Faculty of International Maritime Studies Kasetsart University

Maritime Technology and Research

https://so04.tci-thaijo.org/index.php/MTR

Research Article

BridgeInsight: An asset profiler for penetration testing in a heterogeneous maritime bridge environment

Avanthika Vineetha Harish, Kimberly Tam*[*] and Kevin Jones

University of Plymouth, Drake Circus, Plymouth, United Kingdom

Article information
Received: July 13, 2023
1st Revision: August 31, 2023
Accepted: September 20, 2023

Keywords
Maritime cyber security,
Machine learning,
Asset profiler,
Automated audits,
Pentesting

Abstract

A maritime bridge environment is a heterogeneous ecosystem of complex systems for various operations. As part of new requirements set by the International Association of Classification Societies, ship operators must now maintain an asset inventory aboard vessels specifically to improve their cyber security. This paper discusses the development of a ship-specific asset profiler that will not only identify and record the devices present automatically but also provide an in-depth analysis of their properties and characteristics in an intelligent and user-friendly manner. As cyberattacks increase in the maritime industry, proper testing of ship systems is essential, to ensure vessels remain secure and the risk of a cyberattack is minimized. An asset profiler for the bridge environment would serve as a tool for profiling the devices, helping personnel make faster and well-informed decisions, and could be a component of a wider audit framework. This paper presents a ship bridge profiler (i.e., BridgeInsight) used to identify all devices on the bridge of a vessel automatically and which provides information on them using a generated PDF report that consists of graphs and charts. To do this, it uses the Random Forest classifier algorithm, and the information it provides will enable the auditor or pen tester to perform manual testing or automate audits, while also providing comprehensive information that engineers and mariners can use to comply with regulations.

1. Introduction

Maritime is a complex billion-dollar industry and a crucial part of the global economy. Countries such as the United States import around 90 % of the goods by sea, and China is a heavy importer of resources like oil and iron (Loomis et al., 2021; US Coast Guard, 2021). With the advent of technology, the complex systems on board vessels have adapted new functionalities to make operations easier and better. Along with network connectivity, several emerging topics like Artificial Intelligence (AI) and Machine Learning (ML) have emerged in the traditional operating environment. While these often provide better safety, usability, and comfort, they also introduce several new challenges, like cyber vulnerabilities or flaws onboard critical systems, which then can be exploited by cyber criminals (Bothur et al., 2017).

Cyber security audit is a process that helps identify digital threats within a defined scope. This provides a comprehensive review of a system's vulnerabilities, its compliance with policies and regulations, and an assessment cyber risks. One of the first steps in cyber security audit is

[*] *Corresponding author: University of Plymouth, Drake Circus, Plymouth, United Kingdom*
E-mail address: kimberly.tam@plymouth.ac.uk

information gathering, to identify the scope and assets. The IASME (Information Assurance for Small and Medium Enterprises) Maritime Cyber Baseline, developed by the IASME consortium in November 2021 and supported by The Royal Institution of Naval Architects (RINA), is an audit process that uses a checklist that allows ship owners and operators to show compliance with security controls and process (IASME, 2021a). Under the scope of assessment in the audit process, the checklists ask for asset registers for all information and operational (IT/OT) technology, along with their make, model, and other characteristics. The assessment also requires listing all the networks on the vessel, their functions, how they are segmented, routers, firewalls, and gateways (IASME, 2021b).

Identifying systems and having an equipment inventory are also required to comply with certain requirements, standards, and policies like Unified Requirements (URs) by The International Association of Classification Societies, or IACS. IACS is an organization of classification societies that establish technical standards for vessels and the maritime industry. IACS produces Unified Requirements, or URs, that are adopted resolutions on minimum requirements on matters covered by classification societies (IACS, 2022c). To ensure cyber resilience onboard vessels, IACS has produced two new URs: UR E26, which deals with the Cyber Resilience of Ships, and UR E27, which deals with the cyber resilience of onboard systems and equipment; these come into force from the 1st of January 2024. Both of these URs will be applicable to vessels constructed on, or after, 1 January 2024; UR E26 document mentions minimum requirements to establish a ship as cyber resilient, while UR E27 deals with the establishment of cyber resilience for the systems on board rather than for the vessel itself.

The first goal of ‘Identify’ in UR26 mentions identifying all the onboard computer-based systems (CBS), their interconnections, interdependencies, and resources involved. This includes creating and maintaining an inventory of all CBS onboard and the networks involved, during the entire life of the ship (IACS, 2022a). The UR also stipulates having the system details, such as manufacturer, brand, and model, and logical connections between them on the network. As part of section 3.1 of the UR27 document, information regarding equipment, hardware, operating systems, configuration files, and network flows, as well as plans and policies, are to be submitted to the classification society for review and approval (IACS, 2022b). This is followed by a requirement to maintain an inventory of the name of the device, manufacturer, model, and versions of software, as well as a software inventory that includes at least installation dates, version numbers, maintenance, and access control policies (IACS, 2022b).

Considering that there are more requirements and guidelines introduced in the maritime sector to improve cyber security onboard, which requires having a proper asset management process, this paper will explain how this automated asset profiler- BridgeInsight- can identify and provide information about the assets/devices on board to the tester/auditor who monitors the process. Additionally, the tool helps to audit/identify any unused and unwanted devices connected to the network that could be a point of weakness for the entire environment. The tool generates a condensed, user-friendly PDF report of all asset and network information found and profiled, which could be used in association with maintaining the asset register. BridgeInsight can also be integrated into a future automated penetration testing system to help in the testing of systems for vulnerabilities.

2. Asset identification

There are many types of networks, especially in complex heterogeneous environments. Different systems communicate with different protocols, creating separate subnetworks (traditionally, often IT or OT specific) or clusters of systems (see **Figure 1**). Assets in a ship environment include equipment, communication interfaces, and networks that are essential for the smooth operation of the vessel (Tam & Jones, 2019). Each organisation defines the word asset differently, but in this paper, the term refers to any equipment networked on the bridge of a vessel for bridge operations (e.g., navigation, emergency communication) on a network and which has an assigned IP address for communication.

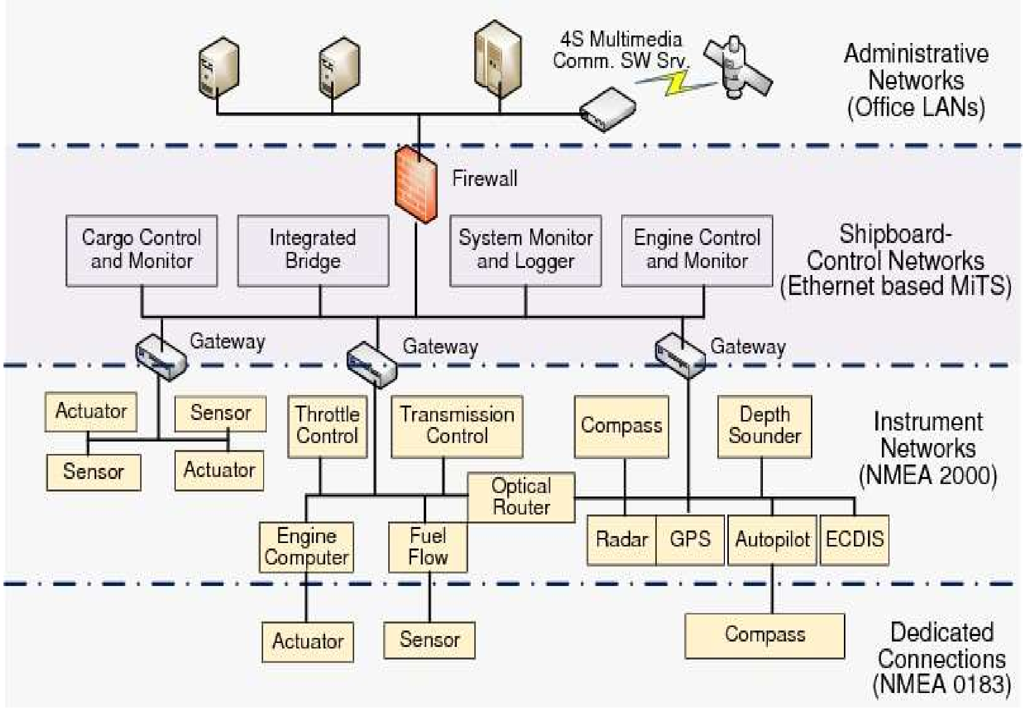

4S Multimedia Comm. SW Srv.
Administrative Networks (Office LANs)
Firewall
Cargo Control and Monitor
Integrated Bridge
System Monitor and Logger
Engine Control and Monitor
Shipboard-Control Networks (Ethernet based MiTS)
Gateway
Gateway
Gateway
Actuator
Sensor
Sensor
Actuator
Throttle Control
Transmission Control
Compass
Depth Sounder
Instrument Networks (NMEA 2000)
Optical Router
Engine Computer
Fuel Flow
Radar
GPS
Autopilot
ECDIS
Actuator
Sensor
Compass
Dedicated Connections (NMEA 0183)

Figure 1 Ship area network (Jeon & Lee, 2014).

The bridge of a vessel typically consists of a variety of equipment, including an Electronic Chart Displaying and Information System (ECDIS), a Voyage Data Recorder (VDR), an Automatic Identification System (AIS), RADAR, VHF equipment, Global Maritime Distress and Safety System (GMDSS), compass, gyroscope, and more. Safety and security standards for vessels mandate certain equipment; however, the type of equipment may differ according to the class, size, and type of vessel. For example, Chapter V of Safety of Life at Sea (SOLAS)- Safety of Navigation requires a Voyage Data Recorder to be fitted on vessels constructed on, or after, 1 July 2002, or ro-ro passenger ships constructed before 1 July 2002, or ships other than passenger ships of 3,000 gross tonnages and upwards constructed on, or after, 1 July 2002.

However, the regulation mentions the vessels may be fitted with an S-VDR (Simplified VDR) that captures less data than a VDR, considering the size and type of vessel. This difference in the equipment type changes the scope and characteristics of management and testing. Therefore, a mechanism that automatically identifies devices and profiles them is a useful reconnaissance tool for engineers/mariners maintaining inventories to comply with regulations. More use cases are discussed below.

2.1 Asset maintenance and inventory listing

An asset inventory provides essential situational awareness for maintenance and in the event of an incident. A device inventory including information about the device type, IP address, MAC address, open ports, manufacturer information, and version number makes it easier for those who have responsibilities to manage those devices. For example, it allows them to identify any obsolete/unused devices connected to the network. The removal of such devices can reduce the network's threat surface without affecting operations. According to the 2020 Global Networks Insights report, which assessed more than 800,000 IT network devices, 47.9 % of the network assets of organisations were obsolete and, on average, have twice as many vulnerabilities per device (42.2) compared to ageing (26.8) and current ones (19.4) (FutureIoT, 2020). Therefore, the profiler can allow seafarers or asset owners to better understand their systems, and maintain the asset inventory for compliance with regulations. The following table maps the new IACS URs to how the proposed asset profiler fulfils them (**Table 1**).

Table 1 Mapping of asset profiler with new IACS Unified Requirements.

UR	Requirement	BridgeInsight
UR 26	• For each CBS: a description of purpose, including technical features (brand, manufacturer, model, main technical data);	Fully Supported
	• A block diagram identifying the logical and physical connections (network topology) among various CBSs onboard, between CBSs, and external devices/networks and the intended function of each node;	Partially Supported
	• For network devices (switches, routers, hubs, gateways etc.), a description of connected sub-networks, IP ranges, MAC addresses of nodes connected, or similar network identifiers;	Partially Supported
	• The main features of each network (e.g., protocols used) and communication data flows (e.g., data flow diagram) in all intended operation modes;	Partially Supported
	• A map of the physical layout of each digital network connecting the CBSs onboard, including the onboard location and network access points;	Not supported
UR 27	• Detailed list of equipment included in the system; may include name, brand/manufacturer (supplier), model or reference; some devices contain several references, current version of the operating system and embedded firmware (software version), and date implemented;	Fully Supported
	• Equipment hardware details (i.e., mother board, storage, interfaces (network, serial), and any connectivity);	Not supported
	• A list of software including: - Operating system/firmware- Network services provided and managed by the operating systems- Application Software- Databases- Configuration files.	Partially Supported
	• Network or serial flows (source, destination, protocols, protocols details, physical implementation)	Partially Supported
	• Network security equipment (including details mentioned above), e.g., traffic management (firewalls, routers, etc.) and packet management (IDS, etc.)	Partially Supported
	• Secure Development Lifecycle Document	Not supported
	• Plans for maintenance of the system	Not supported
	• Recovery Plan	Not supported
	• System Test Plan	Not supported
	• Description of how the system meets the applicable requirements in E27 (i.e., Operation Manual or User Manual, etc.)	Not supported
	• Change Management Plan	Not supported

2.2 Information gathering by pen-testers/auditors

Maritime cyber security for vessels is a relatively new discipline that protects onboard systems and surrounding marine/maritime infrastructure. Understanding gaps and flaws is a key step for this. Penetration testing, or pentesting, is a process where authorised personnel attack the system, within scope, to find exploitable vulnerabilities and threats. There are security frameworks that can assist this, such as Metasploit (Rapid7, 2023). While carrying out penetration testing in a live, complex environment with sector-specific devices, a lack of system knowledge can introduce challenges and disrupt operations. Traditionally, penetration testing was used to test IT systems, whereas these days OT penetration testing, and IT that monitors or controls OT, are becoming more prevalent.

One of the first steps in penetration testing is information gathering, to identify the scope and assets. In an IT environment, this is fairly simple, as most devices would be computers, networking devices, or small IoT devices. In a vessel's bridge environment, networked devices are more bespoke, and for various purposes, which makes it more difficult, but still necessary, to understand the systems and networks in place. Currently, this is done manually, where the pentester, or auditor, goes on board a vessel. This is time-consuming for pen-testers/auditors and requires appropriate technical qualifications and certifications. In a comprehensive literature review by Bolbot et al. (2022), out of 144 papers about maritime cyber security from the period 2010 - 2022, only 13 papers were attributed to penetration testing and vulnerability scanning. This indicates that there is a lack of historical research data available to conduct the testing process. With an automated tool like BridgeInsight, not only does the tester not need to be familiar with all the devices or protocols in the sector they are testing, but the tool can guide a non-expert and be faster than expert manual asset inspection. On a ship's bridge, equipment may also be hidden out of sight, which could also make this a less intrusive and invasive process.

2.3 Background literature

A number of studies have been conducted on identifying devices in IT and IoT networks. A study by Ammar et al. (2019) implemented a network protocol based IoT device identification system for smart home environments, using the features extracted from network packets. Their model extracted features like manufacturer and device name from DHCP information, model, and service names. As a result, a unique feature vector was generated, which represented the device and was used to identify a newly connected device based on previously extracted feature vectors (Ammar et al., 2019). The study used real traffic information from a lab and publicly available IoT data, and the results showed that the model identified 30 devices out of 33 devices. The main goal of this model was to identify a newly connected device in the network, and did not employ any machine learning mechanisms for automation (Ammar et al., 2019).

A similar study by Sivanathan et al. (2017) performed smart device identification using network trace capture over a period of three weeks in a smart city and campus environment with over 20 devices that included cameras, lights, and health monitors. The study used multiple supervised learning algorithms in the Weka tool to classify devices using features like sleep time, active volume, average packet size, active time, number of servers, number of protocols, DNS (Domain Name System) information, NTP (Network Time Protocol) interval, and port information. This Random Forest (RF) classifier's highest accuracy was 97 % in the 10-fold cross-validation test, and it could identify specific IoT devices with over 95 % accuracy in the independent test analysis. Authors in Hamad et al. (2019) carried out device fingerprinting and classification to whitelist approved IoT devices and monitor suspicious ones. A number of sequential network packets were collected to extract features from the packet headers, and different classifiers were applied to identify the device type. A total of 67 features were extracted (e.g., TTL, Ethernet packet size, IP packet, and header sizes and TCP payload size). The authors compared 9 different classifiers with 50 and 100 estimators, and selected RF with 100 estimators as the base classifier, with an average accuracy of 90.3 % in identifying whitelisted devices.

Unlike the previous methods, Sivanathan et al. (2018) used active device identification by probing for open ports one after another and constructing a hierarchical tree. According to the authors, 42 TCP (Transmission Control Protocol) ports were open on 19 devices, and although port combinations which define a device type differed between devices, some similarities were observed between the ones manufactured by the same vendor. Furthermore, the results indicated that port 80 was the most commonly used port (9 out of 19 devices had port 80 open), and while building the hierarchical tree, the script chose port 80 as its root node and probed other ports to identify devices (Sivanathan et al., 2018).

There are a few network topology generators commercially available for IT systems that identify, list, and visualize networks. Currently, the SolarWinds network topology mapper is the only popular tool auditors use to view the status of their networks and monitor them. The tool takes in the IP address of a seed device, typically the main switch in the network, and then scans for devices and draws a map with IP addresses. However, this tool is widely used only in IT environments, where the common devices found are routers, switches, servers, firewalls, VMware hosts, and wireless access points (SolarWinds, 2023). Another tool is Auvik Network mapping & IT asset management, a web-based tool that pulls in information from ARP (Address Resolution Protocol) tables and IP assignments to establish connections between IT devices and draw the topology map (Auvik, 2023). ARP tables store the MAC address and IP address pairs of devices used for communication and sending packets from a source device to a destination device (Auvik, 2022). With several useful features built-in for IT network monitoring, this tool, similar to SolarWinds, focuses on networking devices like firewalls and routers. All these tools are used by network administrators in IT and office environments, where the devices are mostly PCs, routers, firewalls, and network gateways. GRASSMARLIN is an open-source tool developed by the National Security Agency (NSA) for providing network situational awareness in Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) networks (NSA, 2017). The tool discovers network communications and visually displays them as topology maps. One of the limitations of the tool is that it chooses only one signature if a device matches with multiple signatures from over fifty-four integrated signatures in the tool (Acord, 2017). This can be an issue in the maritime sector, where there are duplicate systems used for operations and communicating with different systems using different protocols.

IoT device identification is the subject of a number of research projects, but industry-specific literature is very limited. For the purpose of identifying the threats in the maritime sector, Amro (2021) utilized IoT device identification using both cyber and physical tracking. Cyber properties were extracted using IoT device scanners like Shodan and Censys to identify internet-connected systems that may possess maritime characteristics like the emission of NMEA (National Marine Electronics Association) messages (Censys, 2023; Shodan, 2023). NMEA 0183 is a messaging protocol where data is transmitted in ASCII strings, depending on their purpose, and a few devices that use NMEA 0183 messages to communicate are chart-plotters, radar, depth sounders, and GPS receivers (Bagur, 2023). Marine tracking devices were used to facilitate physical tracking of the systems in order to extract information such as GPS location, speed, and heading. Based on these data, Shodan API queries were then analyzed and, in total, 4942 unique NMEA emitting hosts were discovered, of which 99 % (4897 hosts) were GPS receivers (Amro, 2021). In order to identify vulnerabilities, this data was then cross-checked against the National Vulnerability Database (NVD). NVD is a repository of vulnerability management data, including product names, software flaws, and impact metrics, maintained by the National Institute of Standards and Technology (NIST) (NIST, 2022). In spite of the fact that this method provides an overview of various maritime device types and statistics, it does not include specific devices or those that do not have an internet connection.

The purpose of these studies is to identify and distinguish malicious devices or abnormal network traffic from new devices connected to the network. In a review of machine learning models applied to the identification of IoT devices and rogue devices in the environment conducted by Liu et al. (2022), four categories of detection and classification were identified: device-specific pattern recognition, deep learning-enabled device identification, unsupervised device identification, and abnormal device detection. This survey also discussed a few challenges associated with these methods, such as devices outside the scope of the identification system, devices from the same manufacturer not being identified, the ability to dynamically grow datasets and to learn new devices, and the robustness of features (Liu et al., 2022).

3. BridgeInsight

In the previous section, different methods of device identification were discussed for IoT and IT systems. Based on the gaps identified in the literature, BridgeInsight will be responsible for detecting ship systems. **Figure 2** illustrates the components and inner workings of BridgeInsight asset profiler.

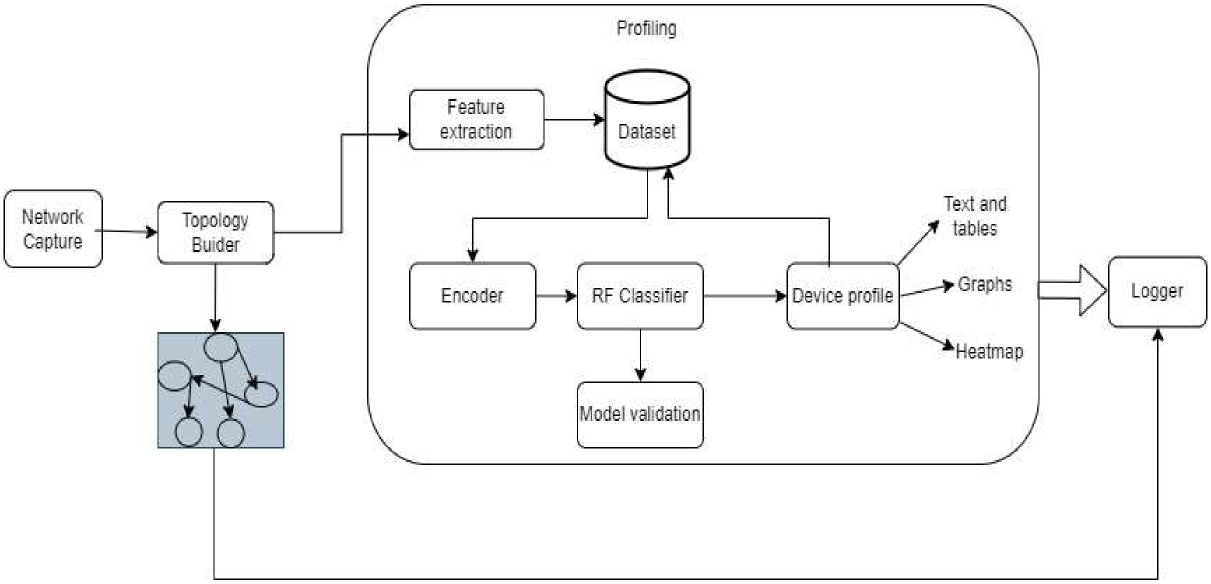


Figure 2 BridgeInsight.

3.1 Components and tools

This section explains the components of BridgeInsight and some of the tools and terminologies.

- Topology Builder: This module creates a network communication flow graph. Upon receiving a network configuration in the form of a network domain address, BridgeInsight captures network traces as Packet Capture (PCAP) files (Keary, 2022). These are processed by the topology builder module, which creates directed topology graphs depicting network connections and how systems communicate.
- Feature Extraction: The network information from the topology builder is passed to a feature extraction module that scans the network, gathering information about open ports, manufacturers, and OS, and extracts their features. Features are then incorporated into a dataset.
- Dataset: Feature datasets can be fed into Machine Learning (ML) modules to analyze data. Recently, ML and Artificial Intelligence (AI) have become increasingly popular in the

shipping industry. These have been used for various purposes, including predicting classes of ships and traffic density, using millions of AIS data records, reducing fuel emissions and avoiding collisions (Kretschmann et al., 2022). However, there is very little data readily available to be used for the profiling and testing of onboard hardware electronic equipment on ships and, thus, there was a need to create a data set to be used for classification and profiling. Dataset creation will be explained in detail in later sections.

- Encoder: This converts and prepares the data in the dataset to be used by the classifier. When a new test is carried out, the information is extracted from the dataset, and fed into the encoder, which converts them into usable information; finally, this is fed into the Random Forest (RF) classifier.
- RF (Random Forest) classifier: In this study, an RF classifier was used for classifying all the devices found, since it has been found to be the best classifier in the literature for yielding accurate results while working robustly with limited data. The classifier then creates profiles for all the devices found in the network, and the output is fed back to the dataset.
- Scikit-learn (Sklearn): To create the training and testing sets, Sklearn library was used. Scikit-learn is a free open-source machine learning library for Python, built on NumPy, SciPy, and Cython (Pedregosa et al., 2011).
- Model validation: This module validates the model and calculates the classification accuracy score. The accuracy score acts as an indicator of the model's performance, with higher accuracy scores indicating that the model is better able to identify devices accurately and distinguish between different devices.
- Visualization and logger: The asset profiler also automatically generates graphs, heatmaps, and other images to visually depict information about the assets which are be included in the report produced after the entire testing process.

4. Network Communication Topology builder

BridgeInsight first performs network reconnaissance and creates a network communication topology map for the devices found. Network communication topology defines how nodes or devices are connected to each other and how they communicate. This is a critical step and, in addition, topology graphs provide a comprehensive view of the network infrastructure to ensure that the devices are functioning properly. The proposed topology builder is kept simple using directed graphs and, to build the topology, the network traffic from the environment is captured and translated into graph format, with nodes and edges where the directed edges represent the source and destination of packets. The graphs produced provide a visual representation of network traffic and its connections and allows auditors or engineers to comprehend a high-level view of the environment. Using the network domain address as input, the framework tool starts capturing network traffic in the form of a PCAP file for a specific period of time, which is then parsed and converted to graphs using Networkx, a Python package for the creation and analysis of networks and graphs (Networkx, 2023). For each new IP address in the PCAP file, a node is created in the graph for the source IP address and the destination IP address, connected by an arrow between them to represent the packet flow direction; if the node already exists, then the connection is marked between existing nodes. This way, once all the entries are drawn as a graph, the entire communication topology can be visualized (see **Figure 3**). The limitation of this approach is that it captures the network for only a limited time period and is static in nature during the testing period; however, this can be changed according to the auditor's needs.

5. Profiling

The next part of BridgeInsight identifies the devices it finds in the ship's bridge network and profiles them based on their characteristics using a Random Forest Classifier.

5.1 Dataset

For maritime bridge device identification, the dataset needs to have data regarding specific characteristics of the found devices that enable the differentiation of maritime bridge-specific devices and generic IT/OT devices. To the best of the authors' knowledge, datasets for profiling maritime equipment do not exist publicly, and therefore a new data set was created to test this framework. As it is challenging, disruptive, and risky to conduct a live scan on a working ship's bridge, data was collected from a cyber-physical test-bed set within the Cyber-SHIP lab at the University of Plymouth. Cyber-SHIP is a maritime-cyber research facility that configures real maritime hardware equipment into an electrically accurate representation of a ship's bridge that can be used for testing (Tam et al., 2019). The equipment and software were configured to act as a ship's bridge in the experiments and, therefore, the network data collected is not simulated and so has high fidelity.

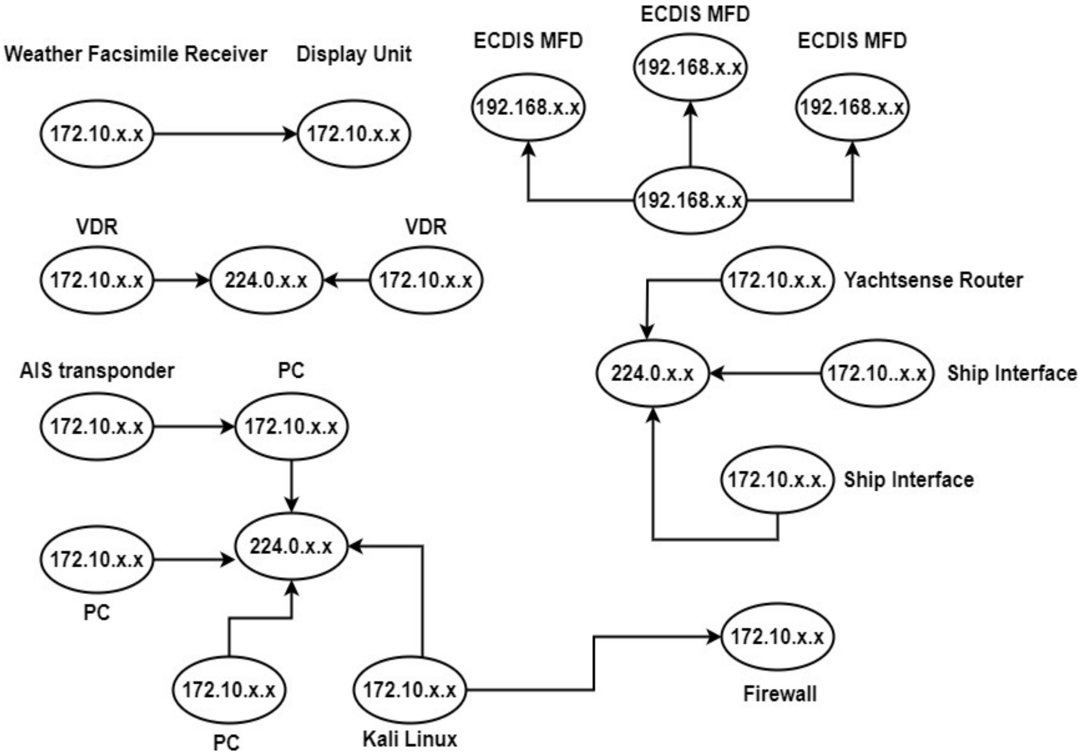


Figure 3 Example of network packet flow topology graph drawn (IP addresses obfuscated for publishing).

As we are validating our framework with our own data, it is important to determine the scope of the dataset before collecting data; (1) how much data is required, (2) what types of data will be collected, and (3) what the expected output will be. The key to identifying maritime equipment is understanding how it differs from its IT counterpart in terms of operations and settings. To account for this, the data attributes gathered were device, different port numbers, operating system, IP address, device type, and manufacturer. A list of common vulnerable network ports was considered for the port numbers attribute. According to Nmap (2023a), a few of the top open TCP ports include port 80 (Hypertext Transfer Protocol or HTTP), 23 (Telnet), 21 (File Transfer Protocol or FTP), 22 (Secure Shell or SSH), 25 (Simple Mail Transfer Protocol or SMTP), 445 (Microsoft SMB), and 53 (Domain), and UDP ports include port 139 (Netbios-ss), port 445 (Microsoft-DS), port 161 (Simple Network Management Protocol or SNMP), port 123 (Network Time Protocol or NTP), etc. This led to having 19 port numbers in the data set as attributes and, based on whether the port is open or closed, within our database, these attributes were denoted with values of either ‘1’ or ‘0’.

A prediction is only as accurate as historical data, and missing values can affect the outcome as, when a dataset is populated, there is the possibility that some values will be missing. A system upgrade, for example, would have little historical data. Since this can lead to difficulties in prediction accuracy, these values are marked as ‘Unknown’ when the profiler lacks confidence. The auditor can then examine them later if necessary, and there is less chance of the profiler tool misguiding an auditor (Altexsoft, 2021). In addition, it is important to consider the different types of data contained in the data set. While port numbers are numeric, operating system data and manufacturer data are alphanumeric categorical data. To make the data consistent, the categorical data was encoded into dummy variables using the Encoder module. This was optimized for the Random Forest classifier in Machine Learning.

5.2 Classifier and experiment setup

As mentioned in section 3, the Random Forest classifier is well suited for this problem for several reasons. Firstly, due to the limited amount of data for profiling ship systems, a module that can work with a limited dataset but still yields high accuracy is important. This is supported by previous related works that looked at identifying IT/IoT devices. What this paper makes clear is that this method works on the less conventional systems in a ship’s bridge environment. Random Forest is a supervised learning algorithm which is an ensemble of multiple decision trees. While decision trees are simple and good classification algorithms, they suffer from the major drawback of overfitting. Overfitting occurs when the model tries to fit the training data to increased accuracy; that is, it attempts to memorize the whole training data, such that it becomes unstable with the introduction of new data. The disadvantages of decision trees are rectified by random forests. During the process of building and splitting the nodes in trees, a random forest generates multiple decision trees based on random sample sizes and a random number of features (Nagesh Singh Chauhan, 2020). Then, the aggregate of all the created decision tree outputs is calculated to classify the data, thus eliminating any bias or chances of overfitting.

In addition to the physical hardware in the Cyber-SHIP lab, a virtual machine running Kali Linux OS was used to collect data from the configured network. The machine was virtually connected to the lab’s ship network, which the topology builder module mapped out automatically. The resulting topology was then used to launch a network port scan using the popular Nmap tool to determine the hosts in the network and their communication protocols (Nmap, 2023b). Nmap also determined which devices were active and up and, once the ping scan was complete, each device from the host list was scanned for open ports and services. The obtained results were filtered and encoded into the dataset, populated from the Nmap and topology builder results.

5.3 Model building

To build and test an accurate model, the dataset needs to be divided into training and testing subsets. Training data is the initial set of data that is fed into the model for learning and finding patterns between the data; that is, it is the historical data that teaches the model to make accurate predictions. The testing data is the set of data used to measure or validate the accuracy of the model. It is the unseen data that can be fed to the model to validate the model. Using train test split () function in the Sklearn library, the data set was split into training and testing sets with a ratio of 70:30, that is, 70 % of the data was split into a training set, while 30 % was reserved for testing. The input of the model was the selected attributes from the dataset, and the output was the ‘Device’ attribute. This again is useful when there is little historical data.

Next, the random forest classifier was built with 100 n estimators, where the number of n estimators denotes the number of decision trees to be built before taking the average of all the outputs and making the prediction. Model fitting is an important step that measures how well the model works with similar data to that of trained data, and a model can be well-fitted, overfitted, or underfitted. Well-fitted models provide accurate predictions or output, while the overfitted model matches the trained data too much, and the underfitted model does not match at all. A random state of value 42 is also provided as the seed of randomness to make sure that the split datasets are the same for every execution. Once the model was trained and fitted, the accuracy score of the model was obtained using the reserved test values and predicted values for 'Device' attributes.

5.4 Classification

For each host found, the model is created sequentially by IP address, trained, and fitted. Once the model has been fitted and tuned using hyper-parameters (explained in detail in section 5.6), the model can be used for profiling. When a new host is identified in the network using the topology builder, the details and characteristics of the host are identified and extracted. This information is written to the dataset with the 'Device' attribute value set to 'dummy'; this information is then encoded to numerical values that act as the input for the model in the form of a list. This input list is then fed into the classifier to make the prediction about the device type, and the output is the value for the 'Device' attribute of the dataset. Once this value is predicted, the 'dummy' value in the data frame is replaced with the predicted output, and then written to the dataset. This enables continuous growth of the dataset and, thus, enables better learning. This process is repeated for all the hosts identified and, at the end of profiling for all devices, the average accuracy score for the model is also calculated.

5.5 Results and findings from the profiling

As mentioned in the previous sections, it is important to profile the asset and show the results in a way that both the technical auditor/tester and the engineer/mariner can understand. To facilitate this, the results are auto-generated in a PDF file. The current experiment set-up in the Cyber-SHIP lab had a bridge network with an average of 30 devices, with a variance of plus or minus 2 devices, depending on the configuration. Input to BridgeInsight was the bridge network's domain address, and the entire process of automated profiling took around 40-45 minutes to complete and produce the PDF report. The Kali Linux virtual machine (VM) that executed the BridgeInsight tool was of 2,048 MB base memory, while the Windows machine that hosted the Kali VM was of Microsoft Windows 11 OS with 32 GB RAM. Results and analysis from the profiler are automatically generated and visually presented in the report by using graphs and charts, and this is discussed in detail in section 6. The model was created with an understanding that maritime equipment will have different characteristics than IT devices, such as different open ports for functionalities. It was also found that certain devices by specific companies had dedicated open ports for configuration and setup. The following results were produced from the analysis of the histogram generated by the profiler.

- The majority of the devices had a web configuration server hosted on port 80; out of 29 devices, 20 had port 80 open.
- All serial-to-IP converters by USR IoT company had port 1501 open, which is assigned to Satellite-data Acquisition System 3, while the ones by Moxa Technologies had port 4,000 open, along with other ports like port 80, for web configuration.
- Another interesting finding was that the VDRs had all open ports, as any Windows PC, even port 3389, used for Remote Desktop Protocol, and port 445, of Windows SMB, implies that a VDR might behave like a PC, and the vulnerabilities and exploits applicable to the Windows system might affect this system as well (Vineetha Harish et al., 2022).
- All the Moxa serial-to-IP converters had port 4900, which is used for firmware upgrade of the device (Moxa, 2023). There are several firmware-related vulnerabilities published in the Common Vulnerabilities and Exposures (CVE) database for Moxa NPort devices, including those that can be crafted and sent via firmware upgrade ports (CVE Mitre, 2020).
- Navigation devices like AIS transponders and weather facsimile receivers by Furuno

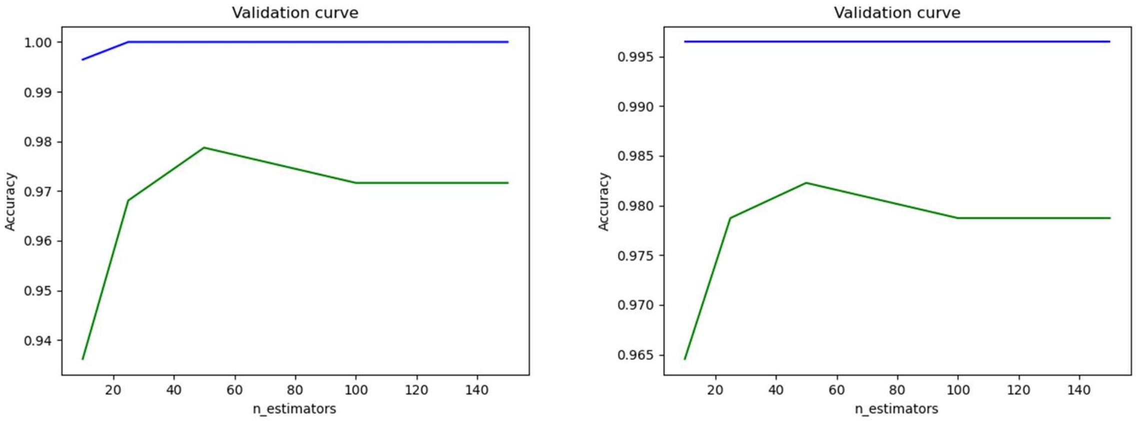
Electric manufacturer had port 10010 open, which is used for broadcasting AIS and NMEA messages.

5.6 Tuning parameters and validation

A major limitation of decision tree algorithms is that they are prone to fitting to extremes; random forest classifiers may reduce these problems to some extent by adding randomness, but may not be free of it entirely. To achieve a balance between overfitting and underfitting, there is a need to adjust and tune the parameters that affect the accuracy and performance of the model, which is known as hyper-parameter tuning. Some of the hyper-parameters used for tuning include n estimators (number of decision trees in the forest), max depth (maximum number of levels allowed in a tree), min samples split (minimum sample required to split a node), min samples leaf (minimum number of samples at leaf nodes), and max features (maximum number of features used in splitting the nodes). Choosing the best value for these hyper-parameters can be done with practical experimentation, trying random values and default values to see how the model performs with those settings. This process can be very tiring and time-consuming; therefore, the better method is to use validation methods, like K-fold cross-validation and validation curves, which help to identify optimal hyper-parameters for the model and diagnose fitting issues. The validation curve plots the performance metrics or the accuracy score of a given model for training and testing data visually against a chosen range of parameters. Analyzing the graph can help in identifying the parameters that may cause underfitting or overfitting of the model.

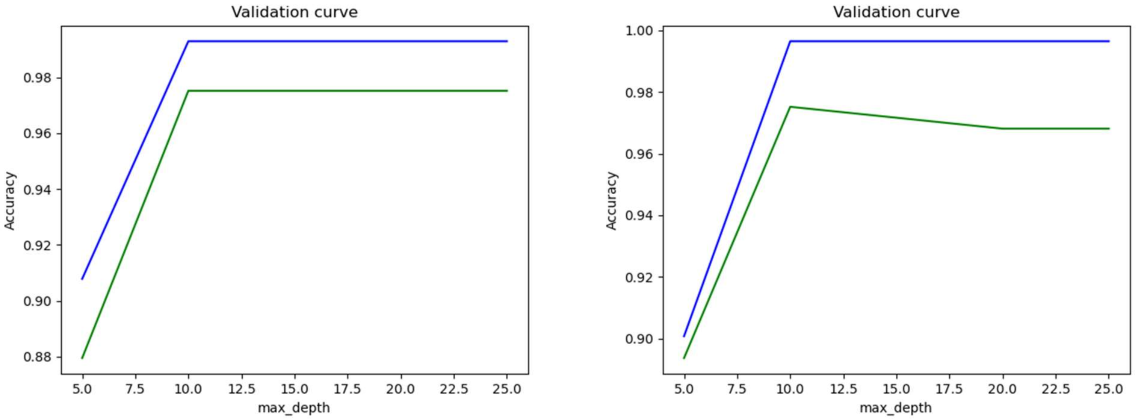
The model was first built with 100 n estimators (default value) and all other hyper-parameters set to the default value. As mentioned in the results section, the average accuracy score for the model with 100 n estimators was 0.988905, and the entire process of classifying all the devices found in the network took 46 minutes. To further refine and tune the model to ensure higher accuracy scores, as well as account for fitting issues and unique features of the devices, validation curves were plotted for different parameters. The blue line in the curves shows the training score, and the green line shows the validation score. If both of these lines are low, the model might be underfitting, and if the training score is high, while the validation score is low, the model might be overfitting. Thus, the optimal value for the hyper-parameter might be the one point where the distance between these lines is shorter, and the accuracy is maximum.

- **n estimators:** n estimators define the number of decision trees built for the forest. To cross-validate and identify the best value for the n estimators, a validation curve was plotted using 2 cross folds (see **Figure 4**), and the values for n estimators considered were 10, 25, 50, 100, and 150. Consider **Figure 4**; in both these graphs, the accuracy score of the cross-validation curve is maximum for the value 50, and then slowly decreases to a stable value. It is important to note that the accuracy score does not change after a particular n estimator value, which means changing the n estimator value does not impact the accuracy score and might indicate overfitting. **Figure 4(a)** shows that the accuracy value changes at 25 n estimators, and thus this value was considered optimal for the model without subjecting the model to overfitting issues. Choosing a lower value for the n estimators might decrease the computational time, while having an effect on the accuracy score.
- **max depth:** Max depth indicates the maximum number of levels the decision trees can have. If set to default value, the model will split until the node attains 100 % purity, or all its data belongs to the same class. To identify the optimal value for the max depth parameter, a validation curve was plotted using two cross-folds (see **Figure 5**), and the values for max depth considered were 5, 10, 15, 20, and 25. As shown in the graphs, the train accuracy score and validation score increase sharply and then stabilize after the max depth value of ten. Therefore, ten was chosen as the optimal value for the max depth parameter, as any greater values do not seem to have an effect on the accuracy of the model.



(a) (b)

Figure 4 Validation curves for n estimator values.

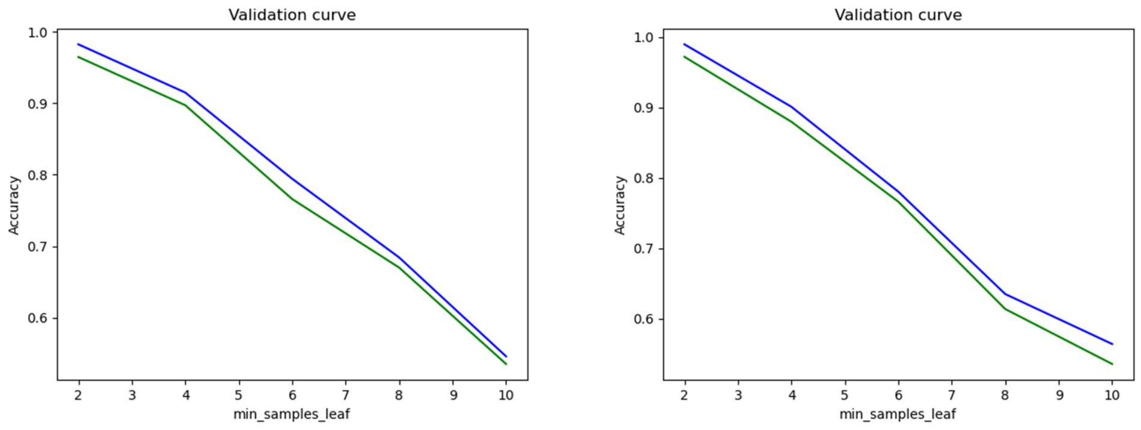


(a) (b)

Figure 5 Validation curves for max depth values.

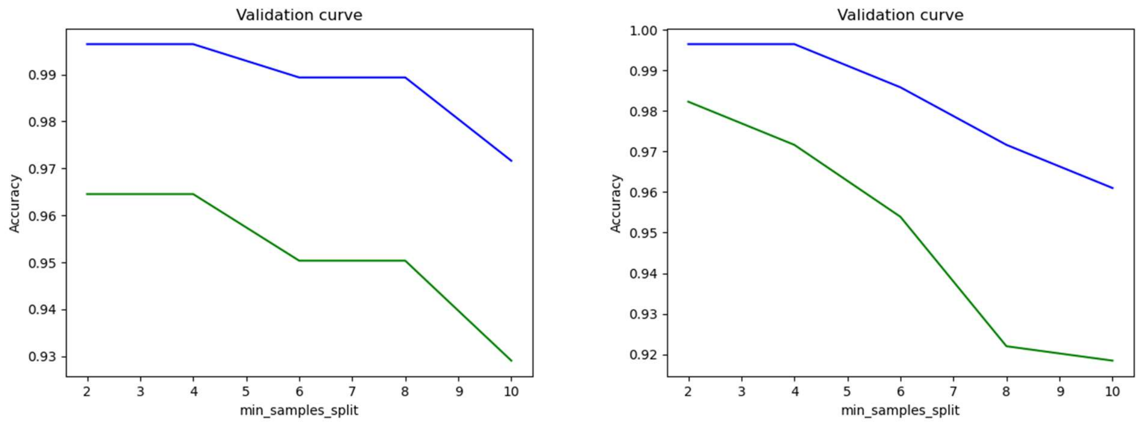
- **min samples leaf:** Min samples leaf value is the minimum number of samples to be present at a leaf node. If, after splitting a node, the internal leaf node has samples less than this value, then it will not be considered as a leaf node, while its parent will be considered as the leaf. This value helps in restricting the size of the tree and the number of levels it grows. Validation curve graphs using the values 2, 4, 6, 8, and 10 were plotted, as shown in the below figures (see **Figure 6**). The default value of min samples leaf in Sklearn is one, which means the leaf node must have at least one sample. The graph plotted clearly shows that the accuracy score value decreases consistently as the min samples leaf value increases. The highest accuracy score is achieved when the min samples leaf value is set to two; therefore, this value was chosen as the best.
- **min samples split:** Similar to the min samples leaf, min samples split represents the minimum number of samples to be present at a node for splitting to happen. After splitting a node, if the number of samples in the internal leaf node is less than this value, then the internal node will not be split. Otherwise, splitting will happen iteratively, until the node is pure. This parameter is also used to limit the growth of the trees and avoid overfitting problems. Validation curve graphs using the values 2, 4, 6, 8, and 10 were plotted, as shown in the below figures (see **Figure 7**). Similar to the previous graphs, the validation curves for this parameter also decrease with the increase in the hyperparameter value. The default value of min samples split in Sklearn is two, and

that value is retained as the optimal value, as the accuracy score is highest compared to when other values are used.



(a) (b)

Figure 6 Validation curves for min samples leaf values.



(a) (b)

Figure 7 Validation curves for min samples split values.

Table 2 Comparison of first classification model and optimized classification model.

Parameters	First classification model	Optimized classification model
n estimators	100 (Default)	25
max depth	None (Default)	10
min samples leaf	1 (Default)	2
min samples split	2 (Default)	2
accuracy score	0.988905	0.986787
average time taken (in minutes)	47	36

Validating and optimizing the model using hyperparameter tuning is necessary, as explained in the previous section. Using the hyperparameters derived from the tuning approach, a new Random Forest classifier model was created. Its accuracy score was 0.9867, while the accuracy score of the model with 100 estimators was 0.98899. The new model took an average of 37 minutes to complete the profiling process, whereas the first model with 100 estimators took 46 minutes. Results showed that, when the estimator values were lowered to 25, accuracy decreased by a minimal value; however, process completion was faster. Therefore, the classifier model needs to be selected based on the purpose of the model considering the accuracy scores and time, while avoiding fitting problems (**Table 2**).

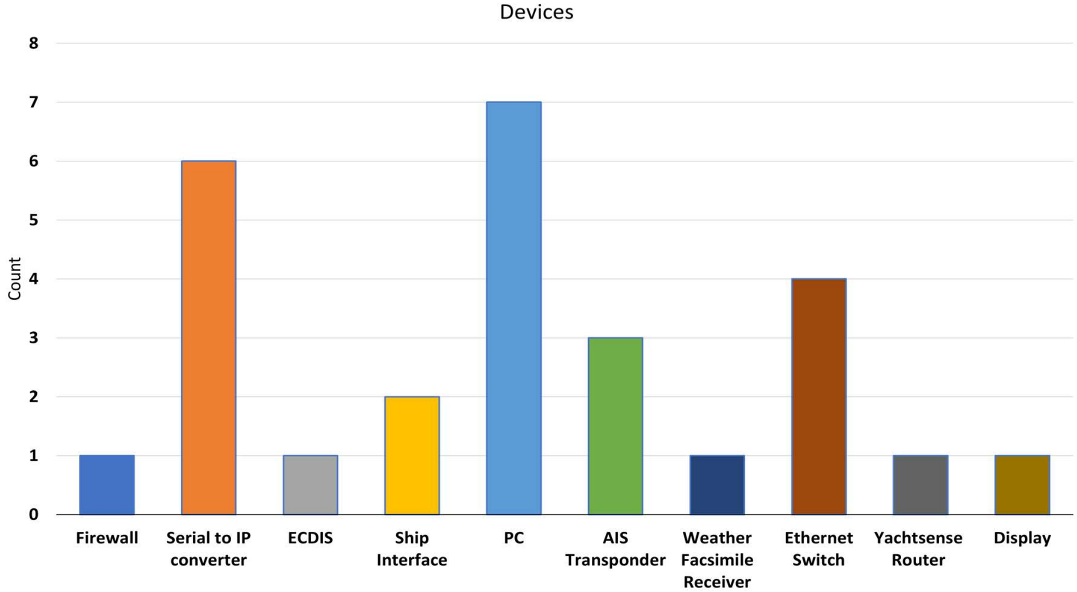


Figure 8 Sample asset count histogram.

6. Visualization and reporting

A penetration tester typically produces security reports manually, highlighting the testing environment, the risks it possesses, its vulnerabilities, and possible mitigations. This paper does this automatically; however, it is important that the quality of information is the same as, or better than, the manual way. The benefit of automation is it makes the pentester/auditor’s job easier and lets them focus on other aspects that AI/ML cannot yet do. Several software programs are available that can automatically generate this report for traditional office-based IT systems. However, it is common for these reports to be very technical in nature, and an individual with a limited understanding of the systems may find them difficult to comprehend. Therefore, when deciding what information to include in a maritime cyber report, it is essential to take into consideration its scope and audience, as the area of maritime cybersecurity is still fairly new. Reports will be more effective in conveying information if they are visually comprehensive, while also including important facts about the vessel’s environment. Moreover, images and graphs are considered better options to convey messages quickly and to a non-cyber-aware audience, like mariners or ship engineers. In order to make the reports more user-friendly, Alharbi (2010) recommends using tables, graphs, bar charts, and pie charts.

6.1 Asset count graph

This graph displays the number of assets for each type. Following the profile and prediction of all hosts with a ’Device’ value in section 5.4, the count of all assets within each device category are shown as a histogram. By using this histogram, auditors and engineers can verify the number of pieces of equipment that are connected to a network on the basis of the type of device. This is useful also to show the changes in a ship over the years. For example, we have seen an increase in IoT devices being added to older ships to improve monitoring and other capabilities; see **Figure 8** for an example.

6.2 Port count graph

An overview of the number of devices that have specified ports open is shown in this graph. A graph like the one below assists in visualizing and reviewing the most commonly open ports in devices, as well as unintentional ports that may be open for testing or auditing purposes. See **Figure 9** for the count of open ports across devices, where the X-axis shows the port numbers, and the Y-axis depicts the number of devices that have the port open.

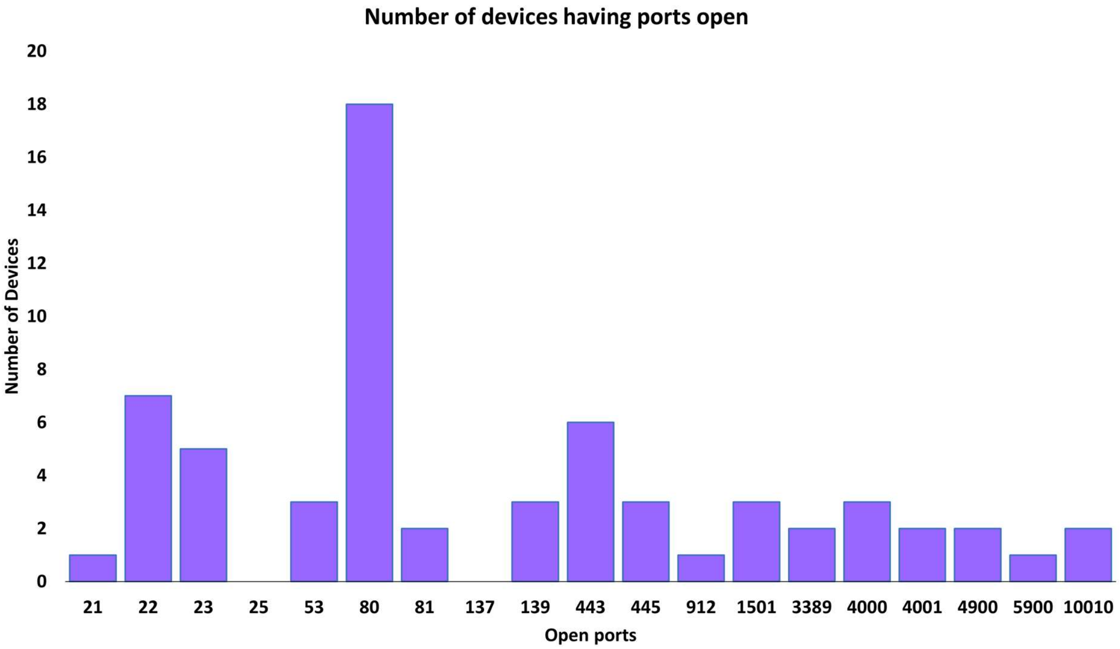


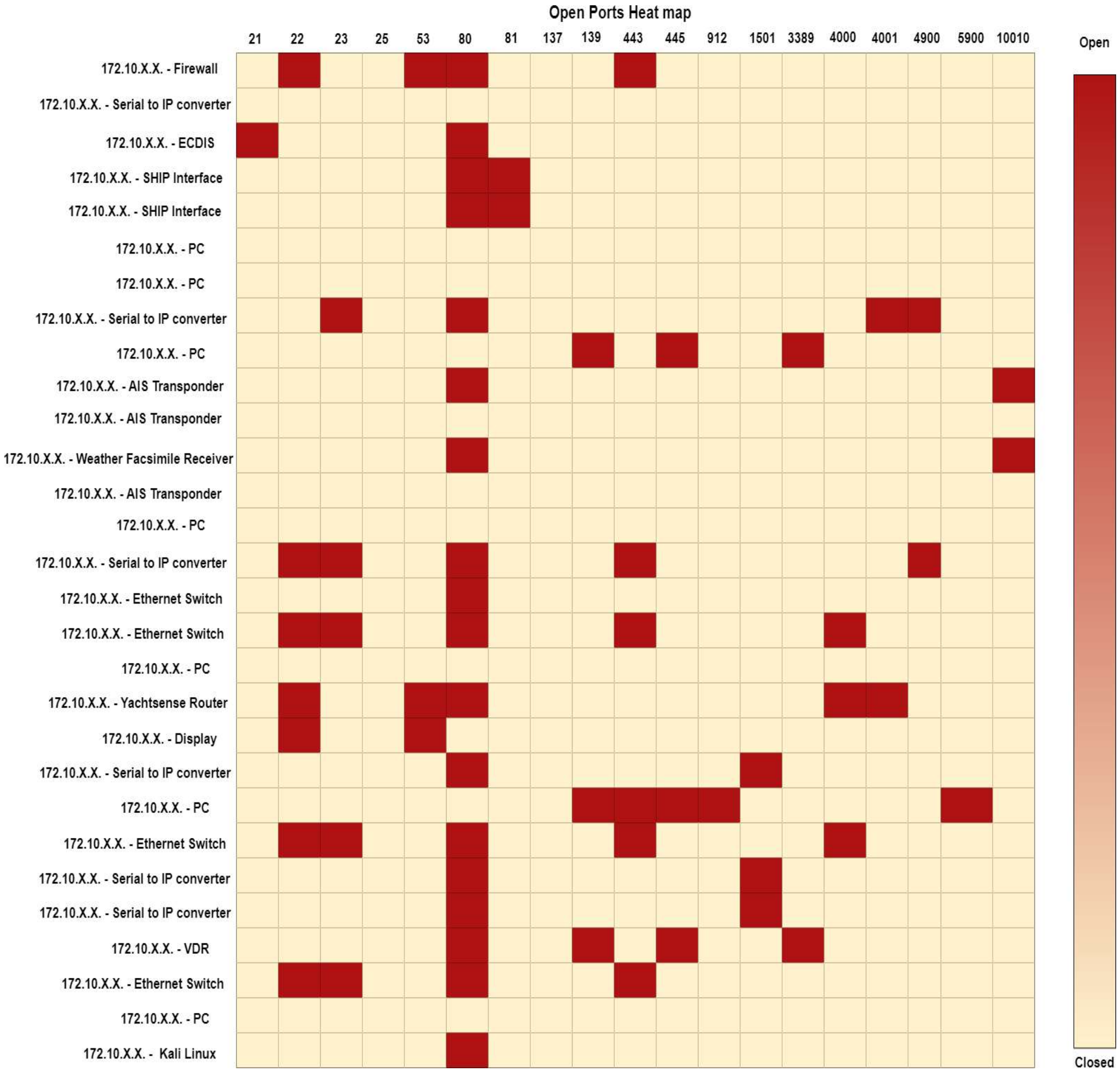
Figure 9 Port count distribution.

6.3 Open ports heat map

Heat maps are visual representations of data using varying colours. This colour coding technique helps the user to understand complex information quickly and easily. Heat maps, when used with suitable color scales and according to similarity, allows the user to be able to see new patterns and structures that are not visible otherwise (Gehlenborg & Wong, 2012). Open ports heat maps illustrate which ports are open on each device. As can be seen in the figure, the X-axis of the map represents different ports, whereas the Y-axis depicts the assets that were profiled previously, along with the predicted device type. Ports in a device that are ‘open’ are coded in red, while those that are ‘closed’ are coded in yellow. As a result, it is possible to understand the characteristics of the various devices in relation to the similarities that exist between the device types and between devices produced by the same manufacturer; see **Figure 10** for an example.

7. Limitations

One of the limitations of the topology builder is its static nature, which means that the user will not be able to edit the topology interactively, but only review and evaluate it based on the graph. The topology builder creates a topology for a given period of time and then produces a network graph of the configuration at the time of the test execution. In the case the user needs to repeat the tests at a later time, the topology builder is executed again, generating a new graph based on the configuration. In addition, there is a possibility that devices will be missing from the topology graph if they are not connected to any other devices and are not communicating. An approach to addressing this would be to obtain the Address Resolution Protocol (ARP) table that contains the list of all the devices, then plot them as single, idle nodes in the graph, and probe the open ports to gather more information. This limitation is somewhat mitigated on ships, as major changes tend to happen around scheduled refits or maintenance, meaning updating the topology can be planned in advance.



	21	22	23	25	53	80	81	137	139	443	445	912	1501	3389	4000	4001	4900	5900	10010
172.10.X.X. - Firewall		🟥			🟥	🟥				🟥									
172.10.X.X. - Serial to IP converter																			
172.10.X.X. - ECDIS	🟥					🟥													
172.10.X.X. - SHIP Interface						🟥	🟥												
172.10.X.X. - SHIP Interface						🟥	🟥												
172.10.X.X. - PC																			
172.10.X.X. - PC																			
172.10.X.X. - Serial to IP converter			🟥			🟥										🟥	🟥		
172.10.X.X. - PC									🟥		🟥			🟥					
172.10.X.X. - AIS Transponder						🟥													🟥
172.10.X.X. - AIS Transponder																			
172.10.X.X. - Weather Facsimile Receiver						🟥													🟥
172.10.X.X. - AIS Transponder																			
172.10.X.X. - PC																			
172.10.X.X. - Serial to IP converter		🟥	🟥			🟥				🟥							🟥		
172.10.X.X. - Ethernet Switch						🟥													
172.10.X.X. - Ethernet Switch		🟥	🟥			🟥				🟥					🟥				
172.10.X.X. - PC																			
172.10.X.X. - Yachtsense Router		🟥			🟥	🟥									🟥	🟥			
172.10.X.X. - Display		🟥			🟥														
172.10.X.X. - Serial to IP converter						🟥							🟥						
172.10.X.X. - PC									🟥	🟥	🟥	🟥						🟥	
172.10.X.X. - Ethernet Switch		🟥	🟥			🟥				🟥					🟥				
172.10.X.X. - Serial to IP converter						🟥							🟥						
172.10.X.X. - Serial to IP converter						🟥							🟥						
172.10.X.X. - VDR						🟥			🟥		🟥			🟥					
172.10.X.X. - Ethernet Switch		🟥	🟥			🟥				🟥									
172.10.X.X. - PC																			
172.10.X.X. - Kali Linux						🟥													

Figure 10 Open ports heat map.

When it came to the profiling phase, the main limitation was the lack of public datasets regarding maritime equipment. This limitation was mitigated with access to a hardware testbed that had ship systems in a ship's bridge configuration. Data could, therefore, still be collected from real systems for the construction of this dataset, followed by verification. A human supervisor will, thus, need to verify the collected data during the initial stages, even if the classification process is automated, in order to ensure accuracy. Once the dataset has been created, the model profiles the hosts found during each test execution, while returning the results back to the dataset, thereby allowing the dataset to continuously grow. However, it is necessary to include large quantities of data from various devices, which is a limitation in terms of the number of devices available to the researcher. This could be resolved in the future by collecting data from live networks onboard ships.

The limited quantity of data available could also introduce overfitting or underfitting problems to the classifier model. Random Forest classifiers perform better than decision trees in incorporating randomness and reducing fitting, but they do not completely eliminate it. In the Random Forest model, there will always be a trade-off between accuracy and computation time. It may take longer to complete the model when using a large number of trees to construct the forest, but the accuracy score may increase as a result. Considering that the accuracy value stops improving after a certain threshold value, it might not be best to have a large number of decision trees. It is also important to tune and optimize other parameters, since trees are sensitive to parameter values. The profiler may take a considerable amount of time when there are a large number of devices in the configuration so, in such a situation, reducing computational power and resources is considered the best solution.

8. Discussion and conclusion

In this paper, we discussed BridgeInsight, an asset profiler that users could use to manage their asset inventories and comply with forthcoming regulations and requirements such as those in IACS UR 26 and UR 27. With a given network configuration, the tool automatically constructs a topology graph of communication flows, and intelligently identifies the devices or assets on the bridge using the Random Forest classifier algorithm. To ensure that potential users (i.e., mariners and engineers) understand the results, BridgeInsight also provides detailed information about the device(s) and their network(s). Onboard crew and engineers can use the graphs and charts produced by the tool to better understand their networks and systems and manage assets more efficiently, and better inform maintenance and security efforts. Generally speaking, we found that this ship-focused tool was more accurate in classifying bridge equipment than similar works designed for IT/IoT environments. We theorize that this could be the result of the number of bespoke and novel system solutions available in maritime space, which therefore provided more unique properties for the ML to process. One possible area of future work is to see if this methodology tends to be highly effective in the wider maritime or cyber-physical topics of cybersecurity.

Security testers and auditors can use the tool on board vessels to gather situational awareness information about the systems and environments they are working in. A tool like this could reduce time and effort, as a manual inspection of systems could be conducted in the order of days instead of minutes, especially if panels need to be removed to access hidden components. An automated, non-intrusive tool can, therefore, speed up the testing process and requires less specialized maritime expertise. We envision automated asset detection and classification to have even more benefit in future work, as pentesters can use those capabilities to build specific exploits for the system and network they are targeting. Many security testing frameworks, such as Metasploit, offer exploit modules for IT devices and OT systems, such as SCADA components. A detailed ship-based asset inventory can also help select the right and most suitable exploit or test type for the device. Future work on building an ethical ship-based penetration testing tool is one way to extend work in this study. This can also help cyber risk assessments, where people responsible for the devices/assets can identify the ones that are critical to operations, ensuring that they are updated and patched and that proper security controls are in place.

Acknowledgement

The authors would like to thank Stephanie Riley and colleagues from the Cyber-SHIP lab, namely Rory Hopcraft and Juan Palbar Misas, for their invaluable support and comments on earlier drafts. The authors would also like to thank Wesley Andrews and Luke Christison for helping to set up the experiment test bed and environment. This research was part of the Cyber-SHIP lab project at the University of Plymouth. The authors are grateful to the project funder- Research England- and our industry partners who supported our research by providing tools and equipment.

References

Acord, J. (2017). *Situational awareness and ICS Using GRASS MARLIN: Infosec*. Retrieved from https://resources.infosecinstitute.com/topics/scada-ics-security/situational-awareness-ics-using-grass-marlin

Alharbi, M. (2010). *Writing a penetration testing report*. Retrieved from https://sansorg.egnyte.com/dl/yNfjHOQix8

Altexsoft. (2021). *Preparing your dataset for machine learning: 10 steps - AltexSoft*. Retrieved from https://www.altexsoft.com/blog/datascience/preparing-your-dataset-for-machine-learning-8-basic-techniques-that-make-your-data-better

Ammar, N., Noirie, L., & Tixeuil, S. (2019). *Network-protocol-based IoT device identification* (pp. 204-209). In Proceedings of the 4th International Conference on Fog and Mobile Edge Computing, Rome, Italy. https://doi.org/10.1109/FMEC.2019.8795318

Amro, A. (2021). *Cyber-physical tracking of IoT devices: A maritime use case. Norwegian ICT conference for research and education*. Retrieved from https://ojs.bibsys.no/index.php/NIK/article/view/961

Auvik. (2022). *What is an ARP Ttable? Address Resolution Protocol 101-Auvik*. Retrieved from https://www.auvik.com/franklyit/blog/what-is-an-arp-table

Auvik. (2023). *Network mapping software: Auvik Networks*. Retrieved from https://www.auvik.com/features/network-navigation

Bagur, J. (2023). *GPS NMEA 0183 Messaging Protocol 101: Arduino Documentation*. Retrieved from https://docs.arduino.cc/learn/communication/gps-nmea-data-101

Bolbot, V., Kulkarni, K., Brunou, P., Banda, O. V., & Musharraf, M. (2022). Developments and research directions in maritime cybersecurity: A systematic literature review and bibliometric analysis. *International Journal of Critical Infrastructure Protection, 39*, 100571. https://doi.org/10.1016/j.ijcip.2022.100571

Bothur, D., Zheng, G., & Valli, C. (2017). *A critical analysis of security vulnerabilities and countermeasures in a smart ship system* (pp. 81-87). In Proceedings of the 15th Australian Information Security Management Conference. Perth, Western Australia.

Censys. (2023). *Exposure management and threat hunting solutions: Censys*. Retrieved from https://censys.io

CVE Mitre. (2020). *CVE-2020-12117*. Retrieved from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-12117

FutureIoT. (2020). *Surge in obsolete network devices pose cybersecurity risk: FutureIoT*. Retrieved from https://futureiot.tech/surge-in-obsolete-network-devices-pose-cybersecurity-risk

Gehlenborg, N., & Wong, B. (2012). Heatmaps. Nat Methods, 9, 213. https://doi.org/10.1038/nmeth.1902

Hamad, S. A., Zhang, W. E., Sheng, Q. Z., & Nepal, S. (2019). *IoT device identification via network-flow based fingerprinting and learning* (pp. 103-111). In Proceedings of the Proceedings 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 13th IEEE International Conference on Big Data Science and Engineering. Rotorua, New Zealand. https://doi.org/10.1109/TrustCom/BigDataSE.2019.00023

IACS. (2022a). *E26 - Cyber Resilience of Ships. London: International Association of Classification Societies*. Retrieved from https://iacs.org.uk/download/14104

IACS. (2022b). *E27 - Cyber Resilience of On-board Systems and Equipment. London: International Association of Classification Societies*. Retrieved from https://iacs.org.uk/download/14105

IACS. (2022c). *Unified Requirements*. Retrieved from https://iacs.org.uk/publications/unified-requirements

IASME. (2021a). *IASME launches cyber security scheme for the maritime industry*. Retrieved from https://iasme.co.uk/cyber-blog/iasme-launches-cyber-security-scheme-for-the-maritime-industry

IASME. (2021b). *Maritime Cyber Baseline Self-Assessment Questions*. Retrieved from https://iasme.co.uk/wp-content/uploads/2022/11/Maritime-Question-Booklet_V1.1.pdf

Jeon, D. K., & Lee, Y. (2014). A ship area network with WiMedia wireless gateway applying a cooperative transmission. *Contemporary Engineering Sciences, 7*(23), 1235-1243. http://dx.doi.org/10.12988/ces.2014.49153

Keary, T. (2022). *PCAP: Packet Capture, what it is what you need to know*. Retrieved from https://www.comparitech.com/net-admin/pcap-guide

Kretschmann, L., Zacharias, M., Klover, S., & Hensel, T. (2022). *Machine Learning in Maritime Logistics*. Retrieved from https://shipzero.com/wp-content/uploads/2022/12/10015_compressed.pdf

Liu, Y., Wang, J., Li, J., Niu, S., & Song, H. (2022). Machine learning for the detection and identification of internet of things devices: A survey. *IEEE Internet of Things Journal, 9*(1), 298-320. https://doi.org/10.1109/JIOT.2021.3099028

Loomis, W., Singh, V., Kessler, G. C., & Bellekens, X. (2021). *Raising the colors: Signaling for cooperation on maritime cybersecurity*. Retrieved from https://www.atlanticcouncil.org/wp-content/uploads/2021/10/Raising-the-colors-Signaling-for-cooperation-on-maritime-cybersecurity.pdf

Moxa. (2023). *Which are the most common TCP and UDP ports used by serial-to-Ethernet device servers?* Retrieved from https://www.moxa.com/en/support/product-support/product-faq/most-common-tcp-udp-ports-used-by-serial-to-ethernet-device-servers

Nagesh Singh Chauhan. (2020). *Decision Tree Algorithm, Explained: KDnuggets*. Retrieved from https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

Networkx. (2023). *NetworkX documentation*. Retrieved from https://networkx.org

NIST. (2022). *National Vulnerability Database (NVD)*. Retrieved from https://www.nist.gov/programs-projects/national-vulnerability-database-nvd

Nmap. (2023a). *Chapter 4. Port Scanning Overview: Nmap Network Scanning*. Retrieved from https://nmap.org/book/port-scanning.html#most-popular-ports

Nmap. (2023b). *Nmap: The Network Mapper - Free Security Scanner*. Retrieved from https://nmap.org

NSA. (2017). *GRASSMARLIN user guide*. Retrieved from https://github.com/iadgov/GRASSMARLIN/blob/master/GRASSMARLINUserGuide.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Rapid7. (2023). *Metasploit Framework: Metasploit Documentation*. Retrieved from https://docs.rapid7.com/metasploit/msf-overview

Shodan. (2023). *Shodan Search Engine.* Retrieved from https://www.shodan.io

Sivanathan, A., Gharakheili, H. H., & Sivaraman, V. (2018). *Can we classify an IoT device using TCP port scan?* In Proceedings of the IEEE 9th International Conference on Information and Automation for Sustainability. Colombo, Sri Lanka. https://doi.org/10.1109/ICIAFS.2018.8913346

Sivanathan, A., Sherratt, D., Gharakheili, H. H., Radford, A., Wijenayake, C., Vishwanath, A., & Sivaraman, V. (2017). *Characterizing and classifying IoT traffic in smart cities and campuses* (pp. 559-564). In Proceedings of the 2017 IEEE Conference on Computer

Communications Workshops. Atlanta, USA. https://doi.org/10.1109/INFCOMW.2017.8116438

SolarWinds. (2023). *Network Topology Mapper - Network Mapping Software*. Retrieved from https://www.solarwinds.com/network-topology-mapper

Tam, K., & Jones, K. (2019). MaCRA: A model-based framework for maritime cyber-risk assessment. *WMU Journal of Maritime Affairs, 18*(1), 129-163. http://dx.doi.org/10.1007/s13437-019-00162-2

Tam, K., Forshaw, K., & Jones, K. (2019). *Cyber-SHIP: Developing next generation maritime cyber research capabilities*. In Proceedings of the International Conference on Marine Engineering and Technology Oman 2019. Muscat, Oman. https://doi.org/10.24868/icmet.oman.2019.005

US Coast Guard. (2021). *Cyber Strategic Outlook AUG 2021*. Retrieved from https://www.uscg.mil/Portals/0/Images/cyber/2021-Cyber-Strategic-Outlook.pdf

Vineetha, H. A., Tam, K., & Jones, K. (2022). *Investigating the security and accessibility of voyage data recorder data using a USB attack* (pp. 74-80). In Proceedings of the 7th International Conference on Cyber-Technologies and Cyber-Systems. Valencia, Spain.