A Zero-Trust Framework Based on Machine Learning for Industrial Internet of Things

Adel Atieh
Faculty of Engineering and IT
University of Technology, Sydney (UTS)
Sydney, Australia
adel.atieh@student.uts.edu.au

Priyadarsi Nanda
Faculty of Engineering and IT
University of Technology, Sydney (UTS)
Sydney, Australia
priyadarsi.nanda@uts.edu.au

Manoranjan Mohanty
Carnegie Mellon University
Qatar
mmohanty@andrew.cmu.edu

***Abstract*— Controlling access to data is essential in ensuring data is only accessed by authorised and trusted users. For these reasons, zero-trust frameworks have been in the centre of interest in the past few years. Zero-Trust frameworks assume that users and systems have been compromised and deal with them as untrusted entities that requires multiple levels of authorisation and security attributes to be compliant in order to be considered trusted. The most used zero trust frameworks use static thresholds to grant levels of access to systems which could introduce false positives and incorrect access privileges to systems/networks. This research paper proposes a machine-learning (ML)-based zero-trust framework that utilises an anomaly detection algorithm. The output of the anomalous detection would inform the observers the trustworthiness of systems in their environments. Moreover, performance, complexity and impact of our proposed scheme is compared against a static threshold zero-trust framework.**

Keywords*— *Zero-Trust, Industrial Control Systems (ICS), Industrial Internet of Things (IIoT), Internet of Things (IoT), Critical Infrastructures, Machine Learning, Anomaly Detection

I. INTRODUCTION

The Industrial IoT (IIoT) model started to be utilized widely across organizations to enhance monitoring and connectivity between organization assets [1]. IIoT involves the replacement of currently deployed Industrial Control Systems (ICS) environments with smarter and more interactive devices to communicate with next-generation IT systems. IIoT systems are deployed in critical infrastructures serving multiple sectors. In Australia and other Commonwealth governments, critical infrastructures are defined as [2]:

'Those physical facilities, supply chains, information technologies and communication networks which, if destroyed, degraded or rendered unavailable for an extended period, would significantly impact the social or economic wellbeing of the nation or affect Australia's ability to conduct national defence and ensure national security'.

A common security model used to secure critical infrastructure environments and systems is called Defence in Depth. The main assumption in this model is that entities contained within these zones are always trusted with respect to the devices in the same zones and in zones with lower security levels [3]. **Therefore, the use of security zoning can allow malicious actors to traverse defences and move laterally within the trusted zones easily using trusted systems in the environment.**

To assist in controlling and stopping sophisticated threats and attacks in the environment, zero-trust frameworks have been developed where any asset in an environment is assumed to be compromised. This assumption allows the continuous evaluation of attributes of users and systems and enforcement of appropriate controls. In our previous research [4], we focused on implementing a zero-trust framework that relied on static thresholds with different access level provided based on different security attributes and security events. However, this method can lead to multiple false positive enforcements due to the hardcoded thresholds that do not change and hence not consider important contextual information about the environment dynamics.

Therefore, this research paper proposes an ML-based zero-trust framework that rely on anomaly detection to determine the change of trust of systems. Specifically, the degree of anomaly relates proportionally to the degree of trust systems are given and hence their levels of access. This paper evaluates multiple anomaly detection algorithms such as Isolation Forest, k-nearest neighbours (KNN), Connectivity-based Outlier Factor (COF), Local Outlier Factor (LOF) on the TON_IoT network dataset produced by UNSW [5]. The best evaluated algorithm is then used within the proposed zero-trust framework to apply the right level of enforcements/access grants based on the resultant anomaly score.

The remaining of this paper is organised into six sections. Section II provides an overview on the concepts of trust and zero-trust frameworks and briefs on anomaly detection algorithms. Section III explains and discusses the existing issues with various trust evaluation models and existing proposed zero-trust frameworks along with our research questions. Section IV provides an in-depth explanation of our proposed evaluation approaches of the anomaly detection algorithms and the access levels the zero-trust framework will be enforced using these algorithms. Section V describes the experimental setup and implementation of the proposed ML-

based zero-trust framework and compares its performance and complexity to a standard static zero-trust framework setup. Section VI provides an analysis of the observed results and the impact of these results on IIoT environments. Finally, Section VII draws conclusions from the results and outcomes on the use of machine learning and anomaly detecting in zero-trust frameworks and discusses future work with potential improvements.

II. Literature Review

A. Trust

Trust has been defined as the belief in the reliability and honesty of the behaviour of another entity. Trust is formed using the following attributes [6,7]:

- Reputation: formed by intel generated by previous and historical interactions with other entities. This includes but not limited to case-studies, customer references in addition to historical bad and good events associated.
- Recommendations: forms an indirect trust relationship with an entity by a trusted third party.
- Sensor & behavioural data: this attribute is formed by collecting data from different assets in an environment. This includes but not limited to authentication information, network and system logs and device health information.

Using these attributes, trust can be defined as the following:

$$reputation \times recommendations \times behaviour = trust \quad (1)$$

A common illustration of Trust is the chaining of certificate issuers in a certificate given to a website issued by trusted a Certificate Authority (CA) [8].

B. Zero-Trust

The Zero Trust concept was first presented in 2004 by Jericho Forum [9] to the expanding utilisation of cloud computing along with the increase of mobility in IT. Zero Trust consists of a series of theories and concepts that aim to lower the likelihood of unnecessary access in the network. The zero-trust model treats all hosts as if they are in compromised networks [6]. It enforces continuous authentication and authorisation to access resources in the network. Zero-Trust have the following fundamental assumptions [10]:

- Threats exist at all times externally and internally with respect to the network.
- Location is not enough for determining the trust level of subjects.
- Every asset in the environment has to be authenticated and authorised with traffic flows between different assets encrypted at all times.
- Zero-Trust uses dynamic trust evaluation and calculation from using different data source to apply policies.

C. Anomaly Detection

An anomaly can be defined as:

- *"An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs" [11]*

Anomalies can be considered to be any of the following anomalies [12]:

- Point Anomaly: represent a single abnormal data instance from the rest of the data.
- Context Anomalies: anomalies only found in a specific scenario/use-cases.
- Collective Anomalies: revealed data points only considered to be anomalous in a collection and not otherwise

There are different types of anomaly detection classes broken down into the following types [12]:

- Supervised anomaly detection: based on supervised machine learning techniques that rely on training labelled data to classify and determine anomalies.
- Unsupervised anomaly detection: based on unsupervised machine learning algorithms, data does not rely on training data.
- Semi-supervised: based on semi-supervised techniques, requires labels only for a normal class data.

The existence of anomalies in data drove the use of anomalous detection in analytics for applications such as detecting unauthorised access [13].

III. Current State of Research & Research Questions

Zero-Trust is still a new and evolving concept that's still being explored and examined by multiple organisations and security vendors around the world. There are still new models and frameworks that are being developed that follow the zero-trust principles. Authors in [14] also proposed a dynamic trust model that utilises fuzzy logic to derive the trust value. In addition, one of the early contributors of the zero trust architectures proposed in the industry is the Google Zero Trust approach called BeyondCorp for enterprise environments [8]. In addition, NASA proposed a zero-trust architecture that could suit their environments and facilities [15].

Our past paper [4] focused on implementing a zero-trust framework that relied on static thresholds with different access levels provided based on different security attributes and security events. From a security perspective, the zero-trust framework we proposed provides the capability required to respond to advanced threats in IIoT environments. However, as the trust is defined solely based on static defined attributes always be a chance of false positives occurring. Also, a disadvantage in the proposed design was the long verification process required to authorise the IIoT device traffic to the cloud IIoT server. In order to enhance the

accuracy and simplicity of the proposed zero-trust framework, **Machine Learning will be used to enhance our zero-trust framework. Particularly, Machine Learning will be used to perform anomalous detection on the selected labelled network dataset. The anomalous detection scoring will be used to determine the evaluated trust level of devices in the network. Therefore, this paper covers the following research questions:**

1. How can anomaly detection algorithms be used to evaluate trust in zero-trust frameworks?
2. Can machine learning improve the performance accuracy of Zero-Trust frameworks for IIoT environments?
3. Does the use of ML-based zero-trust frameworks reduce the operational impact on IIoT systems compared to static threshold zero-trust frameworks?

IV. Proposed Machine Learning Evaluation for Zero Trust Framework

A. Machine Learning algorithms

In order to implement machine-learning-based zero-trust framework, we have evaluated the following algorithms to choose the highest accuracy and using the TON_IoT network dataset:

- Isolation Forest
- KNN
- COF
- LOF

a) Isolation Forest

Isolation forest is an algorithm that utilises the use of binary tree to determine anomalies and normal data. The algorithm constructs a binary search tree for every N random sample of size of M. The anomalous score is calculated from the anomaly score it inserts into each of the binary search trees. The mean insertion depth is used to drive the anomaly score [16]. Isolation Forest has been used in various ways in research such as in Paper [17] where it was used to detect deviation and anomalies of employee behaviours. Authors in [18] implemented an isolation forest algorithm for streaming data using a sliding window. Isolation forest consists of two phases:

- The training phase involves the construction of the forest of random trees.
- The second phase is the anomaly scoring phase of the observations using the constructed tree.

The anomaly score $s(x)$ of an observation x is computed using the isolation forest algorithm by normalising the path length $h(x)$:

$$s(x) = 2^{-\frac{E[h(x)]}{c(n)}} \qquad (2)$$

Where:

- $s(x)$: an anomaly is indicated a score value close to 1 where a score value between 0.5 and 0 indicates a normal observation.
- $E[h(x)]$: isolation trees average path length in the isolation forest
- $c(x)$: unsuccessful searches' average path length in a binary search tree of x observations

b) LOF

Breunig et al. [19] introduced the local outlier factor (LOF) algorithm, which assigns an anomaly score to each data instance. This score represents the ratio of the average local density of the k nearest neighbours of the instance to the local density of the instance itself. To compute the local density of a data instance, the authors determine the radius of the smallest hyper-sphere centred at the instance that encompasses its k nearest neighbours. The local density is then calculated by dividing k by the volume of this hyper-sphere. LOF aims to assign a level of outlierliness to each data point in a multidimensional dataset. The local outlier factor of a data point x is computed as the ratio of its local density to that of its k-nearest neighbours. That is, the local outlier factor of observation x is:

$$LOF_k(x) = \frac{1}{|N_k(x)|} \sum_{o \in N_k(x)} \frac{lrd_k(o)}{lrd_k(x)} \qquad (3)$$

Where:

- $LOF_k(x)$: the local reachability density of an observation x
- $N_k(x)$: represents the k-nearest neighbors of observation x.
- $|N_k(x)|$: is the number of observations in $N_k(x)$.

Normal observations are indicated by $LOF_k(x)$ values that less than or close to 1 whereas anomalies are indicated with $LOF_k(x)$ values greater than 1.

c) KNN

The kNN is a general non-parametric supervised learning technique utilized in classification tasks. It involves predicting the class or label of a new observation by examining the labels of its closest k neighbours within the training dataset [20]. In essence, the one-class kNN rule operates on the premise that similar observations tend to have close neighbours in the training data, while dissimilar ones (anomalies or outliers) are distant from their nearest neighbours. Thus, anomalies exhibit significantly greater distances to their nearest neighbouring training samples compared to normal observations. There are different methods used to calculate distance between observations but the one that was used in this research is Euclidean distance which measures a straight distance between different points. The normalised Euclidean distance is calculated by the following formula:

$$d(A,B) = \sqrt{\frac{\sum_{i=1}^{m}(x_i - y_i)^2}{m}} \quad (4)$$

Where:

- A: feature space consisting of $(x_1, x_2, \dots, x_m)$
- B: feature space consisting of $(y_1, y_2, \dots, y_m)$
- m: is the size of the of the feature space

Normalised Euclidean distance values less than or equal to 1 are classified as anomalies whereas Normalised Euclidean distance values close to 0 are classified as normal.

d) COF

Tang et al. [20] introduced a variant of the LOF method, termed Connectivity-based Outlier Factor (COF). The primary distinction between LOF and COF lies in the approach used to determine the k neighbourhood for each data instance. In COF, the neighbourhood for a given instance is calculated incrementally. Initially, the nearest instance to the given one is included in the neighbourhood set. Subsequently, additional instances are added to the neighbourhood set based on their minimal distance to the existing set among all remaining data instances. The distance between an instance and a set of instances is defined as the minimum distance between the given instance and any instance within the set. This incremental growth process continues until the neighbourhood reaches the size of k. Once the neighbourhood is established, the anomaly score is computed using the same method as LOF. The COF algorithm computes the connectivity-based outlier factor for observations through comparing the neighbouring observations with observation x that is subject to outlier scoring. This is shown in the following formula computing connectivity-based outlier factor at observation x with respect to its k-neighbourhood $COF_k(x)$ [21]:

$$COF_k(x) = \frac{ac - dist_{N_k(x)\cup x}(x)}{\frac{1}{k}\sum_{o\in N_k(x)} ac - dist_{N_k(o)\cup o}(o)} \quad (5)$$

Where $ac - dist_{N_k(x)\cup x}(x)$: average chaining distance between observations x_k and $N_k(x)$ is the set of k nearest neighbours (k-NN).

B. Trust Access Grant Levels

Systems in IT and OT environments are installed and connected to the network using different authentication protocols. Similarly, IIoT devices are installed and connected to the network sending and receving data as part of their functionality. Devices connected to the network are assumed and operating as fully trusted systems. In order to control traffic and access of these IIoT systems, our proposed zero-trust framework consists of the following access grants similarly described in our past paper [4]:

- **Full Access grant**: authorised access to resources in the environment with full permissions based on the IIoT client role.
- **Partial Access grant**: IIoT devices with partial access will report their metrics and status to the IIoT platform but not be able to interact with the environment fully.
- **Limited Access grant**: IIoT devices with limited access will not report their status and metrics to the IIoT platform nor interact with the environment. It will only be accessible by central management.

These access grant levels have been evaluated using static thresholds in our previous paper [4]. However, the use of static thresholds defined in detection rules/conditions have been observed to be ineffective and/or inaccurate in various applications in different industries. Static thresholds are built to monitor enormous number of vectors that can contain false positives which can lead to threat alert fatigue [22]. Cisco produced a report [23] which stats that around 44% of triggered alerts are ignore by security operators as they receive a lot of alerts that required analysis.

With this in mind, the approach for measuring access grant levels have been shifted to rely on detected anomalous behavior of IIoT systems. The more anomalous behavior is observed from an IIoT system, the less trusted the IIoT system is. This is because IIoT systems are design and deployed to be performing certain and defined static actions. Hence, anomalous behaviour in such environments is considered to be an indicator of compromise (IoC). The use of ML-based anomalous detection will dynamically score the anomalous behavior observed from IIoT network traffic. The anomaly scores are normalised to reflect the numeric range from 0 to 1 where 1 is an extremely high anomaly and scores close to 0 are defined as normal/insignificant unusual behaviour. The evaluated trust score will reflect the access grant using the following score ranges to apply the access grant level. This approach creates an access level for these IIoT devices relatable to their trust levels based on the potential of them being compromised.

$$T_{Entity}(x) = \left|T_{Entity}(x-1) - s(x)\right| \quad (6)$$

- $T_{Entity}(x)$: The evaluated trust value of the IIoT entity based on the current observation x
- $T_{Entity}(x-1)$: The evaluated trust value of the IIoT entity based on the previous observation $x-1$
- $s(x)$: anomaly score $s(x)$ of an observation x from the IIoT entity

Table 1 below describes the access grant levels and their associated anomaly scores.

TABLE I. ANOMALY SCORE RANGES PER ACCESS GRANT LEVEL

Anomaly Score	Anomaly Level	Access Grant
0.67 – 1	High	Limited
0.34 – 0.66	Medium	Partial
0 – 0.33	Low	Full

Figure 1 shows a slop diagram visualising the relationship between anomaly score $s(x)$ and the trust level $T_{Entity}(x)$ where the higher the anomaly score the lower the trust is limiting access and vice versa.

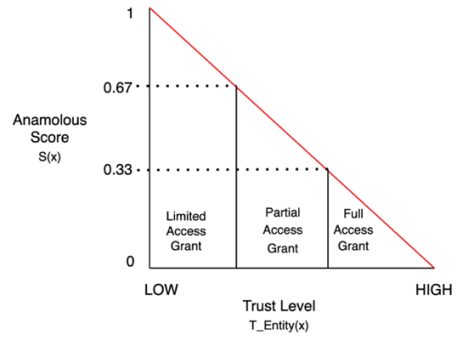

Fig. 1. Slope of Trust vs Anomaly Score with access grant levels

V. Experimental Implementation

The main objective of this experimental implementation is to **measure and compare the performance, complexity and impact between the proposed ML-based zero-trust solution and the static zero-trust solution in our previous paper [4]**. With this in mind, the ML classification algorithms mentioned in the previous section were evaluated to choose the best algorithm to be used in the **ML-based zero-trust solution based on the chosen dataset.** The network dataset used in this research was extracted from one of the UNSW datasets labelled as 'TON_IoT'. The dataset consists of virtual IoT devices that have been deployed in a virtual lab in which device and network activities are gathered. The traffic records contained in this dataset have been labelled to denote normal and attack instances, where '0' signifies normal behavior and '1' indicates attacks. [5]. The experimental implementation of the solution consists of:

- Data Preparation
- Anomaly Detection Evaluation
- Zero-Trust Experimental Implementation
- Zero-Trust Framework Evaluation

A. Data Preparation

Before starting the evaluation steps of this experimental implementation, we defined the scenario in which we were interested to filter the data according and make it relevant to our zero-trust framework. The scenario is focused on IIoT devices being the target of attempted security attacks in the network on different ports and protocols. This is relevant to our framework as the more a device is targeted in IIoT network, the higher the likelihood this device could get compromised in some shape or form. Therefore, we focused on filtering the 'TON_IoT' network dataset associated with the virtual IoT devices where they were the target through filtering the dataset with their destination IP addresses. We then sampled the data to help us implement anomaly detection on 15 minutes samples. With this approach, we had the following dataset associated with 5 IIoT devices shown in Table 2. This dataset will be used the evaluation of the anomaly detection algorithm and the input of the ML-based zero-trust framework.

TABLE II. AUC Value Measurements per Algorithm

IIoT Devices	No. Records	No. Normal Records	No. Attack Records
Dev 1	2612	2599	13
Dev 2	249	242	7
Dev 3	2609	2598	11
Dev 4	328	323	5
Dev 5	2609	2595	14

B. Anomaly Detection Evaluation

The TON_IoT network dataset used in this research was broken down between training and test with different ratios. This is to evaluate the performance of the anomaly detection algorithm used for Zero-Trust evaluations. The following ratios of used data are as follows:

- 90% Train – 10% Test
- 80% Train – 20% Test
- 70% Train – 30% Test
- 60% Train – 40% Test
- 50% Train – 50% Test
- 40% Train – 60% Test

When evaluating the performance of various classification techniques, it's essential to gauge how accurately a classification model assigns records to their respective classes. Area under the ROC curve (AUC) serves as a popular metric for ranking which was used to reflect the overall ranking performance of a classifier. AUC value is theoretically and empirically superior to accuracy in evaluating classifier performance and identifying optimal solutions during classification training [28,29]. Table 3 below shows the AUC value measurements for each algorithm.

TABLE III. Average AUC Value Measurements per Algorithm

Algorithm	MIN	MAX	AVERAGE
KNN	0.4492	0.7670	0.5959
LOF	0.3352	0.7370	0.5570
COF	0.2376	0.6943	0.4932
IForest	0.3545	0.8679	0.6759

Based on the highest AUC measurements, **we have chosen Isolation Forest to be the anomaly detection algorithm for the zero-trust framework with TON_IOT network dataset.**

C. Zero-Trust Experimental Implementation

In our experimental set-up we used Python to simulate network communication between a client and a server. Figure 2 below shows the zero-trust evaluation flow for the proposed ML-based zero-trust framework using Isolation Forest deriving access grant levels.

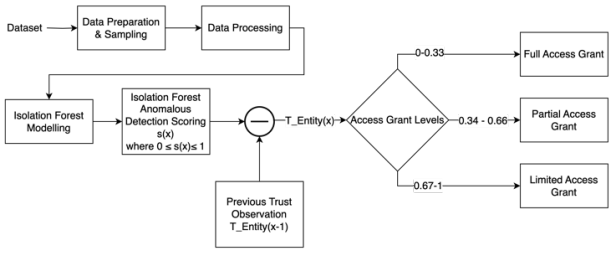

Fig. 2. ML-based zero-trust evaluation flow deriving access grant levels

Figures 3 and 4 show layouts of the static threshold zero-trust architecture and the ML-based zero-trust architecture.

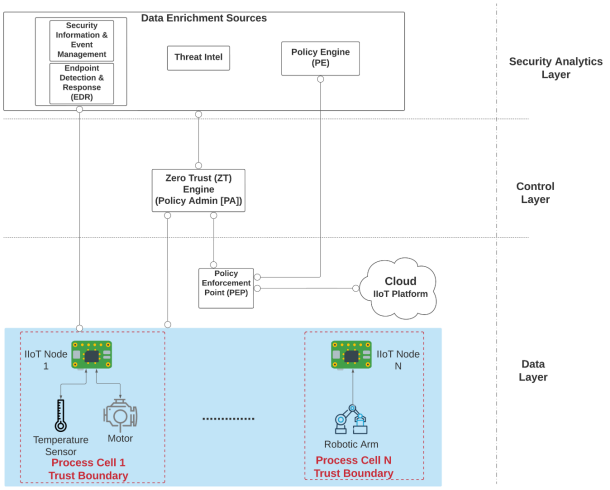

Fig. 3. The static threshold zero-trust architecture

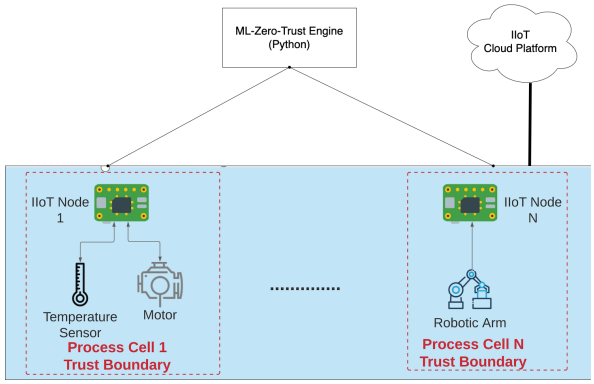

Fig. 4. The proposed ML-based zero-trust architecture

D. Zero-Trust Framework Evaluation

Impact Evaluation is performed by comparing the number of trust value degradation along with the access grant changes for each lab setup. Out of the attack records identified, Table 4 below shows a breakdown of the applied grant levels per device for the ML-based zero-trust lab and the static threshold zero-trust lab. As observed, the static zero-trust lab has more grant level changes than the ML-based zero-trust lab which leads to more impact.

TABLE IV. APPLIED GRANT LEVELS FOR ML-BASED & STATIC THRESHOLD ZERO-TRUST LABS PER DEVICES

IIoT Device	ML-Based Zero-Trust Lab	Static Threshold Zero-Trust Lab
Dev 1	Full Access Grant: 13 Partial Access Grant: 0 Limited Access Grant: 0	Full Access Grant: 10 Partial Access Grant: 3 Limited Access Grant: 0
Dev 2	Full Access Grant: 7 Partial Access Grant: 0 Limited Access Grant: 0	Full Access Grant: 4 Partial Access Grant: 3 Limited Access Grant: 0
Dev 3	Full Access Grant: 11 Partial Access Grant: 0 Limited Access Grant: 0	Full Access Grant: 3 Partial Access Grant: 8 Limited Access Grant: 0
Dev 4	Full Access Grant: 5 Partial Access Grant: 0 Limited Access Grant: 0	Full Access Grant: 2 Partial Access Grant: 3 Limited Access Grant: 0
Dev 5	Full Access Grant: 14 Partial Access Grant: 0 Limited Access Grant: 0	Full Access Grant: 11 Partial Access Grant: 3 Limited Access Grant: 0

This can also be visualised by the following trust value T_Entity(x) over time for Dev 3 where Figure 5 shows the ML-based zero-trust chart and Figure 6 shows the static threshold zero-trust chart.

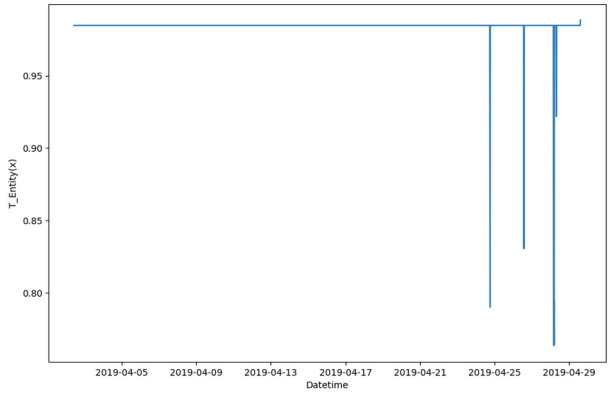

Fig. 5. The evaluated trust value change of Dev 3 using the ML-based zero-trust framework

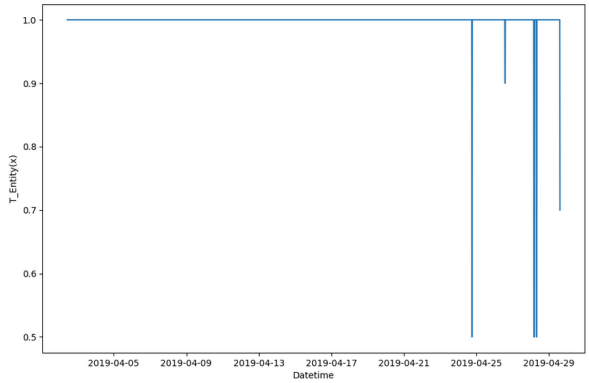

Fig. 6. The evaluated trust value change of Dev 3 using the static threshold zero-trust framework

Machine Learning Performance Evaluation is performed by evaluating the performance of the machine learning algorithm modelling to emphasise the accuracy of the ML-based zero-trust framework. The following evaluation metrics and their formulas are used for evaluating the Isolation Forest model built on the used dataset [26]:

- **Accuracy**: percentage of correction classifications produced by the anomaly algorithm.
- **Precision**: measurement of the correctly predicted positive patterns from the total predicted patterns.
- **F1-score**: measurement of the accuracy of testing.
- **Recall**: ratio of false positives categorised as attacks.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F_1 score = 2 \times (\frac{Precision \times Recall}{Precision + Recall}) \quad (10)$$

Where:

- True Positive (TP): percentage of attack records correctly categorised as attack data
- False Negative (FN): percentage of attack records incorrectly categorised as normal data
- False Positive (FP): percentage of the normal records incorrectly categorised as attack data.
- True Negative (TN): percentage of the normal records correctly categorised as normal data.

Table 5 below shows the performance evaluation metric measurements for the **Isolation Forest model build using a 70%-30% data split of the dataset.**

TABLE V. MACHINE LEARNING EVALUATION METRIC MEASUREMENTS FOR THE ISOLATION FOREST MODEL

Device	Accuracy	Precision	Recall	F1_Score
Dev 1	91.33%	99.58%	91.66%	95.40%
Dev 2	85.33%	98.46%	86.49%	92.09%
Dev 3	90.04%	99.70%	90.25%	94.75%
Dev 4	83.83%	97.65%	85.57%	91.21%
Dev 5	90.20%	99.86%	90.52%	94.96%

Complexity Evaluation is performed by comparing the number of hops for the ML-based and the static-based Zero-Trust implementations to perform analysis and provide trust outcomes. The following interactions for the static threshold and the ML-based zero-trust implementations are shown as follows:

Static threshold zero-trust interactions [7]

1. *IIoT client → Policy Enforcement Point (PEP)*
2. *PEP → ZT-Engine (Policy Admin [PA])*
3. *Verification Request*
   a. *PA → Policy Engine (PE)*
   b. *PA → Endpoint Detection & Response (EDR)*
4. *Valid Verification Response*
   a. *PE → PA*
   b. *EDR → PA*
5. *PEP → IIoT Cloud Firewall*
6. *IIoT Cloud Firewall → Cloud IIoT Server*

ML-Based zero-trust interactions

1. *IIoT Client → ML Zero-Trust Engine*
2. *ML Zero-Trust Engine → IIoT Cloud Firewall*
3. *IIoT Cloud Firewall → Cloud IIoT Server*

Processing Performance Evaluation is performed by sending over 25 data packets from the IIoT client to the zero-trust server utilising both implementations to measure and comparing their processing time. As the static threshold zero-trust implementation has more hops and interactions required to verify and authorise IIoT device, we expect a lower latency from the ML-based implementation than the zero-trust lab. We observed that the ML-based zero-trust setup had a nearly half the processing time of the static zero-trust implementation. Table 6 summarises the results observed from our analysis and tests.

TABLE VI. PROCESS PERFORMANCE EVALUATION RESULTS

Metrics	ML-Based Zero-Trust Lab	Static Zero-Trust Lab
No. of interactions	3	6
Min. processing (ms)	114.1698	357.340008
Max. processing (ms)	44.0998	52.0102081
Avg. processing (ms)	55.2124	98.9972172

VI. ANALYSIS

From a security perspective, the proposed ML-based zero-trust framework provides the capability required to respond to advanced threats and apply restrictions dynamically based on the behaviour of IIoT systems. As anomalies in the behaviours of systems are considered to be key indicators of compromise (IoCs), these anomalies can also be used as indicators of mistrust of which is what has been used here in our zero-trust framework. Particularly as IIoT systems are industrial systems that are programmed to be working in a specific set of patterns, there will be a very high chance of an occurrence of compromise whenever any change of behaviour occurs.

In addition, as observed from the results, the use of **static zero-trust framework can have great impact on the availability and functionality** of critical services delivered by IIoT devices. Limiting access to IIoT devices in any way, shape or form can cause **safety hazards to humans and hence the frequency of these limited accesses has to be kept low at all times**. On the other hand, the ML-based zero-trust framework has applied trust value degradation on the lab IIoT devices while maintaining appropriate level of accuracy and precision of classification of normal vs attack traffic. This can allow security engineers to respond to alerts raised based

on the trust value of these IIoT systems in a safe and accurate manner.

Moreover, the ML-based zero-trust framework exceeded the static zero-trust framework through its enhanced performance and simplicity. However, more exploration is needed to experiment with the use of unsupervised machine learning to remove the dependency for training datasets. Moreover, relating the anomalous detections to the severity of security alerts raised can provide more context and weight to the evaluated trust of IIoT system. For this proposed ML-based to succeed in IIoT environments, the framework is required to be **integrated within a Fog Computing solution to improve the performance of the overall IIoT solution and correlate security findings to improve the evaluated trust values of systems within the environment.**

VII. Conclusion

The convergence of IT and critical industrial environments is becoming more apparent as the need for advanced technologies and integrations increases. Zero-trust is one of the new concepts that are being studied and explored in various avenues. With the use of machine learning within zero-trust, improved security and performance can be applied on IIoT solutions as observed. However, further performance enhancements can be sought with the use of fog computing to reduce the latency via changing the trust evaluation process of the framework. Moreover, incorporation of context awareness gained from insights of access control events and security alerts that can enrich the zero-trust evaluation process and apply more accurate access grants on IIoT systems in the environment.

References

[1] Australian Government - Department of Industry, Science, Energy and Resources, 2018, Industry 4.0, viewed 20/03/2021, https://www.industry.gov.au/funding-and- incentives/industry-40.

[2] Critical Infrastructure Centre Australia 2018, Critical Infrastructure Centre Com- pliance Strategy.

[3] DHS 2016, Recommended Practice: Improving Industrial Control System Cyber- security with Defense-in-Depth Strategies, National Cybersecurity and Communi- cations Integration Center Industrial Control Systems Cyber Emergency Response Team.

[4] A. Atieh, P. Nanda, and M. Mohanty, "A Zero-Trust Framework for Industrial Internet of Things," in *2023 International Conference on Computing, Networking and Communications (ICNC)*, 20-22 Feb. 2023 2023, pp. 331-335, doi: 10.1109/ICNC57223.2023.10074295.

[5] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," *IEEE Access,* vol. 8, pp. 165130-165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[6] C. DeCusatis, P. Liengtiraphan, A. Sager, and M. Pinelli, "Implementing Zero Trust Cloud Networks with Transport Access Control and First Packet Authentication," in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, 18-20 Nov. 2016 2016, pp. 5-10, doi: 10.1109/SmartCloud.2016.22.

[7] Momani, M., Challa, S. & Aboura, K. 2007, 'Modelling trust in wireless sensor net- works from the sensor reliability prospective', Innovative algorithms and techniques in automation, industrial electronics and telecommunications, pp. 317-21.

[8] R. Ward and B. Beyer, "Beyondcorp: A new approach to enterprise security," 2014.

[9] F. E. Grubbs, "Procedures for Detecting Outlying Observations in Samples," *Technometrics,* vol. 11, no. 1, pp. 1-21, 1969/02/01 1969, doi: 10.1080/00401706.1969.10490657.

[10] D. M. Hawkins, "Identification of outliers," ed. London ;: Chapman and Hall, 1980.

[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR),* vol. 41, no. 3, pp. 1-58, 2009.

[12] K. Labib and R. Vemuri, "NSOM: A real-time network-based intrusion detection system using self-organizing maps," *Networks and Security,* vol. 21, no. 1, 2002.

[13] R. Fujimaki, T. Yairi, and K. Machida, "An approach to spacecraft anomaly detection problem using kernel feature space," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 401-410.

[14] A. Selvaraj and S. Sundararajan, "Evidence-Based Trust Evaluation System for Cloud Services Using Fuzzy Logic," *International Journal of Fuzzy Systems,* vol. 19, no. 2, pp. 329-337, 2017/04/01 2017, doi: 10.1007/s40815-016-0146-4.

[15] D. Kay, "Planning for a Zero Trust Architecture Target State," Cybersecurity Standards, Architecture and Engineering, Federal CIO Zero Trust Architecture Summit, 2019. Accessed: 15-12-2021. [Online]. Available: https://www.nccoe.nist.gov/sites/default/files/legacy-files/5_kay_nasa_distribution.pdf

[16] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in *2008 Eighth IEEE International Conference on Data Mining*, 15-19 Dec. 2008 2008, pp. 413-422, doi: 10.1109/ICDM.2008.17.

[17] L. Sun, S. Versteeg, S. Boztas, and A. Rao, "Detecting anomalous user behavior using an extended isolation forest algorithm: an enterprise case study," *arXiv preprint arXiv:1609.06676,* 2016.

[18] Z. Ding and M. Fei, "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window," *IFAC Proceedings Volumes,* vol. 46, no. 20, pp. 12-17, 2013.

[19] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Optics-of: Identifying local outliers," in Principles of Data Mining and Knowledge Discovery: Third European Conference, PKDD'99, Prague, Czech Republic, September 15-18, 1999. Proceedings 3, 1999: Springer, pp. 262-270.

[20] J. Tang, Z. Chen, A. W.-C. Fu, and D. W.-L. Cheung, "Enhancing Effectiveness of Outlier Detections for Low Density Patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2002.

[21] A. Nowak - Brzezińska and C. Horyń, "Outliers in rules - the comparision of LOF, COF and KMEANS algorithms," Procedia Computer Science, vol. 176, pp. 1420-1429, 2020/01/01/ 2020, doi: https://doi.org/10.1016/j.procs.2020.09.152.

[22] W. U. Hassan *et al.*, "NoDoze: Combatting Threat Alert Fatigue with Automated Provenance Triage," *Proceedings 2019 Network and Distributed System Security Symposium,* 2019.

[23] D. Ulevitch, "Cisco 2017 Annual Cybersecurity Report: The Hidden Danger of Uninvestigated Threats," ed, 2017.

[24] H. Jin and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering,* vol. 17, no. 3, pp. 299-310, 2005, doi: 10.1109/TKDE.2005.50.

[25] S. Rosset, "Model selection via the AUC," presented at the Proceedings of the twenty-first international conference on Machine learning, Banff, Alberta, Canada, 2004. [Online]. Available: https://doi.org/10.1145/1015330.1015

[26] N. Elmrabit, F. Zhou, F. Li, and H. Zhou, "Evaluation of Machine Learning Algorithms for Anomaly Detection," in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 15-19 June 2020 2020, pp. 1-8, doi: 10.1109/CyberSecurity49315.2020.9138871.