# Decision Tree-based Detection of Denial of Service and Command Injection attacks on Robotic Vehicles

Tuan Phan Vuong, George Loukas, Diane Gan and Anatolij Bezemskij
Department of Computing and Information Systems
University of Greenwich
London, UK
Email: (t.p.vuong, g.loukas, d.gan, a.bezemskij)@gre.ac.uk

***Abstract*—Mobile cyber-physical systems, such as automobiles, drones and robotic vehicles, are gradually becoming attractive targets for cyber attacks. This is a challenge because intrusion detection systems built for conventional computer systems tend to be unsuitable. They can be too demanding for resource-restricted cyber-physical systems or too inaccurate due to the lack of real-world data on actual attack behaviours. Here, we focus on the security of a small remote-controlled robotic vehicle. Having observed that certain types of cyber attacks against it exhibit physical impact, we have developed an intrusion detection system that takes into account not only cyber input features, such as network traffic and disk data, but also physical input features, such as speed, physical jittering and power consumption. As the system is resource-restricted, we have opted for a decision tree-based approach for generating simple detection rules, which we evaluate against denial of service and command injection attacks. We observe that the addition of physical input features can markedly reduce the false positive rate and increase the overall accuracy of the detection.**

***Index Terms*—Cyber-physical systems; Mobile robots; Intrusion detection; Decision tree; Cyber-physical attack; Network security; Denial of service (DoS); Command injection**

## I. Introduction

While physical damage has been traditionally caused by physical means and cyber damage by cyber means, our increasing dependence on highly automated and networked systems, from industrial control to robotic vehicles, has generated formidable cyber-physical vulnerabilities. The operation of robotic vehicles often depends heavily on computer networks. A cyber-attack against or through an associated network affects the movement of a vehicle in a manner that gives rise to a range of new security challenges. Such incidents have already been reported to have occurred both in the wild [1], [2] and in controlled experimental environments [3], [4] and competitions [5].

Cyber-physical attacks are defined as security breaches in cyberspace that adversely affect physical space [1]. Here, we focus on denial of service attacks [6] that can affect robotic vehicles by delaying or preventing commands from reaching their movement control system, as well as command injection attacks that provide the robot with conflicting control commands. Our aim is to provide mechanisms for such vehicles to rapidly and accurately detect that they are under attack.

The vast majority of cyber-physical security research has focused on industrial control systems. Such systems are inherently predictable. They perform routine operations and any deviation of the network traffic from an expected behaviour can be flagged as suspicious. We argue that the uncertain environment in which robotic vehicles operate makes them less predictable. This is more so for vehicles that are not autonomous but controlled by a user over a communication channel. In previous research, we have identified that, depending on implementation approach and attack type, a robot under a denial of service attack on its communication channel might be forced to shut down, to continue moving blindly, to jitter, or to delay changing direction [7], [8] etc. Here, we try to make use of these physical manifestations of two practically usable cyber attacks [9]: denial of service and command injection to investigate whether we can use them meaningfully as part of an on-board intrusion detection system. We refer to these as physical input features. As cyber input features we refer to the ones traditionally used in intrusion detection, such as inbound and outbound network rates, CPU usage, disk activity etc. Following the terminology proposed by Mitchell and Chen [10] for intrusion detection of cyber-physical systems, we evaluate the combined use of cyber and physical input features on a knowledge-based machine learning algorithm.

## II. Literature Review

Cyber security in vehicles is a relatively new area of study. Relevant research has focused primarily on proof-of-concept attacks [3] on the integrity of sensing and actuation or the availability of communications. Defence is usually about resilience through redundancy [11] or prevention through authentication and encrypted communication [12]. Here, we place our attention instead on the few examples of research on intrusion detection techniques designed specifically for mobile cyber physical and robotic systems (Table I).

Depending on its architecture and application, a robotic vehicle may be able to benefit from communication with other agents (multi-agent) or may need to rely solely on its own sensing capabilities and monitoring processes (single-agent).

This work was supported by the the UoG VC's PhD grant "Cyber-physical security of robotic vehicles"

TABLE I
INTRUSION DETECTION APPROACHES FOR ROBOTIC AND MOBILE CYBERPHYSICAL SYSTEMS

| Ref. | Type | Comms | Location | Attack Types | Input Features | Detection approach |
|---|---|---|---|---|---|---|
| Mitchell, Chen [13], [14], [15] | *Mobile CPS* | *Wireless* | *Host Based, Network Based* | *Bad Command Injection, Node Hijack* | *Position, Battery Exhaustion Rate Nodes Compromised* | *Dynamic IDS Voting, Positional Discontinuity, Enviroconsistency* |
| Fagiolini et al. [16], [17] | *Multi-Robot System* | *Wireless* | *Host Based, Decentralized* | *Misbehaviour* | *Node Reputation, Behaviour score, Distance Estimation* | *Clustered Monitoring, Voting* |
| Bonaci et al. [18] | *Robotic Surgery System* | *Wired* | *Host Based, Network Based* | *Intent Modification, Control Hijack* | *Motor Performance, Network Performance* | *Recommendations for Network Monitoring* |
| Shetty et al. [19] | *Multi-Robot System* | *Wireless* | *Host Based, Network Based Decentralized* | *Denial Of Service* | *Lack of Connectivity* | *Network Monitoring* |
| Vuong et al. [7] | *Remote-controlled Robot* | *Wired* | *Host Based* | *Denial Of Service* | *Motor Perfomance, Network Peformance* | *Rule-based* |
| Zeng et al. [20]<br>Fagiolini et al. [21]<br>Bicchi et al. [22] | *Multi-Robot System* | *Wireless* | *Host Based, Role Based Network Based Decentralized* | *Node Failure, Node Misbehaviour* | *Network Performance, Behaviour Score, Node Reputation, Neighbour State, Neighbour Actions, System Configuration, Agent Position* | *Reputation Based, Consensus Based, Set-Valued Consensus* |

Multi-agent approaches focus on coordination between different agents (e.g, between different driverless vehicles or robots [19]) in their effort to detect one agent's suspicious actions, reports, configuration or location. The detection criteria may include consistency with the laws of physics (e.g., velocity measurements that are physically possible, or location that is consistent with the velocity measured [1]), consistency with the sensor measurements reported by neighbouring agents [13], voting [23], reputation scores etc.

Most of these approaches and detection criteria become impractical in single-agent systems, such as the single remote-controlled robotic system used here. Without the opportunity to coordinate with multiple other robots, the focus has to shift to the identification of relevant characteristics that can be measured by its own on-board systems.

## III. TESTBED

The robotic vehicle that we use as our testbed is illustrated in Figure 1. It is a four-wheel-drive robot controlled via an on-board Intel Atom computer running the Linux operating system. An Arduino micro-controller is responsible for driving the robots motors. The robot is also equipped with a pan/tilt camera for remote navigation and situational awareness. Remote control of the robot can be via Ethernet cable or Wi-Fi, by relaying commands received over a TCP socket to the robots control board. The two rear wheel motors are fitted with magnetic encoders, which provide information on the angular position of each wheel. A detailed schematic of the components of the robot is shown in Figure 2, which also highlights the cyber and physical input features that are collected from different components for the purposes of intrusion detection.

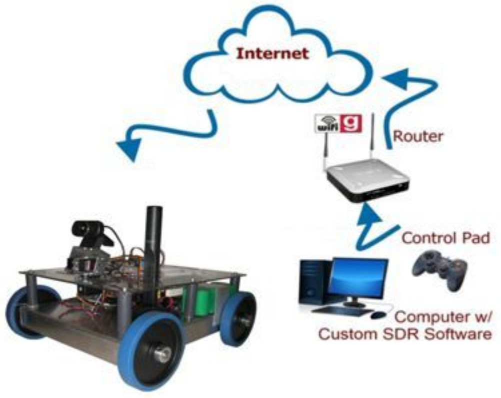


Fig. 1. The testbed system

## IV. EXPERIMENTAL SETUP AND SCENARIOS

Our first aim is to establish whether meaningful detection is feasible with physical input features or, more likely, by combining cyber and physical input features. For this reason, we start with simplified conditions and attack scenarios. To reduce the variable effect of friction, the robot is placed on a stand during the experiments. This also helps with repeatability of scenarios that involve movement with different speeds and different duration in the same laboratory environment. For an effective communication between operator and robot as well as the attacker, we utilise both wired and Wi-Fi communications during the experiment. Finally, for consistency across different scenarios, we power the robot via DC power supply rather than its on-board battery pack. The depletion rate of the latter would affect the consistency of the motors' power and consequently of any measurements coming from the encoders.

Following the same logic of simplified conditions, we conduct experiments where the robot can be in normal operation (no attack), under denial of service (DoS) attack, or under command injection attack, as shown in Table II.

TABLE II
THE FIVE SCENARIOS TESTED EXPERIMENTALLY

| Scenario | s1 | s2 | s3 | s4 | s5 |
|---|---|---|---|---|---|
| Description | DoS | Data Injection "STOP" | Data Injection "LEFT" | Stronger DoS from two PCs | No attack. Normal traffic |
| Duration | 307 s | 173 s | 79 s | 29 s | 221 s |

### A. Denial of Service attack (S1, S4)

The robot is in normal operation and under attack, while it is periodically moving and stopping, as shown in scenarios s1 and s4. For this part, we limit the experimentation to setting the robot to move in a straight line and at constant speed and to halt repeatedly. The attack is a simple denial of service attack at a bit rate of approximately 8.7 MBit/s originating from an attacking machine. For a stronger DoS attack, we use a second PC to direct further illegitimate network traffic to the robot. The aim of the attack is to flood the robots network interface with TCP traffic.

### B. Command injection attack (S2, S3)

Under this attack the robot receives commands to move forward from its legitimate operator, as well as "stop" commands from an attacker, as in scenario s2, or "turn left" commands from an attacker, as in scenario s3.

### C. Normal operation (S5)

Here, there is no attack. The only network traffic involved has to do with transmitting camera feed to the operator and receiving the operator's legitimate commands. This is shown in scenario 5.

## V. INTRUSION DETECTION USING DECISION TREES

In our previous work [7], we have identified the need for a robust classification method for behavioural characteristics of a robotic vehicle under attack using both physical and cyber input features. Towards this goal, we start our investigation with a knowledge-based approach, which depends on the existence of a known attack pattern. As an example of such an approach, we use a decision tree-based algorithm, as in [24]. Decision Tree machine learning is a common method for classifying data with high speed, strong learning ability and simple construction [25]. We have selected the Decision Tree C5.0 to define the rule set for detecting the physical impact as well as the cyber attack on the robotic vehicle testbed. Before applying our C5.0 detection mechanism, we start with a data collection phase (Section V-A).

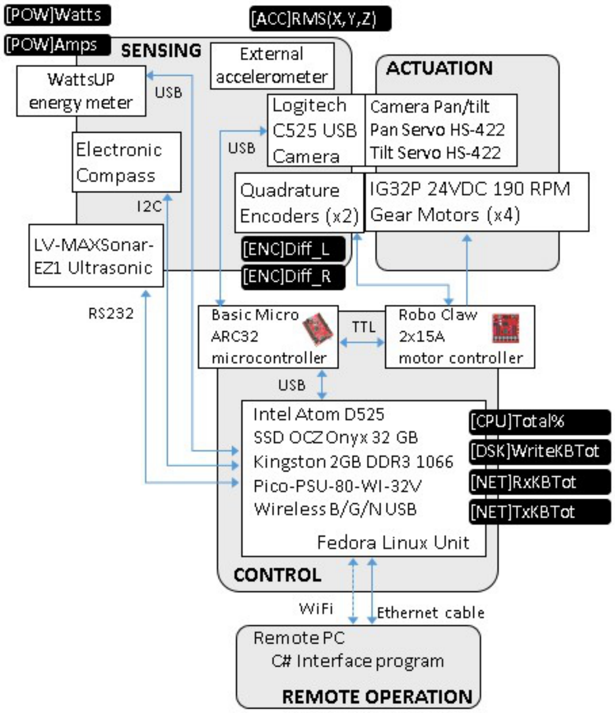


Fig. 2. Detailed diagram of the testbed. The input features used for the detection are shown in black background

### A. Data collection and preparation

We have collected data for eight input features. Four of these are for communication and processing, which we refer to as the cyber input features, and four are for the physical properties of the robot, which we refer to as the physical input features. Each feature is designed to log its value at a specific time within a cycle (period of time T). Table III summarises these. In short, [NET]RxKBTot and [NET]TxKBTot correspond to the received and transmitted network traffic rates; [CPU]Totl% is the CPU utilisation; [DSK]WriteKBTot is the rate that files are written to the disk; DiffLEncoder represents the wheel speed; RMS of (x,y,z) represents the vibration of the chassis (using accelerometer measurements); Watts corresponds to power consumption; and Amps corresponds to current.

The data collection occurs at several different locations within the architecture, as shown in Fig. 2. For example, the encoder value is collected by monitoring scripts embedded within the robot control unit, while Watts and Amps are measured with the WattsUp device [26]. As a result, we have had to process the data offline to address the synchronisation difference between the clocks of these different data collection devices, and to use linear interpolation for devices that would collect data at different time intervals to each other (see Table III - column "Period"). Figures 3 (cyber) and 4 (physical) show a representative run of the experiments using the data after clock synchronisation and interpolation in R.

TABLE III
COLLECTED CYBER AND PHYSICAL FEATURES WITH TIME CYCLE

| | Type | Feature name | Period (T) |
|---|---|---|---|
| Cyber | Network | [NET]RxKBTot | 1.0 s |
| | Network | [NET]TxKBTot | 1.0 s |
| | CPU Data | [CPU]Totl% | 1.0 s |
| | Disk Data | [DSK]WriteKBTot | 1.0 s |
| Physical | Encoder | DiffEncoderL | 30 ms |
| | Accelerometer | RMS of (x,y,z) | 20 ms |
| | Power | Watts | 1.0 s |
| | Current | Amps | 1.0 s |

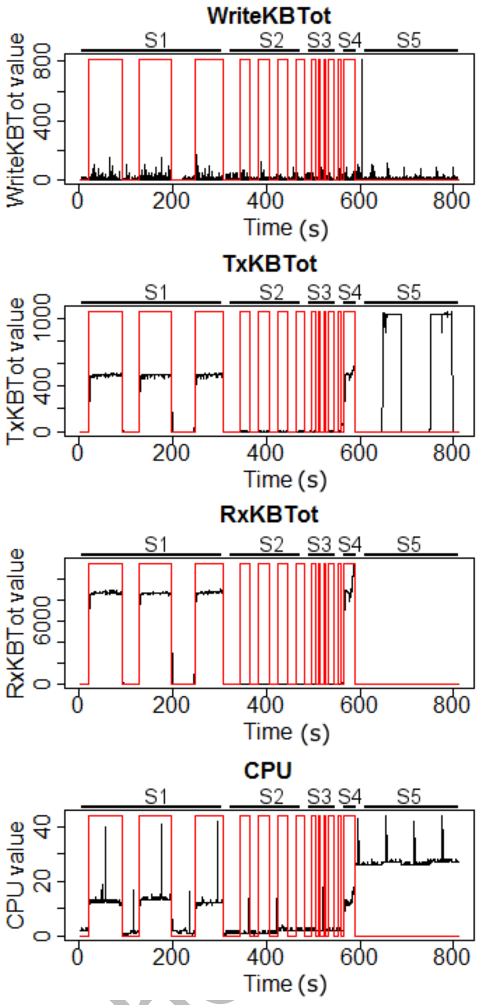


Fig. 3. The data for cyber features collected during the five scenarios (denoted as S1 - S5, and presented one after the other). The overlaid frames denote the periods of time that a cyber attack (denial of service or command injection) is on. Note that there is no attack in S5.

## B. Detection mechanism

Our goal here is (i) to provide a mechanism that can detect a cyber attack against a robotic vehicle and (ii) to explore whether the addition of physical input features can improve its effectiveness. As a representative approach, we used a decision tree learning algorithm for automatically producing detection rules that will be used by the robotic vehicle.

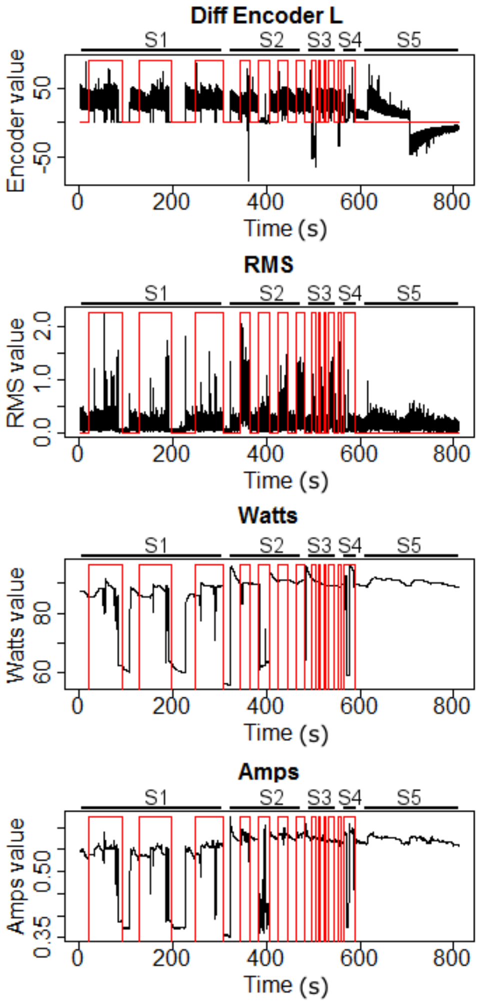


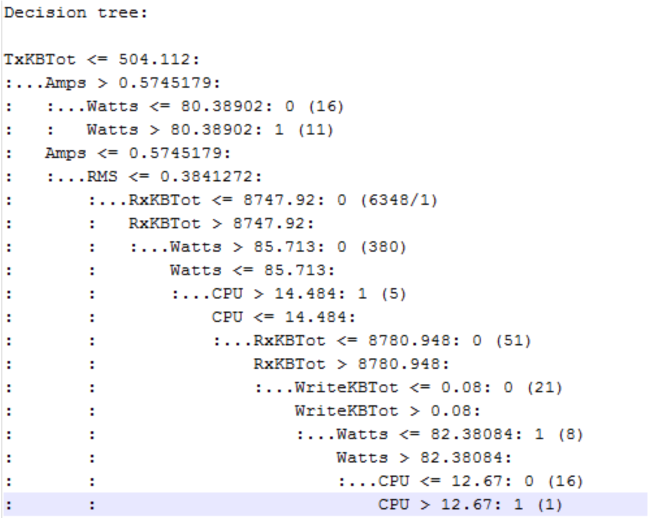
Fig. 4. The data for physical features collected during the five scenarios (denoted as S1 - S5, and presented one after the other). The overlaid frames denote the periods of time that a cyber attack (denial of service or command injection) is on. Note that there is no attack in S5.

*1) Method:* The Decision Tree C5.0 was chosen as the updated version of C4.5, which is improved in terms of speed, memory and efficiency. We used the C5.0 package [27] in R to execute this algorithm. The dataset collected in our experiments had 40,454 data points for each feature. We randomly chose 30% of the data points of the initial attack scenarios (s1 for denial of service, and s2 for command injection) for training and the rest (70% of S1, 70% of S2, and the whole of S3, S4, S5) for testing. In the training datasets, we add the learning flag of the algorithm, which is true when the "ground truth" is that there is an attack and false when there is no attack.

*2) Feature selection:* In line with the second of our goals, we allow our learning algorithm to use all or subsets of our available input features:

- Set 1: All eight cyber and physical features
- Set 2: The four cyber features only
- Set 3: The four physical features only

*3) Decision Tree model:* With each set of features, we fit a Decision Tree model using Quinlan's C5.0 algorithm on the training data. A sample of the tree is depicted in Figure 5. For a given testing data, the attribute usage shows the contribution of features in percentage within the rule-set. Table IV shows the different usages amongst cyber and physical input features for the three different sets above.



```
Decision tree:

TxKBTot <= 504.112:
:...Amps > 0.5745179:
:   :...Watts <= 80.38902: 0 (16)
:   :   Watts > 80.38902: 1 (11)
:   Amps <= 0.5745179:
:   :...RMS <= 0.3841272:
:       :...RxKBTot <= 8747.92: 0 (6348/1)
:       :   RxKBTot > 8747.92:
:       :   :...Watts > 85.713: 0 (380)
:       :       Watts <= 85.713:
:       :       :...CPU > 14.484: 1 (5)
:       :           CPU <= 14.484:
:       :           :...RxKBTot <= 8780.948: 0 (51)
:       :               RxKBTot > 8780.948:
:       :               :...WriteKBTot <= 0.08: 0 (21)
:       :                   WriteKBTot > 0.08:
:       :                   :...Watts <= 82.38084: 1 (8)
:       :                       Watts > 82.38084:
:       :                       :...CPU <= 12.67: 0 (16)
:       :                           CPU > 12.67: 1 (1)
```

Fig. 5. A section of Decision Tree rules generated

## VI. Evaluation

### A. Detection accuracy

After creating the Decision Tree model based on the training data, we evaluate the model by measuring the accuracy of the generated set of rules on the test data. The evaluation relates to the rate of false positives (FPR) and false negatives (FNR) returned with regards to the "ground truth" (as shown by the overlaid frames in Figures 3 and 4, as well as the overall accuracy rate (ACC) of the specific decision tree. In addition, we use Receiver Operating Characteristic (ROC) curves, plotting the true positive rate (TPR) against the FPR for different thresholds as in Figure 6. In an ideal detection result, the curve should have a point (0,1) where the FPR is 0% and TPR is 100%, while the line between (0,0) and (1,1) corresponds to random guess. The metric that represents the quality of detection of the decision based approach is the area under curve (AUC) of the ROC curves.

### B. Results

As mentioned in Section V-B2, the experiment consisted of building three decision trees with different sets of features for denial of service and command injection attack detection.

As shown in Table IV, detection based on physical features only is rather poor (albeit better than random guess). Nevertheless, including physical features together with cyber features provides considerably better results than using only cyber features with regards to ACC and FPR, but is worse in terms of FNR. In terms of the features utilised the most by the decision tree approach, these are the network incoming traffic rate ([NET]RxKBTot) and the energy-related ones (Amps and Watts). While increased vibration of the chassis is visually observed during some of these attacks, this feature is utilised less by the decision tree algorithm. If we remove the physical features, then the algorithm relies almost exclusively on [NET]RxKBTot and the CPU utilisation.

TABLE IV
DoS and command injection detection results

| Features | Attribute usage | FP% | FN% | ACC% |
|---|---|---|---|---|
| All Cyber & Physical (8) | 100.00% RxKBTot<br>99.63% Amps<br>61.93% Watts<br>17.53% TxKBTot<br>5.15% CPU<br>3.03% WriteKBTot<br>0.88% DiffEncoderL | 4.56 | 8.76 | 93.81 |
| Cyber features only (4) | 100.00% RxKBTot<br>60.93% CPU<br>5.31% WriteKBTot<br>0.50% TxKBTot | 25.91 | 3.45 | 82.81 |
| Physical features only (4) | 100.00% Watts<br>85.57% Amps<br>85.26% DiffEncoderL<br>71.50% RMS | 50.21 | 12.59 | 64.40 |

The benefits of adding the physical input features are more clearly seen in Figure 6. Using the ROCR package in R [28], we compare the performance for the three sets of features: cyber and physical; cyber only; and physical only. The three ROC curves in Figure 6 show different TPR, and FNR for different probability thresholds. As can be observed, the area under the curve (AUC) is considerable higher when all eight features are utilised (96.79%) than when only the cyber features (82.38%) or only the physical features are utilised (69.31%). Figure 7 summarises the results for FPR, FNR, ACC and AUC in a single bar chart.

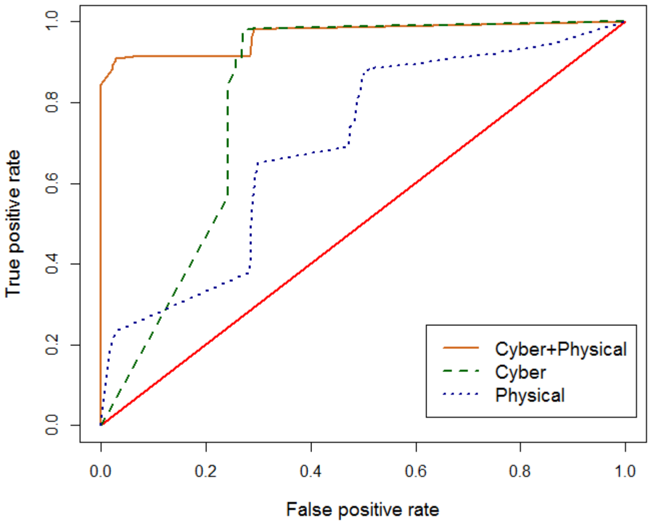


Fig. 6. The ROC curves of the detection rules for the three sets of features: Both cyber and physical; cyber only; and physical only. The (0.0) to (1.1) line is the random guess line.

We consider these results to be promising. As expected, cyber features play the most dominant role in the detection rules identified by the decision tree algorithm. With different training data sets, the decision tree algorithm may provide different rulesets, but it is important to note that the inclusion of physical features has proven beneficial in all experiments conducted regardless of the particular choice of ruleset.

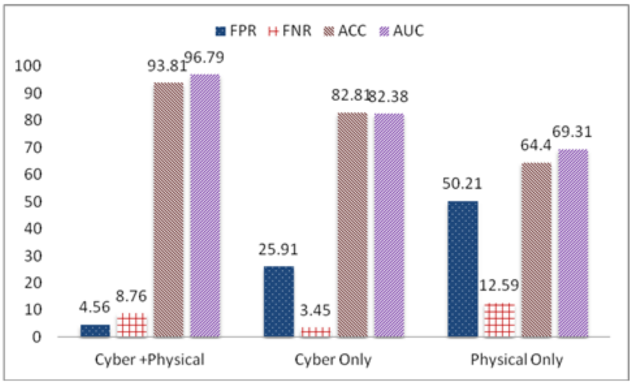


Fig. 7. FPR, FNR, Accuracy and Area Under Curve (AUC) for the three sets of features: Both cyber and physical; cyber only; and physical only.

## VII. Conclusion and Future Work

Intrusion detection for cyber-physical systems is a relatively new area of research. It has been explored to some extent for industrial control systems, but remains at an early stage for mobile cyber-physical systems such as robotic vehicles. Through experimentation, we have observed that different attacks have different impacts on the operation of the robot, and this includes both its cyber (network, CPU, disk data) and physical behaviour (speed, vibration, power consumption). This presents an opportunity, as by leveraging both types of features, one can improve the accuracy of the detection of an attack. We have validated this for an intrusion detection approach based on decision trees and the C5.0 algorithm.

Our next step is to extend the scope of our experiments by applying the same approach on different attack types [9], such as malware infection and communication jamming. We are also working towards testing the hypothesis that the addition of physical input features can improve the accuracy and detection latency of behaviour-based detection approaches, which are more suitable than knowledge-based approaches for zero-day cyber-physical attacks.

## References

[1] G. Loukas, *Cyber-Physical Attacks: A Growing Invisible Threat*. Butterworth-Heinemann (Elsevier), 2015.

[2] S. J. Templeton, "Security aspects of cyber-physical device safety in assistive environments," in *4th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, pp. 53:1–53:8.

[3] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *Security and Privacy (SP), 2010 IEEE Symposium on*, pp. 447–462, IEEE, 2010.

[4] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, pp. 617–636, 2014.

[5] J. Griffiths, "Zhejiang University team scoops 10,600 Yuan for hacking into Tesla Model S," July 17, 2014. South China Morning Post.

[6] G. Loukas and G. Öke, "Protection against denial of service attacks: A survey," *The Computer Journal*, pp. 1020–1037, 2010.

[7] 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*, pp. 338–343, IEEE, 2014.

[8] T. Vuong, G. Loukas, and D. Gan, "Performance evaluation of cyber-physical intrusion detection on a robotic vehicle," in *Proceedings of 13th International Conference on Pervasive Intelligence and Computing (IEEE-PICOM 2015)*, IEEE, 2015.

[9] A. Lang, J. Dittmann, S. Kiltz, and T. Hoppe, "Future perspectives: The car and its ip-address–a potential safety and security risk assessment," in *Computer Safety, Reliability, and Security*, pp. 40–53, Springer, 2007.

[10] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys (CSUR)*, vol. 46, p. bxp078, 2014.

[11] A. Deshpande, O. Obi, E. Stipidis, and P. Charchalakis, "Integrated vetronics survivability: Architectural design and framework study for vetronics survivability strategies," *Computer Standards and Interfaces*, vol. 39, pp. 1–11, 2015.

[12] M. Wolf, A. Weimerskirch, and C. Paar, "Secure in-vehicle communication," in *Embedded Security in Cars*, pp. 95–109, Springer, 2006.

[13] R. Mitchell and I.-R. Chen, "A hierarchical performance model for intrusion detection in cyber-physical systems," in *Wireless Communications and Networking Conference*, pp. 2095–2100, IEEE, 2011.

[14] R. Mitchell and I.-R. Chen, "Effect of intrusion detection and response on reliability of cyber physical systems," *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 199–210, 2013.

[15] R. Mitchell and I.-R. Chen, "On survivability of mobile cyber physical systems with intrusion detection," *Wireless personal communications*, vol. 68, no. 4, pp. 1377–1391, 2013.

[16] 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*, pp. 120–127, IEEE, 2008.

[17] A. Fagiolini, F. Babboni, and A. Bicchi, "Dynamic distributed intrusion detection for secure multi-robot systems," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 2723–2728, IEEE, 2009.

[18] T. Bonaci, J. Herron, T. Yusuf, J. Yan, T. Kohno, and H. J. Chizeck, "To make a robot secure: An experimental analysis of cyber security threats against teleoperated surgical robots," *arXiv preprint arXiv:1504.04339*, 2015.

[19] S. Shetty, T. Adedokun, and L.-H. Keel, "Cyberphyseclab: A testbed for modeling, detecting and responding to security attacks on cyber physical systems," 2014.

[20] W. Zeng and M. Chow, "A reputation-based secure distributed control methodology in d-ncs," *IEEE Transactions on Industrial Electronics*, vol. 61(11), 2014.

[21] A. Fagiolini, G. Dini, and A. Bicchi, "Distributed intrusion detection for the security of industrial cooperative robotic systems," in *World Congress*, vol. 19, pp. 7610–7615, 2014.

[22] A. Bicchi, A. Fagiolini, G. Dini, and I. M. Savino, "Tolerating, malicious monitors in detecting misbehaving robots," in *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, pp. 109–114, IEEE, 2008.

[23] 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.

[24] A. Filippoupolitis, G. Loukas, and S. Kapetanakis, "Towards real-time profiling of human attackers and bot detection," in *Proceedings of 7th International Conference on Cybercrime Forensics Education and Training (CFET)*, 2014.

[25] B. R. Patel and K. K. Rana, "A survey on decision tree algorithm for classification," *International Journal of Engineering Development and Research*, vol. 2, 2014.

[26] Watts up? meters, www.wattsupmeters.com.

[27] M. Kuhn, W. Steve, and N. Coulter, "Package C50," 2014.

[28] T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer, "Rocr: visualizing classifier performance in r," *Bioinformatics*, vol. 21, no. 20, pp. 3940–3941, 2005.