AI-powered Network Security: Approaches and Research Directions

Elisa Bertino
Imtiaz Karim
bertino@purdue.edu
karim7@purdue.edu
Purdue University, Computer Science Department
West Lafayette, Indiana, USA

ABSTRACT

Networks are today a critical infrastructure. Their resilience against attacks is thus crucial. Protecting networks requires a comprehensive security life-cycle and the deployment of different protection techniques. To make defenses more effective, recent solutions leverage AI techniques. In this paper, we discuss AI-based protection techniques, according to a security life-cycle consisting of several phases: (i) Prepare; (ii) Monitor and Diagnose; and (iii) React, Recovery and Fix. For each phase, we discuss relevant AI techniques, initial approaches, and research directions.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; • **Networks → Protocol testing and verification**.

KEYWORDS

protocol analysis, intrusion detection, smart network controllers

ACM Reference Format:
Elisa Bertino and Imtiaz Karim. 2021. AI-powered Network Security: Approaches and Research Directions. In *8th International Conference on Networking, Systems and Security (8th NSysS 2021), December 21–23, 2021, Cox's Bazar, Bangladesh.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3491371.3491384

1 INTRODUCTION

Networking has seen explosive growth and continuous evolution over the past decade, especially in the area of mobile communications. Fourth Generation Long Term Evolution (4G LTE) cellular technology has increased the bandwidth available for mobile devices, in essence, delivering broadband speeds to these devices. 5G New Radio (NR) is further enhancing the transmission speeds and cell capacity, as well as, reducing latency through the use of different radio technologies and is expected to provide Internet connections that are an order of magnitude faster than 4G LTE.

Technology continues to advance rapidly, however, and the next generation, 6G, is already being envisioned. 6G will make possible a wide range of powerful, new applications including holographic telepresence, telehealth, remote education, ubiquitous robotics and autonomous vehicles, smart cities and communities, Internet of things and of biothings, nanonetworks [27], and Industry 4.0, sometimes referred to as the Fourth Industrial Revolution, to name but a few [7, 8, 19, 24]. The advances we will see start at the hardware level and extend all the way to the top of the software "stack" [3]. Some of these advances will reduce power consumption, such as free-space optical communication [6] for indoor use, and multiple-access techniques able to scale up in scenarios in which large numbers of devices try to communicate with the same base station using a low duty cycle [27]. Also application-domain-specific processors are being developed that approach ASIC efficiency, but are relatively flexible [3]. These more flexible processors will enable powerful and efficient software-defined radios (SDRs), which will change how spectrum is used. In addition the development and deployment of software defined networks (SDN) and virtualization of network functions have enhanced the flexibility and customization of networks for different applications and reduced costs due to the use of cloud systems.

However, because all activities we may think of depend on network infrastructure, we can expect that attacks to these infrastructures will not longer be limited to simple (albeit significantly harmful) discrete events [5], such as a distributed denial-of-service attack against a portion of a network. Rather we can expect stealthy, persistent, and sophisticated activities aiming at establishing a foothold in core networks and maintaining such foothold to carry out massive disruption operations or sophisticated data gathering operations. Network security is thus critical. Many defense techniques and security practices have been proposed for network security. However the increasing complexity of network infrastructure makes their security extremely challenging.

We believe that reasoning about and addressing network security requires a security foundation for network protocols and comprehensive security life-cycle framework. The life-cycle we focus on consists of three main phases: (1) *Prepare* - it is at the core of security. It basically makes sure that the network system is best prepared to withstand attacks, failures, etc. (2) *Monitor and diagnose* - as even the best "prepared" network system can still be breached, monitoring activities are critical in order to detect attacks or anomalies that may be indicative of attacks. These attacks/anomalies have to be analyzed to gather information about the root causes, steps of attacks, etc. (3) *React, recover and fix* - once a diagnosis is obtained, actions have to be executed to ensure that the network system continue working, perhaps with reduced functions. These

8th NSysS 2021, December 21–23, 2021, Cox's Bazar, Bangladesh

ACM ISBN 978-1-4503-8737-8/21/12.
https://doi.org/10.1145/3491371.3491384

three phases are continuously executed and depending on the situation may even run in parallel. For example, once attacks/anomalies are diagnosed, the prepare phase is executed again to undertake activities such as patching vulnerabilities - exploited by the attacks, changing permissions, etc., while at the same time activities are executed to contain the attacks/manage the anomalies.

In order to devise more effective defenses, a recent trend is to leverage AI techniques, which are becoming more feasible with recent advances in AI combined with big data collection and analysis capabilities. A major problem is, however, that the application of AI techniques to networks is not trivial. For example, if one would like to apply reinforcement learning to secure a network, one has to understand how to properly design reward functions. At the same time, the use of AI techniques for securing networks offers interesting research directions. The goal of this paper is to discuss AI opportunities and challenges for network security and present some of our initial results.

The paper is organized as follows. In Section 2 we discuss relevant AI techniques relevant for the secure foundations of network protocols and for the security life-cycle; for each such technique we discuss research opportunities. In Section 3 we present a short overview of some of our work. Finally, in Section 4 we outline a few concluding remarks.

2 AI TECHNIQUES - RESEARCH DIRECTIONS

2.1 Secure foundations for Networks

Next-generation networks will have to coordinate communication, computation, caching and control (4C). Therefore, a critical component of the network fabric is represented by the 4C protocols. It is critical that these protocols do not have vulnerabilities that can be exploited by attackers. We thus need systematic approaches supporting comprehensive analyses of those protocols. Such analyses must cover both the specifications of the protocols, which when available are often expressed in natural language, and their implementations.

Well known analysis frameworks, based on formal methods, have been designed by Hussain et al. to analyze the 3GPP natural language specifications of the 4G LTE [10] and the 5G protocols [11]. However such methodologies require as input formal specifications, such as expressed by finite state machines (FSMs) of the analyzed protocols, and the properties to be verified against the specifications. Designing approaches for the (semi-) automatic generation of such specifications and the identification of properties is challenging. However this is an area where AI techniques, properly extended and/or combined with other non-AI techniques, can applied. Examples of such applications include:

- *Extraction of some formal (semi-formal) specifications from natural language descriptions of these protocols and properties to be verified.* Examples of these specifications are the ones provided by standardization bodies. **AI opportunity:** Using natural language processing (NLP) techniques for the extraction of the specifications and the properties. **AI challenges:** (i) Conventional NLP mining techniques are not suitable as standardization documents are very long, convoluted and with many cross-references. We need specialized NLP mining techniques. (ii) NLP mining techniques focus mainly on applications, such as sentiment analysis. In our context, the results of the NLP mining have to be a formal specification, for example expressed as a set of FSMs. We need a new generation of NLP mining able to provide outputs structured according to formats required for formal analyses.
- *Verification of large scale implementations of protocols.* When applying frameworks, like the ones proposed in [10] and [11], to large code bases a major challenge is the extraction of FSMs from the code. It is clear that a manual approach is not scalable. To address such an issue a recent approach leverages the functional conformance testing frameworks developed by protocol standardization bodies and/or commercial test-case developers [13] combined with code instrumentation. However, such an approach is not applicable when the source code is not available and/or when there is no suitable testing framework. **AI opportunity:** (i) Using symbolic learning techniques to learn FSMs from execution traces. (ii) Using machine learning (ML) techniques, combined with techniques such as symbolic execution, to reduce human efforts in generating proper abstractions from protocol implementations. **AI challenges:** So far, no approaches have been proposed addressing (i) and (ii). The challenge is to understand which AI techniques or combination of these can be combined/extended to address these challenges.

In addition to methodologies, like LTEInspector and 5Greasoner, focusing on logical vulnerabilities, other methodologies are available for detecting low level vulnerabilities, such as the memory ones. A popular category of methodologies is based on fuzzing [20]. A critical issue in the use of fuzzing is code coverage and solutions have been recently proposed using AI techniques [1, 21]. However, this is an area where more research is required.

2.2 Security Life-Cycle

As we have mentioned, we consider a network security life-cycle consisting of three phases continuously repeating: (a) prepare; (a) monitor and diagnose; (c) react, recover and fix. Those phases are executed across the different layers/components/subsystems of the network. They are context and situation dependent (e.g., certain situations may require compliance with strict real-time requirements). They are repeated because networks are dynamic and thus security has to adapt accordingly. Each phase requires specific activities, some of which can be supported/enhanced by the use of AI techniques (see Figure 1 for a summary). In what follows, we discuss for each phase relevant AI techniques and challenges in the application of these techniques.

2.2.1 *Prepare Phase.*

For this phase critical activities include:

- *Configuring security appliances, such as firewalls, and specifying security policies.* These policies will be increasingly attribute-based, that is, expressed as conditions against security-relevant properties of entities. Coming up with these configurations and policies is difficult. **AI opportunity:** Using AI for learning these configurations and policies. **AI challenges:** (i) These configurations and policies often need to be expressed in symbolic form (e.g., as rules); thus, we may need to integrate symbolic learning with non-symbolic learning. (ii) Learning those configurations and policies requires

	AI Technique	Purpose
Prepare	(1) Association Rule Mining, Symbolic Learning (2) Reinforcement Learning	(1) Learn Security Policies (e.g., access control policies, firewall rules, security services configuration policies) (2) Application Testing and Fuzzing; Security Resource Allocation
Monitor and Diagnose	(1) Classification Techniques (deep neural networks, SVM, random forest, etc.) (2) Causal Reasoning (3) AI Attention Mechanism (4) Graph Classification Techniques (5) Classification Techniques (deep neural networks, SVM, random forest, etc.)	(1) Anomaly Detection (2) Identify the Executed Attack Steps (3) Predict the Possible Next Attack Steps (4) Identify Malware
React, Recover and Fix	(1) Reinforcement Learning (2) Causal Reasoning (3) AI Attention Mechanism	(1) Identify Attack Containment Actions (2) Identify Recovery Actions (3) Identify the Attack Steps

Figure 1: AI Techniques for Each Phase of the Network Security Life-cycle

training data; an issue is how to get these data. An approach is to use transfer learning techniques by which to adapt rules learned in one context for use in another. Transfer learning has been explored for non-symbolic learning and we need to design transfer learning techniques for symbolic learning.

- *Ensuring that "network programs" are correct and, if not, automatically patching them.* Techniques, like fuzzing techniques, are widely used to identify vulnerabilities in code. However, these techniques have many limitations, such as being unable to identify logical vulnerabilities and also coverage issues. **AI opportunity:** Using AI techniques for enhancing fuzzing; using AI combined with techniques, such as symbolic execution, to abstract code into high level representations suitable for formal method analysis; using AI to automatically repair network program code. **AI Challenges:** (i) Identifying which AI technique, or combination techniques, is more suitable for those tasks. (ii) Designing approaches for generating training data and techniques for learning with few data.

2.2.2 Monitor and Diagnose Phase. For this phase critical activities include:

- *Continuously monitoring the network to detect anomalies at different layers/components that can be indicative of potential attacks.* **AI opportunity:** This is an area where machine learning techniques have been used. However, such techniques need to be extended to deal with: (i) large-scale complex systems; (ii) detection of anomalies in complex phenomena (e.g., finding anomalies in the physical layer in ultra-dense cells) in rapidly changing environments. **AI challenges:** (i) How to use AI for learning "behavior models" of physical layers that can be used as baselines for anomaly detection. (ii) How to support ultra-fast federated learning combining information from different local ML models. (iii) How to continuously adapt and evolve ML models.
- *Detecting entities (e.g., IoT devices, users, applications) present in the network.* This is challenging for dynamic complex environments but it is critical for security - to detect for example the presence of un-authorized devices in a certain area. **AI opportunities:** Using ML techniques for multi-dimensional fingerprinting of entities to be able to infer their presence and their characteristics (such as communication protocols they use) and also relational features (e.g., which entity communicates with which other entity). **AI challenges:** How to quickly identify and classify entities based on their own characteristics and their communication patterns (both physical and logical); perhaps the use of multi-dimensional embedding techniques could help.
- *Analyzing information reported by monitors.* Such analysis should also based on domain knowledge (e.g., in a single hop network, a black hole attack cannot happen [15]) about the current status of the network portion of interest, etc. The goal is to assess whether an anomaly or a problem, for example a node not responding, is due to an attack and if so diagnose the attack (e.g. understand the type of attack, the steps done by the attacker etc.), and possibly predict the next steps of the attacker. This is quite challenging and would require combining different techniques and identifying suitable processes to follow. **AI opportunity:** Using causal reasoning techniques to determine the actions/states that have resulted in the anomaly/problem. If the anomaly/problem is due to an attack, using causal reasoning techniques to determine the steps that the attacker has followed and using prediction techniques to determine which the next steps would be. **AI challenges:** (i) Information collected to diagnose anomalies/problems may be uncertain; therefore, multiple possible root causes and sequences of attack steps may be identified and one would need to associate some confidence levels to the various possibilities. (ii) Prediction techniques must be developed for complex processes involving adversarial parties. (iii) Both (i) and (ii) need to be executed very fast as network statuses continuously change and thus a diagnosis executed much later may be less accurate; also, for the react phase, a quick and correct diagnosis is critical.

2.2.3 React, Recover and Fix Phase. For this phase critical activities include:

- *Deciding actions to block the attack.* Such decisions must be based on the security goal (for example minimize data losses, do not disrupt certain critical communications and applications), and the anticipated steps of the attack (if such knowledge is available). **AI opportunity:** Using reinforcement learning (RL) techniques [26] to decide actions to contain the attack. **AI challenges:** (i) How to quickly train RL agents - perhaps transfer learning techniques can be used. (ii) How to deal with dynamic changes in the reward functions, for example, when the strategy to contain the attack may have to dynamically change. (iii) How to deal with very large state spaces.
- *Deciding actions to bring the system back to its "normal behavior".* Such actions may include using auxiliary resources and backups, and shifting activities to portions of the system not affected by the attack. This is a complex process which is application dependent. **AI opportunity:** Using sequential decision processes such as the ones based on RL. **AI challenges:** (i) How to include domain knowledge in RL systems; approaches may include constraining the exploration and/or properly formulating the reward function. (ii) Formulating reward functions may be complex; we need approaches to determine when a reward function is not the correct one and automatically modify it.
- *Permanently removing vulnerabilities/mis-configurations exploited by attacks.* This activity loops back to the prepare phase with additional knowledge gained by the attacks. **AI opportunity:** Using AI for forensic processes and postmortem analyses. **AI Challenges:** These analyses need to be specialized the specific network "planes" (e.g., user plane, data plane, control plane, management plane) and may require identifying suitable specific AI techniques and possibly enhance these.

3 OVERVIEW OF RECENT RESEARCH PROJECTS

We now present an overview of some initial research efforts along the research directions discussed in the previous sections.

3.1 Learning Access Control Rules from Data

Access control is gaining relevance for enhancing network security by restricting accesses to networks (or portion of them). It is especially critical in the context of zero-trust architecture (ZTA) [2]. ZTA has been introduced as a fine-grained defense approach paradigm shifting defenses from static, network-based perimeters to users, assets, and resources [18]. It assumes that no entities outside and inside the protected system can be trusted and therefore requires articulated and high-coverage deployment of access control.

However, because ZTA requires fine-grained access controls, we can expect that huge numbers of access control rules would have to be generated. These rules will likely be attribute based, that is, based on properties of subjects, protected resources, and contexts. Therefore a critical issue is the generation of these rules. It is clear that a manual approach to generate such rules is not feasible. An interesting direction to address such an issue is to use AI techniques to learn rules from data.

One such approach, the Polisma framework [12], has been recently designed to learn attribute-based access control rules from logs of past decisions. An important requirement in the design of Polisma has been that the results of the learning process be expressed in symbolic form, that is, as a set of rules so that these rules can be directly provided as input to access control enforcement engines and other security appliances, such as firewalls. It is important to also mention that in addition to logs of data on past access control decisions, one may have available other information, such directories and organizational charts. It is critical that these additional resources, if available, be also used. In addition, the rules should be of "good quality" [4].

To address such requirements, Polisma includes several steps (see Figure 2). The first step uses the well known association rule mining technique to extract an initial set of access control rules. However, these initial rules are often overfitted and unable to cover requests for which no past decisions are present in the log. The second step addresses such issue by generalizing the initial set of rules; generalization however has to be careful as generalizing too much may lead to very permissive rules. Therefore Polisma adopts a very careful generalization strategy that aims at the minimal possible generalization. Such a strategy has two variants, depending on whether additional information is available from organizational directories and charts. Once the rules have been properly generalized, Polisma executes a third step to adds some negative rules; this step is only significant if the considered access control model supports both permit and deny rules. Finally, in order to improve the completeness of the learned rule set, Polisma applies a ML classifier on log requests not covered by the learned set of rules. Then Polisma uses the result of the classification to generate data concerning the uncovered requests and generates additional rules in an "ad-hoc" manner. Polisma has been evaluated using two datasets (one real and the other synthetic). Experimental results show that Polisma is able to generate rules that accurately control access requests and outperforms existing approaches. The experimental results also show that after the first step the accuracy is low and steps 2 and 4 are instrumental in greatly increasing the accuracy.

Even though Polisma has been designed for controlling accesses to resources such as files, one interesting research direction is its use for generating access control rules for networks by taking advantages of logs of messages exchanged in networks.

3.2 Transfer Learning Techniques for Training Network Intrusion Detection Systems

Deep learning (DL) techniques have been shown to be highly effective for assisting network intrusion detection systems (NIDSs). Training DL classification models, however, requires vast amounts of labeled data which is often expensive and time-consuming to collect. Also, DL models trained using data from one type of network may not be used to detect attacks on other types of network or identify new families of attacks discovered over time.

An approach to address such drawbacks has been proposed by Singla et al. [22] based on transfer learning. Transfer learning refers to adapting a model learned in one domain, referred to as

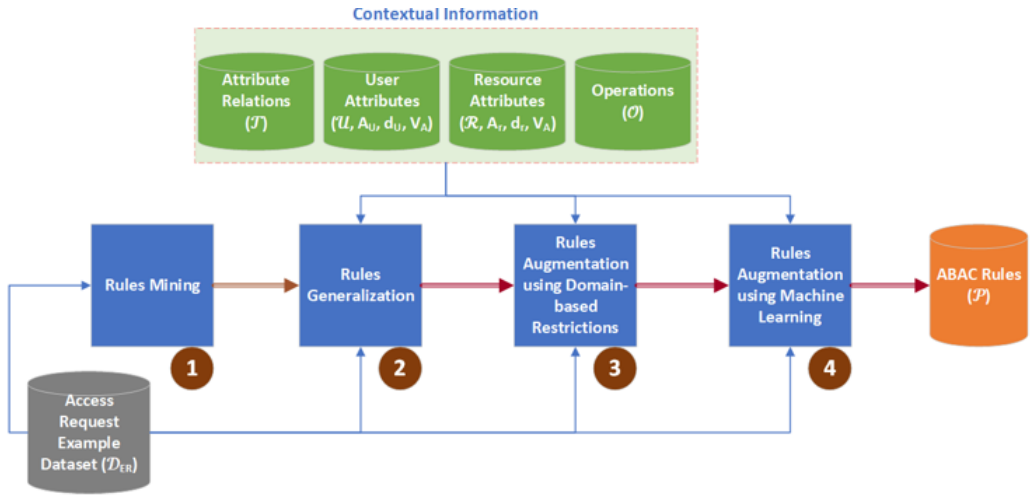

Figure 2: Learning Pipeline for Access Control Rules from [12]

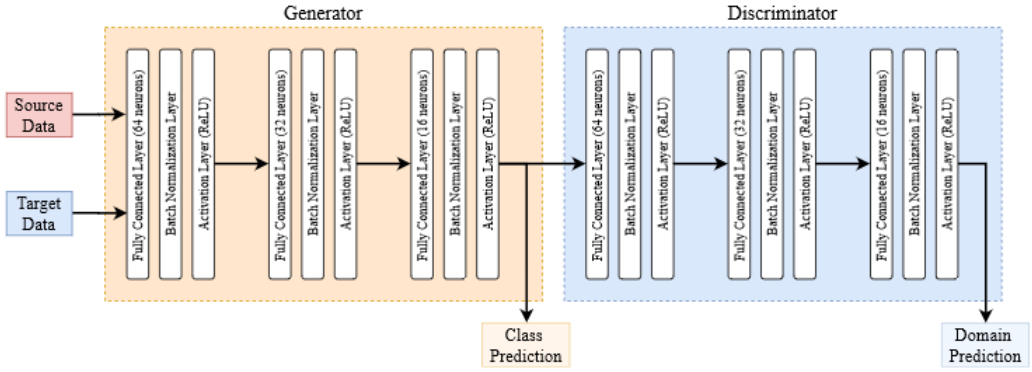

Figure 3: GAN architecture for adversarial domain adaptation from [22]. The GAN is trained to minimize the class prediction and domain prediction loss. The adversarial DA process outputs the generator as a classifier for the target dataset.

source domain for use in other domain, referred to as *target domain*. Transfer learning is particularly useful when the target domain has a small number of labeled data and thus using these data would not allow one to train an accurate model.

The approach by Singla et al. uses a domain adaptation (DA) technique based on a generative adversarial networks (GAN). The GAN basically creates a domain-invariant mapping of the source and target datasets. The GAN-based DA technique consists of two main components, namely a generator and a discriminator (see Figure 3). The goal of the generator is to take samples from the source and target datasets and convert them into a domain-invariant representation to fool the discriminator into misclassifying the generated representations. The mapping is also used as an input to a classifier which predicts which class the sample belongs to. The discriminator's goal is to identify whether the representation provided by the generator belongs to the source or the target dataset. The generator also has the additional goal of being able to distinguish between the classes on the source and target data distributions. The generator and the discriminator are trained simultaneously to get better at their respective tasks. The final goal is to minimize both the domain prediction and the class prediction loss. Once the training process is completed, the adversarial DA process outputs the generator as a classifier for the target dataset. We refer the reader to [22] for details on the loss functions used by the generator and discriminator and the training algorithm.

The GAN-based DA approach has been evaluated on two well know datasets used for NIDSs evaluation, that is, the *KDD-CUP99* [14] and the *UNSW-NB15* dataset [16]. It is important to mention that even though these datasets both deal with network attacks, they have some different features. However, the experimental results show that even when dealing with source and target datasets with some different features, the GAN-based DA approach is able to train an accurate target model even with few labeled target data.

Those results are promising. However there are some open interesting research directions. The first direction is related to the privacy of the source dataset, in that the GAN-based DA approach

requires that the source dataset be available to the party performing the transfer learning. Such a party in most cases is a party in the target domain. Therefore approaches must be investigated to ensure the privacy of the source dataset in the TL process. Another possibility is to perform the TL process in a cloud, in which case the privacy of both the source and target datasets must be ensured. The second direction is related to transfer learning processes using multiple source domains to train the target model. As more datasets and training models become available, it is interesting to take advantage of such resources to further reduce the numbers of data required to train a model at a target domain. However, there are issues to investigate, including modifications required to the GAN architecture, shown in Figure 3, approaches to determine which source datasets are actually beneficial for a specific target domain, and how to deal with the case in which the target domain has only unlabeled data.

3.3 Security-Driven Reinforcement Learning for Software Defined Networks

The control plane is a component of software-defined networks (SDNs) to which AI techniques have been applied. The goal is to make network controllers "smarter" in taking critical decisions related to traffic engineering, such as how to maximize quality of service.

As discussed by Mudgerikar et al. [17], even though ML techniques can model complexity, they need sufficient training datasets, which may be difficult to gather for large scale networks with a diversity of traffic behavior. RL, on the other hand, learns optimal policies online, through explorations, based on the system state using a model-free approach. These policies are more likely to transfer over to new situations and contexts, and these characteristics make them more suitable for network control. Therefore, among the various AI techniques, RL [26] has been proposed for various key applications, including routing [23], traffic rate control [9] and load balancing [25]. However applications of RL to network control have not considered security; it is critical that achieving a given optimization goal, such as minimizing latency, is not at the expense of security. A recent approach by Mudgerikar et al. [17], focusing on intelligent rate control, addresses such a requirement by constraining explorations based on security policies. Those security policies are learnt in a semi-supervised manner in the form of 'partial attack signatures', learned by using a deep q-network from packet captures of an IDS dataset. They are then encoded in the objective function of the RL based optimization framework. Experimental results show that such an approach is effective. There is however more work to be done including extensions to such a security-constrained approach for other network control functions and its integration with open source SDN software. Also its scalability and robustness need to be investigated.

4 CONCLUSIONS

In this paper we have discussed a few research directions concerning the application of AI techniques to network security and presented a short overview of some of our related projects. It is important to emphasize that, unlike many other application domains, networks are large scale complex dynamic systems, with many different stakeholders, and therefore a successful application of AI requires the combination of several AI techniques and models. Also those techniques and models must be very efficient in providing recommendations, decisions, classification results, and predictions, and need to be robust against attacks.

ACKNOWLEDGMENTS

The work reported in this paper has been funded by NSF under Grants IIS-2112471 and DGE-2114680.

REFERENCES

[1] Mansour Ahmadi, Reza Mirzazade farkhani, Ryan Williams, and Long Lu. 2021. Finding Bugs Using Your Own Code: Detecting Functionally-similar yet Inconsistent Code. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*.

[2] Elisa Bertino. 2021. Zero Trust Architecture: Does It Help? *IEEE Security & Privacy* 19, 5 (2021).

[3] Elisa Bertino, Daniel Bliss, Daniel Lopresti, Larry Peterson, and Henning Schulzrinne. [n. d.]. Computing Research Challenges in Next Generation Wireless Networking - A Computing Community Consortium (CCC) Quadrennial Paper. https://arxiv.org/ftp/arxiv/papers/2101/2101.01279.pdf.

[4] Elisa Bertino, Amani A. Jabal, Seraphin Calo, Dinesh Verma, and Christopher Williams. 2018. The Challenge of Access Control Policies Quality. *ACM Journal on Data and Information Quality* 10, 2 (2018).

[5] John Brtis and Michael Mcevilley. [n. d.]. Systems Engineering for Resilience. The MITRE Corporation: Bedford, MA, 2013.

[6] Klaus David and Hendrik Berndt. 2018. 6G Vision and Requirements: Is There Any Need for Beyond 5G? *IEEE Vehicular Technology Magazine* 13, 3 (2018), 72–80.

[7] Nate Foster, Nick McKeown, Guru Rexford, Jennifer Parulkar, Larry Peterson, and Oguz Sunay. 2020. Using Deep Programmability to Put Network Owners in Control. *ACM SIGCOMM Computer Communication Review* 50, 4 (2020).

[8] Marco Giordani, Michele Polese, Marco Mezzavilla, Sandeep Rangan, and Michele Zorzi. 2020. Towards 6G Networks: Use Cases and Technologies. *IEEE Communications Magazine* 58, 3 (2020).

[9] Xiaohong Huang, Tingting Yuan, Guanhua Qiao, and Yizhi Ren. 2018. Deep reinforcement learning for multimedia traffic control in software defined networking. *IEEE Network* 32, 6 (2018), 35–41.

[10] Syed Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. [n. d.]. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. In *Network and Distributed Systems Security (NDSS) Symposium 2018, San Diego, CA, USA, February 18-21, 2018.*

[11] Syed Hussain, Mitziu Echeverria, Imtiaz Karim, Omar Chowdhury, and Elisa Bertino. [n. d.]. 5GReasoner: A Property-Directed Security and Privacy Analysis Framework for 5G Cellular Network Protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019.*

[12] Amani Jabal A., Elisa Bertino, Jorge Lobo, Mark Law, Alessandra Russo, Seraphin Calo, and Dinesh Verma. [n. d.]. Polisma - A Framework for Learning Attribute-Based Access Control Policies. In *Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part I. Lecture Notes in Computer Science 12308, Springer 2020.*

[13] Imtiaz Karim, Hussain Syed, and Elisa Bertino. [n. d.]. ProChecker: An Automated Security and Privacy Analysis Framework for 4G LTE Protocol Implementations. In *Proceedings of the 41st IEEE International Conference on Distributed Computing Systems, ICDCS2021, July 07-10, 2021.*

[14] MIT Lincoln Labs. 1999. KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/task.html

[15] Daniele Midi, Antonino Rullo, Anand Mudgerikar, and Elisa Bertino. [n. d.]. Kalis - A System for Knowledge-Driven Adaptable Intrusion Detection for the Internet of Things. In *Proceedings of the 37th IEEE International Conference on Distributed Computing Systems, ICDCS2017, July 05-08, 2017.*

[16] Nour Moustafa and Jill Slay. 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Military Communications and Information Systems Conference (MilCIS), 2015.* IEEE, 1–6.

[17] Mudgerikar, Anand and Bertino, Elisa, and Lobo, Jorge and Verma, Dinesh. [n. d.]. A Security-Constrained Reinforcement Learning Framework for Software Defined Networks. In *Proceedings of the IEEE International Conference on Communications, ICC2021, June 14-23, 2021.*

[18] NIST. [n. d.]. Zero trust architecture. [Online]. Available from: https://csrc.nist.gov/publications/detail/sp/800-207/final.

[19] University of Oulu. [n. d.]. Key Drivers and Research Challenges for 6G Ubiquitous Wireless Intelligence. http://jultika.oulu.fi/files/isbn9789526223544.pdf.
[20] Mathias Payer. 2019. The Fuzzing Hype-Train: How Random Testing Triggers Thousands of Crashes. *IEEE Security & Privacy* 17, 1 (2019).
[21] Sameer Reddy, Caroline Lemieux, Rohan Padhye, and Koushik Sen. [n. d.]. Quickly generating diverse valid test inputs with reinforcement learning. In *Proceedings of the 42nd International Conference on Software Engineering, ICSE2021, June 27-July 19, 2021.*
[22] Ankusu Singla, Elisa Bertino, and Dinesh Verma. [n. d.]. Preparing Network Intrusion Detection Deep Learning Models with Minimal Data Using Adversarial Domain Adaptation. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, ASIACCS '20, Taipei, Taiwan, October 5-9, 2020.*
[23] Giorgio Stampa, Marta Arias, David Sánchez-Charles, Victor Muntés-Mulero, and Albert Cabellos. 2017. A deep-reinforcement learning approach for software-defined networking routing optimization. *arXiv preprint arXiv:1709.07080* (2017).
[24] C. Emilio Strinati, Sergio Barbarossa, Jose Luis Gonzalez-Jimenez, Dimitri Ktenas, Nicolas Cassiau, and Cedric Dehos. [n. d.]. 6G: The next frontier. https://https://arxiv.org/abs/1901.03239.
[25] Penghao Sun, Zehua Guo, Gang Wang, Julong Lan, and Yuxiang Hu. 2020. MARVEL: Enabling controller load balancing in software-defined networks with multi-agent reinforcement learning. *Computer Networks* (2020), 107230.
[26] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press.
[27] Tataria, Harsh and Shafi, Mansoor and Molish, Andreas F. and Dohler, Mischa and Sjöland, Henrik, and Tufvesson, Fredrik. 2021. 6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities. *Proceedings of IEEE* 109, 7 (2021), 1166–1199.