SUPPLY CHAINS AND CONTROL SYSTEMS

Machine Learning-based Intrusion Detection Systems

Deployment Guidelines for Industry

Authors

Shreevanth Gopalakrishnan

Dr Nilufer Tuptuk

Prof. Stephen Hailes

University College London

THE PETRAS NATIONAL CENTRE OF EXCELLENCE FOR IoT SYSTEMS CYBERSECURITY

Contents

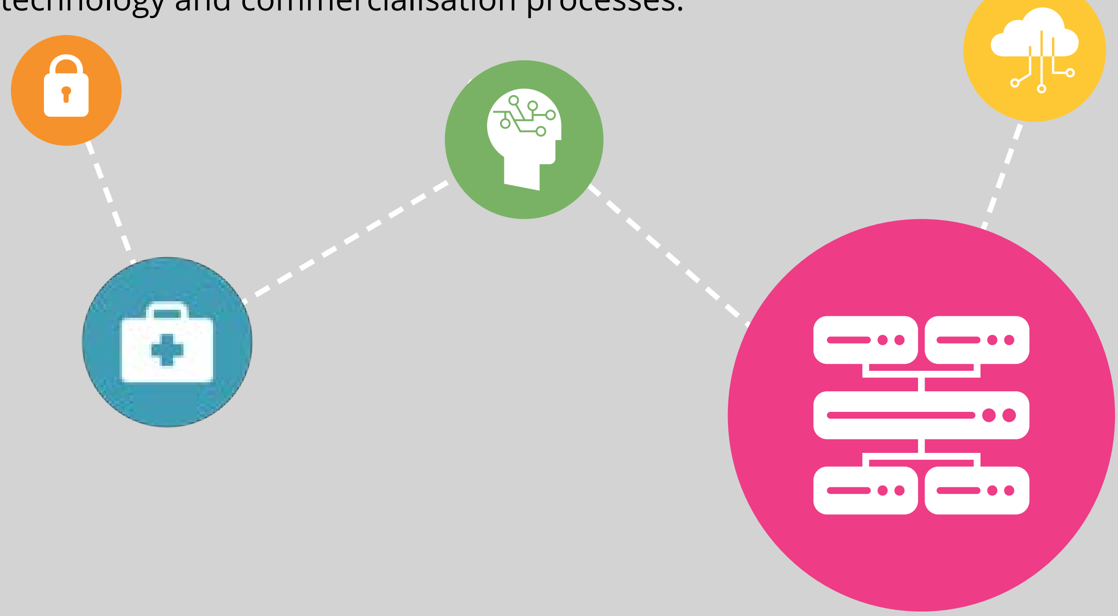
About PETRAS

The PETRAS National Centre of Excellence for IoT Systems Cybersecurity exists to ensure that technological advances in the Internet of Things (IoT) are developed and applied in consumer and business contexts, safely and securely. This is done by considering social and technical issues relating to the cybersecurity of IoT devices, systems and networks.

To achieve our objectives, PETRAS works in collaboration with academia, industry and government partners to ensure our research can be directly applied to benefit society, business and the economy.

The Centre is a consortium of 23 research institutions and the world's largest socio-technical research centre focused on the future implementation of the Internet of Things. The research institutions are: UCL, Imperial College London, University of Bristol, Cardiff University, Coventry University, University of Edinburgh, University of Glasgow, Lancaster University, Newcastle University, Northumbria University, University of Nottingham, University of Oxford, University of Southampton, University of Surrey, Tate, the University of Warwick and Keele University.

As part of UKRI's Security of Digital Technologies at the Periphery (SDTaP) programme, PETRAS runs open, national level funding calls which enable us to undertake cutting edge basic and applied research. We also support the early adoption of new technologies through close work with other members of the SDTaP programme, such as InnovateUK, supporting demonstrations of new technology and commercialisation processes.



Abbreviations

AI	Artificial Intelligence	AIDS	Anomaly-based Intrusion De-tection System
API	Application Programming Interfacing	ARIMA	Autoregressive Integrated Moving Average
AUC	Area Under Curve	BYOD	Bring Your Own Device
CGI	Common Gateway Interface	CIA	Confidentiality, Integrity, Availability
COPOD	Copula-Based Outlier Detection	CPS	Cyber-Physical Systems
CRM	Customer Relationship Management	CV	Cross-Validation
DCS	Distributed Control System	DFT	Discrete Fourier Transform
DL	Deep Learning	DMZ	Demilitarised Zone
DNP3	Distributed Network Protocol 3	DNS	Domain Name Service
DoS	Denial of Service	DPIT	Differential Pressure Indicator and Transmitter
DWT-MLEAD	Discrete Wavelet Transforms and Maximum Likelihood Estimation	ECG	Electrocardiogram
EMS	Energy Management System	ERP	Enterprise Resource Planning
EUC	Equipment Under Control	F1	F1 Score
Fast-MCD	Fast Minimum Covariance Determinant	FDI	False Data Injection
FFT	Fast Fourier Transform	FN	False Negative
FP	False Positive	FPR	False Positive Rate
FTP	File Transfer Protocol	GAN	Generative Adversarial Network
HBOS	Histogram-Based Outlier Score	HIDS	Host-based Intrusion Detection System
HIF	Hybrid Isolation Forest	HMAD	Hidden Markov Anomaly Detection
HMI	Human Machine Interface	HTTP	Hypertext Transfer Protocol
I/O	Input/Output	ICMP	Internet Control Message Protocol
ICS	Industrial Control Systems	IDS	Intrusion Detection System
IoT	Internet of Things	IP	Internet Protocol
IT	Information Technology	KNN	K-Nearest Neighbours
K-S	Kolmogorov Smirnov	LaserDBN	Laser Dynamic Bayesian Network

LIME	Local Interpretable Model Agnostic Explanations	LIT	Level Indicator and Transmitter
LoC	Loss of Control	LOF	Local Outlier Factor
LSTM-AD	Long Short-Term Memory Anomaly Detection	ML	Machine Learning
MultiHMM	Multivariate Hidden Markov Model	NF	Normalizing Flows
NIDS	Network-based Intrusion Detection System	NoveltySVR	Novelty Support Vector Regression
OCSVM	One-Class Support Vector Machine	OS	Operating System
OT	Operational Technology	PCI	Prediction Confidence Interval
PCS	Process Control System	PERA	Purdue Enterprise Reference Architecture
PLC	Programmable Logic Controllers	Pr	Precision
PR	Precision-Recall	PST	Probabilistic Suffix Trees
Re	Recall	RNN	Recurrent Neural Network
RobustPCA	Robust Principal Components Analysis	ROC	Receiver Operating Characteristic
RTU	Remote Terminal Unit	SaaS	Software-as-a-Service
SCADA	Supervisory Control and Data Acquisition	SDM	Statistical Division Multiplexing
SHAP	Shapely Additive Explanations	SIS	Safety Instrumented System
sk-learn	scikit-learn	SMB	Service Message Block
SNMP	Simple Network Management Protocol	SOA	Service-Oriented Architecture
SOC	Security Operations Centre	SQL	Structured Query Language
SR	Spectral Residual	STOMP	Explainable Artificial Intelligence
SVM	Support Vector Machines	SWaT	Secure Water Treatment
TCP	Transmission Control Protocol	TDM	Time Division Multiplexing
TLS	Transport Layer Security	TN	True Negative
TP	True Positive	TPR	True Positive Rate
TSMC	Taiwan Semiconductor Manufacturing Company	UDP	User Datagram Protocol
USB	Universal Serial Bus	VALMOD	Variable-Length Motif and Discord discovery
VAR	Vector Autoregression	VNC	Virtual Network Computing
XAI	Explainable Artificial Intelligence		

1. Executive Summary

Industrial Control Systems (ICS) are increasingly becoming the subject of high-profile attacks. The motivations for these attacks can range from disgruntled employees, financial, socio-political, military advantage, and corporate advantage, amongst others.

Historically, intrusion detection systems (IDS) have not been widely used to protect ICS. For years, security for ICS was achieved through obscurity and isolation due to wide use of legacy systems that were not connected to wider networks and use of proprietary communication protocols. However, to improve cost-efficiency and productivity, ICS are becoming more connected to other systems via open communication protocols and use of smart devices such as Internet of Things (IoT). This new design has made securing ICS more challenging, and in need of security tools and techniques to increase visibility and protect against evolving threats.

In the coming decade, due to increasing sophistication of attackers and their attack methods, it is critical that security measures also advance and have the ability to accurately detect and prevent threats. Machine Learning (ML) is one such promising technology. ML systems can be trained to automatically learn patterns of behaviour directly from network and/or physical data to detect malicious activity, and optionally, faults, and then deploy them to make inferences about new patterns in service. While the use of ML has advantages such as faster creation of attack detection models, building and deploying ML systems have significant challenges.

This report aims to prepare ICS end-users to have technical discussions and make informed decisions about creating and deploying ML-based IDS into a business. There is also guidance on which detection tools to choose from in the presence of a plethora of commercial and open-source options.

This report is meant to serve as a guide for:

- Operators, managers of ICS or those responsible for making decisions related to designing, installing, purchasing, or maintaining the performance of IDS.
- ICS suppliers, component designers, and others working on design/architecture definition processes.
- Decision-makers at the boardroom level when taking high-level decisions about the security of their ICS facilities.

This report provides tools for selecting and deploying Machine Learning-based anomaly detection tools into a business:

- The types of ML-based anomaly detection tools available
- Aspects to consider while selecting one/discussing about them
- How to define well-rounded performance
- Options for deploying them and maintaining them when they are in use
- Limitations - both at an algorithm-level and at a domain-level. Also, Cyber-Physical Systems which encompass autonomous cars, robots, etc., are broader than only ICS. They are not targeted by this guidelines report.

For further details regarding the parent project, refer to Section APX A.

2. Key Recommendations

- **Dataset:** The most important starting point is to identify the critical assets and processes with regards to safety and security. This will allow data capture at suitable points around the system to maximise protection. Fortunately, with ICS, changes upstream can be felt downstream; therefore, an anomaly detector with wide enough focus will be also able to capture this context-based anomaly. More detailed considerations will need to be made about physical measurement units, sampling frequency and data concentration based on accessibility and fidelity requirements.
- **Model Selection:** Semi-supervised and unsupervised detectors with their ability to detect zero-day attacks should make them the preferred choice, despite their tuning challenges. Furthermore, if hand-labelling is feasible to some extent, semi-supervised learning would be the best option. However, in a novelty detection form, this approach would first require an anomaly-free dataset for the model to learn normal behaviour.
- **Interpretability/Usability:** For the models to be usable, it is essential that the performance metrics utilised account for false positive rates in the imbalanced training dataset. If the model(s) is too sensitive, then this could lead to the operators getting distracted and/or not taking the model seriously. On the other hand, maximising interpretability will allow the operator who has the final say to select better decisions regarding corrective actions/ countermeasures. Knowledge of “where”, “when” and “why” the anomaly was detected can add to model trustworthiness. Several options were presented including appropriate model choice, data visualisation and separate explainability tools.
- **Maintenance:** Post-deployment, to keep up with process changes and to ensure that the system remains performant, it is essential that periodic and rigorous re-assessment, model updating (and documentation) is carried out. If

feasible, online training with fresh data from the field could make the system robust to operational drift.

Stage	References
Acquire a dataset from operation	3.1, 3.2, & 5.1.1
Verify veracity of data, and conduct pre-processing and exploratory data analysis	5.1.1, 5.1.2, & 5.3
Carry out feature engineering & complete data preparation	5.1.3, 5.3
Devise machine learning strategy and identify a suite of detection models	5.2, 5.3, & 5.4
Select evaluation metrics	5.2, 5.4, & 5.5.2
Train (& validate) detection models, & tune hyperparameters	5.5.1, 5.5.2
Evaluate and compare models' performance and suitability	5.4, 5.5.2, & 5.7
Deploy to field for real-time use & consider maintainability aspects	5.6, 5.7

Table 1: References for ML-based anomaly detection system development flowchart

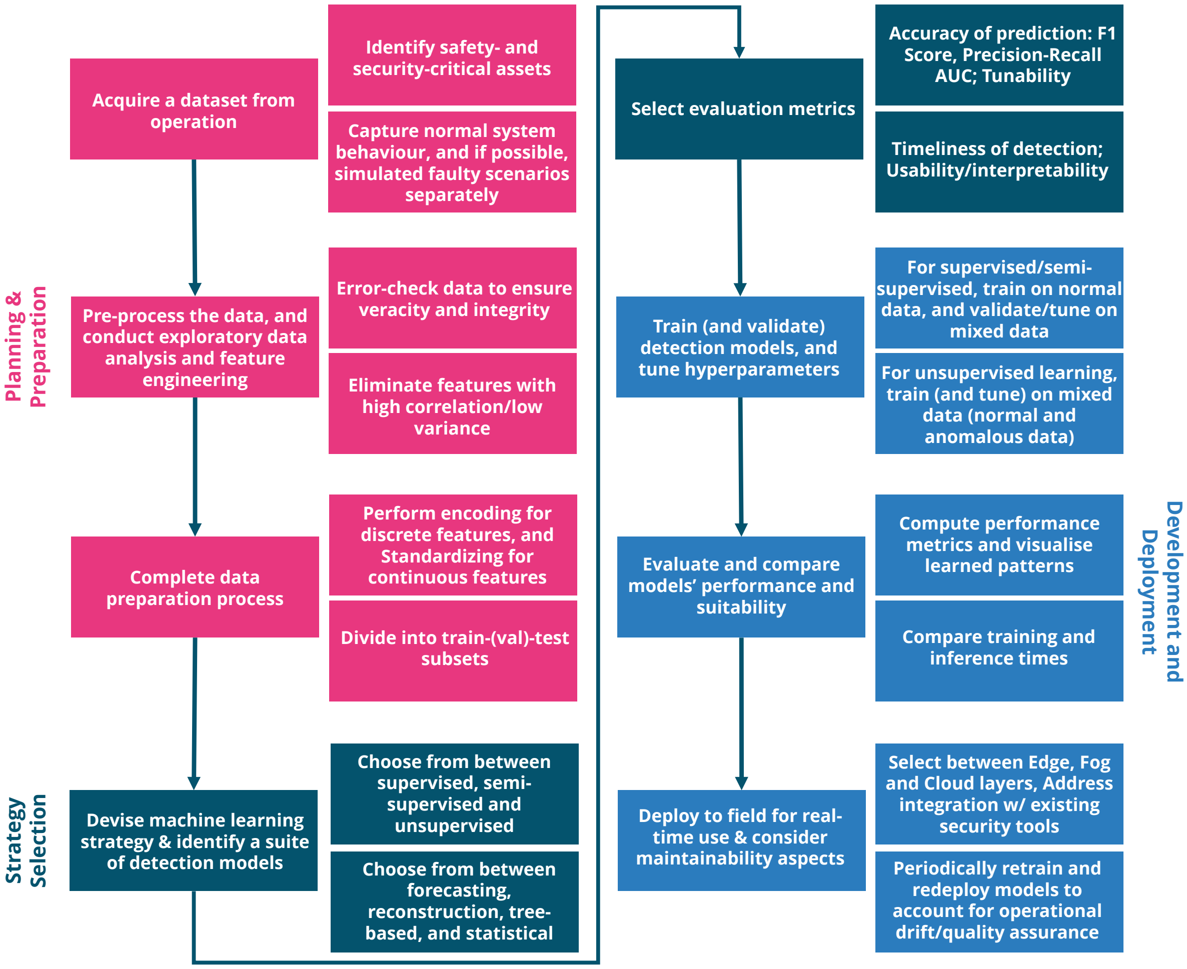

Planning & Preparation
Acquire a dataset from operation
Identify safety- and security-critical assets
Capture normal system behaviour, and if possible, simulated faulty scenarios separately
Pre-process the data, and conduct exploratory data analysis and feature engineering
Error-check data to ensure veracity and integrity
Eliminate features with high correlation/low variance
Complete data preparation process
Perform encoding for discrete features, and Standardizing for continuous features
Divide into train-(val)-test subsets
Strategy Selection
Devise machine learning strategy & identify a suite of detection models
Choose from between supervised, semi-supervised and unsupervised
Choose from between forecasting, reconstruction, tree-based, and statistical
Select evaluation metrics
Accuracy of prediction: F1 Score, Precision-Recall AUC; Tunability
Timeliness of detection; Usability/interpretability
Train (and validate) detection models, and tune hyperparameters
For supervised/semi-supervised, train on normal data, and validate/tune on mixed data
For unsupervised learning, train (and tune) on mixed data (normal and anomalous data)
Evaluate and compare models' performance and suitability
Compute performance metrics and visualise learned patterns
Compare training and inference times
Development and Deployment
Deploy to field for real-time use & consider maintainability aspects
Select between Edge, Fog and Cloud layers, Address integration w/ existing security tools
Periodically retrain and redeploy models to account for operational drift/quality assurance

Figure 1: Machine-Learning-based anomaly detection system development flowchart

3. Background

Industrial Control Systems (ICS) Control Systems (ICS) are networked information systems that control and monitor physical processes. They are a critical part of a wide variety of products and services we rely on every day. From a high-level view, the monitor and control functions involve:

- **Monitor:** part of a control loop; keeping an eye on a critical value and comparing it against a predefined threshold to automatically compute the error
- **Control:** through error feedback; moving/activating things, and initiating processes

In tandem, an operator typically watches the current state of the automation system in real-time and intervenes when necessary, i.e., they supervise the process.

Hence, ICS is an umbrella term used for various automation systems and their devices, e.g., Programmable Logic Controllers (PLC), Building Management Systems (BMS), Human Machine Interfaces (HMI), Supervisory Control and Data Acquisition (SCADA), Remote Terminal Units (RTU), etc.

The rest of this section provides a brief review of the landscape of cyber threats associated with ICS before diving into the details of IDS from Section 4 onwards.

3.1 Attacks on ICS

3.1.1 Motivation, Attacker Types, and Attack Vectors

ICS have been the subject of several attacks in recent years. Figure 1 shows some of the different sources of cyber threats. Each of these attack groups (or “cyber-criminal” groups) have different motivations and levels of sophistication/technologies at their disposal. These motivations could be a financial, socio-political, military, corporate advantage, damage to reputation, etc.

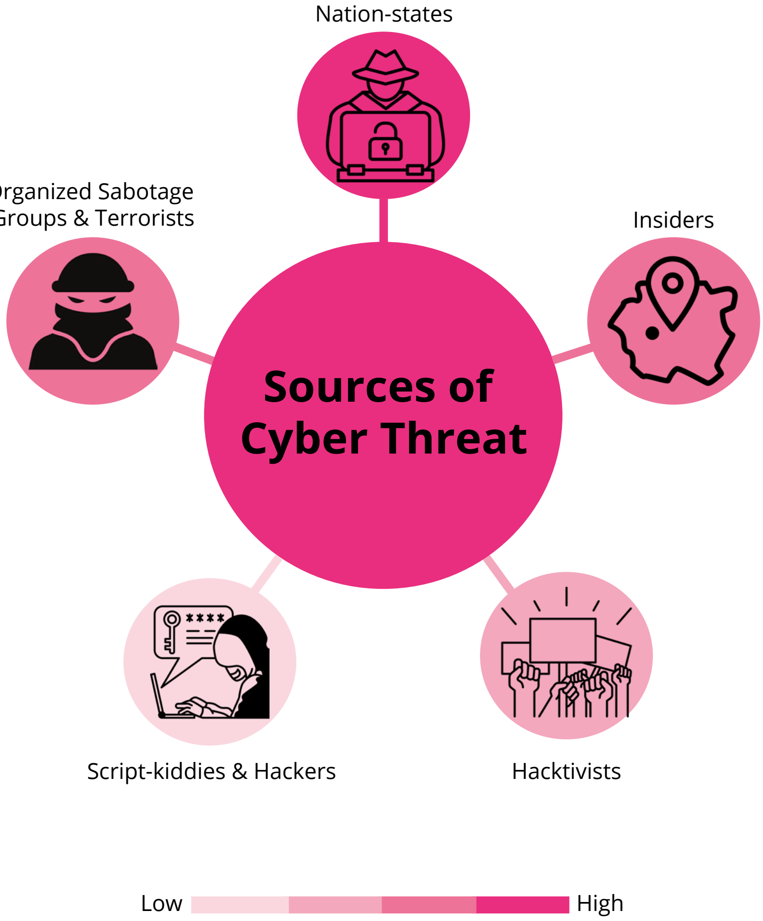


Figure 2: Categories of cyber threat sources based on level of sophistication

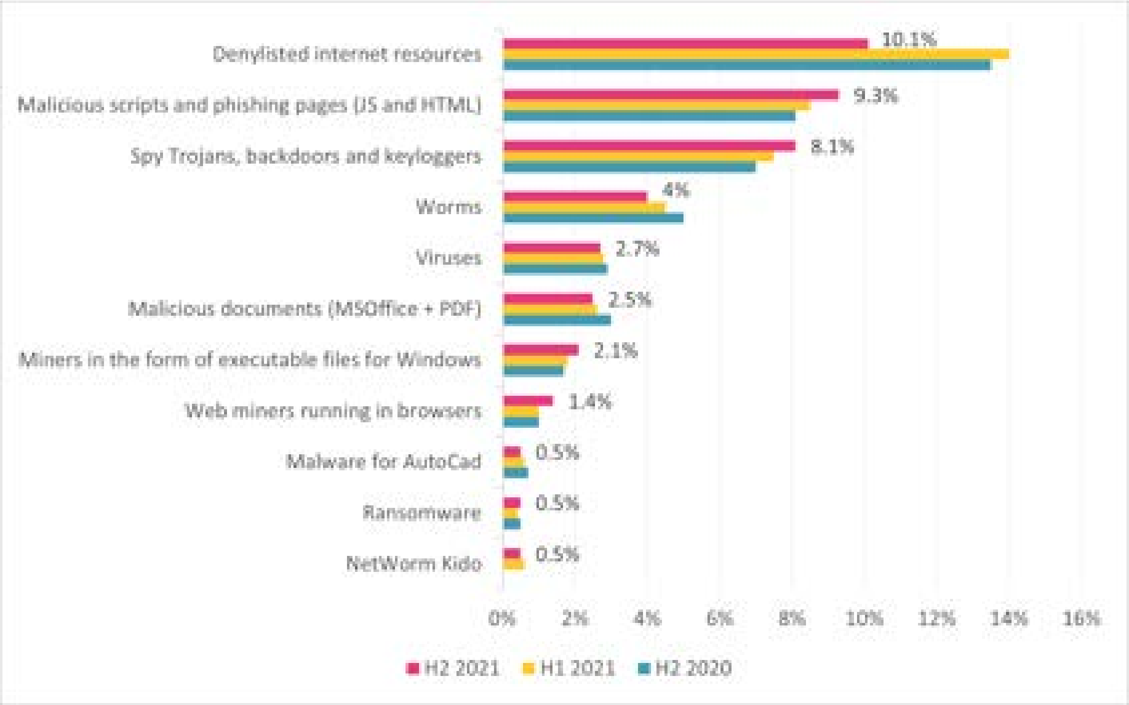


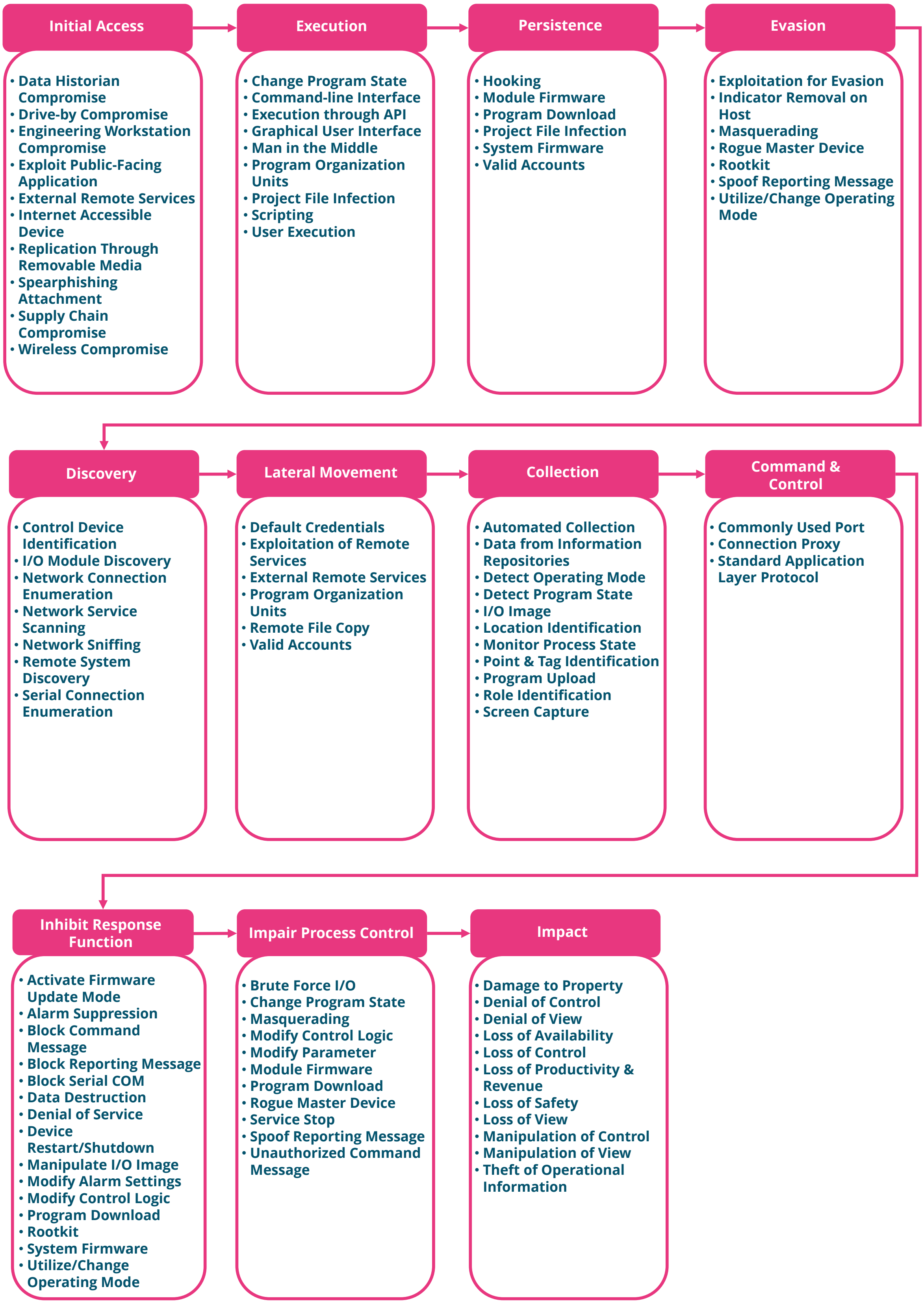
Figure 3: Perpetrators of cyberattacks choose from a variety of attack vectors. Adapted from [1]

Based on the tools at their disposal, their ability to conduct reconnaissance/available intelligence, and differing motivations, perpetrators of cyberattacks choose from a variety of attack vectors. Figure 3 provides details of the commonly observed attack methods based on Kaspersky's annual ICS threat landscape report 2022 [1].

Most attacks appear to involve compromising the corporate IT network e.g., phishing pages, software vulnerabilities. Once attackers have a foothold in a network, they systematically move "laterally" (Cyber "Kill-Chain") across the network to compromise a particular system or a combination of systems along the way. The MITRE ATT&CK framework for ICS shown in Figure 4 is a generalized model of the movement strategy. The MITRE Corporation also maintains a globally centralized, publicly accessible online database of Common Vulnerabilities and Exposures (CVEs) related to information security whereby known vulnerabilities are attributed to official CVE IDs [2], [3]. The database is searchable by vendor name and product type, and provides information regarding available patches.

Examples of these scenarios with different components in the Operational Technology (OT) side are shown in Table 2 and Figure 5., Attacks are often top-down (IT to OT), however, OT sub-network attacks can also lead to penetration of enterprise IT systems (Purdue model [4]).

In recent times, OT security has had to adapt due to the additional attack vectors introduced by IT-OT convergence and the Industrial Internet of Things. Further details are provided in Section APX D. ii).



Initial Access

- Data Historian Compromise
- Drive-by Compromise
- Engineering Workstation Compromise
- Exploit Public-Facing Application
- External Remote Services
- Internet Accessible Device
- Replication Through Removable Media
- Spearphishing Attachment
- Supply Chain Compromise
- Wireless Compromise

Execution

- Change Program State
- Command-line Interface
- Execution through API
- Graphical User Interface
- Man in the Middle
- Program Organization Units
- Project File Infection
- Scripting
- User Execution

Persistence

- Hooking
- Module Firmware
- Program Download
- Project File Infection
- System Firmware
- Valid Accounts

Evasion

- Exploitation for Evasion
- Indicator Removal on Host
- Masquerading
- Rogue Master Device
- Rootkit
- Spoof Reporting Message
- Utilize/Change Operating Mode

Discovery

- Control Device Identification
- I/O Module Discovery
- Network Connection Enumeration
- Network Service Scanning
- Network Sniffing
- Remote System Discovery
- Serial Connection Enumeration

Lateral Movement

- Default Credentials
- Exploitation of Remote Services
- External Remote Services
- Program Organization Units
- Remote File Copy
- Valid Accounts

Collection

- Automated Collection
- Data from Information Repositories
- Detect Operating Mode
- Detect Program State
- I/O Image
- Location Identification
- Monitor Process State
- Point & Tag Identification
- Program Upload
- Role Identification
- Screen Capture

Command & Control

- Commonly Used Port
- Connection Proxy
- Standard Application Layer Protocol

Inhibit Response Function

- Activate Firmware Update Mode
- Alarm Suppression
- Block Command Message
- Block Reporting Message
- Block Serial COM
- Data Destruction
- Denial of Service
- Device Restart/Shutdown
- Manipulate I/O Image
- Modify Alarm Settings
- Modify Control Logic
- Program Download
- Rootkit
- System Firmware
- Utilize/Change Operating Mode

Impair Process Control

- Brute Force I/O
- Change Program State
- Masquerading
- Modify Control Logic
- Modify Parameter
- Module Firmware
- Program Download
- Rogue Master Device
- Service Stop
- Spoof Reporting Message
- Unauthorized Command Message

Impact

- Damage to Property
- Denial of Control
- Denial of View
- Loss of Availability
- Loss of Control
- Loss of Productivity & Revenue
- Loss of Safety
- Loss of View
- Manipulation of Control
- Manipulation of View
- Theft of Operational Information

Figure 4: The MITRE ATT&CK framework for ICS. Adapted from [5].

Attack Point	Example Attack
1 – Sensors	Compromised sensor and false data injection (FDI) attack, causing the control logic to act on malicious data.
2 – Communications Media (S to C)	Man-in-the-middle between sensor and controller causes a delay or full obstruction, e.g., stale data and denial-of-service (DoS) attacks.
3 - Controller	Compromised controller and incorrect signals (false data injection attack) sent to the actuators leading to damage of assets.
4 – Communications Media (C to A)	Man-in-the-middle between the controller and actuator could perform delay, replay or denial-of-control (DoS of the actuator) attacks.
5 - Actuators	Compromised actuator and execution of control actions different to the instructions of the controller, e.g., zero dynamics attacks.
6 – Physical Processes	Encompasses physical attacks damaging the system. Carried out in isolation or in combination with a cyber-attack (as a hybrid).
7 – Communications Media (Sup to C)	Man-in-the-middle between the supervisory control layer or configuration devices and the controllers. E.g., delay, DoS, etc.
8 – Supervisory Controls	Compromised supervisory controls (SCADA system) or configuration devices. E.g., the Ukraine power grid attack led to a service disruption for customers for several hours and loss of data.

Table 2: Example "Active" attack scenarios at different points (Impact/Exploitation stage). Adapted from [6].

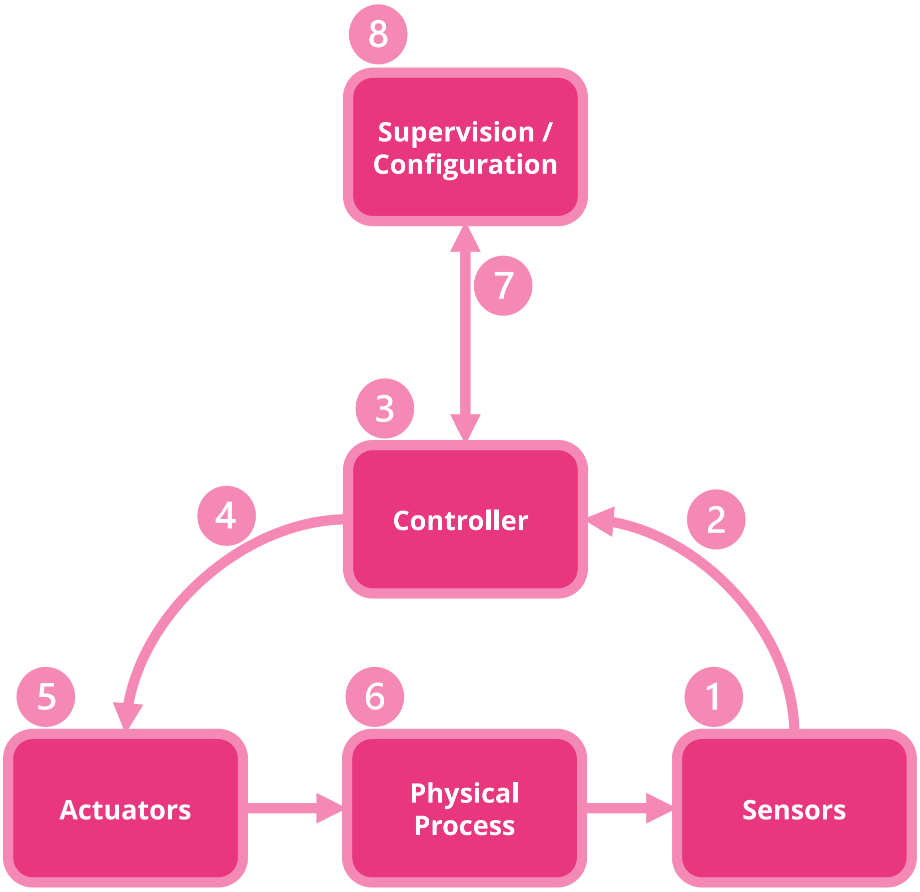


Figure 5: Attack points in a Cyber-Physical or Industrial Control System [6].

3.1.1 Motivation, Attacker Types, and Attack Vectors

Attacks can be broadly divided into two categories based on the attackers' desired consequences. **It is evident from recent events (Section 3.2) that the attackers primarily seek to cause physical damage to assets, exfiltrate data, and/or disrupt critical operations/service to customers**.

1. **Passive Attacks:** Seek to undertake stealthy operations while remaining in the computer or ICS networks, e.g., reconnaissance, data exfiltration. Alternatively, they could seek to deceive relevant employees or mobilise them for malicious activities, e.g., phishing or social engineering.
2. **Active Attacks:** Actively seek to cause damage to physical assets, disrupt operations, damage the reputation of the company, etc. They tend to have crossed the IT and DMZ layers and operate in the lower layers of the Purdue model (APX B). Common examples include (as seen in Table 2) False Data Injection, Denial of Service, Replay, Logic Bomb, Spoofing etc.

3.2 Recent Attacks

Several academic works have reviewed the major historical ICS attacks. Table 3 discusses them.

Year	ICS	Attack vector/method	Consequences
2003	Davis–Besse nuclear power plant, Ohio, US	(Microsoft SQL) Slammer worm injected through direct (T1) remote connection – able to bypass firewall and access control	Safety Parameter Display System and plant process computer being inaccessible for around 6 hours each
2005	Daimler-Chrysler automobile plants, US	Zotob worm exploited vulnerabilities in Windows 2000 industrial PCs	Stopped production at 13 sites for up to 1 hour. Affected around 50,000 workers
2010	Natanz nuclear facility, Iran	Stuxnet malware (4 zero-day exploits, stolen authentication certificates) injected via USB drive. Targeted Siemens ICS systems	Destruction of centrifuge tubes at the nuclear facility
2014	German steel mill, Germany	Social engineering targeted ICS operators to deliver malware for Windows PCs in OT networks. Remote PLC reprogramming	Blast furnaces were shut down causing loss of control, interruption to process and physical damage to system
2014	Energy companies in US and Europe	Phishing emails, ICS-related software containing Havex malware planted in vendor websites (“watering hole”)	Remote access Trojan used for reconnaissance (stealing), software download and code execution. Could have disrupted energy supplies
2015	Kemuri Water Company (KMC), US	Vulnerabilities in payment portal exploited using SQL injection & phishing. Credentials for AS400 operating system stolen from a server	Personal identifiable information of 2.5 million customers leaked. Setpoints for water treatment chemicals altered (detected by KMC, so no impact)

2015	Power grid, Ukraine	Plant login credentials stolen using spear phishing (BlackEnergy3). KillDisk malware deployed to wipe industrial PC memory and prevent rebooting. Telephony DoS used to jam customer reporting	Malware disconnected electrical substations causing power outage for around 225,000 users. SCADA firmware infected
2016	Power Grid, Ukraine	Vulnerabilities in Siemens SIROPROTEC relays exploited to infiltrate substation and create backdoors. Fully automated and “persistent” attack (Industroyer)	20% of Ukraine’s capital, Kiev was disconnected from the grid for over an hour
2017	Ukrainian public and private sector, and multinational companies (e.g., Maersk, Merck)	NotPetya ransomware spread through a centralised update to MeDoc tax accounting system. Utilised the EternalBlue exploit.	Major economic losses (combined 10 billion dollars) through collected ransoms and irreversibly encrypted critical data. Radiation monitoring system at Chernobyl went offline
2017	Petro-chemical plant, Saudi Arabia	Remote access methods used to infect a Safety Instrumentation System (SIS) Windows workstation, and reprogram the SIS to not function correctly	Temporary disruption of industrial processes – no physical damage. Accidentally triggered automatic shutdown of processes allowing operators to be alerted

Table 3 - 1: History of major ICS cyberattacks ([7]–[17])

Year	ICS	Attack vector/method	Consequences
2018	Taiwan Semiconductor Manufacturing Company, Taiwan	WannaCry ransomware (variant) introduced by malicious software installed by supplier on IT network	Virus spread to more than 10,000 machines. Shut down several production plants for 1 day (revenue loss of ~$256 million)
2019	Hoya Corporation, Thailand	Corporate network used to spread unnamed virus capable of stealing user access credentials	Partial shutdown of a large section of its factories for three days. Up to 100 computers infected for the purpose of cryptocurrency mining
2021	The Colonial Pipeline Company, Georgia, US	Compromised/stolen credentials for virtual private network account (for remote access) used to deploy ransomware	100GB data stolen; entire pipeline shutdown for 6 days (major supply disruption in 4 states); ransom ~$4.4 million in bitcoins paid

Table 3 - 2: History of major ICS cyberattacks ([18]–[20])

Some fundamental network hygiene and best practices that we know today could have lowered the chances of these attacks occurring, for example:

1. Timely firmware and software patching of industry-grade computers. Given the availability requirements of ICS, this would need to be facilitated through redundant stand-in systems.
2. Improved password management and two-factor authentication.
3. Staff training to recognise phishing attacks.
4. Provision for manual overrides and fail-safe modes so that the operators can enable safe shutdown of the system if necessary after they have detected tampering of any kind.
5. Stricter policies on the usage of removable media (e.g., USB drives) and personal devices (“Bring Your Own Device” [BYOD]).

A common pattern of all the attacks was to reach down into the critical physical/field layer (or the SIS) of the Purdue model and attempt to cause disruption/damage to assets. In many cases, better isolation of the physical plant with the corporate network through a firewall, and Safety Instrumentation System (SIS) from the DCS could have hindered the attackers. Figure 6 shows the top six targeted ICS sectors including building automation systems, oil & gas, manufacturing, and energy. Raising awareness of these trends will allow the respective industries to be better prepared.

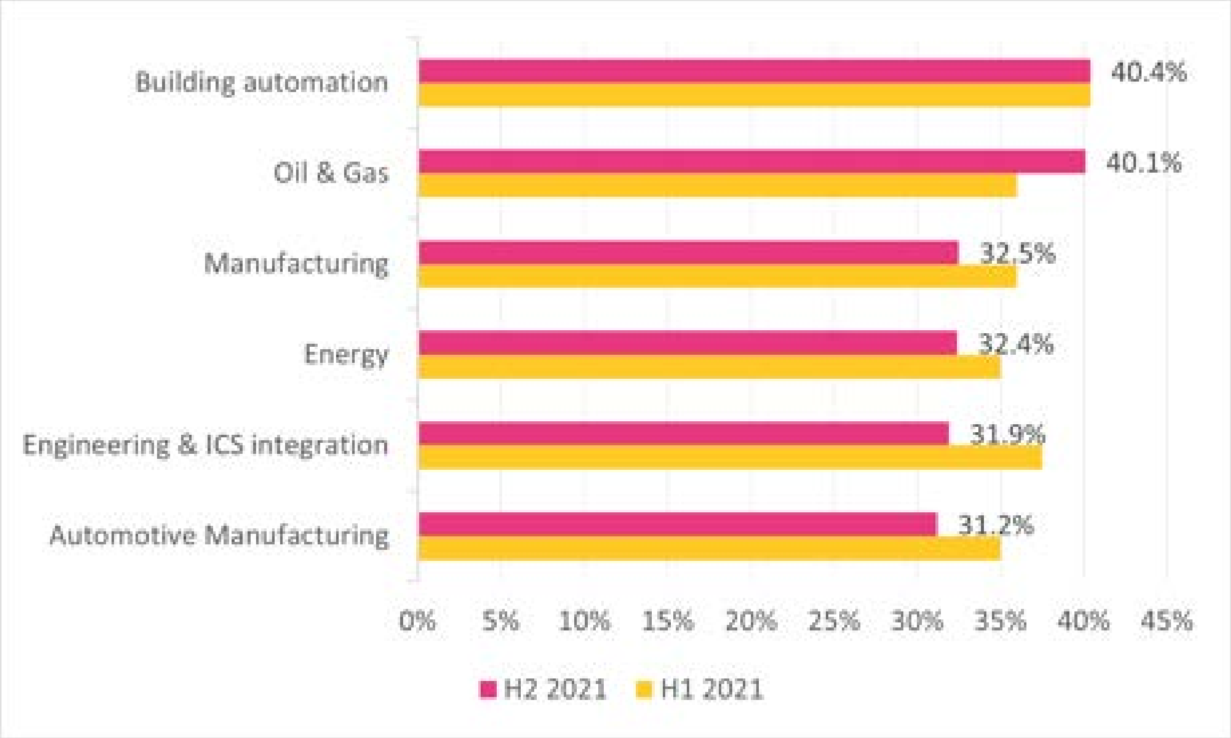


Figure 6: Percentage of ICS computers on which malicious objects were blocked. Adapted from [1].

All the attack groups and attack methods discussed so far have become more sophisticated in the current decade. Therefore, more advanced security tools and techniques, such as ML will be necessary. **In particular, this report will address anomaly-based detection applied to the predominantly insecure interactions in the lowest levels of the Purdue model.**

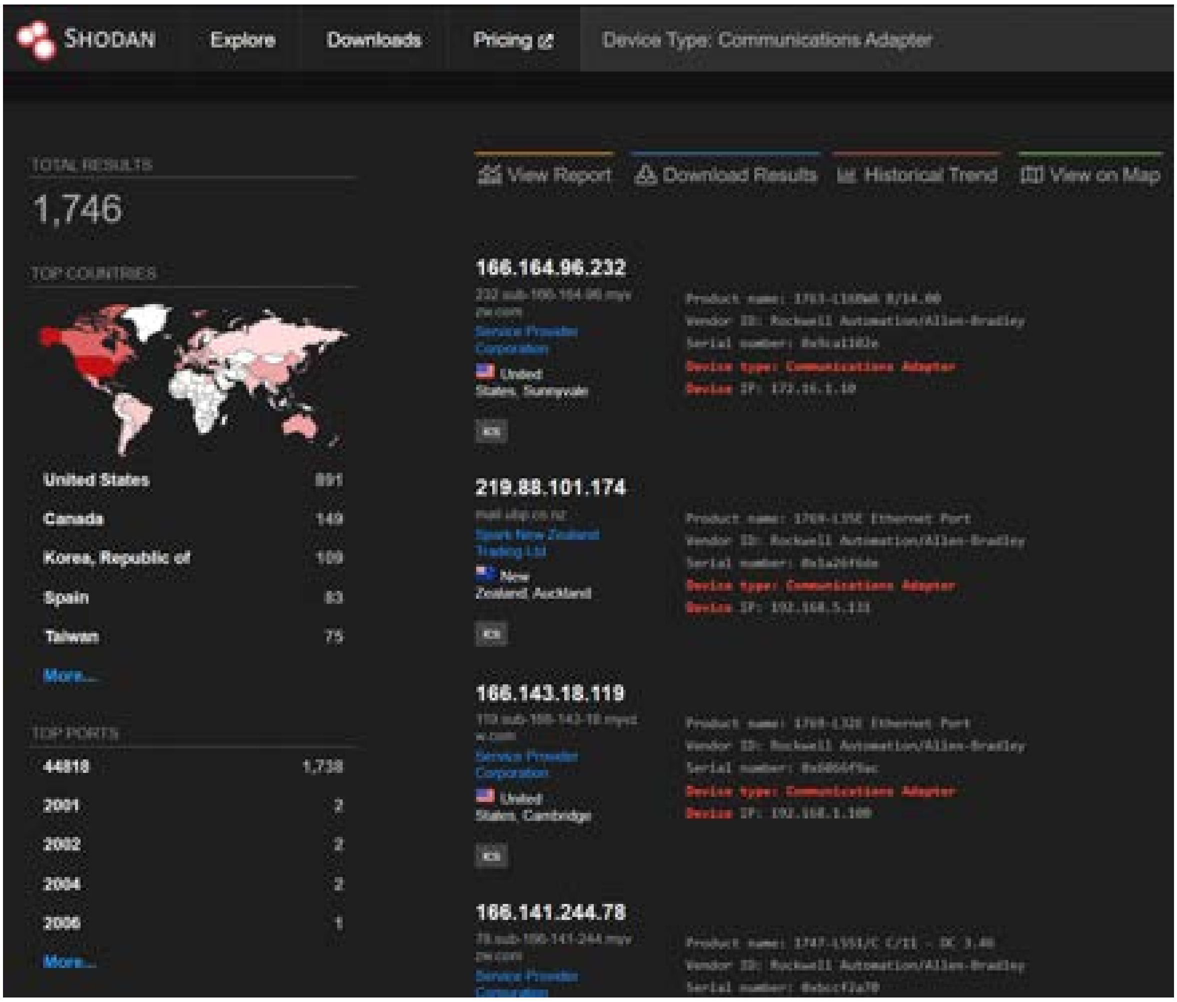
4. Intrusion Detection Systems

4.1 What Are They

Intrusions can be broadly defined as unauthorised activities of any kind regardless of whether they cause damage. This definition is valid in IT and ICS domains. However, as defenders of systems, we should be more cautious of malicious activities in ICS networks whose purpose is to intentionally cause damage. Attacks vectors are becoming more sophisticated, and perpetrators of cyberattacks are using different techniques to evade detection. In this scenario, the main challenges are to identify unknown and concealed malware/attack patterns. Traditional firewalls are unable to cope with these advanced attack techniques.

Formally, an Intrusion Detection System (IDS) can be defined as a monitoring device or software that detects suspicious activities and generates alerts as early as possible when they are detected. Depending on the context of their deployment, an operator in a security operations centre (SOC), an independent incident responder, or other security staff can investigate the issue and take the appropriate actions. This is vital in order to satisfy the security requirements of availability, integrity and confidentiality of an Industrial Control System [21].

Overall, IDS can be useful in achieving real-time visibility into instances of potential compromises in an ICS, even if they are not linked to malicious activity. IDS can be deployed in different layers of the Purdue model as described in previous sections. Although, historically, it has been primarily seen as an IT cybersecurity measure, it is playing an increasing role in OT security as well.



TOTAL RESULTS

1,746

TOP COUNTRIES

United States	891
Canada	149
Korea, Republic of	109
Spain	83
Taiwan	75
More...	

TOP PORTS

44818	1,738
2001	2
2002	2
2004	2
2006	1
More...	

166.164.96.232
232.sub-166-164-96.myvzw.com
Service Provider Corporation
United States, Sunnyvale

```
Product name: 1763-L16BWA B/14.00
Vendor ID: Rockwell Automation/Allen-Bradley
Serial number: 0x9ca1102e
Device type: Communications Adapter
Device IP: 172.16.1.10
```

219.88.101.174
mail.ubp.co.nz
Spark New Zealand Trading Ltd
New Zealand, Auckland

```
Product name: 1769-L35E Ethernet Port
Vendor ID: Rockwell Automation/Allen-Bradley
Serial number: 0x1a26f6de
Device type: Communications Adapter
Device IP: 192.168.5.131
```

166.143.18.119
119.sub-166-143-18.myvzw.com
Service Provider Corporation
United States, Cambridge

```
Product name: 1769-L32E Ethernet Port
Vendor ID: Rockwell Automation/Allen-Bradley
Serial number: 0x6066f5ac
Device type: Communications Adapter
Device IP: 192.168.1.100
```

166.141.244.78
78.sub-166-141-244.myvzw.com
Service Provider Corporation

```
Product name: 1747-L551/C C/11 - DC 3.46
Vendor ID: Rockwell Automation/Allen-Bradley
Serial number: 0xbccf2a70
```

Figure 7: Example of publicly accessible IoT devices using Shodan [22]

4.2 Types of Intrusion Detection Systems

In this section, the different kinds of IDS are discussed, and where appropriate, the limitations of each of these models has also been presented. Based on where they are deployed, i.e., the type of data they see, IDS can be classified into the following [23], [24]:

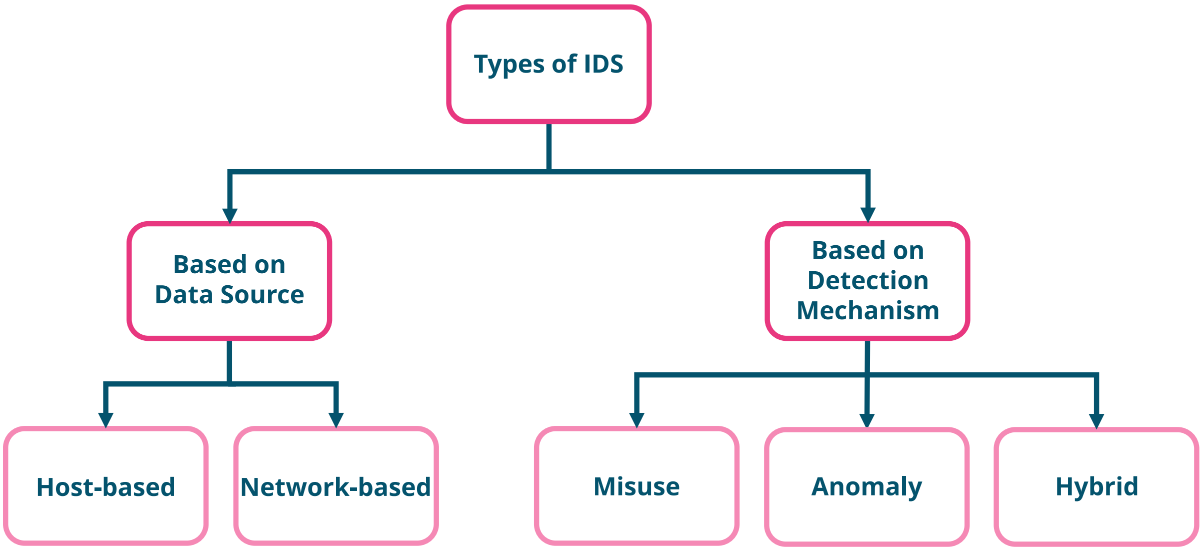


Figure 8: Categorisation of Intrusion Detection Systems. Adapted from [24].

- **Host-based (HIDS):** This system monitors the host/device/endpoint it is deployed in. Its functions include monitoring and analysing internals of a host such as configuration files and application activity, and comparing them with previous “snapshots” where applicable; and monitoring the host’s participation in the network, etc. Therefore, a HIDS’ visibility is limited to the host only. Scenarios they can deal with include unexpected modifications such as deletion, overwriting sensitive system files and access rights of locked ports.
- **Network-based (NIDS):** This system is focussed on monitoring an entire network it is placed within. Its functions involve monitoring and analysing real-time network traffic passing through firewalls/through pre-defined sensitive nodes for suspicious behaviour. The NIDS makes decisions based on packet metadata (headers, etc.) and contents where suitable. However, in contrast to HIDS, these systems lack the internal visibility into the network’s hosts themselves.

It is therefore evident that the deployment of both solutions is necessary for the complete protection of the industrial control system’s assets and network. Further, based on the technique they employ, IDS can be classified into three main categories [25], [26]:

- **Misuse-/Signature-based:** They use patterns/rulesets related to known threats from prior experience to define intrusions. Once an attack has been identified, the specific patterns or instructions related to it are used to generate a signature. This is then added to a defined list of known attack vectors and is used by the IDS solution to test incoming new content. This kind of system is easy to implement. While it has a high attack detection rate with a low false positive rate because all alerts are generated based upon detection of known-malicious content, it has a low detection rate for (unknown) threats, e.g., zero-day vulnerabilities.
- **Behaviour-/Anomaly-based:** They look for abnormalities, in the form of deviations from "normal system behaviour". These solutions employ some learning techniques to build a model of normal behaviour of the protected system/network/device under conditions that are assumed to be attack-free. Subsequently, all future behaviour is then compared to this model. Any anomalies are labelled as potential attacks and alerts are generated. As a result, this approach can detect formerly unknown zero-day intrusions. However, this kind of model requires building an accurate model of normal behaviour in a way that balances incorrect alerts (or false positive predictions) with missed anomalies (or false negatives).
- **Hybrid/Specification-based:** This approach was proposed to minimise the number of false positive associated with anomaly-based IDS, whilst being better than knowledge-based approaches. According to this approach, a formal description of the base specification of the normal behaviour of the system is constructed using the support of an expert, and the IDS looks for deviation from this model. Hence, while this model requires more effort (human) to be set up and maintained, it can help improve the reliability of the IDS.

While it appears that specification-based IDS are the optimal solution, they have not been most popular in recent times since they are less effective against novel attacks than anomaly-based methods. Therefore, research has focussed on building more efficient data-driven Anomaly Detection Systems.

For the lower layers of the Purdue model, these anomaly detectors are built to take network traffic of physical parameters from the OT/industrial automation networks as input. **Generally, anomaly detectors are designed to detect any abnormal behaviour. Based on the dataset and how they are trained, they could be used for fault detection and as a security tool.**

4.3 Comercially Available IDS Solutions

Table 4 provides a ready-made summary of some of the commercially available (mostly free) sole IDS solutions for different-sized businesses from an IT perspective. They could be used for network intrusion detection in the higher layers of the Purdue model. Different tools are suitable for different scenarios, operating systems, and vulnerability scenarios. *Please note that these examples have only been provided for reference. It is recommended that the knowledge gained from this report be used along with expert knowledge of the threat landscape of the system(s) before making decisions regarding which solution to use/vendor to work with*.

Tool Name	Best for	Type of IDS	Features
Suricata	Medium and Large Businesses (Free)	NIDS	Data collection at the application layer (unlike Snort); network security monitoring: TLS/SSL, HTTP, DNS logging and analysis; inline intrusion prevention; understands higher-level (SMB, FTP, HTTP) and lower-level protocols (UDP, ICMP); integration with third-party tools (e.g., Anaval, Squil); scripting module
Manage-Engine Log360	Small to Large Businesses (Paid)	NIDS	Real-time incident management/ event correlation, integration with ticketing tools, diverse logging and log parsing, privileged user monitoring; forensic reporting for SOC; in-built ticketing
Zeek	Any Business (Free)	NIDS	Passive network traffic logging and signature analysis; monitor SNMP, FTP, DNS, and HTTP traffic; event engine to track triggering events; policy scripts for mining event data
Snort	Small and Medium Businesses (Free)	NIDS	Core intrusion prevention using pre-defined rules based on live threat intelligence, combined with packet sniffing and logging; options for anomaly detection; able to detect a variety of events including operating system fingerprinting, protocol probes, common gateway interface attacks, stealthy scans etc.

Table 4: Some commercial NIDS solutions for different-sized businesses ([27]–[31])

Tool Name	Best for	Type of IDS	Features
Open Source Security (OSSEC)	Medium and Large Businesses (Free)	HIDS	Client/server-based logging including mail, FTP, and web server data; monitor unauthor-ised registry modifications and unauthorized access attempts to root account; rootkit detec-tion and real-time alerting
SolarWinds Security Event Manager	Large Businesses (Paid)	HIDS/NIDS	Collects data from network and infrastructure logs to determine the amount and types of attacks on the network as part of a proactive detection and response system; low requirement for operator detection/ effort
Security Onion	Medium and Large Businesses (Free)	HIDS, NIDS	Free open-source Linux distribution for log management, enterprise security monitoring, and threat hunting (e.g., proactively searching for malicious attempts to compromise the system)

Table 5: Some commercial HIDS solutions for different-sized businesses ([27], [32]–[34])

Online searches for OT environment-specific IDS present limited results. On the other hand, most security providers tend to provide integrated security solutions for Cloud, Endpoint, Data, Network, etc., together with Threat Intelligence, Asset Management, Incident Response etc. in different combinations. Some examples of leading companies there are providing commercially available solutions as shown in market reports include CyberArk [35], Sophos [36], Kaspersky [37], BAE Systems [38], Dragos [39], SCADAfence [40], Forescout [41]. Some of these companies are dedicated to cybersecurity (OT and/or IT), whereas others are extensions of their component lines, e.g. ABB [42], Rockwell Automation [43], Cisco [44].

A minimisation of the number of inter-provider (of tools) interfaces could result in more efficient transport and utilisation of data with regards to building situational awareness and timely alarm reporting. Hence, this trend is advantageous.

4.4 Anomaly-Based Intrusion Detection Systems

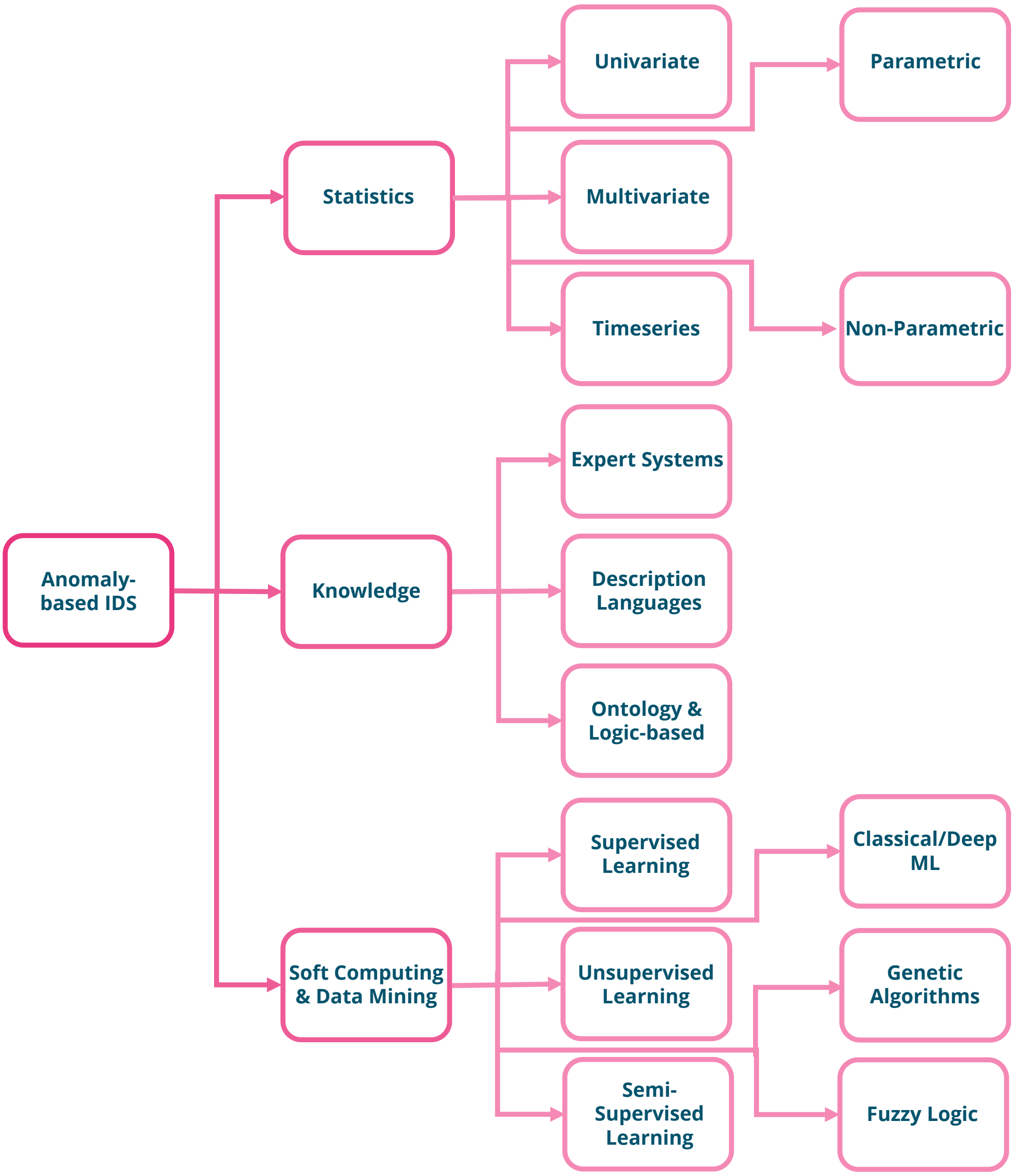


Figure 9: Categorisation of Anomaly Detection Systems

The focus of this work is specifically on "Machine Learning" models that sit under the **soft computing and data mining** category in Figure 8. These cover algorithms which undertake data-driven learning to learn characteristics enabling them to detect anomalies in previously unseen data. The **statistics-based** approach uses statistical properties such as mean, standard deviation, seasonality, etc., to build a statistical model/threshold for system behaviour. **Knowledge-based** approaches rely on prior knowledge from human experts, and previous experience, e.g., network traffic corresponding to attacks, etc. Figure 10 depicts the general concept of using an anomaly detection system – i.e., using inferred statistics, prior knowledge, or a model of system behaviour, to classify new data as an intrusion/ anomaly or not.

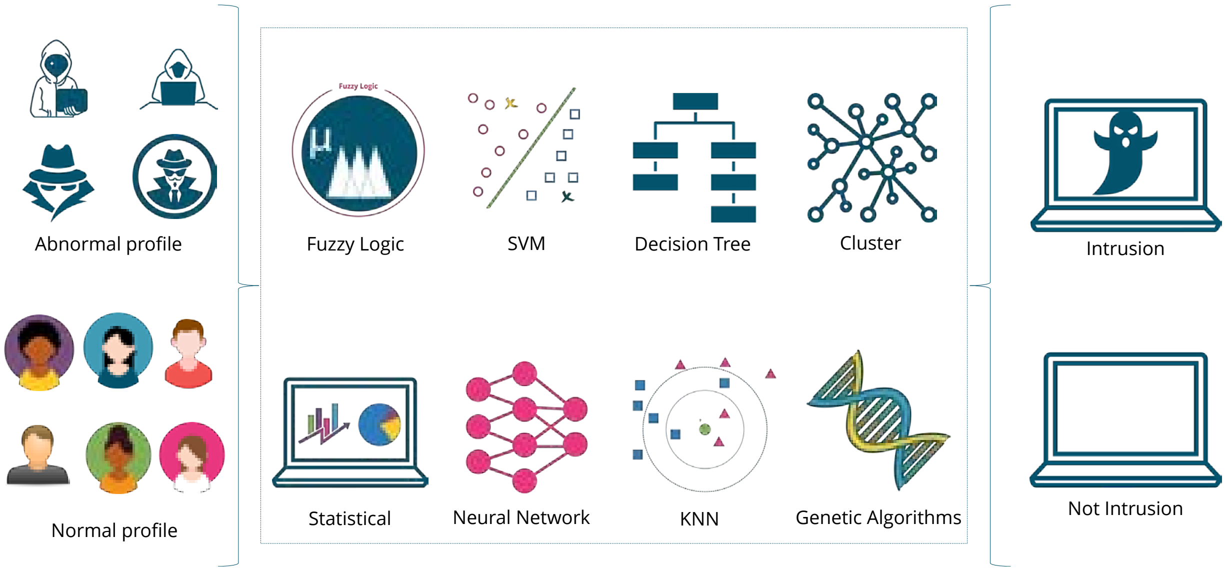


Figure 10: Concept of using anomaly detection systems. Adapted from [21].

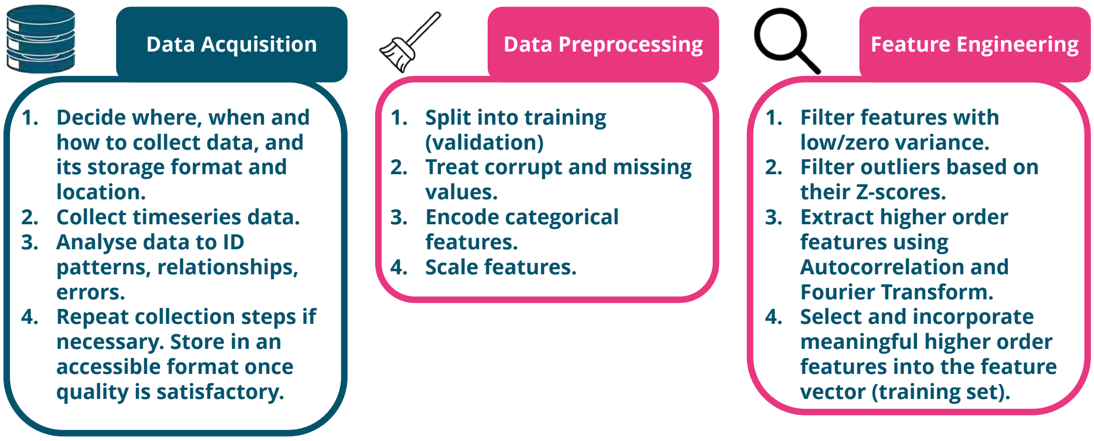
5. Application of Machine Learning

This section describes how to acquire and prepare a dataset for ML model development, the types of detection models available, what to consider when attempting to select one (or a set), how to train it, how to assess its performance, and how to continuously monitor it and make improvements.

5.1 Acquiring and Preparing Machine Learning Datasets

5.1.1 Data Acquisition and Exploratory Data Analysis

Following the adage: "garbage in, garbage out" used in data-driven scenarios, the key to effective anomaly detection is in the dataset used for learning.



Data Acquisition

1. Decide where, when and how to collect data, and its storage format and location.
2. Collect timeseries data.
3. Analyse data to ID patterns, relationships, errors.
4. Repeat collection steps if necessary. Store in an accessible format once quality is satisfactory.

Data Preprocessing

1. Split into training (validation)
2. Treat corrupt and missing values.
3. Encode categorical features.
4. Scale features.

Feature Engineering

1. Filter features with low/zero variance.
2. Filter outliers based on their Z-scores.
3. Extract higher order features using Autocorrelation and Fourier Transform.
4. Select and incorporate meaningful higher order features into the feature vector (training set).

Figure 11: Stages of data acquisition (blue) and preparation (green). Adapted from [45].

The steps of the data acquisition and preparation processes are shown in the Figure 11. The fastest way to build a prototype of a detection model is to use a ready-made dataset generated synthetically or collected as part of a different experiment in a related setting. Some useful online repositories for finding open-source datasets [46] are:

- Google Dataset Search [47]
- Amazon AWS Marketplace [48]
- Kaggle [49]
- Microsoft Research Open Data [50], and
- scattered GitHub repositories [51]

However, collecting and using own data is the best way to make the models' learning represent the unique use cases of systems more accurately. There are other security reasons for doing so – they are discussed in Section 5.7.

Three key aspects need to be considered:

1. **"where" to collect data from? :** Near safety-/security-critical assets/processes
2. **"how" often to sample? :** Related to the frequency of state changes of the system
3. **in "what" format to store the data? :** Either 2D tabular form, or 3D time-series format where the dimensions are [*number of samples, length of sequence, number of signals*].

Depending on the learning method which will be discussed in the next section, it may be necessary to collect a dataset that represents "normal" system behaviour – i.e., it is attack and fault-free.

Real-world cybersecurity testbed

An example is the widely used Secure Water Treatment (SWaT) dataset, created by the iTrust research centre in Singapore. They utilised the testbed as shown in Figure 12 for data generation. It demonstrates how the potential cyber-attack points were considered while instrumenting the testbed for data collection. The full dataset contains fifty-one tags (25 sensors & 26 actuators) sampled every second [52]. Many other public research datasets are also collected from running small-scale physical testbeds, or from simulations mimicking real-world processes.

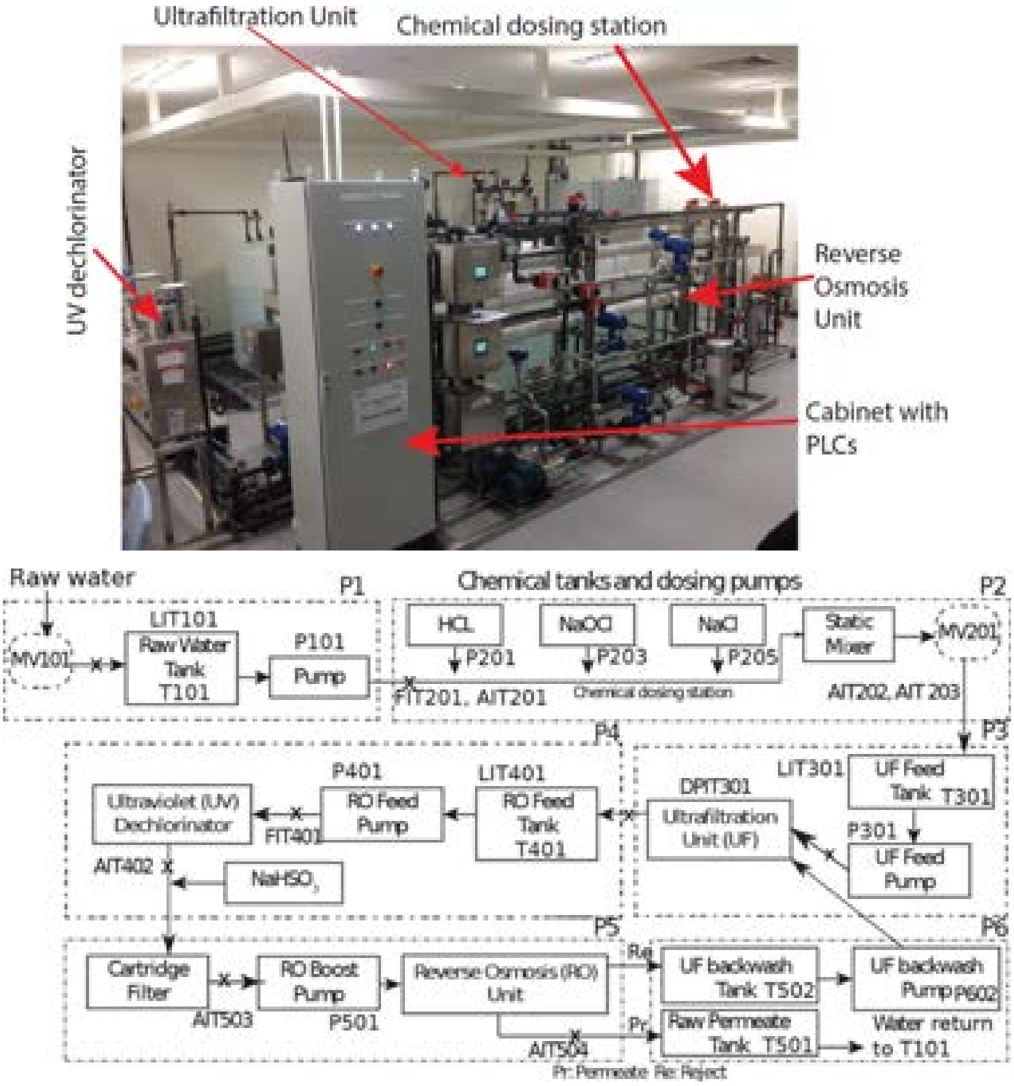


Figure 12: SWaT testbed used for creating their dataset. Physical water treatment process in SWaT and attack points used in the case study. P1 though P6 indicate the six stages in the treatment process. Solid arrows indicate flow of water or chemicals in the dosing station. Dashed arrows indicate potential cyber-attack points. LIT: Level Indicator and Transmitter; Pxxx: Pump; AITxxx: Property indicator and Transmitter; DPIT: Differential Pressure Indicator and Transmitter [52].

In most ICS scenarios, since the order of the data has significance (i.e., temporal correlations exist), it is recommended that the timestamp of sample collection is included as a feature in the dataset. This is referred to as the “timeseries” format. Further, while anomaly detection and ML-based classification on unstructured image/video data is a growing field, in the world of intrusion detection systems for cybersecurity, the datasets used are still predominantly numeric and can be easily displayed in tabular form, i.e., as structured data. Hence, the recommended format is a structured timeseries – this guideline will focus on this format.

There are three paths with respect to data collection which depend on the abilities to collect normal operational data (i.e., fault and attack-free) and abnormal data (faulty, attack) in isolation, and thereby affect machine learning modelling strategy (section 5.3). This is shown in Figure 13. For modelling purposes, the dataset would be comprised of one/two parts: a features dataset (signals) and, optionally, a label dataset (categorizing data into normal (0) or anomalous (1)). Note that, some open-source software libraries use a different labelling convention: -1 for anomalous, and 1 for normal data.

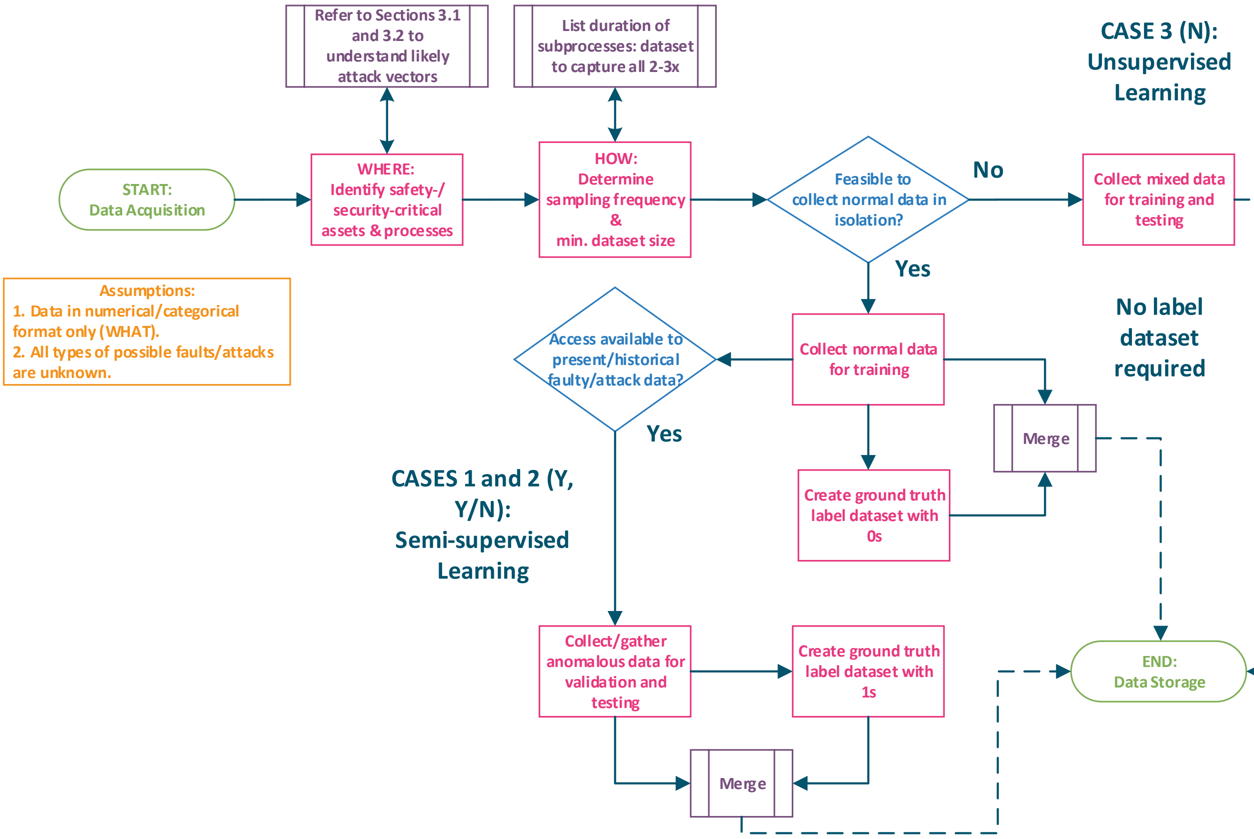

Refer to Sections 3.1 and 3.2 to understand likely attack vectors
List duration of subprocesses: dataset to capture all 2-3x
CASE 3 (N): Unsupervised Learning
START: Data Acquisition
WHERE: Identify safety-/ security-critical assets & processes
HOW: Determine sampling frequency & min. dataset size
Feasible to collect normal data in isolation?
No
Collect mixed data for training and testing
Yes
Assumptions:
1. Data in numerical/categorical format only (WHAT).
2. All types of possible faults/attacks are unknown.
Access available to present/historical faulty/attack data?
Collect normal data for training
No label dataset required
Merge
Yes
CASES 1 and 2 (Y, Y/N): Semi-supervised Learning
Create ground truth label dataset with 0s
Collect/gather anomalous data for validation and testing
Create ground truth label dataset with 1s
END: Data Storage
Merge

Figure 13: Data acquisition process decisions with recommendations regarding ML strategy

The term “Exploratory Data Analysis” (EDA) in data science refers to the initial process of analysing and investigating the acquired data using summary statistics and data visualisation techniques in order to discover patterns, spot outliers, test hypotheses/check assumptions and identify potential methods of modelling. For example, it would be useful to compute the following on the dataset: length of the dataset and corresponding timestamp range, the datatypes (e.g., integer, floating-point, Boolean/categorical, etc.), number of signals/features, range and measures of central tendency (mean/median) for each feature, number of missing values, and number of outliers/if any (visually).

An important part of the data acquisition process is ensuring that the collected data meets the required quality standards by either EDA or by running validation/consistency checking scripts. For example, if there is a requirement for the data to represent only normal behaviour, at the time of data collection – were there any abnormal activities such as system/component faults, maintenance activities, or exceptionally, a security attack. Some stages of the data acquisition process may need to be repeated as **the integrity and veracity of the training dataset are particularly vital for IDS**. An important assumption, valid in most cases, is that the defender does not know of all possible faults and attacks which could affect their system.

Finally, the validated data is stored on the cloud or on premises servers, within a data warehouse (more likely for structured data), a data lake or elsewhere, in an accessible manner. While this section focusses on developing the models for the first time, the same pipelines and methodologies would apply after model deployment (only inferences, no training). However, post-deployment (Section 5.6), the quality checks with regards to eliminating outliers on the collected (“testing”) dataset tend to become less rigorous as the model has already learned to identify anomalies.

Refer to Figure 11 again for a summary of the different sub-stages of data preparation: data pre-processing, and feature engineering (filtering and extraction). Specifically, for ICS datasets, there are some special statistical characteristics that need to be considered when attempting anomaly detection:

- Due to the repetitive nature of the process, there is a need to preserve the temporal coherence while splitting the dataset and training the model.
- Due to the high interdependence of devices in ICS, a high correlation between dataset features (columns) is desired.

5.1.2 Data Preprocessing

As seen in Figure 13, **there is a cyclic dependency between model choice and data collection pipeline and therefore, they would need to be considered in parallel.** For example, with unsupervised ML, there is no need for a separate training set or for the samples to be labelled in advance. An unsupervised model would learn to separate normal from anomalous instances by itself. On the other hand, with supervised ML/semi-supervised ML, there is a need for separate and labelled training, validation, and testing datasets. More details about this are provided in Section 5.3.

If dataset splitting is required, the rule of thumb is to split the dataset in a ratio of 80% (train): 20% (test; validate). The train/test split would need to happen first; subsequently, the new training set is once again split to create a validation set. After splitting the dataset, it is recommended that incomplete or corrupt time-series sequences are addressed. If the Python language is selected, there are some easy-to-use, in-built tools within the *Pandas* library to deal with them.

Features could either be numeric or categorical. In case they are categorical, to be able to be understood by a Machine Learning algorithm, they need to be encoded – i.e., into binary values split over multiple features based on the number of possible categories (“one-hot-encoding”); or ordinally (ordinal encoding). Finally, since different features could be measured in different units, it is recommended that the datasets are standardised or min-max normalised based on the training dataset’s statistics. However, this would once again depend on the choice of algorithm and the distribution of the data:

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation}(x)} \qquad \text{OR} \qquad x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

5.1.3 Feature Engineering

Two stages of feature engineering are used to prepare the training dataset for machine learning. The first stage is filtering where the goal is to eliminate features/data points that do not change in the entire dataset or conversely, those whose statistical distributions in the dataset partitions vary significantly. Note that these operations need to be carried out without snooping bias based on the category of ML detection algorithm, i.e., selecting models to employ based on the test set (inadvertently maximising performance).

For features which do not vary much or in contrast vary significantly, variance thresholds/z-score (Z)-based filtering for individual points can be utilised.

$$\text{if } Z \begin{cases} \geq 3 : \text{discard} \\ < 3 : \text{keep} \end{cases} \quad \text{where } Z = \frac{x - \text{mean}(x)}{\text{standard deviation}(x)}$$

The second and final stage (optional) is related to extracting higher order features from the existing set of features with the help of expert knowledge. One possibility is to extract the repetitive actions that the ICS performs in the form of additional features using autocorrelation and the Fourier Transform.

$$autocorr_{x_w,k} = \frac{\sum_{i=w-W+1}^{w-k}(x_i - \text{mean}(x))(x_{i+k} - \text{mean}(x))}{\sum_{i=w-W+1}^{w}(x_i - \text{mean}(x))^2}$$

Autocorrelation is defined as the correlation of a signal with a delayed version of itself, as a function of *k* (lag parameter). For a timeseries, the delay or lag can be applied over a time window *W*. Formally, autocorrelation can be defined as shown above, where x_w is the value that the feature takes at instant $_w$. Autocorrelation could be applied to find repeating patterns in the signal from sensors and actuators.

The Fourier Transform is a mathematical tool that allows us to decompose a signal into the frequencies from which it is formed, i.e., convert from the time domain to the frequency domain. The Discrete Fourier Transform (DFT) could be used with a numeric timeseries feature as in the ICS case where x_w is the width of the wth sample of *x* and *W* is the sample count.

$$\overline{DFT}_{x_w,k} = \sum_{j=w-W+1}^{w} x_j e^{-\frac{2\pi i}{W}k(j-(w-W+1)}$$

Together, these latter two techniques are meant to provide information about the periodicity of the signal, in terms of each feature. Each of them produces a vector output for each feature to which statistical methods such as mean, standard deviation, minimum, maximum, and range can be applied. The result is 10 new features (5 each), which can be added to the dataset after inspection.

5.2 Types of Anomalies

Anomalies (or outliers – used interchangeably) are often defined as data instances that significantly deviate from most other instances in the dataset. Following a behaviour-driven taxonomy proposed by [53], anomalies can be divided into two branches as described below. Refer to Figure 14 and Figure 15 for visual representations:

1. Point-wise:

a. *Global*: Points ("spikes") that significantly deviate from the rest of the points.

b. *Contextual*: Points that deviate from their corresponding context, where context is defined as neighbouring samples within a certain range, with respect to time.

2. Pattern-wise:

a. *Shapelet*: Sub-sequences with dissimilar shapelets compared with the normal shapelet. A shapelet is a small and unique (sub-sequence) descriptor of a time-series.

b. *Seasonal*: Subsequences with unusual seasonalities compared with the overall seasonality. Retains the same shapelet and trend (long-term movement pattern).

c. *Trend*: Subsequences that significantly alter the trend of their parent time-series, resulting in a permanent change of the mean. Retains the shapelet and seasonality.

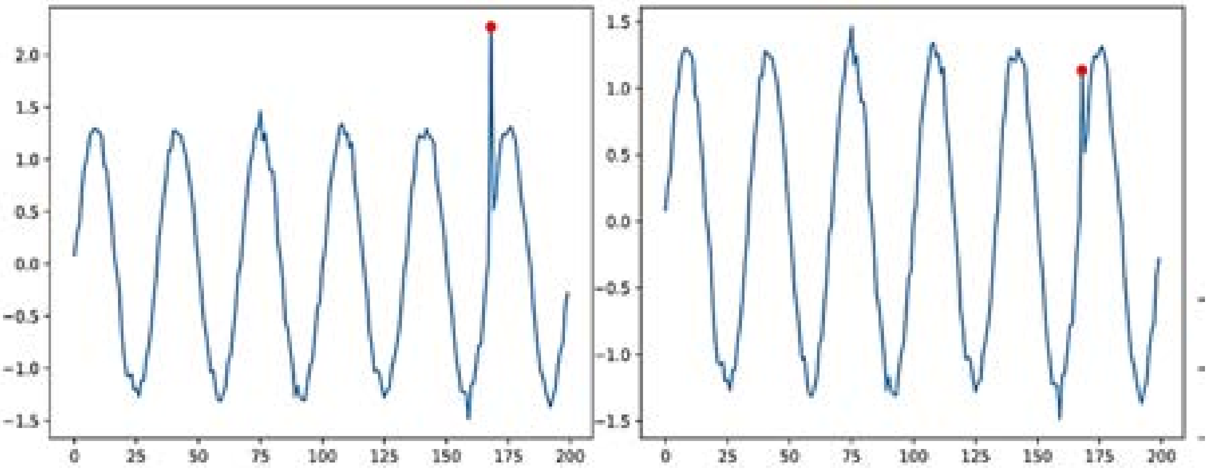


Figure 14: Point-wise anomalies: Global (Left) and Contextual (Right) [53].

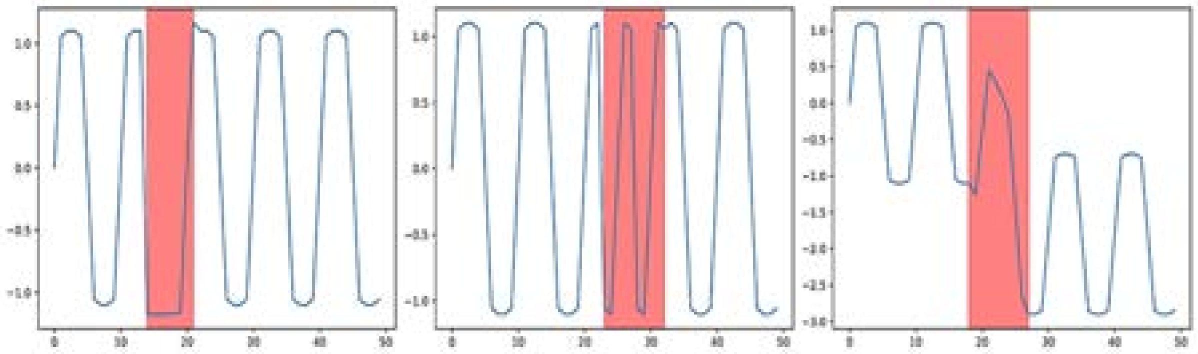


Figure 15: Pattern-wise anomalies: Shapelet (L), Seasonal (Middle), and Trend (Right) [53].

5.3 Types of Detection Models Available

This section provides details about the types of detection models available with a focus on the likely profile of the dataset as shown in Table 6.

Property	Value
Temporal	Timeseries (Time Period ranging from minutes to hours)
Number of features	Multivariate with high number of features (>20)
Data types	Combination of continuous and discrete (categorical)
Volume	High (likely to be growing due to frequent sampling)
Stationarity (of mean)	Yes, however, operational drift possible resulting in moving mean
Purity	Contaminated with outliers like faults (depending on collection range)
Correlations	High degree of inter-feature-correlation

Table 6: Possible profile of collected ICS dataset

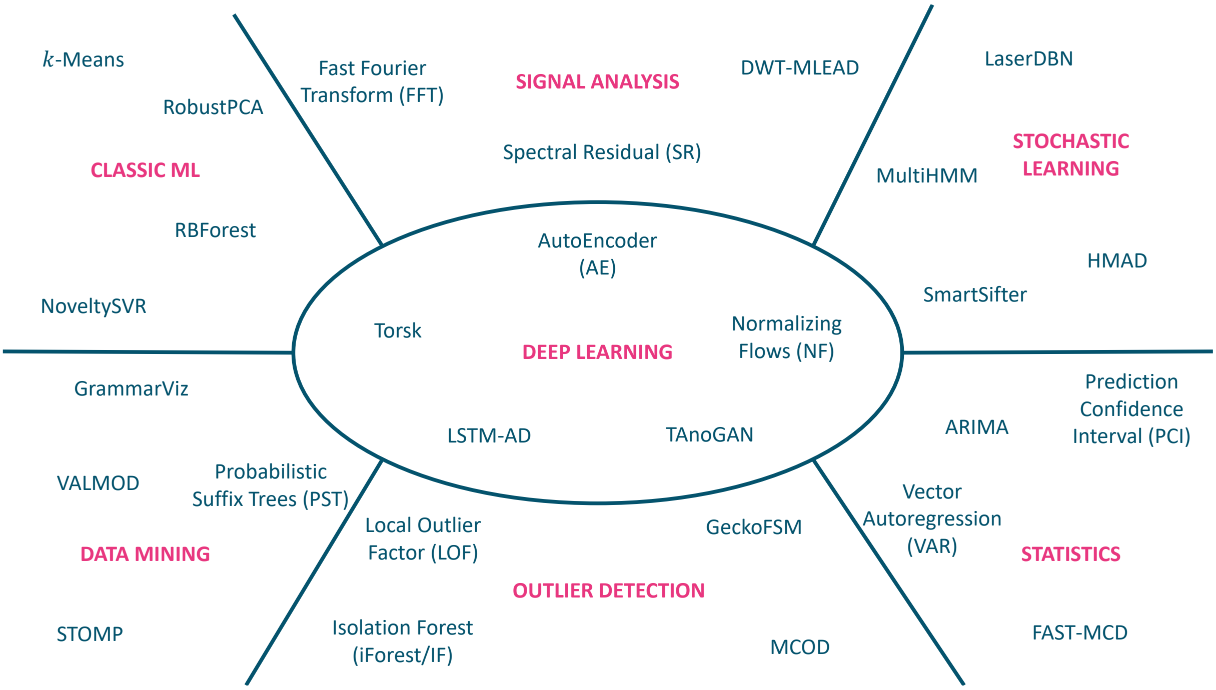


Figure 16: Families of anomaly detection algorithms with popular examples. References can be found in Table 7.

There is an ocean of possible timeseries anomaly detection as shown in Figure 16, and they can be categorised along different dimensions. Only one such method is utilised in this document for clarity.

Firstly, there are two key dimensions along which models can be classified: dimensionality (univariate and multivariate), learning method (supervised, semi-supervised and unsupervised). Where, dimensionality of the data refers to the number of features in the timeseries dataset, and learning method refers to the presence of labels (normal or anomalous) in the dataset which in turn corresponds in the way the model is trained. Note the earlier recommendations made in Figure 13. The difference between them is as follows:

- **Supervised Learning**: Requires labelled training data as shown in Figure 15. For each instance of the training dataset, this method uses *n* features from the feature vector *X*, i.e., [x_1,x_2,x_3,x_4,...,x_n], to learn the class variable (label *Y*). Hence, as with other categories, the model learned refers to a function f mapping from vector *X* to *Y* such that *Y=f(X).*
- **Unsupervised Learning**: Uses feature vector *X* without a corresponding class/ label dataset *Y*. The assumption is that the dataset contains both 'normal' and 'anomalous' data and the algorithm would need to appropriately discriminate between them. The trained model can then be used to make inferences on real-time data at a sample-by-sample basis (Fig. 16).
- **Semi-Supervised Learning ("some supervision")**: This is the category which sits between supervised and unsupervised learning, i.e., the training dataset involves a small number of labelled examples and a large number of unlabelled examples. In one form, training is split into three phases as seen in Figure 19. This is particularly relevant to practical scenarios where there may be a large quantity of data, but not a viable means to label each sample. **However, the form of semi-supervised learning of relevance in this report is referred to as "novelty detection" wherein the training dataset represents normal operation (with limited/no outliers) only. The models then search for "novelties" at test time**.

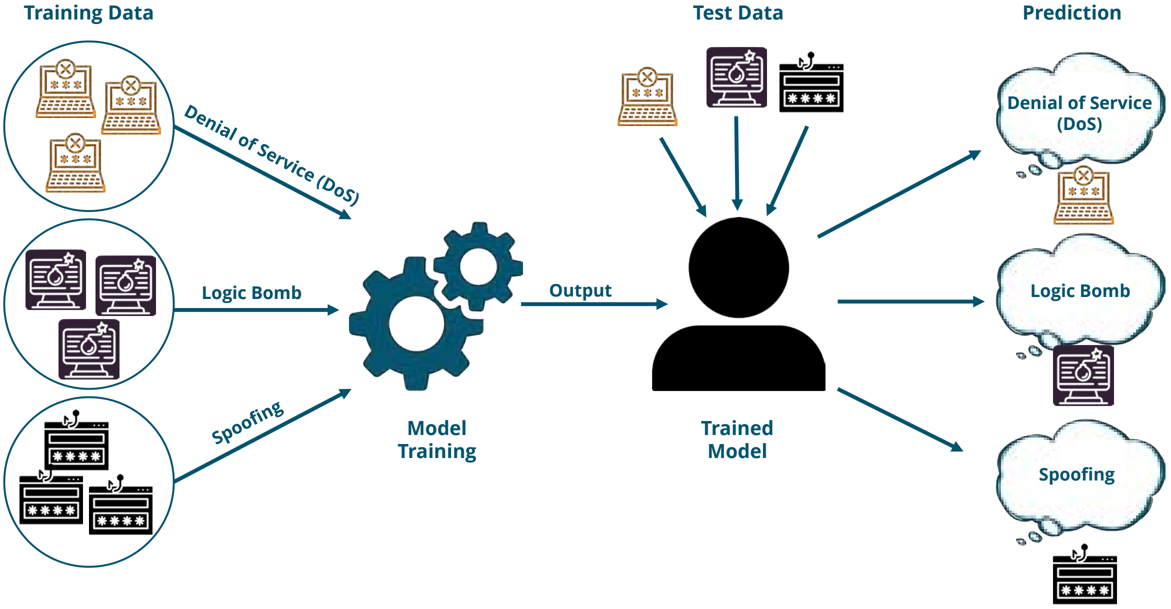


Figure 17: Depiction of supervised learning

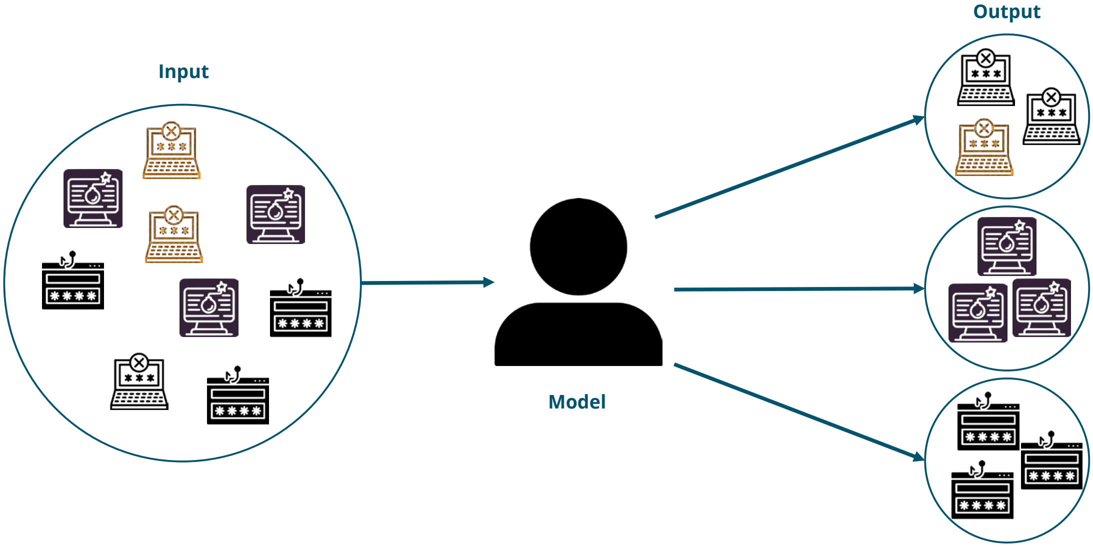


Figure 18: Depiction of unsupervised learning

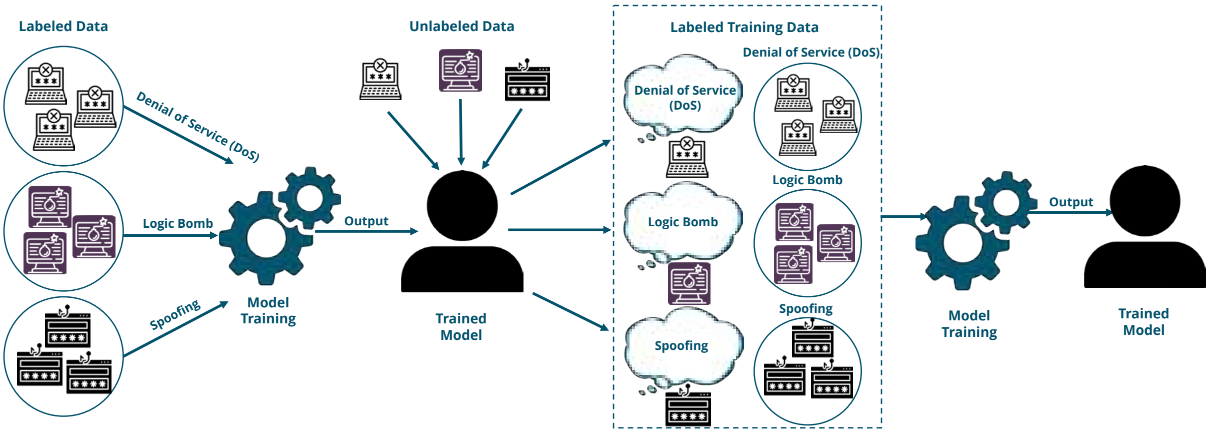


Figure 19: Depiction of semi-supervised learning (traditional)

In Machine Learning, more generally, there is another family of learning methods called reinforcement learning, however, anomaly detection tools using this approach are less common.

The third and final dimension which will be used to classify ML-based anomaly detectors is based on how they categorise data points methodically. Figure 20 depicts this categorisation.

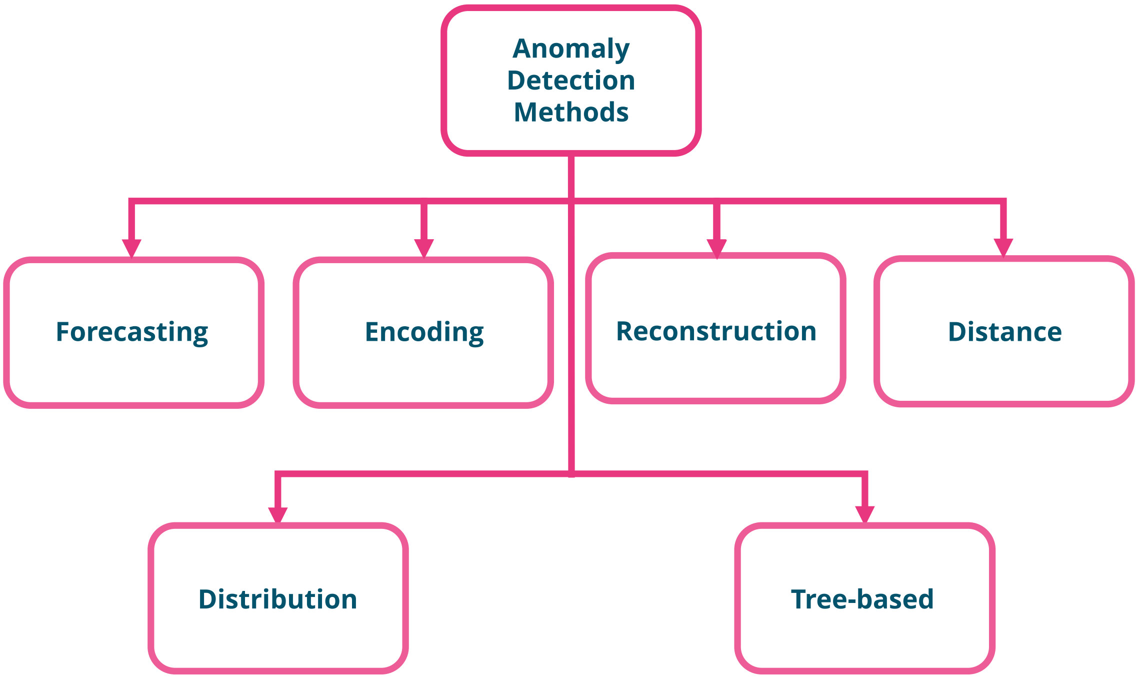


Figure 20: Techniques for timeseries anomaly detection

For each of these categories (references in Table 7), the most common training methodology is presented in parenthesis:

- **Forecasting (Semi-supervised):** Use a learned model to forecast several steps based solely on the other timeseries data points in the preceding context window. Recurrent Neural Network (RNN)-based models and ARIMA are examples.
- **Encoding (Semi-supervised):** Build a model of normal behaviour by learning to convert normal training timeseries subsequences into lower dimensional latent space (encoding). Anomaly scores are then computed based on these latent space representations. GrammarViz, LaserDBN, and MultiHMM are examples.
- **Reconstruction (Semi-supervised):** Like Encoding-based models, however, they also subsequently use the learned model to reconstruct other encoded

subsequences when the model is deployed (testing phase) which are compared against the original, observed values. Some examples are Autoencoder and Generative Adversarial Network (GAN)-based models.

- **Distance (Unsupervised):** Use specialised distance metrics to compare points or subsequences from one timeseries with another. Generally, anomalous subsequences are expected to have larger distances. Several classical ML algorithms such as K-Nearest Neighbours (KNN), Local Outlier Factor (LOF), One-Class SVM (OCSVM [55]; normally, semi-supervised) are in this category.
- **Distribution (Unsupervised):** Estimate the distribution of the data or fit a distribution model to the dataset. Abnormality is assessed based on frequency (probabilities, likelihoods, distances to prior). Some examples include COPOD, Fast-MCD, etc.
- **Tree-based (Unsupervised):** Build an ensemble of random trees that attempt to partition the points or subsequences of the test timeseries through recursive selection of random features and split values. Abnormality is based on the number of splits required to isolate the point – tends to be easier to isolate anomalies. Versions of Isolation Forest (iForest) fall into this category.

Most of these algorithms rely on comparing (i.e., similarity) prediction/ reconstruction/ encoding, etc. with an expected value. To do this, they systematically compute an anomaly score. For example, forecasting-based methods use the trained model to predict some future samples ((x_i)) and then compare them with the observed test values (x_i). A subsequence $S=x_p,\ldots,x_{p+n-1}$ of length *n* is therefore anomalous if:

$$\sum_{i=p}^{p+n-1}\|x_i - \hat{x}_i\| > \tau, \text{ where } \tau \text{ is a pre-defined (tuneable) threshold}$$

There are several approaches for selecting a value for *τ*; most often it is considered a hyperparameter to be adjusted during validation to minimise error-rate.

Dimensions	Learning Type	Method	Area	Family
Multivariate	Unsupervised	k-Means [56]	Classical ML	Distance
Multivariate	Semi-supervised	RobustPCA [57]	Classical ML	Reconstruction
Univariate	Unsupervised	NoveltySVR [58]	Classical ML	Distance
Multivariate	Supervised	XGBoost [59]	Classical ML	Trees
Univariate	Unsupervised	GrammarViz [60]	Data Mining	Encoding
Univariate	Unsupervised	VALMOD [61]	Data Mining	Distance
Univariate	Unsupervised	PST [62]	Data Mining	Trees
Univariate	Unsupervised	STOMP [63]	Data Mining	Distance
Univariate	Unsupervised	ARIMA [64]	Statistics	Forecasting
Multivariate	Unsupervised	VAR [65]	Statistics	Forecasting
Multivariate	Semi-supervised	Fast-MCD [66]	Statistics	Distribution
Univariate	Unsupervised	PCI [67]	Statistics	Reconstruction
Univariate	Unsupervised	Sub-LOF [68]	Outlier Detection	Distance
Multivariate	Unsupervised	iForest [69]	Outlier Detection	Trees
Multivariate	Unsupervised	COPOD [70]	Outlier Detection	Distribution
Multivariate	Supervised	HIF [71]	Outlier Detection	Trees
Multivariate	Supervised	MultiHMM [72]	Stoch. Learning	Encoding
Multivariate	Semi-supervised	LaserDBN [73]	Stoch. Learning	Encoding
Multivariate	Semi-supervised	HMAD [74]	Stoch. Learning	Encoding
Multivariate	Unsupervised	SmartSifter [75]	Stoch. Learning	Distance
Univariate	Unsupervised	FFT [76]	Signal Analysis	Reconstruction
Univariate	Unsupervised	SR [77]	Signal Analysis	Reconstruction
Univariate	Unsupervised	DWT-MLEAD [78]	Signal Analysis	Distribution
Multivariate	Unsupervised	Torsk [79]	Deep Learning	Forecasting
Multivariate	Supervised	NF [80]	Deep Learning	Distribution
Multivariate	Semi-supervised	TAnoGan [81]	Deep Learning	Reconstruction
Multivariate	Semi-supervised	AE [82]	Deep Learning	Reconstruction
Multivariate	Semi-supervised	LSTM-AD [83]	Deep Learning	Forecasting

Table 7: Examples of Univariate and Multivariate anomaly detectors belonging to different families

Although most of the algorithms highlighted so far work with semi-supervised and unsupervised learning methods, there are several options for supervised learning which can simply be considered a two-class/multi-class classification problem. Some examples are Support Vector Machines, Random Forest, and Deep Learning algorithms such as Dense Feed-forward and RNN-based networks. Table 7 shows some more examples of timeseries anomaly detectors paired along with the family of algorithms and area they come from, with the first two dimensions on the left.

5.4 Selecting a Detection Model

Several review papers [84], [85], [86], [54], [87], [88] have attempted to benchmark the performance of these different categories of algorithms periodically. The common conclusion from all their works can be summarised in one line: "There is no one-size-fits-all problems algorithm". **Each algorithm has its own strengths and weaknesses. As stated before, the purpose of this work is to allow decision-makers to ask the right questions and to have a birds-eye view when talking about anomaly detection algorithms**. Therefore, instead of recommending a particular learning method (supervised, semi-supervised or unsupervised) and a particular algorithm, this section will suggest different aspects to consider when making this decision – refer to Table 8.

One of these reviews' result is discussed for reference [53]. It found that classical (ML and statistical) algorithms outperform Deep Learning (DL) models in most of the real-world datasets they considered, e.g., Autoregressive (AR), iForest, OCSVM. Particularly, GANs (GANs) were unable to detect any outliers due to the complexity of real-world anomalies. Note that, in this work, "real world datasets" refers to the credit-card fraud detection, IoT for drinking water monitoring, server attack monitoring, and extreme space weather detection.

Operations **1.** Ease of deployment & maintenance {5.6}. **2.** Model size {5.7}. **3.** Inference time {5.6}. **4.** Ease of retraining {5.6} **5.** 5Usability/Interpretability {5.5.2}	**Performance {5.5.2}** **1.** High performance metric (accuracy) to detect relevant anomaly types. **2.** Acceptable number of False Positives and False Negatives to prevent information overload.
Data **1.** Amount of data required. **2.** Labelling effort. **3.** Ability to work with lots of features. **4.** Ability to work with a large dataset.	**Robustness** **1.** Performance with unseen anomaly types – e.g., zero-day attacks {5.7}. **2.** Performance with limited training data. **3.** Resilient to adversarial attacks (data poisoning and noise) {5.7}.

Table 8: Decision matrix for selecting anomaly detectors. References to future sections in curly braces.

Other noteworthy points from common knowledge amongst the community is that OCSVM is memory efficient, works well with large number of features, however, it struggles with long training times with large datasets, and often do not meet real-time performance requirements. On the other hand, KNNs are simplest to implement but struggle with computational inefficiency. A visual example is shown here of how one algorithm's strengths are another's weakness in Figure 21.

The advice for selecting the most appropriate ML model(s) is to try as many of them as possible, and then decide which would work best based on the decision matrix provided in this section. It must also be emphasised that the model selection process could conclude by saying that multiple models are needed to be used in parallel. This same framework applies in case they are deployed independently (without interaction). .

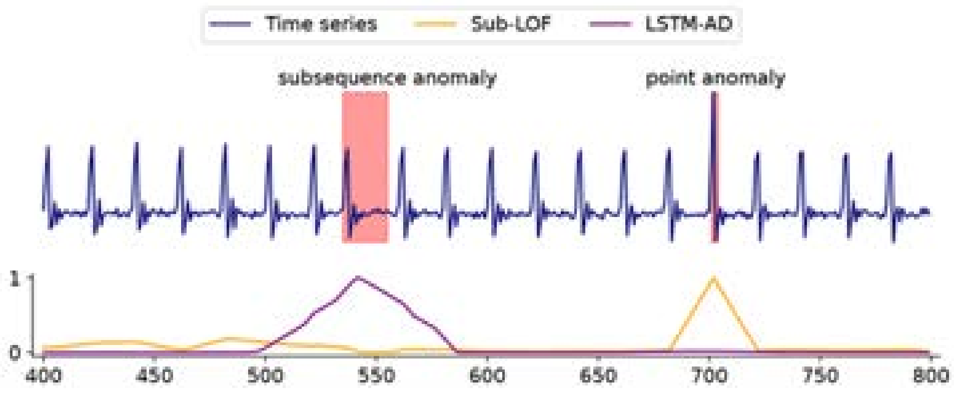
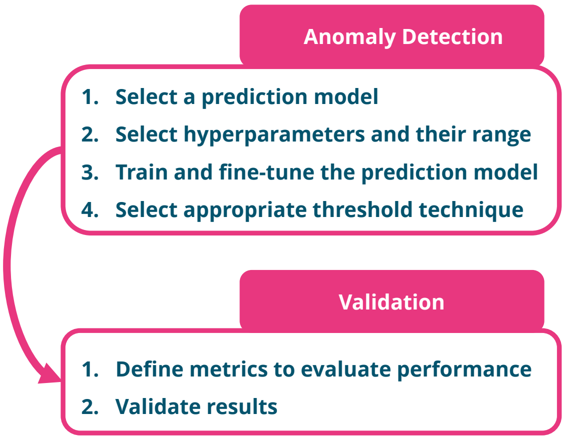


Figure 21: Synthetic ECG data shows how Sub-LOF detects point anomaly, and LSTM-AD detects subsequence anomaly, but not vice-versa. Plot on the bottom shows the computed anomaly score computed against time (x axis).[89]

5.5 Training the Detection Model and Assessing Performance

5.5.1 Training and Validation Plan

With regard to implementing the shallow Machine Learning-based models, there are several off-the-shelf methods of utilizing them – the most common of which is the trusted scikit-learn (sk-learn) library in Python. Deep Learning models on the other hand have several open-source implementations on GitHub, however, they would need to be carefully selected after inspection, or built from scratch. Figure 22 provides an overview of the training and validation process.


Anomaly Detection

1. Select a prediction model
2. Select hyperparameters and their range
3. Train and fine-tune the prediction model
4. Select appropriate threshold technique

Validation

1. Define metrics to evaluate performance
2. Validate results

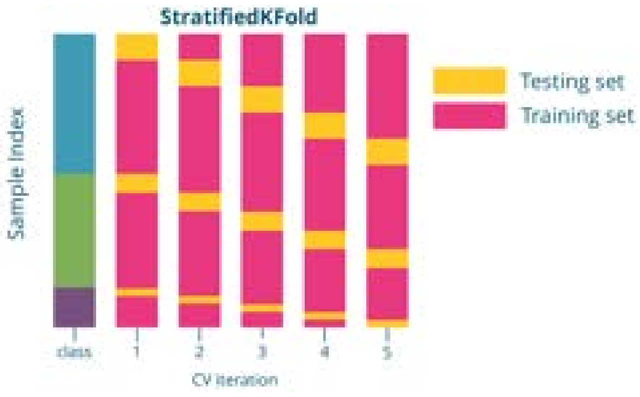


Figure 22 (Left): Approach for Supervised and Semi-Supervised Learning [45] & Figure 23 (Right): Depiction of Stratified K-Fold Cross-Validation (CV) [12] for multi-class problems [90]

Staying with the example of the ready-to-use Machine Learning models available from *sk-learn*, this library provides separate functions which can be called for fitting the model and making predictions with the trained model on the test dataset. Each of these models tend to have hyperparameters which could be tuned according to the dataset to maximise performance.

With supervised learning models (fully labelled dataset) and semi-supervised learning, cross-validation is the commonly used method to fine-tune hyperparameters. Since there is considerable class imbalance in ICS datasets, i.e., only a small number of anomalous samples, it is recommended to perform Stratified K-Fold Cross Validation. Figure 22 shows an example where the data is sorted according to the class labels and how this method preserves the class frequencies in each "fold".

With semi-supervised learning (novelty detection format) the formulation often utilised for anomaly-based intrusion detection is one-class classification, i.e., the model is trained the learn the normal behaviour (one class) of the system only, and its parameters (including detection threshold τ) are tuned according to a mixed dataset with anomalies.

However, with unsupervised learning (unlabelled dataset), it is not feasible to tune hyperparameters as there is no prior knowledge about which samples/ sequences are normal or anomalous. In this case, there is no conventional validation step. Hence, this does not require a separate validation subset. This format is also referred to as "outlier detection" since the problem is now about being able to detect any observable (by the model) outliers from a dataset. The training process therefore simply consists of fitting the model to a training dataset to allowing it to form a decision boundary(ies) between the two classes.

5.5.2 Assessing Model Performance

As noted earlier, anomaly detection is an imbalanced problem, i.e., 90+ percent of the time, the system is operating as "normal", and only the small remaining percentage of the time faults and anomalies, which would raise alarms, can be seen. Therefore, suitable metrics must be selected to account for this.

Some preliminaries used in the following definitions:

- **True Positive (TP):** Number of anomalies properly detected.
- **True Negative (TN):** Number of non-anomalies properly classified as

anomalous.

- **False Positive (FP):** Number of non-anomalies wrongly classified as anomalous.
- **False Negative (FN):** Number of anomalies wrongly classified as non-anomalous.

Based on these definitions, the most used metrics for performance assessment are:

1. **Precision (Pr)**: "how many of the detected anomalies are anomalous".

$$\mathrm{Pr} = \frac{TP}{TP + FP}$$

2. **Recall (Re)**: "how many of the anomalies are detected". Also called sensitivity.

$$\mathrm{Re} = \frac{TP}{TP + FN}$$

In isolation, Precision and Recall do not provide the complete picture of detector performance. For instance, it is possible to achieve 100% recall by classifying every item as anomalous, and it is possible to achieve 100% precision by detecting only a small number of extremely likely anomalies as anomalous. Therefore, other metrics which combine these metrics as shown below are necessary.

3. **F1 Score (F1)**: Combines and equally balances precision and recall through their harmonic mean. A better model will have a higher F1 score and vice-versa.

$$F1 = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

4. **Area Under Curve – Receiver Operating Characteristic (AUC-ROC)**: This is a performance measurement at various classifier threshold (τ) settings. ROC is a probability curve, and AUC represents the degree of separability between the anomalies and the normal instances. It is plotted as False Positive Rate (FPR) versus True Positive Rate (TPR). An excellent model has AUC near 1 and vice-versa. Figure 24 shows a scenario where the probability distributions of normal and anomalous samples overlap leading to some FNs and FPs. There are several approaches such as using the Geometric Mean of sensitivity and specificity to tune the detection threshold.

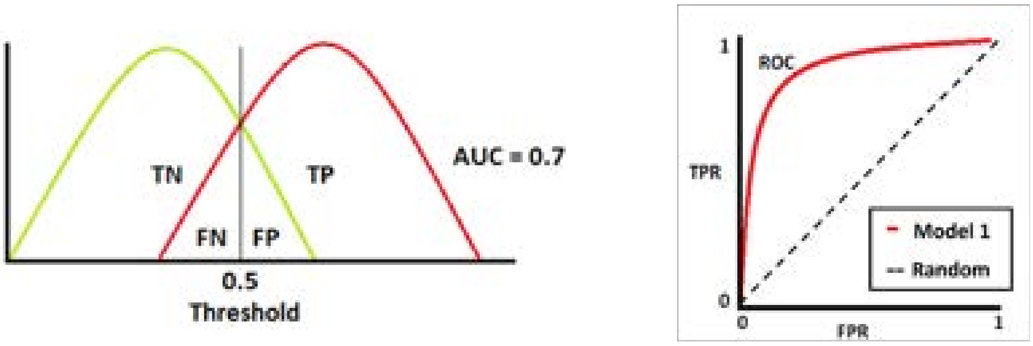


Figure 24: Impact of overlapping distribution of two classes affects the AUC curve [91]

$$TPR = Recall$$

$$Specificity = \frac{TN}{TN + FP}$$

$$FPR = 1 - Specificity = \frac{FP}{TN + FP}$$

Alternatively, a Precision-Recall (PR) AUC can also be used. The decision about which to use would depend on the quantity of anomalous data (faults, attacks, etc) available. For instance, ROC-AUC can be optimistic on severely imbalanced datasets. Therefore, it would be better than PR-AUC for cases with more anomalous data and vice-versa. Beyond these, there are several OT-specific metrics which could be created ad-hoc to measure early detection, inference time, etc. in accordance with Table 8.

Global Intepretation

Being able to explain the conditional interaction between dependent (response) variables and independent (predictor or explanatory) variables based on the complete dataset. Helpful in explaining the context of the decision classification.

Local Intepretation

Being able to explain the conditional interaction between dependent (response) variables and independent (predictor or explanatory) variables for a single row or a subset of rows. Helpful in identifying local trends and intuitions.

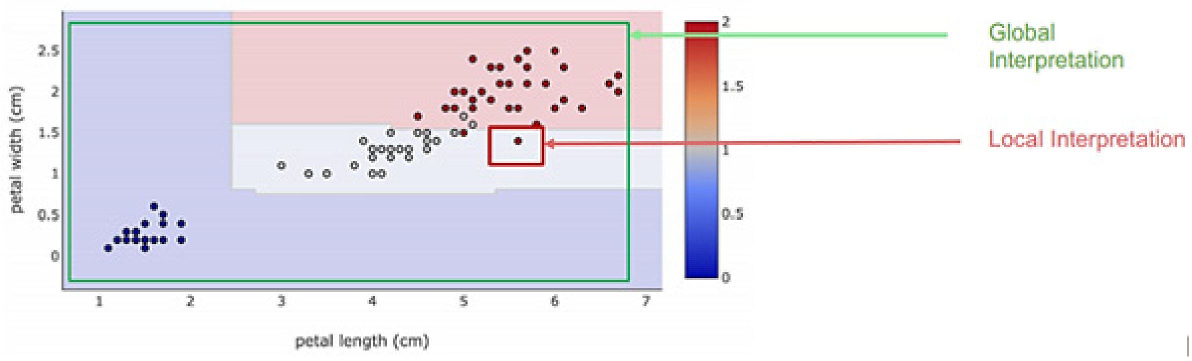


Figure 25: Global and local levels of model interpretability [92]

The final aspect to be considered when assessing model performance is its interpretability. In the avenue of anomaly detection for cybersecurity, interpretability of models is vital as it allows an operator/analyst to select suitable countermeasures where appropriate and appropriately direct their attention. There are three pillars of interpretation:

1. Transparency.
2. Ability to question.
3. Ease of understanding

→ Being able to understand why the model made a certain prediction, i.e., causality.

Further, interpretability can be assessed at a global level and a local level as shown in Figure 25.

Unfortunately, there tends to be a trade-off between model accuracy and interpretability as shown in Figure 25. However, Explainable AI (or XAI) is a growing field which is working to eliminate the perception of ML models (particularly DL) as “black box” predictors. There are currently three different approaches which can be used to maximise usability of the discussed ML algorithms:

- **Data Visualization:** Dependent on the number of features, it could be possible to map the model predictions back to the feature set. Dimensionality reduction techniques may need to be employed. A simple example of a three-class classification of flower based on dimensions of petal and sepal length and width, i.e., 4 features.
- **Model Selection:** There are some models which are easier to interpret than others. An example is Random Forest (Ensemble) which can clearly associate its prediction with feature importance values.
- **Separate Tools (“Post-hoc”)**: Work is being done to use mathematics and game theory to associate a feature’s “contribution” to a model prediction independently. Examples of this are Local Interpretable Model-Agnostic Explanations (LIME), Shapely Additive Explanations (SHAP), SKATER (e.g., in Figure 26), etc.

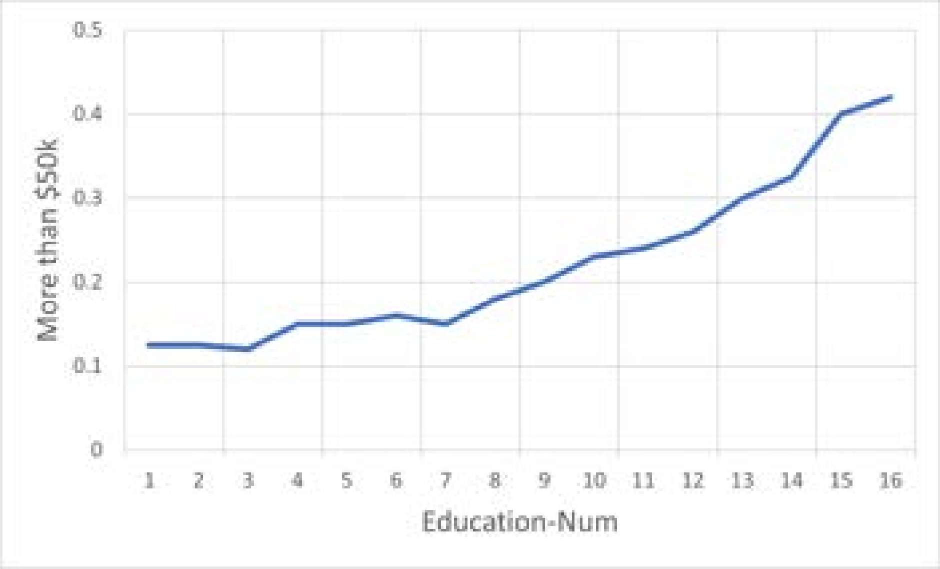


Figure 26: Example application of SKATER: Partial Dependence Plot of the relationship between Number of years of Education vs Probability of Income More than $50k. [93]

5.6 Deploying and Maintaining the Model

Model deployment refers to the phase of the model's lifecycle where it is operationally used for securing the ICS. Through discussions with industrial partners and a review of available commercial options, it was observed that there are three possible locations where the model can be deployed:

1. **Cloud**: This method involves sending the process dataset (and/or network dataset) to the cloud where the data preparation pipelines, and machine learning algorithms can be implemented.
2. **Fog**: Decentralised computing (multiple "nodes") placed close to the edge. A virtual machine may need to be instantiated on a fog device to run analysis.
3. **Edge**: Closest possible to the edge devices. Loosely connected structure whereby these devices tend to work with the data independently.

	Cloud	Fog	Edge
Latency	High	Medium	Low
Scalability	High, easy to scale	Easy within network	Hard to scale
Distance from Edge	High	Close to edge	Zero, at the edge
Analysis	Less time-sensitive data processing; permanent storage.	Real-time; flexible – sends data to cloud or processes locally	Real-time; allows instantaneous decision-making
Compute	High	Limited by device	Very limited
Interoperability	High	High	Low

Table 9: Comparison options for anomaly detector deployment [94], [95]

These fundamental differences, as shown in Table 9, inherently make these options be better suited to different use cases discussed next. Focussing on IDS, due to the limited compute and data storage capabilities of the Edge devices, the candidates for deploying Machine Learning-based anomaly detection will be the Cloud and Fog levels. The final choice will be once again specific to the requirement of the cybersecurity operation. Some scenarios are presented below. Note that time-criticality is closely linked to safety-criticality.

- **security-critical AND time-critical:** Deploy in Fog layer. Concentrate and process data locally to drive any necessary control response to ensure safety.
- **security-critical AND NOT time-critical:** Deploy either in Fog or Cloud layers dependent on the nature and size of the dataset, and the type of model selected for analysis.
- **NOT (security-critical OR time-critical):** This could be considered as the Optimisation case. Once again, this can be done either in the Fog or Cloud (preferred) layers.
- **NOT security-critical AND time-critical:** This could be considered as the Fault Detection case, but not all faults are time-critical. Due to time constraints, it is recommended that the model is deployed at the Fog layer so that the operator can respond as soon as possible.

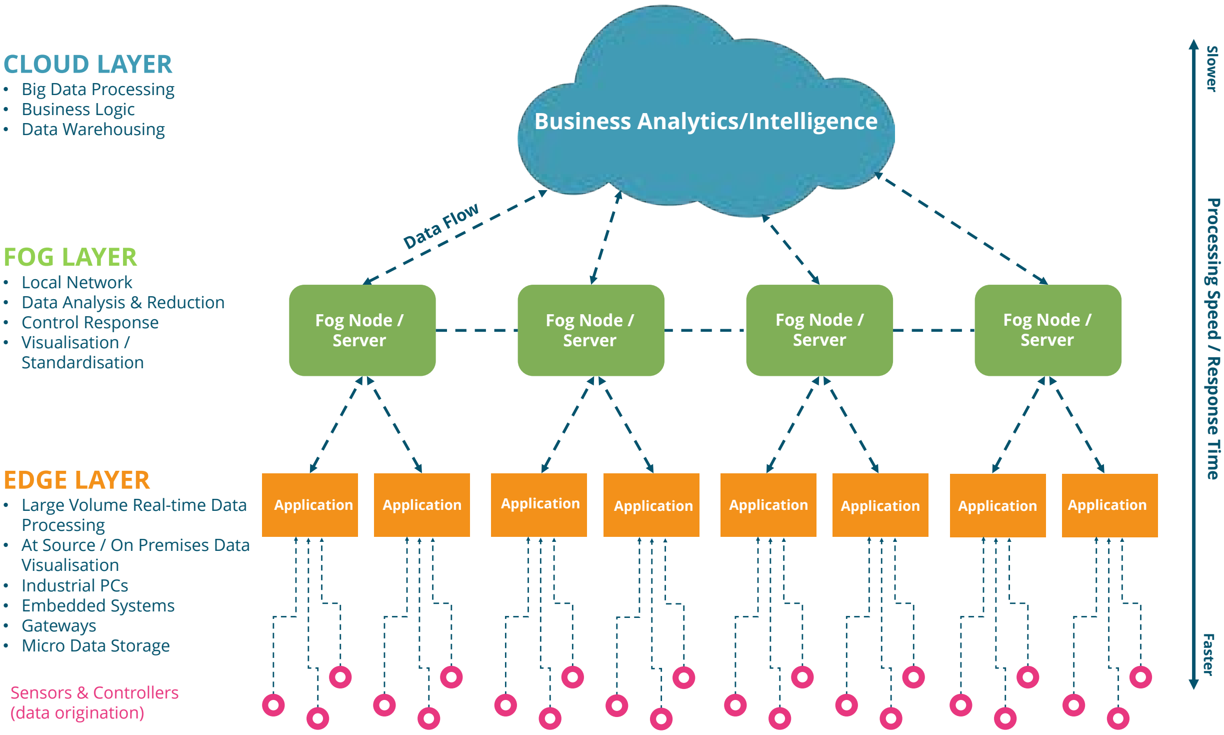


Figure 27: Industrial IoT (IIoT) Data Processing Stack. Adapted from [96].

As stated earlier, it is possible that there is a requirement to run multiple independent models at different layers, each prioritizing different subsets of the entire (large) feature set. This is feasible as well and can be realised by following the same framework presented in this document.

While Figure 27 may indicate that, to leverage the benefits of any of these layers, they need to be used together with the other layers – this is not necessarily true. There are several different configurations wherein Edge, Fog and Cloud capabilities can be obtained either in isolation or as pairs, e.g., Edge-Cloud: data collected by IIoT devices in the field is transferred to the cloud for optimisation.

Having deployed the model at the suitable juncture, the next consideration which is important concerns how to ensure the model stays relevant and accurate over its lifetime. Figure 29 presents this unified picture of **Online Learning**. Particularly for ICS, it is essential that the deployed Machine Learning models and the datasets are monitored and kept up to date with any changes made to the process itself.

Online Learning, as the name suggests, is a proactive approach whereby the learning continues even after the system is online. This means more data that the system capture is fed back into the data preparation and model training (and validation) periodically. As a result, the anomaly detector's detection thresholds are in tune with the current state of the system. As a simple example, if some known faults (but previously unconsidered) are observed in the field, these can be added to the validation set and the model's hyperparameters and detection threshold can be tuned to further increase generalisability of the model.

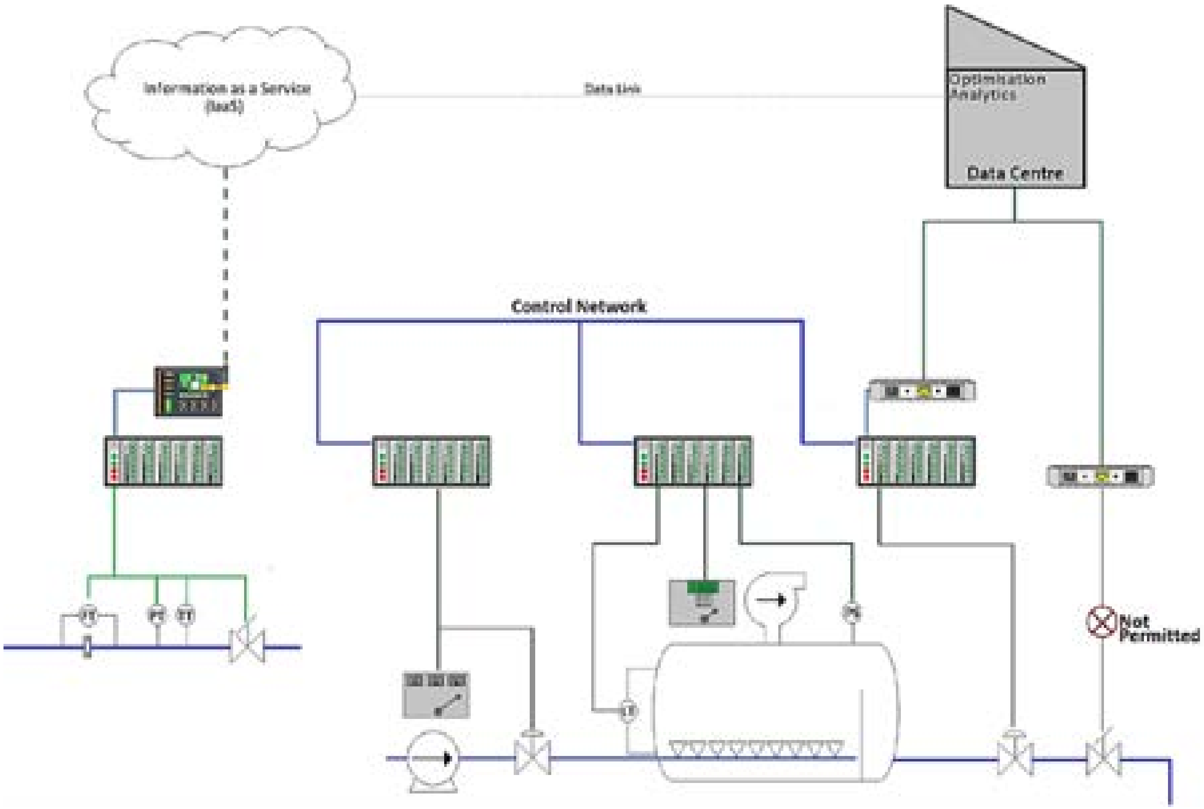


Figure 28: Edge-Cloud case [97]: Data collected from the Field layer/Control Network (pressure transducers, motors, etc.) is fed to Cloud-based Information-as-a-Service (IAAS) platforms and/or to Data Centres for optimisation/analytics.

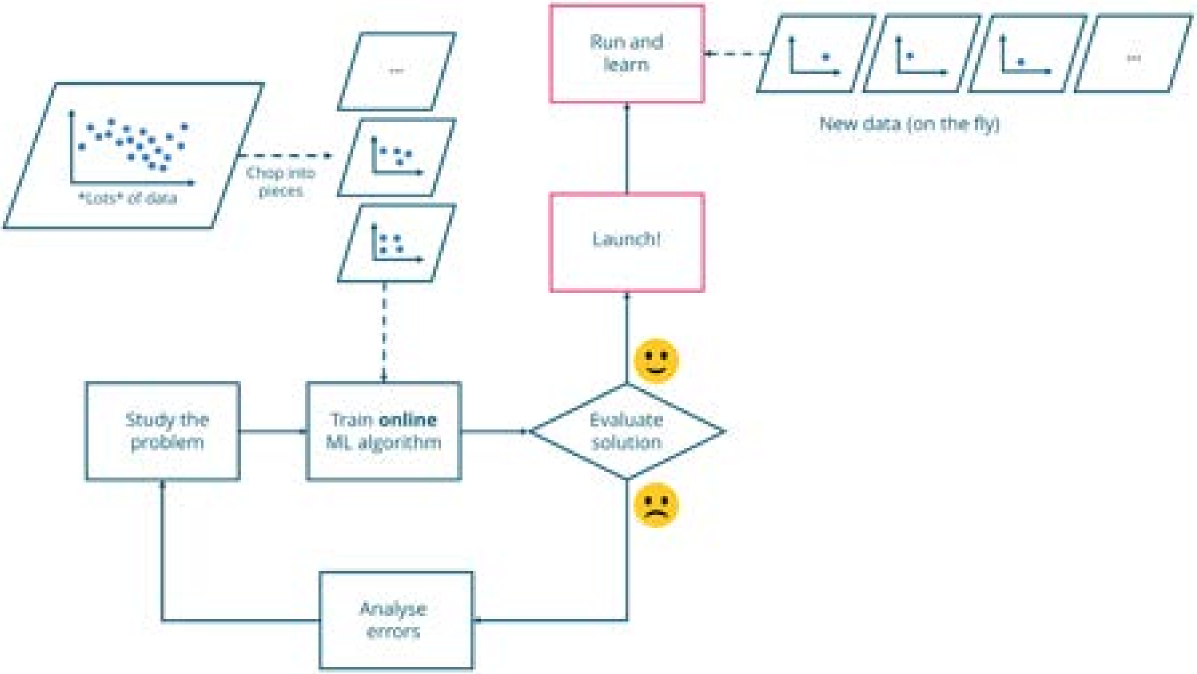


Figure 29: Adaptive, online learning framework. Red represents the online learning section; remainder is done offline. Adapted from [98].

Another option is a reactive approach whereby the performance of an anomaly detector is monitored over time, and if there are unwarranted but statistically significant changes in the detections/detection metrics, then the system can be re-trained (re-fitted) with a fresh batch of data. This would still classify as **Offline Learning**; however, it is also a suitable way to keep the detector up to date.

5.7 Challenges/Limitations of ML-based Anomaly Detection

While Machine Learning for Anomaly Detection has shown tremendous promise for securing industrial control systems in this new IT-OT converged era and increasing penetration of IoT, there are still some key challenges which remain. This section talks about the common issues faced by all Machine Learning algorithms.

5.7.1 Dataset

The dataset used to train the model is of paramount importance. Misrepresentations of operations of any kind could lead to the model learning unimportant and incorrect patterns. For example, if a semi-supervised learning model is selected in a one-class learning format, it is essential that during the data collection phase, the system is kept as close to error and fault-free as possible.

Considering this, one of the attack objectives for cybersecurity breaches is to gain access to the dataset to “poison” it and thereby compromise its integrity and reliability. This is referred to as the **Dataset Poisoning** attack [99].

More generally, one of the key drawbacks of Machine Learning and even more so for Deep Learning is the amount of data they require for learning. Hence, **lack of availability of sufficient training data** is a major issue – particularly for anomalous samples. For academia, this is due to the lack of good open-source datasets, and for industry, this is due to challenges associated with constructing adequate data acquisition and preparation pipelines, and the low incidence of cyberattacks from which training/testing datasets might be established. If supervised and semi-supervised learning approaches are chosen, it must be noted that **hand-labelling data is expensive** (time-consuming). From an attacker’s standpoint, training real-life ICS with open-source datasets poses further challenges as it allows them to find weaknesses in the IDS through **adversarial transfer learning** [100], i.e., different models trained using the same dataset tend have similar trends in detection performance.

5.7.2 Model

As discussed previously, one of the major challenges with the Machine Learning models themselves is the need to **trade-off between accuracy and interpretability**. Several approaches have been recommended in this document; however, more research is needed in this area of Machine Learning before mature solutions are made available for use in industry.

The second challenging area is **adversarial attacks** on machine learning algorithms. An example of this with image data is shown in Figure 30. In this example, a small amount of adversarial noise is added to the image to confuse the image classifier. The same principle applies on ICS data such that malicious attackers will attempt to launch stealthy attacks against a plant by first creating an attack to deceive the anomaly-based detector.

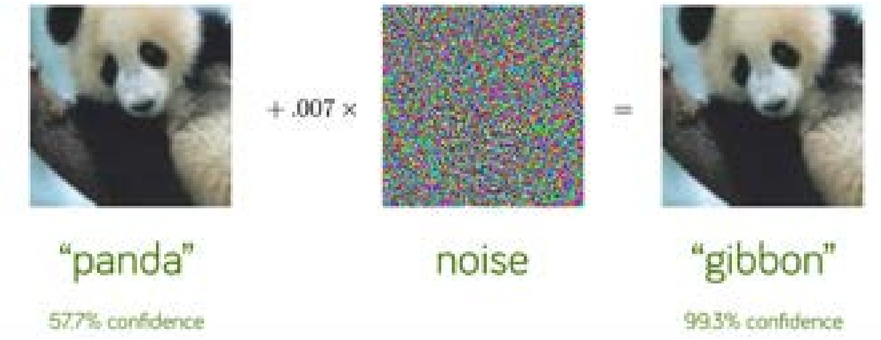


Figure 30: An example of adversarial noise being used to deceive an image classifier [101].

Fortunately, this is also another active area of research. Methods such as adversarial re-training are well-established whereby a small percentage of adversarial data is mixed in with the training dataset to teach the machine learning model to be more robust to adversarial examples, whilst only slightly compromising (if any) detection performance.

The third challenging area is to do with the generalisability of models, given the heterogeneity of physical processes and/or network data seen in an ICS plant. There is another interdependency here between a **model's representation power vs training time and model size**. As a general trend, shallower models tend to have lower representation capabilities compared to deeper models which use non-linear combinations of the input features to learn. With supervised learning-based detectors, the possibility of overfitting (model learns the given dataset too well and will not perform on unseen data) needs to be accounted for using regularisation.

Finally, most pertinent to ICS is the **ability to detect zero-day and previously unseen attacks**. Supervised learning algorithms cannot do this. Performant semi-supervised and unsupervised learning algorithms should, however, be able to detect them. The advantage of supervised learning approaches is that if all possible attack patterns are known beforehand, it could result in an optimal solution. The downside with semi-supervised and unsupervised learning algorithms is their high reliance on an operator tuning their detection thresholds. If this is not done correctly, they tend to produce a **high number of errors** (FPs and FNs). While False Positives are more distracting for an operator, reducing the model's sensitivity could result in (increased False Negatives) imply that critical attacks and faults would not be recognised.

6. Closing Remarks

To keep up with the shifting attack landscape, operators of industrial control systems need to move away from the traditional principle of security by isolation. With the increasing digitization of OT, cyber defence techniques such as intrusion detection (IDS), more prevalent in the IT space, are becoming relevant for OT and industrial control automation. ML-based IDS have proven to be a promising technology within the research literature and are being gradually introduced in industry to early adopters through specialist cybersecurity vendors. Their strengths are in being able to learn and separate normal from abnormal system behaviour in a data-driven manner, with limited expert insights and user-specified rules – often required with other types of intrusion detection systems. This makes them – in particular, semi-supervised and unsupervised methods – more resilient to previously unseen attack vectors such as those based on zero-day vulnerabilities.

In this light, the purpose of this report is to inform wider industry about this type of technology and provide them with a set of principles based on which they can make decisions regarding its adoption. The reader is furnished with knowledge regarding the types of anomaly-based IDS and their principles of operation, some commonly used algorithms, the process acquiring and preparing data to train these algorithms, the methods/metrics which can be used to evaluate their performance and suitability, the tools to augment their usability, and finally, the deployment lifecycle and post-deployment aspects such as maintainability. It is important to highlight that anomaly-based IDS are not a silver bullet for all scenarios; however, they are a viable option which, when tuned/adapted well, can provide tremendous value and contribute towards increased situational awareness for security analysts and/or a security operations centre. After all, a cyber defence tool must be able to integrate into a larger socio-technical operational environment.

7. References

[1] 'Threat landscape for industrial automation systems. Statistics for H2 2021 | Kaspersky ICS CERT', Mar. 03, 2022. https://ics-cert.kaspersky.com/publications/threat-landscape-for-industrial-automation-systems-statistics-for-h2-2021/ (accessed Oct. 28, 2022).

[2] 'CVE - Search CVE List'. https://cve.mitre.org/cve/search_cve_list.html (accessed Mar. 23, 2023).

[3] 'CVS : Security vulnerabilities'. https://www.cvedetails.com/vulnerability-list/vendor_id-442/CVS.html (accessed Mar. 23, 2023).

[4] T. J. Williams, 'The Purdue Enterprise Reference Architecture', *IFAC Proc. Vol.*, vol. 26, no. 2, Part 4, pp. 559–564, Jul. 1993, doi: 10.1016/S1474-6670(17)48532-6.

[5] O. D. Alexander, M.-H. Belisle, and J. Steele, 'MITRE ATT&CK® for industrial control systems: Design and philosophy', 2020.

[6] 'CyBOK – The Cyber Security Body of Knowledge v1.1'. https://www.cybok.org/knowledgebase1_1/ (accessed Oct. 28, 2022).

[7] Tofino Security, 'Case Profile: Davis-Besse Nuclear Power Plant | Tofino Industrial Security Solution'. https://www.tofinosecurity.com/why/Case-Profile-Davis-Besse-Nuclear-Power-Plant (accessed Jan. 26, 2023).

[8] P. F. Roberts, 'Zotob, PnP Worms Slam 13 DaimlerChrysler Plants', *eWEEK*, Aug. 18, 2005. https://www.eweek.com/security/zotob-pnp-worms-slam-13-daimlerchrysler-plants/ (accessed Jan. 26, 2023).

[9] S. Collins and S. McCombie, 'Stuxnet: the emergence of a new cyber weapon and its implications', *J. Polic. Intell. Count. Terror.*, vol. 7, no. 1, pp. 80–91, Apr. 2012, doi: 10.1080/18335330.2012.653198.

[10] International cyber law: interactive toolkit contributors, 'Steel mill in Germany (2014)', *International cyber law: interactive toolkit*, Jun. 09, 2021. https://cyberlaw.ccdcoe.org/wiki/Steel_mill_in_Germany_(2014) (accessed Jan. 26, 2023).

[11] A. Pichel, 'HAVEX Targets Industrial Control Systems', *Threat Encyclopedia*, Jul. 14, 2014. https://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/139/havex-targets-industrial-control-systems (accessed Jan. 26, 2023).

[12] J. Leyden, 'Water treatment plant hacked, chemical mix changed for tap supplies', *Security*, Mar. 24, 2016. https://www.theregister.com/2016/03/24/water_utility_hacked/ (accessed Jan. 26, 2023).

[13] International cyber law: interactive toolkit contributors, 'Power grid cyberattack in Ukraine (2015)', *International cyber law: interactive toolkit*, Jun. 04, 2021. https://cyberlaw.ccdcoe.org/wiki/Power_grid_cyberattack_in_Ukraine_(2015) (accessed Jan. 26, 2023).

[14] International cyber law: interactive toolkit contributors, 'Industroyer – Crash Override (2016)', *International cyber law: interactive toolkit*, Jun. 04, 2021.

https://cyberlaw.ccdcoe.org/wiki/Industroyer_%E2%80%93_Crash_Override_(2016) (accessed Jan. 26, 2023).

[15] International cyber law: interactive toolkit contributors, 'NotPetya (2017)', *International cyber law: interactive toolkit*, Nov. 14, 2022. https://cyberlaw.ccdcoe.org/wiki/NotPetya_(2017) (accessed Jan. 26, 2023).

[16] S. Gibbs, 'Triton: hackers take out safety systems in "watershed" attack on energy plant', *The Guardian*, Dec. 15, 2017. Accessed: Jan. 26, 2023. [Online]. Available: https://www.theguardian.com/technology/2017/dec/15/triton-hackers-malware-attack-safety-systems-energy-plant

[17] International cyber law: interactive toolkit contributors, 'Triton (2017)', *International cyber law: interactive toolkit*, Jun. 04, 2021. https://cyberlaw.ccdcoe.org/wiki/Triton_(2017) (accessed Jan. 26, 2023).

[18] M. Kumar, 'TSMC Chip Maker Blames WannaCry Malware for Production Halt', Aug. 07, 2018. https://thehackernews.com/2018/08/tsmc-wannacry-ransomware-attack.html (accessed Jan. 26, 2023).

[19] Cyware Labs, 'Lens manufacturer Hoya Corporation suffers cyber attack causing partial factory shutdown | Cyware Hacker News', *Cyware Labs*. https://cyware.com/news/lens-manufacturer-hoya-corporation-suffers-cyber-attack-causing-partial-factory-shutdown-bc642d97 (accessed Jan. 26, 2023).

[20] International cyber law: interactive toolkit contributors, 'Colonial Pipeline ransomware attack (2021)', *International cyber law: interactive toolkit*, Mar. 07, 2022. https://cyberlaw.ccdcoe.org/wiki/Colonial_Pipeline_ransomware_attack_(2021) (accessed Jan. 26, 2023).

[21] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, 'Survey of intrusion detection systems: techniques, datasets and challenges', *Cybersecurity*, vol. 2, no. 1, p. 20, Jul. 2019, doi: 10.1186/s42400-019-0038-7.

[22] P. Ackerman, *Industrial Cybersecurity: Efficiently secure critical infrastructure systems*. Packt Publishing, 2017.

[23] 'What is an Intrusion Detection System (IDS)?', *Check Point Software*. https://www.checkpoint.com/cyber-hub/network-security/what-is-an-intrusion-detection-system-ids/ (accessed Oct. 28, 2022).

[24] N. S. Sulaiman et al., 'Intrusion Detection System Techniques : A Review', *J. Phys. Conf. Ser.*, vol. 1874, no. 1, p. 012042, May 2021, doi: 10.1088/1742-6596/1874/1/012042.

[25] R. Mitchell and I.-R. Chen, 'A survey of intrusion detection techniques for cyber-physical systems', *ACM Comput. Surv.*, vol. 46, no. 4, p. 55:1-55:29, Mar. 2014, doi: 10.1145/2542049.

[26] N. Tuptuk, P. Hazell, J. Watson, and S. Hailes, 'A Systematic Review of the State of Cyber-Security in Water Systems', *Water*, vol. 13, no. 1, Art. no. 1, Jan. 2021, doi: 10.3390/w13010081.

[27] 'Top 10 BEST Intrusion Detection Systems (IDS) [2022 Rankings]', *Software Testing Help*. https://www.softwaretestinghelp.com/intrusion-detec-

tion-systems/ (accessed Oct. 28, 2022).

[28] 'Home - Suricata'. https://suricata.io/ (accessed Jan. 26, 2023).

[29] 'A comprehensive SIEM solution | ManageEngine Log360'. https://www.manageengine.com/log-management/siem-solution-log360.html?utm_source=sth&utm_medium=website-cpc&utm_campaign=Log360-ids (accessed Jan. 26, 2023).

[30] 'The Zeek Network Security Monitor', *Zeek*. https://zeek.org/ (accessed Jan. 26, 2023).

[31] 'Snort - Network Intrusion Detection & Prevention System'. https://www.snort.org/ (accessed Jan. 26, 2023).

[32] 'OSSEC - World's Most Widely Used Host Intrusion Detection System - HIDS', *OSSEC*. https://www.ossec.net/ (accessed Jan. 26, 2023).

[33] 'Intrusion Detection Software – IDS Security System | SolarWinds'. https://www.solarwinds.com/security-event-manager/use-cases/intrusion-detection-software (accessed Jan. 26, 2023).

[34] 'Security Onion Solutions'. https://securityonionsolutions.com/ (accessed Jan. 26, 2023).

[35] 'Homepage', *CyberArk*. https://www.cyberark.com/ (accessed Oct. 31, 2022).

[36] 'Cybersecurity as a Service Delivered', *SOPHOS*. https://www.sophos.com/en-us (accessed Oct. 31, 2022).

[37] 'Kaspersky Industrial CyberSecurity', *Kaspersky Industrial CyberSecurity | Holistic approach to Industrial Cybersecurity*. https://ics.kaspersky.com/ (accessed Oct. 31, 2022).

[38] 'Cyber security & intelligence', *BAE Systems | United Kingdom*. https://www.baesystems.com/en-uk/what-we-do/cyber-security---intelligence (accessed Oct. 31, 2022).

[39] 'About Dragos, Your Ally Against Industrial Cyber Threats | Dragos'. https://www.dragos.com/about/ (accessed Oct. 28, 2022).

[40] 'Securing Building Management Systems', *SCADAfence*. https://www.scadafence.com/securing-building-management-systems/ (accessed Oct. 17, 2022).

[41] 'Forescout – Automated Cybersecurity Across Your Digital Terrain', *Forescout*. https://www.forescout.com/ (accessed Oct. 28, 2022).

[42] 'ABB Cyber Security Services', *Process Automation*. https://new.abb.com/process-automation/process-automation-service/advanced-digital-services/cyber-security (accessed Oct. 31, 2022).

[43] 'Industrial Cybersecurity Solutions', *Rockwell Automation*. https://www.rockwellautomation.com/en-us/capabilities/industrial-cybersecurity.html (accessed Oct. 31, 2022).

[44] 'Cisco Industrial Security for your IoT, OT, and ICS', *Cisco*. https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-security.html (accessed

Oct. 31, 2022).

[45] Á. L. Perales Gómez, L. Fernández Maimó, A. Huertas Celdrán, and F. J. García Clemente, 'MADICS: A Methodology for Anomaly Detection in Industrial Control Systems', *Symmetry*, vol. 12, no. 10, Art. no. 10, Oct. 2020, doi: 10.3390/sym12101583.

[46] A. Braun, 'How to collect data for a Machine Learning model', *CodeX*, Jul. 21, 2021. https://medium.com/codex/how-to-collect-data-for-a-machine-learning-model-2b152752a15b (accessed Oct. 28, 2022).

[47] 'Dataset Search'. https://datasetsearch.research.google.com/ (accessed Oct. 28, 2022).

[48] 'AWS Marketplace: Search Results'. https://aws.amazon.com/marketplace/search/results?FULFILLMENT_OPTION_TYPE=DATA_EXCHANGE&CONTRACT_TYPE=OPEN_DATA_LICENSES&filters=FULFILLMENT_OPTION_TYPE%2CCONTRACT_TYPE (accessed Oct. 28, 2022).

[49] 'Find Open Datasets and Machine Learning Projects | Kaggle'. https://www.kaggle.com/datasets (accessed Oct. 28, 2022).

[50] 'Microsoft Research Open Data'. https://msropendata.com/ (accessed Oct. 28, 2022).

[51] 'GitHub: Where the world builds software', *GitHub*. https://github.com/ (accessed Oct. 28, 2022).

[52] A. P. Mathur and N. O. Tippenhauer, 'SWaT: a water treatment testbed for research and training on ICS security', in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater),* Apr. 2016, pp. 31–36. doi: 10.1109/CySWater.2016.7469060.

[53] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu, 'Revisiting Time Series Outlier Detection: Definitions and Benchmarks', presented at the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1), Jan. 2022. Accessed: Oct. 28, 2022. [Online]. Available: https://openreview.net/forum?id=r8IvOsnHchr

[54] S. Schmidl, P. Wenig, and T. Papenbrock, 'Anomaly detection in time series: a comprehensive evaluation', *Proc. VLDB Endow.*, vol. 15, no. 9, pp. 1779–1797, Jul. 2022, doi: 10.14778/3538598.3538602.

[55] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, 'Support Vector Method for Novelty Detection', in *Advances in Neural Information Processing Systems*, MIT Press, 1999. Accessed: Mar. 28, 2023. [Online]. Available: https://papers.nips.cc/paper_files/paper/1999/hash/8725fb777f25776ffa9076e44fcfd776-Abstract.html

[56] T. Yairi, Y. Kato, and K. Hori, 'Fault Detection by Mining Association Rules from House-keeping Data', 2001. Accessed: Mar. 28, 2023. [Online]. Available: https://www.semanticscholar.org/paper/Fault-Detection-by-Mining-Association-Rules-from-Yairi-Kato/730898e5a7e655cc8359d496f26ffb6ecfc8850b

[57] R. Paffenroth, K. Kay, and L. Servi, 'Robust PCA for Anomaly Detection in Cyber Networks'. arXiv, Jan. 04, 2018. doi: 10.48550/arXiv.1801.01571.

[58] J. Ma and S. Perkins, 'Online novelty detection on temporal sequences', in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, in KDD '03. New York, NY, USA: Association for Computing Machinery, Aug. 2003, pp. 613–618. doi: 10.1145/956750.956828.

[59] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[60] P. Senin et al., 'Time series anomaly discovery with grammar-based compression'. OpenProceedings.org, 2015. doi: 10.5441/002/EDBT.2015.42.

[61] M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh, 'Matrix profile goes MAD: variable-length motif and discord discovery in data series', *Data Min. Knowl. Discov.*, vol. 34, no. 4, pp. 1022–1071, Jul. 2020, doi: 10.1007/s10618-020-00685-w.

[62] P. Sun, S. Chawla, and B. Arunasalam, 'Mining for Outliers in Sequential Databases', in *Proceedings of the 2006 SIAM International Conference on Data Mining (SDM)*, in Proceedings. Society for Industrial and Applied Mathematics, 2006, pp. 94–105. doi: 10.1137/1.9781611972764.9.

[63] Y. Zhu et al., 'Exploiting a novel algorithm and GPUs to break the ten quadrillion pairwise comparisons barrier for time series motifs and joins', *Knowl. Inf. Syst.*, vol. 54, no. 1, pp. 203–236, Jan. 2018, doi: 10.1007/s10115-017-1138-x.

[64] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[65] C. A. Sims, 'Macroeconomics and Reality', *Econometrica*, vol. 48, no. 1, pp. 1–48, 1980, doi: 10.2307/1912017.

[66] P. J. Rousseeuw and K. V. Driessen, 'A Fast Algorithm for the Minimum Covariance Determinant Estimator', *Technometrics*, Mar. 2012, Accessed: Mar. 28, 2023. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00401706.1999.10485670

[67] Y. Yu, Y. Zhu, S. Li, and D. Wan, 'Time Series Outlier Detection Based on Sliding Window Prediction', *Math. Probl. Eng.*, vol. 2014, p. e879736, Oct. 2014, doi: 10.1155/2014/879736.

[68] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, 'LOF: identifying density-based local outliers', in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, in SIGMOD '00. New York, NY, USA: Association for Computing Machinery, May 2000, pp. 93–104. doi: 10.1145/342009.335388.

[69] F. T. Liu, K. M. Ting, and Z.-H. Zhou, 'Isolation Forest', in *2008 Eighth IEEE International Conference on Data Mining*, Dec. 2008, pp. 413–422. doi: 10.1109/

ICDM.2008.17.

[70] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, 'COPOD: Copula-Based Outlier Detection', in *2020 IEEE International Conference on Data Mining (ICDM)*, Nov. 2020, pp. 1118–1123. doi: 10.1109/ICDM50108.2020.00135.

[71] P.-F. Marteau, S. Soheily-Khah, and N. Béchet, 'Hybrid Isolation Forest - Application to Intrusion Detection'. arXiv, May 10, 2017. doi: 10.48550/arXiv.1705.03800.

[72] J. Li, W. Pedrycz, and I. Jamal, 'Multivariate time series anomaly detection: A framework of Hidden Markov Models', *Appl. Soft Comput.*, vol. 60, pp. 229–240, Nov. 2017, doi: 10.1016/j.asoc.2017.06.035.

[73] A. Ogbechie, J. Díaz-Rozo, P. Larrañaga, and C. Bielza, 'Dynamic Bayesian Network-Based Anomaly Detection for In-Process Visual Inspection of Laser Surface Heat Treatment', in *Machine Learning for Cyber Physical Systems*, J. Beyerer, O. Niggemann, and C. Kühnert, Eds., in Technologien für die intelligente Automation. Berlin, Heidelberg: Springer, 2017, pp. 17–24. doi: 10.1007/978-3-662-53806-7_3.

[74] N. Goernitz, M. Braun, and M. Kloft, 'Hidden Markov Anomaly Detection', in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, Jun. 2015, pp. 1833–1842. Accessed: Mar. 28, 2023. [Online]. Available: https://proceedings.mlr.press/v37/goernitz15.html

[75] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne, 'On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms', *Data Min. Knowl. Discov.*, vol. 8, no. 3, pp. 275–300, May 2004, doi: 10.1023/B:DAMI.0000023676.72185.7c.

[76] F. Rasheed, P. Peng, R. Alhajj, and J. Rokne, 'Fourier transform based spatial outlier mining', in *Proceedings of the 10th international conference on Intelligent data engineering and automated learning*, in IDEAL'09. Berlin, Heidelberg: Springer-Verlag, Sep. 2009, pp. 317–324.

[77] H. Ren et al., 'Time-Series Anomaly Detection Service at Microsoft', in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in KDD '19. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 3009–3017. doi: 10.1145/3292500.3330680.

[78] M. Thill, W. Konen, and T. Bäck, 'Online Adaptable Time Series Anomaly Detection with Discrete Wavelet Transforms and Multivariate Gaussian Distributions', *Arch. Data Sci. Ser. Online First*, vol. 5, no. 1, p. 04, 2018, doi: 10.5445/KSP/1000087327/04.

[79] N. Heim and J. E. Avery, 'Adaptive Anomaly Detection in Chaotic Time Series with a Spatially Aware Echo State Network'. arXiv, Sep. 02, 2019. doi: 10.48550/arXiv.1909.01709.

[80] A. Ryzhikov, M. Borisyak, A. Ustyuzhanin, and D. Derkach, 'NFAD: Fixing anomaly detection using normalizing flows', *PeerJ Comput. Sci.*, vol. 7, p. e757,

Nov. 2021, doi: 10.7717/peerj-cs.757.

[81] M. A. Bashar and R. Nayak, ‘TAnoGAN: Time Series Anomaly Detection with Generative Adversarial Networks’, in *2020 IEEE Symposium Series on Computational Intelligence (SSCI),* Dec. 2020, pp. 1778–1785. doi: 10.1109/SSCI47803.2020.9308512.

[82] M. Sakurada and T. Yairi, ‘Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction’, in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, in MLSDA’14. New York, NY, USA: Association for Computing Machinery, Dec. 2014, pp. 4–11. doi: 10.1145/2689746.2689747.

[83] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, ‘Long Short Term Memory Networks for Anomaly Detection in Time Series’, presented at the The European Symposium on Artificial Neural Networks, 2015. Accessed: Mar. 28, 2023. [Online]. Available: https://www.semanticscholar.org/paper/Long-Short-Term-Memory-Networks-for-Anomaly-in-Time-Malhotra-Vig/8e54dd2b426b8d656a77c155818a87dd34c40754

[84] A. M. Y. Koay, R. K. L. Ko, H. Hettema, and K. Radke, ‘Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges’, *J. Intell. Inf. Syst.*, Oct. 2022, doi: 10.1007/s10844-022-00753-1.

[85] G. R. M. R., C. M. Ahmed, and A. Mathur, ‘Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation’, *Cybersecurity*, vol. 4, no. 1, p. 27, Aug. 2021, doi: 10.1186/s42400-021-00095-5.

[86] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, ‘Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations’, *Int. J. Crit. Infrastruct. Prot.*, vol. 38, p. 100516, Sep. 2022, doi: 10.1016/j.ijcip.2022.100516.

[87] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, ‘A Review on Outlier/Anomaly Detection in Time Series Data’, *ACM Comput. Surv.*, vol. 54, no. 3, p. 56:1-56:33, Apr. 2021, doi: 10.1145/3444690.

[88] G. Pang, C. Shen, L. Cao, and A. van den Hengel, ‘Deep Learning for Anomaly Detection: A Review’, *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439950.

[89] P. Wenig, S. Schmidl, and T. Papenbrock, ‘TimeEval: a benchmarking toolkit for time series anomaly detection algorithms’, *Proc. VLDB Endow.*, vol. 15, no. 12, pp. 3678–3681, Aug. 2022, doi: 10.14778/3554821.3554873.

[90] ‘Data Splitting Strategies — Applied Machine Learning in Python’. https://amueller.github.io/aml/04-model-evaluation/1-data-splitting-strategies.html (accessed Oct. 28, 2022).

[91] S. Narkhede, ‘Understanding AUC - ROC Curve’, *Medium*, Jun. 15, 2021. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

(accessed Oct. 28, 2022).

[92] guest_blog, 'Explainable AI | Explainable AI to Explain the Working of Your Model', *Analytics Vidhya*, Jan. 07, 2021. https://www.analyticsvidhya.com/blog/2021/01/explain-how-your-model-works-using-explainable-ai/ (accessed Oct. 28, 2022).

[93] D. (DJ) Sarkar, 'Model Interpretation Strategies', *Medium*, Dec. 24, 2018. https://towardsdatascience.com/explainable-artificial-intelligence-part-2-model-interpretation-strategies-75d4afa6b739 (accessed Oct. 28, 2022).

[94] Y. Kalyani and R. Collier, 'A Systematic Survey on the Role of Cloud, Fog, and Edge Computing Combination in Smart Agriculture', *Sensors*, vol. 21, no. 17, p. 5922, Sep. 2021, doi: 10.3390/s21175922.

[95] 'Differences Between Cloud, Fog and Edge Computing', *Digiteum*, May 04, 2022. https://www.digiteum.com/cloud-fog-edge-computing-iot/ (accessed Mar. 23, 2023).

[96] 'Cloud Computing - Fog Computing & Edge Computing: WinSystems'. https://www.winsystems.com/cloud-fog-and-edge-computing-whats-the-difference/ (accessed Oct. 28, 2022).

[97] Yorkshire Water, 'IT, OT and ICS Technical Governance'. Yorkshire Water, 2022.

[98] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed., . 1st ed., O'Reilly Media, Inc., 2017.

[99] M. Kravchik, B. Biggio, and A. Shabtai, 'Poisoning attacks on cyber attack detectors for industrial control systems', in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, in SAC '21. New York, NY, USA: Association for Computing Machinery, Mar. 2021, pp. 116–125. doi: 10.1145/3412841.3441892.

[100] A. Erba et al., 'Constrained Concealment Attacks against Reconstruction-based Anomaly Detectors in Industrial Control Systems', in *Annual Computer Security Applications Conference*, in ACSAC '20. New York, NY, USA: Association for Computing Machinery, Dec. 2020, pp. 480–495. doi: 10.1145/3427228.3427660.

[101] I. J. Goodfellow, J. Shlens, and C. Szegedy, 'Explaining and Harnessing Adversarial Examples'. arXiv, Mar. 20, 2015. doi: 10.48550/arXiv.1412.6572.

[102] E. J. M. Colbert and A. Kott, *Cyber-security of SCADA and Other Industrial Control Systems*, 1st ed., . 1st ed., Springer Publishing Company, Incorporated, 2016.

APX Appendices

A. Project Details

This document was created as part of a PETRAS project being carried out by University College London and its User Partners. Further details are provided below:

Project title (with acronym): Early Anomaly Detection for Securing IoT in Industrial Automation (ELLIOTT)

Type of PETRAS project: SRF1

Project Start Date: 01/02/2020

Project End Date: 31/01/2023

Research Organisation(s): Department of Computer Science, University College London

Funded staff: PI (Prof. Stephen Hailes), Co-I (Dr Nilufer Tuptuk), RA (Shreevanth Gopalakrishnan).

User Partners: Cube Controls, Rockwell Automation.

B. Purdue Reference Architecture

The Purdue reference model, which was adopted from the Purdue Enterprise Reference Architecture (PERA) model by ISA-99 [4], is a well-established concept model of network segmentation which also ties together these different components comprising an ICS architecture. As shown in Figure A-1, the PERA model hierarchically orders an ICS environment into six logically separate layers. Within each, it defines the broad ICS functionality required. The layers are as follows:

Enterprise Zone (IT side):

> **Level 5:** *Enterprise Network*: Where the business systems such as Enterprise Resource Planning (ERP) and SAP which span multiple facilities sit.
> **Level 4:** *Site Business Planning and Logistics Network*: Home to all IT systems that support the production process in a plant or facility.

Industrial Demilitarised Zone: Separates and allows a secure connection between two distinct networks (IT and OT sides).

"Manufacturing" Zone (OT side):

Level 3: *Site Operations*: Contains systems that support plant-wide control, monitoring and data aggregation functions.

Level 2: *Area Supervisory Control*: Similar functions and systems as in Level 3, but targeted towards a smaller subset or area of the overall system.

Level 1: *Basic Control*: Contains all the controlling equipment. E.g., devices to open valves, move actuators, start motors, etc.

Level 0: *Process*: Home to the actual process equipment being controlled and monitored from higher levels, hence, it is also called Equipment Under Control (EUC).

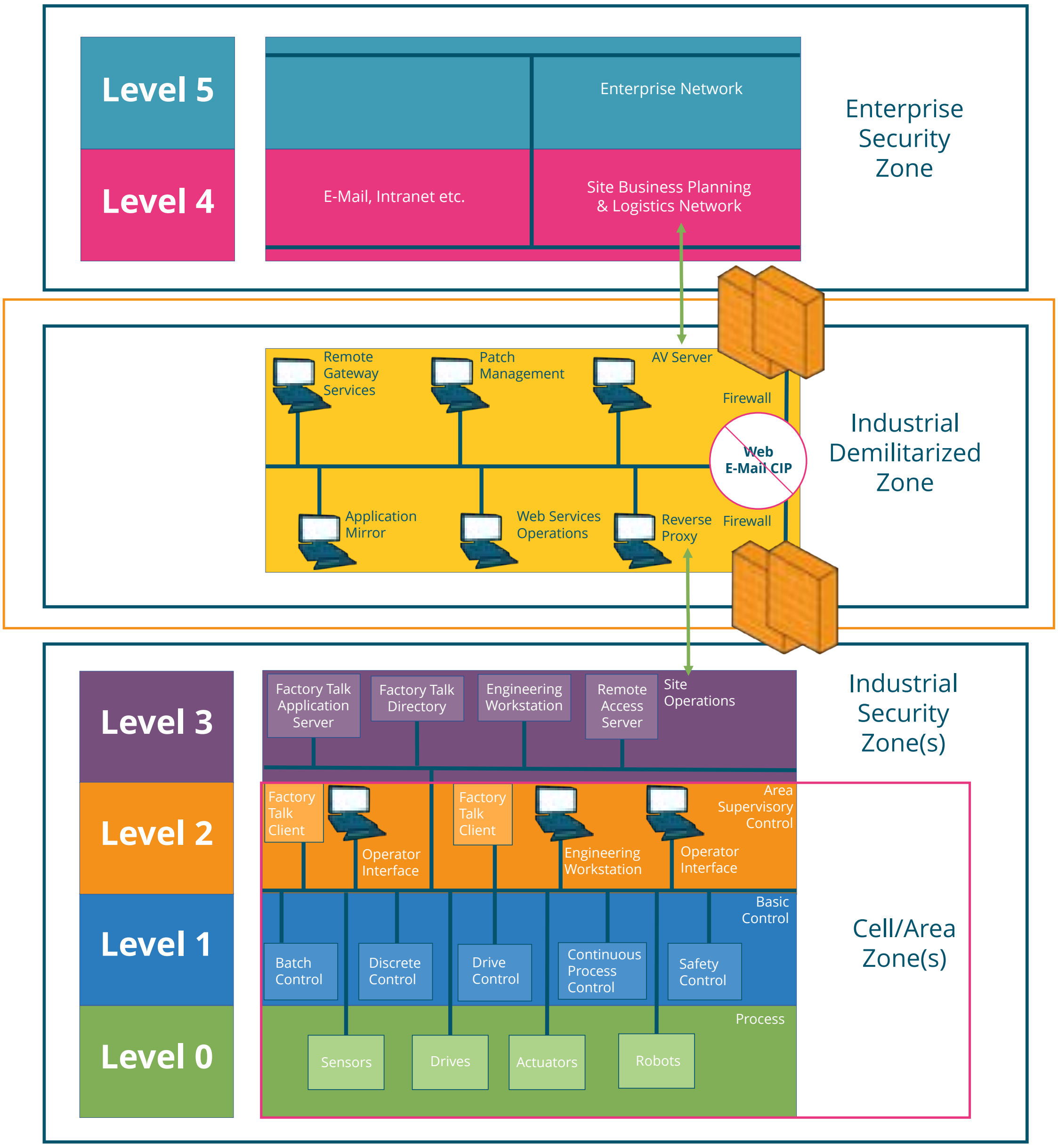


Figure A-1: ICS Architecture based on the PERA model. Adapted from [22].

C. Differences between IT and OT systems

In comparison to pure IT systems, ICS systems are considerably different and therefore have different set of security requirements which would apply to them [85]. Refer to Figure A-2 for a depiction.

1. System must **operate uninterrupted** without even needing to stop for security patches. Hence, unexpected outages of the sub-systems that monitor and control the processes are unacceptable. The ICS operates in a repeatable and predictive manner, and its components require **deterministic responses** with **minimal levels of jitter and delay**.
2. The Confidentiality, Integrity, Availability (CIA) security model is instead perceived as **AIC** for OT systems in the order of importance. For example, it might be desired to ensure the integrity of sensed data, however, the confidentiality of data might not be a major concern.
3. In ICS, the primary **focus is on safeguarding the physical assets** (e.g., PLCs, sensors, actuators), the **environment, and the human operators** involved. On the other hand, in an IT system, the focus could be on the data itself and IT assets through which the data is moved. Simultaneously, companies will look to maintain their **reputation** which takes longer to build than any plant.
4. This implies that a **successful attack on an ICS could have a severer impact** than on an IT system since damage to physical assets could result in service disruption, damage to the environment and may even impact human life.
5. An **ICS packet's payload is shorter** than an IT packet. Further, data captured at different places on site tend to be **highly correlated**. They all need to obey the laws of physics and the system design specifications.
6. An ICS operates in a **significantly resource-constrained environment** and the usage of third-party applications (i.e., deployment of software) is restricted.
7. **Communications protocols are unique** in comparison and in some cases they are proprietary. However, due to the time-constrained nature of operations, these protocols have historically been decided to operate without encryption.

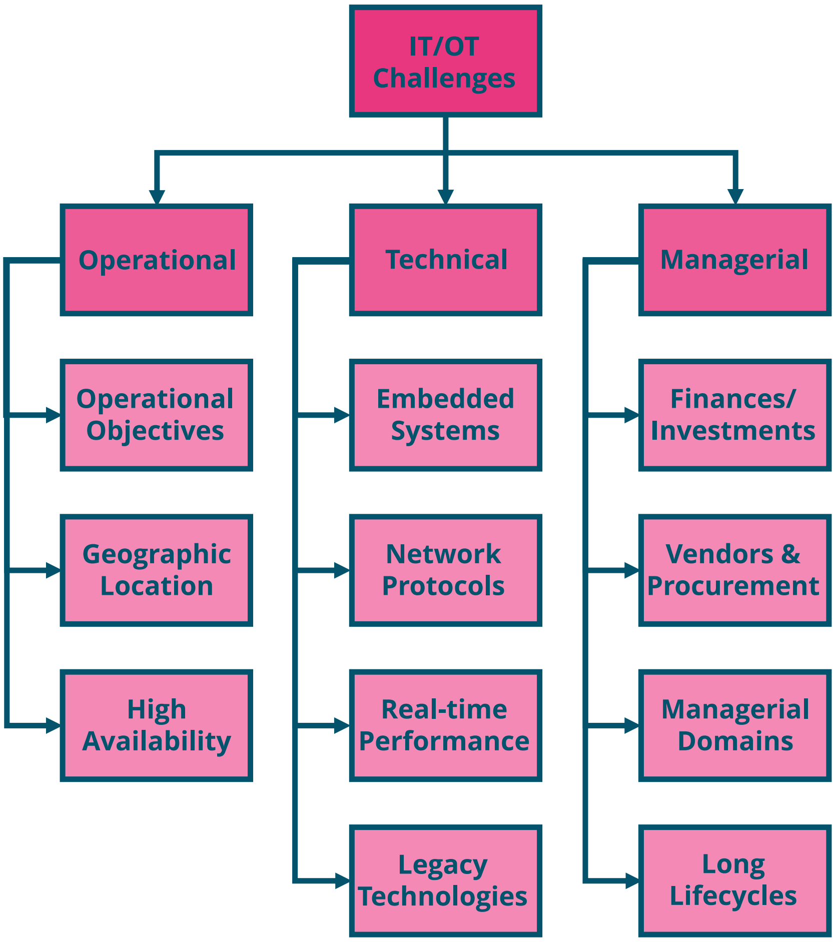


Figure A-2: Summarises some of the key differences and novel challenges when moving from IT to OT. Adapted from [102].

D. Security Issues and Challenges of ICS

D. i) Insecure-by-Design

Building on some of the points highlighted in the previous section, it can also be added that security is not a priority in legacy ICS infrastructure by design [84]. For example, there are many instances where processes are run with escalated privileges in "always on" mode on devices, and these devices are accessible to many users; devices and applications are designed for long lifetimes and high availability, and not necessarily to be robust to modern cyber threats; many OT environments have backdoors to enable remote support through insecure protocols such as TeamViewer, FTP, VNC, etc. Old-fashioned reliance on air-gapped OT networks will not be suitable in this new era. Table A-1 presents some of the key vulnerabilities in each layer of the Purdue reference model discussed previously.

Component	Vulnerabilities/Risks
PLC	Risk of code corruption or modification, and configuration manipulation
HMI	Difficulty of patching operating system
Sensors/Actuators	Hard to guarantee integrity and authenticity of data sent to PLC/controllers
Safety System (SIS)	Monitored less than the plant. Same, limited cybersecurity protection
Historian	Same vulnerabilities as common database platforms
RTU	Possible authentication bypass, data manipulation, malformed packets etc.
Eng. Workstation	Insecure remote access, software vulnerabilities, USB insertions, etc.

Table A-1: List of some well-known vulnerabilities of each layer of the Purdue model. Adopted from [84].

D. ii) IT-OT Convergence

Industry 4.0 has led to the gradual convergence of IT and OT networks to. Due to technological advances in the IT domain (e.g., Internet of Things (IoT)) and increasingly ubiquitous communications technologies (e.g., Internet Protocol (IP), Ethernet, etc.), several avenues opened for reducing operating cost, simplifying maintenance procedures, and increasing visibility in the OT world. Evidently, this required bringing across technologies from IT to OT.

Some examples of this include [102]:

1. Protocol migration from serial to IP: e.g., DNP3 which was designed for remote communication in utilities.
2. Deterministic time division multiplexing networks (TDM) to non-deterministic statistical time division multiplexed networks (SDMs): Particularly with reference to the Physical layer – Ethernet, used for IP communications, is an example of SDMs.
3. Mobile Computing within ICS: for granting engineers and maintenance personnel with easier access to system information and control function.
4. Cloud and Fog Computing: For central (cloud) and/or distributed optimisation and analytics.
5. Internet of Things (IoT): It is an extension of the Internet, i.e., a global network of “things”/objects connected to the physical world exchanging information. IIoT is a subset of this, referring to the use of IoT for the industrial sector. They provide capabilities such as acquisition of data, controlling operations, and edge computing and optimisation.

However, IIoT and being internet connected present significant new attack vectors. In some cases, they can be a publicly accessible back-door to critical ICS networks. A simple search in the Shodan search engine for a communications adapter produces the following results as in Figure 7. Internet-connected Ethernet/IP devices would have an open TCP or UDP port 44818. The results indicate that, with little effort, an outsider could conduct reconnaissance about an ICS plant, and maybe even manipulate some of the devices.

These devices, applications, and related cloud functionalities are often deployed as an end-to-end third-party solution. The take-away message is that while these solutions can be useful and convenient, they should be adopted with caution if they are to work alongside reliability and/or safety-critical ICS networks.