشف عن أمان صحة عميل الشبكة في بنية الثقة الصفرية

منتصر عيسى محمد طنينه

د. حذيفة الاشقر

د. محمد حمارشة

د. نائل ابو الحلاوة

الملخص

تواجه المؤسسات الحديثة تحديات متزايدة في مجال الأمن السيبراني نتيجة التطور السريع في تقنيات العمل عن بُعد، والاعتماد المتزايد على الحوسبة السحابية، وانتشار ممارسات إحضار الجهاز الشخصي للعمل(BYOD)، مما أدى إلى توسّع سطح الهجوم بشكل كبير. في هذا السياق، برزت بنية الثقة الصفرية (ZTA) كنموذج أمني حديث يعتمد مبدأ «عدم الثقة المسبقة والتحقق المستمر» لكل مستخدم وجهاز. ومع ذلك، لا تزال هناك فجوة بحثية واضحة تتعلق بكيفية تقييم الحالة الأمنية للأجهزة الطرفية بشكل ديناميكي ودمجها بفعالية ضمن سياسات الثقة الصفرية.

تهدف هذه الرسالة إلى اقتراح إطار عمل قائم على تقنيات تعلم الآلة لتقييم الصحة الأمنية للأجهزة الطرفية بشكل مستمر، ودمج هذا التقييم ضمن محركات اتخاذ القرار في بيئات الثقة الصفرية. يعتمد الإطار المقترح على جمع بيانات متعددة المصادر من الأجهزة، تشمل حالة التحديثات الأمنية، وضع برامج الحماية، مؤشرات الثغرات الأمنية، وسلوكيات النظام واستهلاك الموارد. تم تطوير مقياس كمي لمخاطر الجهاز يجمع بين احتمالية التعرض للهجوم وتأثيره المحتمل، ويُستخدم هذا المقياس في عملية توصيف البيانات ودعم سياسات التحكم في الوصول.

نظرًا لمحدودية البيانات الواقعية، تم استخدام تقنيات توليد البيانات الاصطناعية، مثل نماذج توليد البيانات، لمعالجة عدم توازن الفئات وزيادة تنوع البيانات مع الحفاظ على خصائصها الأمنية. بعد

ذلك، تم تدريب وتقييم مجموعة من نماذج تعلم الآلة، بما في ذلك نماذج الأشجار، وخوارزميات التجميع، والنماذج المعتمدة على التعلم الإشرافي، لتصنيف الأجهزة إلى أجهزة آمنة (قبول) أو غير آمنة(رفض).

أظهرت النتائج أن دمج تقييم صحة الجهاز القائم على تعلم الآلة ضمن بنية الثقة الصفرية يؤدي إلى تحسين دقة قرارات التحكم في الوصول وتقليل المخاطر الأمنية. كما أثبتت الدراسة أن استخدام مؤشرات أمنية مدروسة ومقاييس مخاطر كمية يسهم في تعزيز القرارات التكيفية وفي تقليل الاعتماد على السياسات الثابتة. تقدم هذه الرسالة مساهمة علمية وتطبيقية في مجال الأمن السيبراني من خلال توفير إطار عملي يمكن اعتماده في المؤسسات لتعزيز أمن الشبكات ودعم تطبيقات الثقة الصفرية الذكية.

الكلمات المفتاحية: فحص صحة امان الجهاز، بنية الثقة الصفرية، تعلم الآلة، تقييم المخاطر، التحكم في الوصول إلى الشبكة.