Security Assurance in Modern IoT Systems

Nicola Bena*, Ruslan Bondaruc* and Antongiacomo Polimeno*
*Department of Computer Science
Università degli Studi di Milano
Milan, Italy
Email: {firstname.lastname}@unimi.it, ruslan.bondaruc@studenti.unimi.it

***Abstract*—Modern distributed systems consist of a multi-layer architecture of IoT, edge, and cloud nodes. Together, they are revolutionizing our lives, bringing intelligence to existing processes (e.g., smart grids) and enabling novel, efficient and effective processes (e.g., remote surgery). This transition however does not come without drawbacks, due to the ever-increasing reliance on devices whose security and safety are, at least, questionable. In this context, research is in its infancy, struggling to adapt successful practices applied, for instance, in cloud systems. Security of modern IoT systems still relies on old-fashioned approaches, mostly static assessments considering only very specific parts of the target system, rather than assessing the system as a whole. In this paper, we put forward the idea of security assurance for IoT, as a higher-level assurance process evaluating the target system at different layers and different moments of its lifecycle, then implemented by a flexible assurance framework. The quality of our approach is evaluated in a real-world smart lighting system.**

***Index Terms*—Assurance, Cloud-Edge, IoT, Security**

I. Introduction

Novel networking paradigms such as Network Functions Virtualization (NFV) and 5G, together with novel computing paradigms such as edge computing, are revolutionizing the way IT services are engineered, delivered, and utilized. The confluence of these technologies point to Internet of Things (IoT) systems that no longer consist of "smart things" only, but rather embrace multi-tiered systems where data collected by sensors on the field are analyzed with low latency at the edge, and are the basis for further optimization at the cloud. These systems provide unmatched simplifications to our lives, from the delivery of novel services such as remote healthcare, to the optimization of existing ones, such as smart grid. Altogether, the estimated economic impact of IoT by 2025 is in the range of 2.7 to 6.2 trillion dollars [1].

The price we pay is however not negligible, with increasing risk of safety, security, and privacy, due to the pervasiveness of these systems as well as the interplay with their physical counterparts. Continuous assessment of the behavior of such systems becomes therefore fundamental. Security assurance is one of the preferred solutions in this regards, providing processes and corresponding frameworks evaluating the compliance of IT systems with respect to user-defined requirements. Security assurance has been consistently applied in service and cloud-based systems [2], [3], [4], yet it is in its infancy when it comes to IoT systems. Static and punctual assessments have been the preferred approach, but no longer are a viable option [5]. Some preliminary solutions have been defined [5], [6], [7], [8], but alternatives are still lacking. In general, they consider low-level details and traditional CIA properties (*Confidentiality*, *Integrity*, *Availability*) only, or provide a system overview which is too abstract to be really useful.

In this paper, we build on the security assurance framework in [5], putting forward the idea of IoT assurance driven by a *holistic view* of the target IoT system. Our approach defines and executes assurance evaluations matching and exploiting the architecture of the target system, distributing and aggregating computation accordingly. It considers the whole system lifecycle, and produces results at different granularity according to punctual as well as novel artificial intelligence-based evaluations. This approach enables the assessment of the system as a whole still considering its peculiarities, with negligible interference with the system working.

The contribution of this paper is twofold. We first define our novel approach for IoT assurance driven by artificial intelligence, focusing both on the methodological and on the technological point of view. We then present an experimental evaluation thereof in a real-world IoT system.

The remainder of this paper is organized as follows. Section II discusses the state of the art of security assurance. Section III presents our approach for an effective IoT assurance. Section IV introduces a refinement of our previous assurance framework in [5] to support this novel paradigm. Section V includes an initial experimental evaluation of our approach. Finally, Section VI draws our conclusions.

II. Background and Motivations

We consider modern three-tiered IoT systems, where resource-constrained devices collect data and operate on the field; these devices are controlled with low latency by one or more edge nodes, whose computational power vary from medium-sized servers to microcontrollers; the overall business process is then analyzed and optimized at the cloud [9].

Security assurance is one of the means to assess the behavior of such systems to increase their trustworthiness [3]. An as-

Research supported, in parts, by EC H2020 Project CONCORDIA GA 830927, GARR through the Orio Carlini scholarship 2020-2021, project "MIND FoodS HUB (Milano Innovation District Food System Hub): Innovative concept for the eco-intensification of agricultural production and for the promotion of dietary patterns for human health and longevity through the creation in MIND of a digital Food System Hub", cofunded by POR FESR 2014-2020_BANDO Call HUB Ricerca e Innovazione, Regione Lombardia, and Università degli Studi di Milano under the program "Piano sostegno alla ricerca".

surance evaluation can be described at two levels: *i) assurance process*, that is, the (formal) description of the assurance evaluation; *ii) assurance framework*, that is, the implementation of the assurance process [10]. In particular, assurance processes for IoT mainly revolve around vertical aspects. Compliance to CIA requirements is still the focus of the state of the art, such as the assessment frameworks in [6], [11], the assurance framework in [7], the integrity of the software installed on edge nodes [8], the quality of service [12], the absence of vulnerabilities or weaknesses [13]. Risk-based approaches have also been proposed, such as [14], [15]. Finally, *assurance cases* are high-level objectives supported by evidence prepared by the system developers [16]. They have been successfully applied in self-adaptive systems [17], [18], and the research community is investigating their adoption for IoT assurance as well [9], [19], [20]. Assurance frameworks are typically strictly-coupled with the corresponding processes they implement, thus making difficult to reuse the same flexible implementation for different processes [5].

In short, current approaches for security assurance in IoT systems are not generic enough, limiting their evaluations to specific aspects of the target systems. In turn, this hinders the applicability of assurance in the real world. An effective evaluation requires instead higher level processes with less formalisms than in the past, while corresponding assurance frameworks should be flexible enough to accommodate different requirements instantiated on different target systems.

III. Assurance Evaluation

An effective assurance evaluation for IoT must be *i) multi-layer*, *ii) multi-phase*, and *iii) holistic*, as follows.

A. Multi-Layer Evaluation

The three layers composing modern IoT systems have intrinsic differences from the functional and non-functional point of view. For instance, cloud and edge layers often rely on well-known and (partially) trusted providers. Instead, IoT devices are resource-constrained and *opaque*, that is, little is known about their internals. This makes it difficult to define a proper assurance process even at high level, and the corresponding results may be a false negative (e.g., the IoT device does not respond because it is busy, not because it is offline). For these reasons, we put forward the idea of *lightweight IoT assurance* in [5]: our assurance evaluations target IoT devices *indirectly*, and rely on the edge and cloud layers to infer the behavior of the IoT layer. We note that this typically translates into querying some metrics already offered by the edge layer (as discussed in Section V), and the goal is to evaluate the target system as a whole, rather than individual nodes. In addition, we note that traditional and well-known assurance evaluations can still be used against the cloud layer [3].

B. Multi-Phase Evaluation

Edge and cloud layers are increasingly developed using approaches such as DevOps [21], and operated with orchestration platforms such as *Kubernetes*. In this scenario, development artifacts become part of the system artifacts, for instance, continuous delivery pipelines and Kubernetes deployment files (i.e., *manifests*). These development artifacts describe the complete system deployment, and therefore become relevant for assurance, in terms of requirements to verify at run time, and of targets of evaluation. For instance, an assurance evaluation for *availability* can be executed against the manifest looking for High-Availability policies, in addition or in replacement of being executed against the system [22]. This brings crucial advantages when the system cannot be assessed directly (e.g., because of resource constraints), and constitutes a complementary and less-invasive means of evaluation.

C. Holistic Evaluation

The dynamic boundaries and complex interactions of IoT imply that rigid modeling of assurance processes (e.g., in terms of non-functional properties [23]) is no longer viable. Instead, assurance processes must rely more on the *behavioral analysis* of the target system to build an *holistic view* thereof, relying on and adapting, for instance, solutions based on anomaly and intrusion detection. In this context, the literature points to high-accuracy solutions based on artificial intelligence (AI) [24], [25]. However, accuracy is not the only indicator that matters [26]; other aspects to consider include *explainability*, *robustness*, and *edge-readiness*, as detailed in the following.

Explainability refers to understanding the logic behind a prediction of an AI model (*local explainability*), or how the model works in general (*global explainability*) [27], [28], [29]. Explainability is fundamental to increase trust in AI in any domains [28], and is even more important in the assurance domain, where assurance evaluations can result in the release of certificates that, by definition, should be trusted [3]. Here, we require *local interpretability*, which can be achieved either by *i) white-box models*, that is, models explainable by definition, such as decision trees [28]; *ii) open black-box models*, that is, techniques explaining neural networks-based models such as *surrogate white-box models* [28]. Intuitively, solutions based on white-box models should be preferable, as they are also simpler. We note that explainable models [28] based on the paradigm of *attention* [30] are being explored in the context of intrusion detection (e.g., [31]) and could boost IoT as well. We finally note that explainability is related also to requirements such as law compliance, accountability, fairness [28], [29].

Robustness refers to the *resistance* of AI models against data poisoning and adversarial attacks [32]. In fact, models can be attacked at training time by injecting poisoned inputs in the training set, or at inference time by sending specially-crafted inputs. This scenario becomes even more serious when working in an adversarial setting [32]. The assurance scenario is in fact adversarial, for instance in terms of a target system wishing to be certified without supporting the corresponding requirements [33]. Here, we require *robustness to inference-time attacks*, assuming that the training set is under the control of the entity performing the assurance evaluation. Currently, the preferred method to achieve robustness is *adversarial*

training, which augments the training set with adversarial data points [34], [35], [36].

Edge-readiness refers to AI models and corresponding deployments that are ready for an edge scenario. Typically, an IoT system can be divided in different *vertical* subsystems, according to, for instance, the location; each subsystem controlled by one or more edge nodes. Edge nodes are the *first* system component that can be directly targeted by assurance evaluations (Section III-A) and also the *last* where IoT traffic can be seen, it therefore is a crucial point. This leads to a behavioral assurance grounded on *i)* distributed and continuous learning/inference, and *ii)* lightweight models. *Distributed inference and continuous learning* refer to AI models deployed nearby edge nodes, analyzing the corresponding subsystems. Models are then fine-tuned and continuously re-trained according to the subsystems' peculiarities. Behavioral evaluations are then aggregated at a central location in the assurance framework. This paradigm can be easily realized with *ensembles*, that is, individual models whose predictions are combined in an improved model [37]. *Lightweight models* refer to AI models suitable for execution on resource-constrained edge nodes. There exist several techniques in this regards, often based on *compression* [38].

The above three aspects are strictly related one to each other. On the one hand, white-box and simple models are typically more lightweight than black-box models, hence fitting resource-constrained scenarios. In addition, black-box models are typically not necessary to obtain the holistic view [25]. On the other hand, their susceptibility to adversarial attacks vary. For instance, decision tree-based models are still vulnerable to adversarial attacks as any other neural network-based models [39], yet ensembles can improve their robustness [40].

IV. Assurance Framework

We refine the assurance framework in [5] to support the assurance approach in Section III. The framework in Figure 1 is based on Kubernetes, which natively supports edge computing [41] and guarantees the high scalability needed to evaluate large and complex target systems. The framework is implemented as a set of REST API-based microservices written mostly in Go. Its deployment is split between the edge, where collected data are preprocessed, and the cloud, where aggregated results are analyzed and stored, as follows.

Edge components include components operating on the edge layer nearby the target systems. Assurance evaluations are implemented in scripts called *probes*, packaged as containers executed by Kubernetes. The framework ships traditional assurance evaluations (dashed lines) targeting the cloud layer of the target system, and behavioral AI models (double dashed lines) targeting the remaining layers. The result produced by each evaluation is called *evidence* and consists of three parts: *i)* a Boolean value indicating the success or the failure of the evaluation; *ii)* a human-readable text briefly indicating reason behind the result's outcome; *iii)* additional low-level

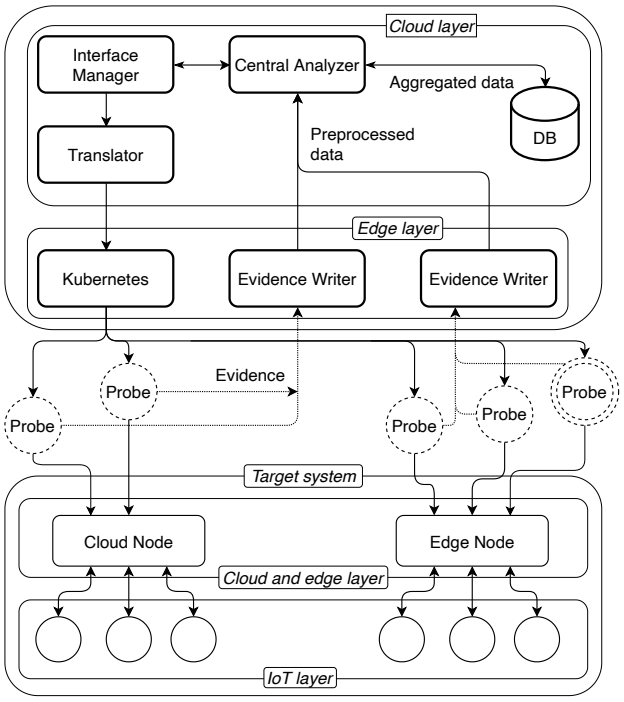

Figure 1. Architecture of the proposed framework.

details. These results are cleaned, aggregated, and stored by the *Evidence Writers* according to the evaluations' peculiarities.

Cloud components include components operating on the cloud layer of the assurance framework. Stored results are processed asynchronously by the *Central Analyzer*, taking the final decision of whether the assurance evaluation is successful or not, and re-orchestrating the evaluation process accordingly. The assurance framework interacts with end users by exposing a REST API service called *Interface Manager*, receiving requests for assurance evaluations and showing the corresponding results. Those requests are sent from *Interface Manager* to the *Translator*, where they are mapped into low-level Kubernetes objects to be executed. In particular, *Translator* encodes and schedules the execution of the probes as `Job` or `CronJob`.

V. Experiments

We evaluated the quality of our approach in a simulated IoT environment closely resembling a real-world smart lighting system for road illumination. Figure 2 shows the architecture of the system, consisting of: *i)* IoT nodes (microcontrollers) directly controlling street lamps; *ii)* edge nodes implemented with the *Mainflux* platform managing IoT nodes by sending and receiving data and commands; *iii)* cloud node where historical data from the below layers are aggregated and visualized in a web frontend (dashboard). Edge and cloud nodes are microservices implemented in Go and deployed as 28 *pods* on a Kubernetes cluster.

Evaluations. We executed 6 probes against the system, as follows: *i) Robust-TLS* checking that mutual TLS authen-

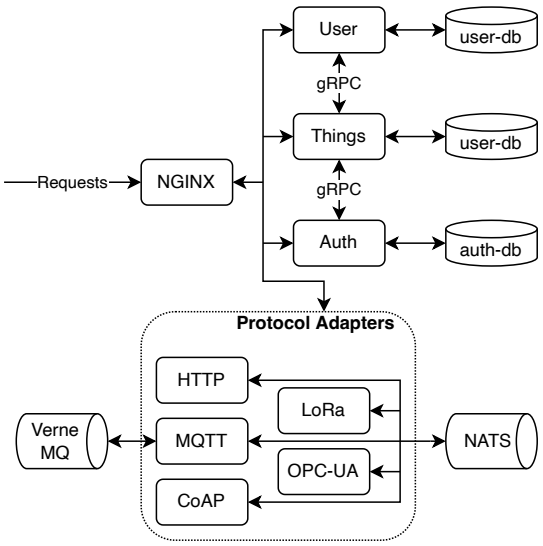

Figure 2. Architecture of the lighting system.

Table I
SUMMARY OF EXPERIMENTAL RESULTS

Probe	Passed	Reason
Robust-TLS	✗	Mutual TLS is configured
Availability	✓	System and its node always up
Manifest-Checker	✗	377/1904 sub-checks failed
Backend-Dependency	✗	22 vulnerabilities found
Frontend-Dependency	✗	20 vulnerabilities found
Image-Scanner	✗	165 vulnerabilities found

Table II
EXCERPT OF RESULTS OF PROBES MANIFEST-CHECKER AND BACKEND-DEPENDENCY

Issue	N. of findings (%)
AppArmor and *Seccomp Any*	100
No CPU and *Memory Limit*	100
No CPU and *Memory Requests*	89

(a) *Manifest-Checker*

Severity	N. of findings
High	3
Medium	19

(b) *Backend-Dependency*

tication is enabled between edge and IoT nodes, to ensure that only authorized IoT devices can connect to the edge nodes; *ii) Availability* checking that edge nodes and, indirectly, IoT nodes are healthy and stable; *iii) Manifest-Checker* checking adherence of Kubernetes manifests to best practices; *iv) Backend-Dependency* and *v) Frontend-Dependency* checking the dependencies of cloud and edge microservices, and of the dashboard respectively, looking for vulnerable libraries; *vi) Image-Scanner* checking the *images* of the microservices, looking for vulnerable software. The probes are executed periodically, with interval varying according to their purpose, from 1 minute for *Availability* to automatic CI/CD triggers for *Manifest-Checker*, *Backend-Dependency*, *Frontend-Dependency*, and *Image-Scanner*. We chose these probes to maximize coverage and diversity. Probes *Robust-TLS* and *Availability* evaluate the target system at different layers, exploiting the edge layer to infer information on the IoT layer (*multi-layer evaluation* in Section III-A). Such probes also cover the *execution* phase of the system lifecycle, while the remaining probes cover the preceding phases, that is, development and deployment (*multi-phase evaluation* in Section III-B). In addition, *Availability* partially covers *holistic evaluation* in Section III-C, despite not using any AI models.

Results. Table I shows a summary of the results, where column "Passed" indicates the Boolean result of the evaluation, and column "Reason" indicates why the evaluation succeeded or failed (Section IV). Altogether, these evaluations show that the security of the lighting system is lacking, with only 1 out of 6 being successful. They also show that there is no much discrepancy between development, deployment, and execution time. In fact, only one execution-time probe successfully concluded (*Availability*), while all development and deployment-time probes concluded with a negative result.

Table II(a) shows an excerpt of the issues found in the manifests by *Manifest-Checker*, in percentage over the number of pods. In particular, *AppArmor Any* and *Seccomp Any* mean that all pods' containers do not have any restrictions to resources and system calls, respectively, therefore having a large attack surface. *No CPU* and *Memory Limit* mean that all pods do not have any upper bounds on the CPU and memory they can use, eventually exhausting the system. *No CPU* and *Memory Requests* mean that most of the pods do not have any lower bounds on the CPU and memory they are assigned with, eventually not being able to complete their computation.

Table II(b) shows the number of vulnerabilities found by *Backend-Dependency*, categorized according to the CWE score (a community-driven list of software and hardware weaknesses). In particular, the probe revealed 3 vulnerabilities with *high* severity, referred to a weak pseudorandom number generator and to a possible incorrect TLS configuration. Although these vulnerabilities might not be a problem in practice, they are still indicators of a weak security posture.

Discussion. From the above results, it emerges that a holistic approach for security assurance in IoT systems is needed. An assessment based on a limited and traditional range of threats does not provide a complete security perspective. A comprehensive assessment must instead encompass the complete system lifecycle, inspecting also development and deployment artifacts. This approach is non-invasive with respect to runtime evaluation, and can uncover issues earlier. For instance, our experiments uncovered several relevant issues without stressing the system execution. In this context, DevSecOps stands out as a cornerstone, to the point that assurance evaluations *can* become a part of a DevSecOps pipeline [22]. At the same time, the system must be evaluated at all layers, possibly exploiting the edge layer to (indirectly) assess IoT nodes without impacting on their resources.

VI. Conclusions

Modern IoT systems promise unmatched benefits to our lives, yet they are perceived as unsafe and untrustworthy. To address this gap and fully unleash their potential, security assurance must be redesigned towards less rigid and static solutions. The approach in this paper provides a first boost in this direction, proposing an assurance evaluation based on multi-layer, multi-phase, and behavioral activities. As a future work, we aim to put more emphasis on the latter, evaluating AI models matching the posed requirements.

References

[1] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs, *Disruptive technologies: Advances that will transform life, business, and the global economy*. McKinsey Global Institute San Francisco, CA, USA, 2013.

[2] M. Anisetti, C. A. Ardagna, and E. Damiani, "A Low-Cost Security Certification Scheme for Evolving Services," in *Proc. of IEEE ICWS 2012*, Honolulu, HI, USA, June 2012.

[3] C. Ardagna, R. Asal, E. Damiani, and Q. Vu, "From Security to Assurance in the Cloud: A Survey," *ACM Computing Surveys*, vol. 48, no. 1, August 2015.

[4] M. Anisetti, C. A. Ardagna, F. Gaudenzi, and E. Damiani, "A Certification Framework for Cloud-Based Services," in *Proc. of ACM SAC 2016*, Pisa, Italy, April 2016.

[5] M. Anisetti, C. A. Ardagna, N. Bena, and R. Bondaruc, "Towards an Assurance Framework for Edge and IoT Systems," in *Proc. of IEEE EDGE 2021*, Guangzhou, China, December 2021.

[6] K. C. Park and D.-H. Shin, "Security assessment framework for IoT service," *Telecommunication Systems*, vol. 64, no. 1, 2017.

[7] C. A. Ardagna, E. Damiani, J. Schütte, and P. Stephanow, *A Case for IoT Security Assurance*. Springer Singapore, 2018.

[8] M. Aslam, B. Mohsin, A. Nasir, and S. Raza, "FoNAC - An automated Fog Node Audit and Certification scheme," *Computers & Security*, vol. 93, June 2020.

[9] O. Jaradat, I. Sljivo, I. Habli, and R. Hawkins, "Challenges of Safety Assurance for Industry 4.0," in *Proc. of EDCC 2017*, Geneva, Switzerland, September 2017.

[10] M. Anisetti, C. Ardagna, N. Bena, and E. Damiani, "Stay Thrifty, Stay Secure: A VPN-based Assurance Framework for Hybrid Systems," in *Proc. of SECRYPT 2020*, Lieusaint - Paris, France, Jul. 2020.

[11] Z. Han, X. Li, K. Huang, and Z. Feng, "A Software Defined Network-Based Security Assessment Framework for CloudIoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, 2018.

[12] K. Govindan and A. P. Azad, "End-to-end service assurance in IoT MQTT-SN," in *Proc. of IEEE CCNC 2015*, Las Vegas, NV, USA, January 2015.

[13] O. Abu Waraga, M. Bettayeb, Q. Nasir, and M. Abu Talib, "Design and implementation of automated iot security testbed," *Computers & Security*, vol. 88, 2020.

[14] S. N. Matheu-García, J. L. Hernández-Ramos, A. F. Skarmeta, and G. Baldini, "Risk-based automated assessment and testing for the cybersecurity certification and labelling of IoT devices," *Computer Standards & Interfaces*, vol. 62, February 2019.

[15] B. Ali and A. I. Awad, "Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes," *Sensors*, vol. 18, no. 3, 2018.

[16] R. Hawkins, I. Habli, T. Kelly, and J. McDermid, "Assurance cases and prescriptive software safety certification: A comparative study," *Safety Science*, vol. 59, 2013.

[17] R. Calinescu, D. Weyns, S. Gerasimou, M. U. Iftikhar, I. Habli, and T. Kelly, "Engineering Trustworthy Self-Adaptive Software with Dynamic Assurance Cases," *IEEE Transactions on Software Engineering*, vol. 44, no. 11, 2018.

[18] S. Jahan, I. Riley, C. Walter, R. F. Gamble, M. Pasco, P. K. McKinley, and B. H. Cheng, "MAPE-K/MAPE-SAC: An interaction framework for adaptive systems with security assurance cases," *Future Generation Computer Systems*, vol. 109, 2020.

[19] V. Sklyar and V. Kharchenko, "Challenges in assurance case application for industrial IoT," in *Proc. of IEEE IDAACS 2017*, Bucharest, Romania, September 2017.

[20] ——, *Green Assurance Case: Applications for Internet of Things*. Springer International Publishing, 2019.

[21] H. Myrbakken and R. Colomo-Palacios, "DevSecOps: A Multivocal Literature Review," in *Proc. of SPICE 2017*, Palma de Mallorca, Spain, October 2017.

[22] M. Anisetti, C. A. Ardagna, F. Gaudenzi, and E. Damiani, "A Continuous Certification Methodology for DevOps," in *Proc. of MEDES 2019*, Limassol, Cyprus, November 2019.

[23] M. Anisetti, C. A. Ardagna, E. Damiani, and F. Gaudenzi, "A Semi-Automatic and Trustworthy Scheme for Continuous Cloud Service Certification," *IEEE Transactions on Services Computing (TSC)*, vol. 13, no. 1, 2020.

[24] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, 2016.

[25] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Applied Soft Computing*, vol. 10, no. 1, 2010.

[26] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," *arXiv preprint arXiv:1702.08608*, 2017.

[27] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A Survey of Methods for Explaining Black Box Models," *ACM Computing Surveys*, vol. 51, no. 5, 2018.

[28] N. Burkart and M. F. Huber, "A Survey on the Explainability of Supervised Machine Learning," *Journal of Artificial Intelligence Research*, vol. 70, 2021.

[29] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy Artificial Intelligence: A Review," *ACM Computing Surveys*, vol. 55, no. 2, 2022.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Proc. of NeurIPS 2017*, Long Beach, CA, USA, December 2017.

[31] M. Tan, A. Iacovazzi, N.-M. M. Cheung, and Y. Elovici, "A Neural Attention Model for Real-Time Network Intrusion Detection," in *Proc. of IEEE LCN 2019*, Osnabrueck, Germany, October 2019.

[32] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain," *ACM Comput. Surv.*, vol. 54, no. 5, 2021.

[33] P. Stephanow, G. Srivastava, and J. Schütte, "Test-Based Cloud Service Certification of Opportunistic Providers," in *Proc. of IEEE CLOUD 2016*, San Francisco, CA, USA, June–July 2016.

[34] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. of ICLR 2014*, Banff, Canada, April 2014.

[35] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *Proc. of ICLR 2015*, San Diego, CA, USA, May 2015.

[36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *Proc. of ICLR 2018*, Vancouver, BC, Canada, April 2018, poster.

[37] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, no. 2, 2020.

[38] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine Learning at the Network Edge: A Survey," *ACM Computing Surveys*, vol. 54, no. 8, 2021.

[39] H. Chen, H. Zhang, D. Boning, and C.-J. Hsieh, "Robust Decision Trees Against Adversarial Examples," in *Proc. of ICML 2019*, Long Beach, CA, USA, December 2019.

[40] B. Huang, Z. Ke, Y. Wang, W. Wang, L. Shen, and F. Liu, "Adversarial Defence by Diversified Simultaneous Training of Deep Ensemble," in *Proc. of AAAI 2021*, Virtual, February 2021.

[41] H. Fathoni, C.-T. Yang, C.-H. Chang, and C.-Y. Huang, "Performance Comparison of Lightweight Kubernetes in Edge Devices," in *Proc. of I-SPAN 2019*, Naples, Italy, 2019.