Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle

Anatolij Bezemskij, George Loukas, Richard J. Anthony, Diane Gan
Department of Computing & Information Systems
University Of Greenwich
London, United Kingdom
Email: {a.bezemskij, g.loukas, r.j.anthony, d.gan}@gre.ac.uk

***Abstract*—Security is one of the key challenges in cyber-physical systems, because by their nature, any cyber attack against them can have physical repercussions. This is a critical issue for autonomous vehicles; if compromised in terms of their communications or computation they can cause considerable physical damage due to their mobility. Our aim here is to facilitate the automatic detection of cyber attacks on a robotic vehicle. For this purpose, we have developed a detection mechanism, which monitors real-time data from a large number of sources onboard the vehicle, including its sensors, networks and processing. Following a learning phase, where the vehicle is trained in a non-attack state on what values are considered normal, it is then subjected to a series of different cyber-physical and physical-cyber attacks. We approach the problem as a binary classification problem of whether the robot is able to self-detect when and whether it is under attack. Our experimental results show that the approach is promising for most attacks that the vehicle is subjected to. We further improve its performance by using weights that accentuate the anomalies that are less common thus improving overall performance of the detection mechanism for unknown attacks.**

1. Introduction

Vehicular cyber security has traditionally focused on passive attacks and especially on protecting the confidentiality of communications between vehicles or vehicles and smart infrastructures. However, over the last few years, autonomous vehicles have become a routine target for experimental cyber attacks, as demonstrated as early as 2009 by the University of Washington [1], [2] and in numerous blackhat conferences since then. As a result, there is a need for protection systems appropriate for active attacks against an autonomous vehicle's integrity or availability, and the corresponding impact on its actuation. Assuming that some attacks do get through regardless of the preventive measures, one needs to equip a vehicle with a mechanism to detect when this happens and potentially alert an operator or trigger some automated countermeasure. The focus of this work is on the real-time detection of the existence of an attack against a robot. We address both cyber-physical attacks, which are security breaches in cyber space that have an adverse effect in physical space, and physical-cyber attacks which are the reverse [3]. For this, we have developed an autonomous robotic vehicle with a variety of sensor and communication technologies typically found in the industry.

To ensure that any solutions developed are highly practical, we have set the following requirements:

- Detection should be real-time, so as to be able to support rapid and effective countermeasures.
- Detection should be carried out by the vehicle itself, so as to be applicable to autonomous vehicles with limited or no communication with their human operators.
- Detection should not rely on the availability of knowledge of previous attacks, so as to be applicable to unknown attacks too.

We do not rely on attacks on cyber-physical systems being frequent enough to allow for the gathering of a realistic body of knowledge on their impact. To meet these requirements, detection should be behaviour-based rather than knowledge-based. To address the above requirements, we have produced an onboard mechanism that monitors data related to cyber (communication and computation) and physical (actuation and sensing) features of the robot in real-time. During the training phase, the robot learns the normal range for the values of each feature monitored. In actual operation, it tracks the cyber and physical features that are in an abnormal state (beyond their learnt range) and accordingly reasons on whether a vehicle is in an attack state or not. The overall emphasis of the mechanism towards more tolerance for false positives or more tolerance for false negatives is configured by a sensitivity index, which determines the length of the normal range considered by the robot. We further improve on the detection accuracy achieved with this approach by also utilising individual weights for each feature, which are finetuned in a dedicated configuration phase.

1.1. Related Work

While very mature for conventional computer systems, the field of intrusion detection is relatively new in the area of cyber-physical systems, such as vehicles and mobile robots. A relatively common approach is to use a human expert to first specify the safe and unsafe states of the vehicle and

determine a large number of rules that cover all potential states, in what is known as behaviour-specification intrusion detection [4], [5]. Rules can also be determined through a more automated learning phase without the involvement of a human expert: The vehicle is subjected to a series of different attacks, observing their impact and training a machine learning system to recognise these. Examples of such supervised learning approaches for the detection of attacks against robotic vehicles can be found in [6], [7], [8], where the rules are formed by a decision tree, which takes into account both cyber and physical features. Real-time capture of an attack's physical impact, such as vibration of the chassis due to repetitively entering and existing safe mode during a denial of service attack, has been shown to improve detection accuracy and latency.

When a vehicle does not operate in isolation, but belongs to a team of vehicles, which can make similar observations about their environment and each other, intrusion detection can be based on the identification of misbehaviour of one of the members of the team. There, reputation-based approaches [9] and voting algorithms [10] can prove very useful. For instance, if one vehicle veers off the pre-defined route or reports very different sensor data, this can be considered as an indication that it may have been compromised.

Most of the research presented above makes assumptions that are largely unrealistic in the operational environment of a cyber-physical system such as a robotic vehicle (whether autonomous or not). Assuming that a new attack will look like one that has been seen before is reasonable for conventional computer networks, where millions of variations of the same attacks can be seen in the same year. For cyber-physical systems, this is less so, because attacks are less common and have very a different impact depending on the type of system targeted. As a result, knowledge-based approaches, where the vehicle is trained to see specific attacks perform poorly when they encounter new types of attack. At the same time, assuming that a robotic vehicle will belong to a team, where group observation can help spot signs of cyber compromise can be unrealistic in many operational environments.

Researchers have experimented with methods for detecting anomalies, but usually only for a particular aspect of a vehicle's operation. An example for aircraft is the detection of false automatic dependent surveillance-broadcast (ADS-B) messages, used by aircraft to broadcast their position to other aircraft and to air traffic control. Strohmeier et al. [11] achieve detection by monitoring statistics regarding the received signal strength (RSS), as it is assumed that false signals would be coming from the ground and thus would have different RSS than signals coming from aircraft. A similar logic can be followed to protect autonomous vehicles that rely on GPS signals, as, coming from satellites, legitimate GPS signals are naturally much weaker than spoofed signals that would come from a terrestrial source [12].

A first attempt to provide completely sensor-agnostic and onboard intrusion detection that is applicable to unknown threats and takes into account both cyber and physical sources has been made in [13]. Here, we extend this work considerably by providing a method to quantify the degree to which a vehicle is likely to be under attack without relying on a learning phase, and further improve it with a mechanism that assigns weights to the different data sources. We validate this approach with real-world experiments involving a variety of normal and attack conditions.

2. Robotic Testbed System Design

Our testbed is a highly modular robotic vehicle developed from the ground up for the purposes of this research (Figure 3). It contains a large variety of sensors, actuators and communication channels widely used in the industry. The latter include CAN, RS-485, WiFi and ZigBee.

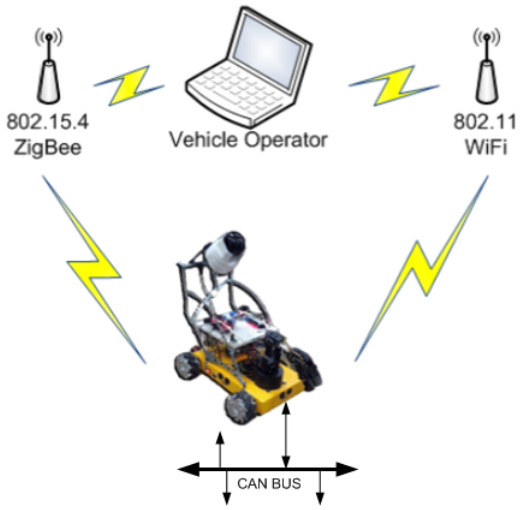

Figure 1. High-level communication diagram

The various sub-systems are integrated such that system components produce signals and feedback used by other system components to change overall system behaviour. Several components that are mentioned in Table 1 produce instrumentation data which is used as cyber or physical domain indicators. The combination of such indicators can produce additional meta-data that can be used to identify a particular behaviour of a system. All indicators that are used are generalised and are treated as a data source for example the compass bearing signal output represents the orientation of the vehicle in degrees, but in an autonomous system it is treated as a stream of numerical values without context or units. Processing is distributed across the various embedded processors on the testbed platform (Table 1). Overall the system contains six processing nodes, five of which are AVR-CAN development boards clocked at 16 MHz, and one STK300 Kanda board powered by Atmel ATMega1281 chip clocked at 8 MHz.

The vehicle is able to undertake a variety of autonomic tasks, such as navigation based on the logical mission layer that represents a sequence of steps given to the testbed. Sensors allow the vehicle to navigate autonomously in an environment using the compass bearing to keep track of

TABLE 1. EQUIPMENT INSTALLED IN THE ROBOTIC VEHICLE TESTBED

Feature	Purpose
CAN bus	Internal communication
ZigBee	External communication
WiFi	Media streaming
Compass Bearing	Navigation correction
DC Motors	Movement
Ultrasonic Rangers	Collision avoidance

the direction, ultrasonic rangers for collision detection and avoidance, and pitch and roll sensors to make direction corrections and inform the system of environment volatility. There is also a sensor that measures the temperature of the heat sink connected to the on-board voltage regulators which supply power for the camera and robotic arm. In this way, the system is able to determine if these heavy-current-drawing system components are in use. These sensors and additional meta-data extraction allow automatic characterisation of the real-time behavioural profile of the vehicle whilst in operation.

To gather the data for off-line analysis, we use an external workstation. Sensor data from the vehicle is collected and stored in a knowledge base. Communication between the workstation and the vehicle is achieved using a dedicated ZigBee network. The ZigBee connection also enables us to transmit commands to the testbed (e.g. to initiate missions). The camera is a self-contained unit; its audio and video feeds are streamed using a standard WiFi protocol. An overview of high-level communication architecture between workstation and robotic testbed vehicle can be seen in Figure 1.

The robotic vehicle is capable of accepting both simple remote commands in terms of navigation, camera streaming and the operation of the robotic arm that is attached to it and complex missions uploaded to it. For security purposes, commands received are executed only if the sender is within a list of authorised ZigBee nodes and the command is in the correct format. The robotic vehicle testbed does not send any commands to any external nodes within the ZigBee network. The testbed will only periodically report its instrumentation data to a verified connected workstation. The instrumentation report periodicity is 1 s, due to the low bandwidth ZigBee protocol and unique ZigBee ZE10 module behaviour. Therefore higher-rate sample aggregation is performed on-platform on the sensor hosting nodes.

For communication between system components, the testbed uses a CAN bus. This bus is used to share overall sensor data from data sources, including additional meta-data extracted during data analysis by the processing nodes. The internal communication architecture is shown in Figure 2. This data is retransmitted to other nodes through gateways and is collected at the reporting node which transmits data to the workstation when appropriate.

The software structure of the robotic vehicle testbed uses a layered architecture, which separates the different levels of reasoning from the lowest physical sensor level, represented by individual embedded nodes performing analog to digital conversions interpreting signals into an understandable soft-

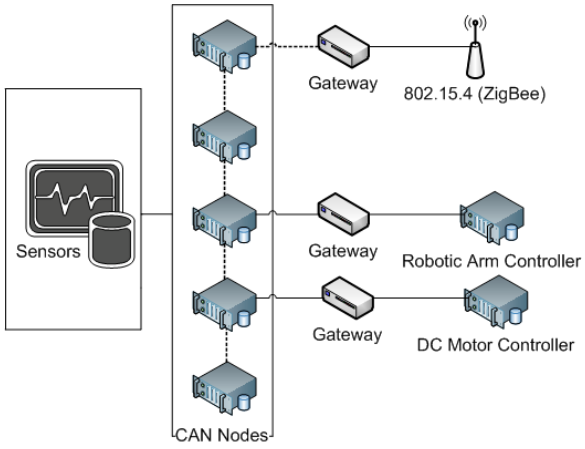

Figure 2. Internal Communication: gateways connect different subsystems

ware language. The next-higher level is the classification layer where data is analysed using statistical analysis approaches, such as exponential smoothing to determine the trends in the data. A level higher, we have an autonomous module controller layer which controls actuating capabilities based on the data received from the lower layers of the model. The autonomic module controller layer is a set of autonomic controllers that are carrying out their defined tasks, such as robotic arm movement or navigational control. A mission layer then collects knowledge from autonomic controllers and evaluates if the expected mission goal has been achieved. The layered software approach improves flexibility and maintainability in terms of software development for the robotic vehicle testbed, as all these layers are implemented as a set of libraries that can be extended further.

Figure 3. Robotic vehicle testbed

3. Experimental Environment

The training and testing phases of the experiments have been conducted in the Queen Mary Building at Greenwich University. The irregular surface of the uneven stone flooring dents and lumps (Figure 4) provides the desired stochasticity

for the different data sources, as well as a challenging environment for the mobility of the vehicle. At the same, it is a controlled environment where we can ensure repeatability of the experiments without any foreign objects or weather modifying the parameters of each iteration. This also allows us to identify the behavioural profile of an environment based on the data source information. The corridor has a set of inset door openings on either side, which facilitate physical observation of the effects of different attacks and of the periodic behaviour of the sensors (especially the ultrasonic ones), as the vehicle passes by.

Figure 4. Experimental environment. An old corridor with irregular surface and uneven stone flooring dents and lumps at the University of Greenwich.

The corridor is 28m long and the distance from wall to wall is 2m and constant. The experiments were repeated eight times to ensure that the collected data set is representative and can be used for creation and evaluation of the behavioural profile. The behavioural profile is built using patterns of the variation and background noise in data sources; mainly we are looking at the spikiness of the data variations and the variety of deviations. The experimental environment facilitates repeatability and contains static elements that can be used as guideline features during analysis of gained data, but it also introduces significant stochastic elements which are essential for understanding the normal levels of noise and variability in sensor signals.

The experimental scenario evaluated in this paper is a mission in which the robotic vehicle testbed has to reach the end of the corridor using its own sensing capabilities. The complexity of such a mission is not obvious. The uniqueness of the flooring surface disrupts the direction of the vehicle, forcing it to continuously adapt the speed of its motors and its direction and ensure that it maintains a safe distance from the walls during operation. The scenario was chosen due to the structural uniqueness of the vehicle, and as such the scenario exercises all sensor capabilities. The experiment is organised in two phases. The first is a training phase, where over several runs a learning data set is collected that allows us to create a "normal" behavioural profile. The second phase is to evaluate the recognition of this profile.

4. Methodology

4.1. Signature of normal behavioural profile

In [13], we have described how signatures are formed and can be used for anomaly or threat identification. This learning phase is shown as "L1" in Figure 5. It is based on an initial signature generation to establish the normal behaviour profile of the sensors on the system. The data from the sensors is transformed into a generic data source format that allows the system to reason about them identically. The learning phase forms a normal behaviour profile based on signature characteristics of each data source and forms normal behaviour variation that is used during the validation phase.

After the learning phase, dynamically detected values are compared with learnt normal behavioural profile signatures. The term "anomaly" is used to denote that a signal characteristic has been measured to be outside its expected normal range. The signature is formed of 11 characteristics which facilitate learning the normal value range limits, as shown in Table 2. Differences and deviations from the standard deviation are called spikes and these characteristics can be seen in Table 2. The validation phase, shown as "V" in Figure 5, classified behaviour based on an overall anomaly index represented by a number of anomalies in the system.

TABLE 2. SIGNATURE CHARACTERISTICS MONITORED IN REAL-TIME ONBOARD THE VEHICLE

Value Type	Characteristic
Raw	Minimum Maximum
Exponential Smoothing	Minimum Maximum Lowest Difference Highest Difference
Deviation	Standard Deviation
Spike Areas	50% - 100% 100%-150% 150%-200% Over 200%

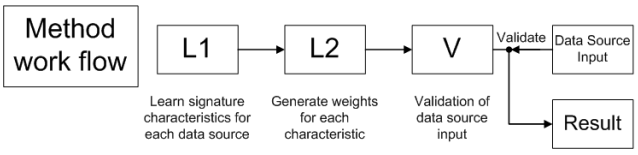

Figure 5. Methodology work flow

Each deviation of a signature characteristic is counted to represent an outgoing level of threat from the data source. These deviations are summarized to produce an anomaly index for the data source, this index represents the deviation level.

4.2. Anomaly Weighing and Indicator Confidence

To strengthen the detection performance, we have introduced an additional learning phase, shown as "L2" in Figure

5, which tunes the system by assigning weights to sources according to their likelihood of appearing anomalous in some normal scenarios too. The focus of this phase is to learn the number of individual signature characteristic anomalies that may be encountered in a non-attack condition, arising due to environmental noise.

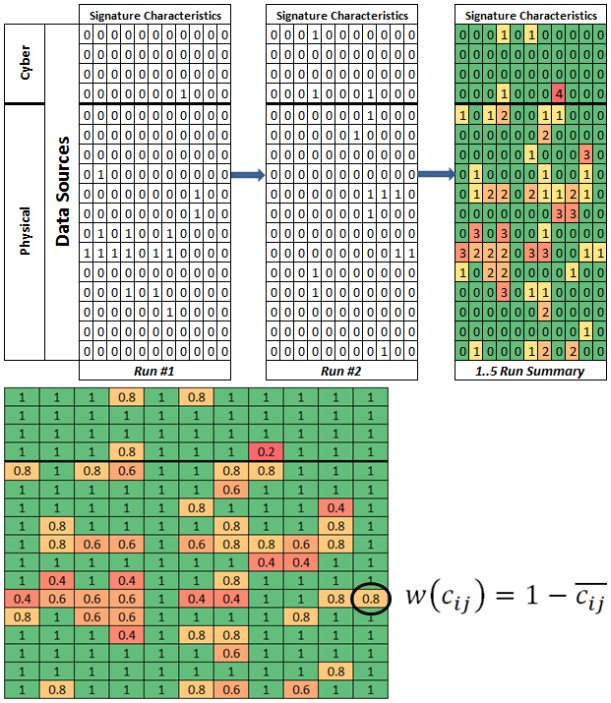

Figure 6. Matrix of the anomalies identified. Each row corresponds to a data source and each column to a signal characteristic measured for each source. The colour coding indicates anomalies

In Figure 6, we demonstrate how we summarise anomalies by taking the system data source signatures from five non-attack situations. Only two are shown here as a demonstration. To reduce the importance of anomalies that tend to occur in a non-attack environment we calculate the weight of each signature characteristic anomaly sample $w(c_{ij})$ in the following way: for a number of n scenarios S_ℓ, $1 \leq \ell \leq n$, we take the complement of the mean of each signature characteristic anomaly sample $c_{ij(\ell)}$, where i represents a data source and j represents a signature characteristic:

$$w\left(c_{ij}\right) = 1 - \overline{c_{i,j}} = 1 - \left(\frac{\sum_{\ell=1}^{n} c_{ij(\ell)}}{n}\right)$$

which produces the weight of an anomaly sample for the signature characteristic. This allows the system to derive a more precise score taking into account the anomalies that tend to be less indicative of an attack as they persist in a non-attack conditions.

The calculated value represents the weight of a signature characteristic. If the system learns that a particular signature characteristic has a high probability of anomaly occurrence in a non-attack mission scenario then the importance of such anomaly is reduced. This generates a lower anomaly index for the data source's signature. The sum of all weighted anomalies generates an overall anomaly index that is used as a reference in the intrusion detection mechanism. To improve the methodology further we introduce a dynamic variable that acts as a controller of the "normality" threshold. The "normality" variation is formed during the "L2" phase. The overall anomaly index generated from the non-attack experiments is used as a mean reference and the dynamic variable controls the variation. This allows the detection mechanism to identify anomalous behaviour in two cases: when multiple anomalies are detected generating a high overall anomaly index, as well as when anomalies are not detected, therefore generating a low overall anomaly index. An overview of a work flow of the intrusion detection mechanism can be seen in Figure 7.

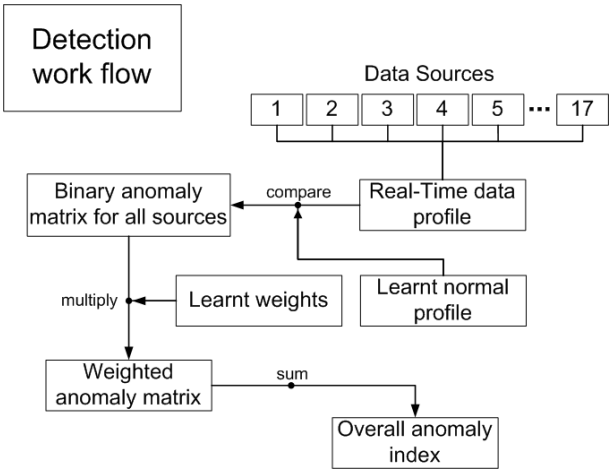

Figure 7. Intrusion detection mechanism

When the learnt weights scheme is applied to detected characteristic anomalies in several attack and mechanical failure experiments, the anomalies that have higher weight are accentuated and the importance of anomalies that tend to occur during a non-attack mission scenario are reduced. This reduces the anomaly score of the system in a non-attack scenario and ensures the score increases when abnormal circumstances occur on characteristics that should not otherwise change during non-attack experiments.

Figure 8. Matrices of anomalies spotted for each of the incidents in the experiments (compass manipulation, rogue node, replay packet injection, wheel failure)

In Figure 8, we observe 4 different scenarios when the robotic vehicle testbed is in operation and is under cyber-physical attacks (Replay packet injection and rogue node), a physical-cyber attack (compass manipulation) or during an unexpected mechanical failure. The result in each cell is rounded to the nearest whole number for presentation purposes. The overall anomaly index is derived from a sum of all signature characteristic anomaly results when the weighting scheme is applied. This index is used by the intrusion detection system to reason the behaviour of the robotic testbed vehicle at the level of an overall anomaly index observation, reducing the need to analyse each signature characteristic as they are updated, thereby reducing computational requirements.

5. Experimental scenarios

We have evaluated the performance of the proposed system using a prototype implementation on our autonomous vehicle, against different incidents, which the robot had not previously been subjected to.

5.1. Cyber-physical Attack C1: Replay Packet Injection

In this scenario, data previously collected are replayed with different intensities (1, 2, or 3 packets/s).

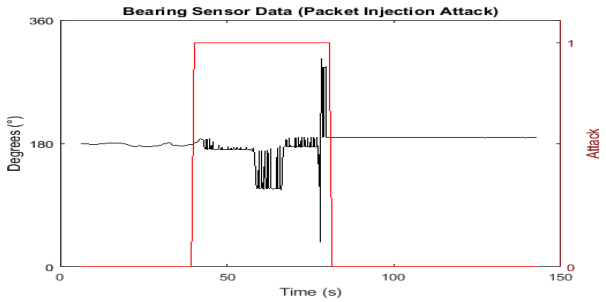

Figure 9. Bearing Sensor Data (Packet Injection)

The attack lasts for 40 s and starts 40 s after the start of the robotic vehicle's mission. Its impact on the senor data bearing is shown in Figure 9.

5.2. Cyber-physical Attack C2: Rogue Node

Here, a rogue node is integrated in the system and starts replaying packets within the internal communication network. The attack has variable intensities and uses amplification of the packet rate, where it replays each packet that is captured multiple times. Each attack is repeated twice using various amplifications. The experiment includes two iterations of 25 s of normal operation and 25 s of attack, as shown in Figure 10.

The particular attack achieves an impact similar in nature to that of a denial of service attack on a conventional computer system [14], causing rapid increase of network utilisation.

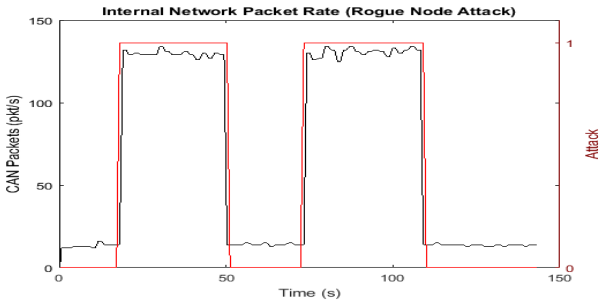

Figure 10. Internal Network Packet Rate (Rogue Node)

5.3. Physical-cyber Attack P1: Compass manipulation

In this experiment, a magnet is placed close to the compass sensor while it is in operation. After 40 s of normal operation, the magnet is applied. It is removed after a further 40 s.

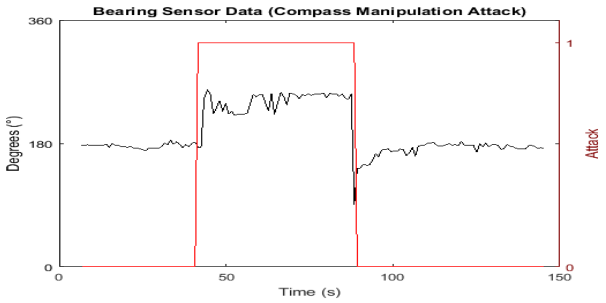

Figure 11. Bearing Sensor Data (Compass Manipulation)

The impact on the bearing reported can be seen in Figure 11. Such a change affects dramatically the vehicle's orientation ability.

5.4. Normal Failure F1: Broken wheel

This experiment occurred unexpectedly, as based on the learnt knowledge there was a high number of false-positive alarms during an experiment. Further investigation showed that a mechanical fault had occurred, as the mecanum wheel holder screws had worked loose due to the vibrations caused by the previous experiments. These runs have been included in our analysis to explain the methodology and also to demonstrate the ability of this model not only to identify an attack, but also to identify mechanical faults as well. The fact that our signature method revealed this unexpected issue, validates the sensor-agnostic approach.

6. Performance evaluation

Figures 8, 9, 10 and 11 show example experimental runs for different cyber and physical incidents and their impact on the cyber and physical data sources used by the detection

mechanism. Green corresponds to a normal state with no noticeable impact from an attack. The color shift to red indicates a higher anomaly occurrence detection in the set of attack scenarios as well as during the accidental mechanical failure. For evaluation of the detection performance, we have used receiver operating characteristics (ROC) curves and measured their area under curve (AUC) (Figure 14). AUC values near 0.5 indicate random detection, while near 1 indicate completely correct detection. Table 3 shows the AUC values measured in our experiments for compass manipulation (CM), packet injection (PI) and rogue node (RN) attacks, for both the simple approach with the signature and the extended with the application of the learned weights. The latter appears to be particularly useful for the case of the more challenging attack to detect, which is the rogue node. In that case, the AUC increased from 0.406 to 0.875.

TABLE 3. OVERALL PERFORMANCE BASED ON ROC AREA UNDER CURVE FOR COMPASS MANIPULATION, PACKET INJECTION AND ROGUE NODE ATTACKS

	CM	PI	RN
Signature + Weights	1.0	1.0	0.875
Signature	0.938	1.0	0.406

Detection accuracy for the compass manipulation attack is shown in Figure 12, demonstrating that the impact of this attack was high and affected multiple data sources producing a high score that exceeded the allowed threshold score drastically.

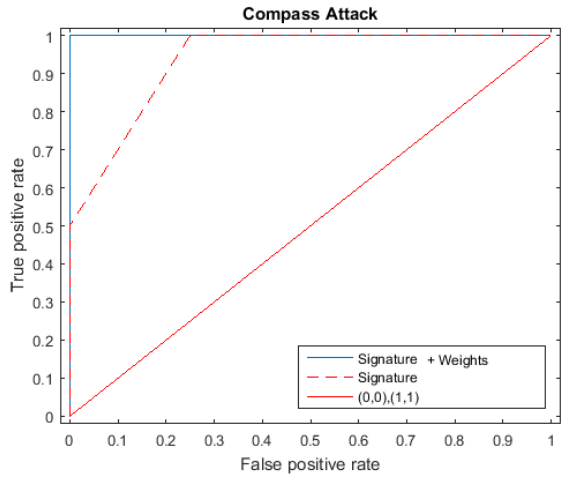

Figure 12. Compass Attack ROC Performance

The performance of the system for the packet injection attack yielded an AUC score of 1 (Figure 13). In this experiment the physical impact on the vehicle while low as external communication was under attack, and the vehicle relied only on the internal network communication data to detect it. The high score was achieved due to large deviations on multiple data sources, to such an extent that the level of false-positive detections was impossible to achieve.

Using the sensor-agnostic approach an issue can arise when deviations are detected in the data source without

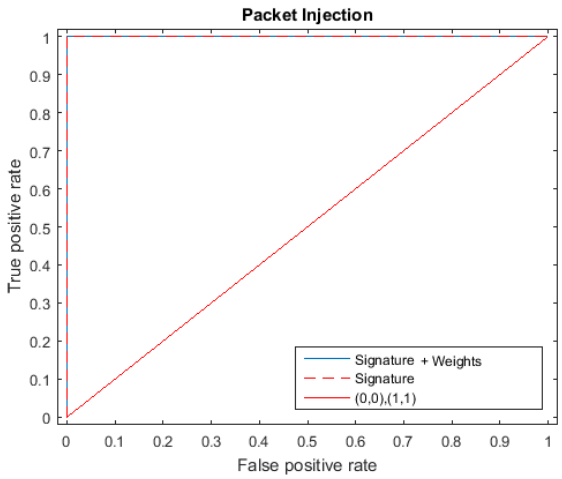

Figure 13. Packet Injection Attack ROC Performance

the impact on a robotic vehicle testbed, increasing the false-negative detection rate. Introducing weights in to the detection technique, improved the detection rate for the rogue node attack as shown in Figure 14. This technique emphasises the anomalies that are not frequently seen in non-attack scenarios.

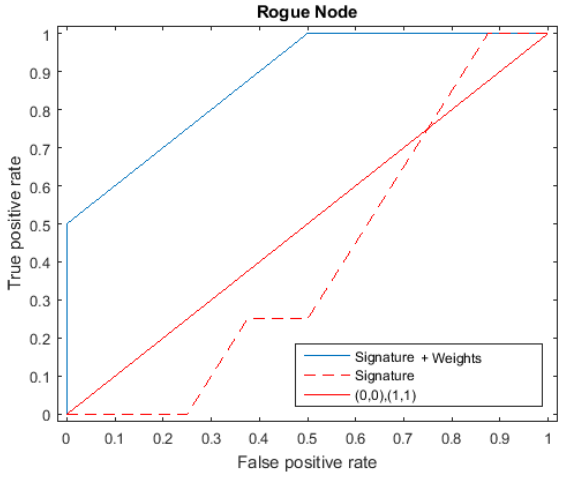

Figure 14. Rogue Node Attack ROC Performance

7. Conclusion

We have presented a sensor-agnostic methodology that is demonstrably able to run onboard a resource-constrained autonomous vehicle and detect in real-time attacks, which it has not been exposed to before. We achieve this by learning the normal ranges for a wide variety of cyber and physical data sources and applying weights to fine tune their importance for detecting anomalous behavioural profiles. The approach has shown to be promising for the variety of different (unknown to the vehicle) attacks that we experimented with, including an unplanned physical failure.

Having shown the feasibility of the approach, our future work will include a wider variety of attack scenarios during missions of different types of autonomous actuation and navigation. We will focus especially on cases where the proposed methodology's detection rate is expected to drop. An example of this would be using a packet injection attack. The current high detection rate is achieved as, in the experiments performed, deviations were high from multiple data sources. However, in the case where an attacker would replay an attack with the data collected during the same ongoing mission, the deviations would be lower and that would reduce the overall anomaly index score as only certain cyber features would be affected. Also if one data source starts acting abnormally and affects other data sources a situation could arise where they will cancel each other out, producing fewer anomalies. Such scenarios will help us acquire a clearer picture of the limits of the approach in challenging conditions.

Acknowledgments

This research has been funded and supported by the Defence Science and Technology Laboratory. We thank Robert Sayers for his invaluable assistance and feedback.

References

[1] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 447–462.

[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Usenix Security Symposium*, 2011.

[3] G. Loukas, *Cyber-Physical Attacks: A Growing Invisible Threat*. Butterworth-Heinemann, 2015.

[4] R. Mitchell and R. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," *Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 5, pp. 593–604, 2014.

[5] R. Mitchell and I.-R. Chen, "Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems," *Dependable and Secure Computing, IEEE Transactions on*, vol. 12, no. 1, pp. 16–30, 2015.

[6] T. Vuong, A. Filippoupolitis, G. Loukas, and D. Gan, "Physical indicators of cyber attacks against a rescue robot," in *IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2014, pp. 338–343.

[7] T. P. Vuong, G. Loukas, D. Gan, and A. Bezemskij, "Decision tree-based detection of denial of service and command injection attacks on robotic vehicles," in *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*. IEEE, 2015, pp. 1–6.

[8] T. P. Vuong, G. Loukas, and D. Gan, "Performance evaluation of cyber-physical intrusion detection on a robotic vehicle," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2106–2113.

[9] A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi, "Consensus-based distributed intrusion detection for multi-robot systems," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 120–127.

[10] R. Mitchell and I.-R. Chen, "Adaptive Intrusion Detection of Malicious Unmanned Air Vehicles Using Behavior Rule Specifications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, p. 1, 2013.

[11] M. Strohmeier, V. Lenders, and I. Martinovic, "Intrusion detection for airborne communication using phy-layer information," in *12th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*. Springer, 2015.

[12] J. Warner and R. Johnston, "GPS spoofing countermeasures," *Homeland Security Journal*, vol. 25, no. 2, pp. 19–27, 2003.

[13] A. Bezemskij, R. J. Anthony, D. Gan, and G. Loukas, "Threat evaluation based on automatic sensor signal characterisation and anomaly detection," in *Proceedings of The Twelfth International Conference on Autonomic and Autonomous Systems*. IARIA, 2016, pp. 1–7.

[14] G. Loukas and G. Oke, "Likelihood ratios and recurrent random neural networks in detection of denial of service attacks," in *Proceedings of International Symposium of Computer and Telecommunication Systems (SPECTS)*. SCS, 2007.