Research Challenges for Applying Machine Learning in Cybersecurity

Fabio Massimo Zennaro
fabiomz@ifi.uio.no

University of Oslo

February 9, 2018

Aim and Organization

In this presentation we are going to survey research topics at the intersection of **machine learning** and **computer security**.

1. Concepts from machine learning
2. Machine learning for computer security
3. Security in machine learning
4. Safety of machine learning

1. Concepts from Machine Learning

What is machine learning?

ML is the field studying *automated induction procedures to develop useful models*.

- *Automated procedures*: algorithms
- *Induction*: from particular (data) to general (model)
- *Models*: abstractions of a phenomenon [Floridi, 2011]
- *Useful*: allowing us to explain/predict/control [Floridi, 2011]

What is model?

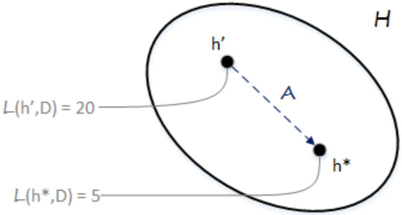
A **model** is a mathematical representation of a phenomenon.

$$f : X \to Y \qquad \begin{matrix} P(X) \\ P(X, Y) \\ P(Y|X) \end{matrix}$$

- How do we learn a model?
- How do we evaluate a model?

How do we learn? (I)

1. *Data* $\mathcal{D}$
2. *Family of models* or *hypothesis space* $\mathcal{H}$
3. *Loss/objective/reward function* $\mathcal{L}(h, \mathcal{D})$
4. *Exploration strategy* of the hypothesis space $\mathcal{A}$



Learning means solving an **optimization problem**:

$$h* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \mathcal{L}(h, \mathcal{D})$$

How do we learn? (II)

Example: Learning to discriminate digits using a neural network

$$f : \text{Image} \rightarrow \text{Label}$$

1. *Data:* $\mathcal{D} = \{\text{Set of digits and labels}\}$
2. *Hypothesis space:* $\mathcal{H} =$ approximate continuous functions on compact subsets of $\mathcal{R}^n$ [Cybenko, 1989]
3. *Loss function:* $\mathcal{L} =$ mean squared error in prediction
4. *Exploration strategy:* $\mathcal{A} =$ gradient descent

$$h* = \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{L}(h, \mathcal{D})$$

How do we evaluate?

We want **generalization**, a model that explains not only the data used to learn, but all possible data produced by the same phenomenon.

1. *Training data*: used to learn
2. *Test data*: used for evaluation

In general, to be meaningful training and test data must be *independent samples from the same distribution*:

$$p(X^{tr}) = p(X^{te})$$

Remarks on learning (I)

- Hypothesis space, loss function and exploration strategy are usually tightly bound and comes as a *machine learning algorithm*.
- There are three popular flavours of learning algorithms:

Supervised	$f : X \to Y$
Unsupervised	$f : X \to Z$
Reinforcement	$\pi(a\|s)$

- There are two main stages in the lifecycle of machine learning

Learning:	learning a specific model
Inference or *deployment*:	using the model

Some generic challenges in ML

- A model must be built on *assumptions* [MacKay, 2003].
- Only what can be induced from the data can be learned; there must be *meaningful* relationship or correlations in the data.
- There is no thing such *THE* model of the data [Wolpert and Macready, 1997].
- A model is not correct or wrong; it must be properly *evaluated*.
- There are always *trade-offs* to consider:
  Expressivity vs Efficiency
  Performance vs Interpretability
  Training performance vs Test performance [Domingos, 2012]

2. Machine Learning for Computer Security

ML for Computer Security

ML can be used for computer security whenever we can define and learn *models of malicious behaviour*:

$f : X \to Y$	A relationship between DNS queries and malware categories
$P(X)$	A probability distribution over user behaviours being malicious

This models are not going to be specified explicitly, but inferred from data.

ML for Computer Security

- **Network models** [Gardiner and Nagaraja, 2016]
  - Generic communication patterns
  - Specific traffic types
  - Temporal patterns
  - Spatial patterns
- **Host models**
- **User models**

Generic Communication Patterns

Model malwares wrt their communication behaviour and content of the packets.

- Detection of hosts participating in malicious P2P networks based on the packets sent and received [Rahbarinia et al., 2013]
- Evaluation of reputation of nodes from network flows [Zhang et al., 2014]
- Clustering of hosts based on destination, payloads and OS [Yen and Reiter, 2008]

Specific traffic types

Model malwares wrt to specific types of traffic, such as DNS queries and domains requested.

- Detection of *command and control systems* from DNS queries [Lison and Mavroeidis, 2017; Schiavoni et al., 2014]
- Detection of *command and control systems* from passive DNS analysis [Bilge et al., 2011]

Temporal patterns

Model malicious servers wrt temporal patterns of requests.

- Identification of malicious servers from netflow data describing client access and temporal pattern of exchanges [Bilge et al., 2012]
- Detection of *fast flux networks* from the dynamics of IP addresses queried [Perdisci et al., 2012]

Spatial patterns

Model malwares wrt the spatial network patterns they instantiate.

- Detection of botnets through an analysis of connected graphs [Collins and Reiter, 2007]
- Identification of malicious domains through belief propagation of reputation [Manadhata et al., 2014]

Challenges in applying ML to computer security

- Learning happens in an adversarial environment
  Adversarial Learning [Goodfellow et al., 2014a]
- Behaviours are highly adaptive
  Robust Learning [Sugiyama and Kawanabe, 2012]
  Continuous Learning
  Active Learning
- Limited data
- Scalability and relevance of data

3. Security in Machine Learning

Security in Machine Learning [Papernot et al., 2016]

Attack Surface:

- *Data:* collection and processing of data $\mathcal{D}$
- *Model:* including hypothesis space $\mathcal{H}$, loss function $\mathcal{L}$ and learning strategy $\mathcal{A}$

Adversary Goal:

- *Confidentiality-Privacy:* extracting data or information about the model
- *Integrity-Availability:* compromise learning or inference

Adversary Capability:

- *White-box knowledge at learning time*
- *Black-box knowledge at learning time*
- *White-box knowledge at inference time*
- *Black-box knowledge at inference time*

Integrity attacks at learning time

Attacks aimed at derailing learning.

- *Label manipulation:* harmful perturbation of labels given partial or full knowledge of a model [Biggio et al., 2011; Mozaffari-Kermani et al., 2015]
- *Direct data poisoning:* insertion of spurious data points in the data set to compromise learning [Kloft and Laskov, 2010; Mei and Zhu, 2015; Steinhardt et al., 2017]
- *Indirect data poisoning:* malicious modification of the data generating process to generate inconsistent data [Perdisci et al., 2006]
- *Subversion of distributed learning:* compromising the learning updates computed by distributed machines [Blanchard et al., 2017; Ghodsi et al., 2017]

Integrity attacks at inference time

White-box attacks attacks exploiting knowledge of the inference model:

- *Direct poisoning using adversarial examples:* generation of adversarial data points exploiting gradient [Szegedy et al., 2013; Goodfellow et al., 2014b]
- *Indirect poisoning using adversarial examples:* insertion of adversarial examples in the data processing pipeline [Kurakin et al., 2016]

Black-box attacks attacks without knowledge of the inference model:

- *Adversarial example transferability:* use of adversarial data points generated on an approximate substitute model [Szegedy et al., 2013]

Privacy attacks at inference time

Attacks aimed at extracting sensitive information.

- *Membership test:* querying the model to discover if specific data points were part of the training set
- *Statistical property test:* querying the model to determine statistical properties of the training set [Ateniese et al., 2015]
- *Model inversion attack:* recovering information about the inputs from the outputs [Fredrikson et al., 2014]
- *Model extraction:* retrieving value of model parameters from outputs [Tramèr et al., 2016]

Challenges in securing ML applications

- Optimistic environment assumptions
- Open systems
- Trade-off between performance and security
- Lack of quantitative measures for security

4. Safety of Machine Learning

AI Safety

Study of the broad impact of machine learning on the environment in which it is deployed.

- *Long-term AI safety:* concerned with existential risks [Bostrom, 2014]
- *Concrete AI safety:* current safety problem in machine learning [Amodei et al., 2016b]

Concrete AI Safety

- **Catastrophic Loss Function Misspecifications** [Amodei et al., 2016b]
  - *Incorrect formal loss function*
    - Negative side effects
    - Reward hacking
  - *Unlearnability of the loss function*
    - Scalable oversight
  - *Incorrect specification of the model*
    - Safe exploration
    - Robustness to distribution shift
- **Interpretability of the Learned Model**
- **Fairness of the Learned Model**

Other related topics: *ethics*; *privacy*; *policy*; *accountability*.

Avoiding Negative Side Effects

How do we guarantee that an agent will not cause bad side effects while pursuing its aim?

Example: If we train a cleaning robot whose loss function is proportional to the rubbish in a room, how do we guarantee it will not knock down furniture while cleaning up?

- Define or learn a reward function that penalizes changes to the environment
- Minimize *empowerment* of an agent [Salge et al., 2014]
- Combine different reward functions of multiple agents [Hadfield-Menell et al., 2016]
- Make reward function uncertain

Reward Hacking

How do we guarantee that an agent will not trick its loss function?

Example: If we train a cleaning robot whose loss function is proportional to the rubbish in a room, how do we guarantee it will not just disable its vision system?

- Adaptive or adversarial reward function
- Providing limited or blinded information about the environment
- Setting a cap on reward [Ajakan et al., 2014]
- Combine multiple reward functions [Deb, 2014]
- Instantiating trip wires

Scalable Oversight

How do we guarantee that an agent will learn every relevant aspect of its aim with a limited oversight?

Example: If we train a cleaning robot whose loss function is proportional to the rubbish in a room, how do we guarantee it will learn not to destroy valuable stray items on the floor?

- Train using aggregate or noisy information [Mann and McCallum, 2010]
- Hierarchical learning [Dayan and Hinton, 1993]

Safe Exploration

How do we guarantee that an agent will not undertake catastrophic actions while exploring?

Example: If we train a cleaning robot, how do we guarantee it will insert a wet mop into a plug?

- Use a risk-sensitive reward function accounting for worst-case scenario [Garcıa and Fernández, 2015]
- Learn from near-optimal demostrations [Abbeel and Ng, 2005]
- Train in a simulated environment
- Bound exploration
- Rely on human oversight [Saunders et al., 2017]

Robustness to Distribution Shift

How do we guarantee that an agent will behave consistently when the environment changes?

Example: If we train a cleaning robot in a house room, how do we guarantee it will behave safely in a factory?

- Rely on *covariate shift adaptation* [Sugiyama and Kawanabe, 2012]
- Devise algorithms to detect out-of-distribution conditions and devise appropriate strategies
- Increase and extend the training data [Amodei et al., 2016a]
- Model through counterfactual reasoning

Interpretability

How do we guarantee that decisions of machine learning systems can be explained and understood?

Example: If we use a machine learning model to decide on a loan, how do we guarantee the decision can be understood?

- Favour simple interpretable models [Lou et al., 2012; Caruana et al., 2015]
- Compress complex models
- Improve visualization techniques [Vellido et al., 2012]
- Use specific tools to get insights into complex models (e.g.: *saliency maps*) [Simonyan et al., 2013; Montavon et al., 2017]
- Interpret models locally [Ribeiro et al., 2016] [1]

[1] Thanks to Pierre Lison for pointing out this work.

Fairness [Kusner et al., 2017]

How do we guarantee that decisions of machine learning systems do not create or spread biases?

Example: If we use a machine learning model to choose an employee, how do we guarantee it will not be affected by racial prejudices?

$$f : (X, A) \to Y$$

- Fairness through unawareness
- Individual fairness
- Demographic parity
- Equality of opportunity
- Counterfactual fairness [Pearl, 2009; Kusner et al., 2017]

Thanks!

Thank you for listening!

References I

Pieter Abbeel and Andrew Y Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 1–8. ACM, 2005.

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182, 2016a.

References II

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016b.

Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *Asian Conference on Machine Learning*, pages 97–112, 2011.

References III

Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. Exposure: Finding malicious domains using passive dns analysis. In *Ndss*, 2011.

Leyla Bilge, Davide Balzarotti, William Robertson, Engin Kirda, and Christopher Kruegel. Disclosure: detecting botnet command and control servers through large-scale netflow analysis. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 129–138. ACM, 2012.

Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, pages 118–128, 2017.

References IV

N. Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014. ISBN 9780199678112. URL `https://books.google.no/books?id=7_H8AwAAQBAJ`.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

M Patrick Collins and Michael K Reiter. Hit-list worm detection and bot identification in large networks using protocol graphs. In *International Workshop on Recent Advances in Intrusion Detection*, pages 276–295. Springer, 2007.

References V

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4): 303–314, 1989.

Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In *Advances in neural information processing systems*, pages 271–278, 1993.

Kalyanmoy Deb. Multi-objective optimization. In *Search methodologies*, pages 403–449. Springer, 2014.

Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

Luciano Floridi. *The philosophy of information*. Oxford University Press, 2011.

References VI

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*, pages 17–32, 2014.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Joseph Gardiner and Shishir Nagaraja. On the security of machine learning in malware c&c detection: A survey. *ACM Computing Surveys (CSUR)*, 49(3):59, 2016.

References VII

Zahra Ghodsi, Tianyu Gu, and Siddharth Garg. Safetynets: Verifiable execution of deep neural networks on an untrusted cloud. In *Advances in Neural Information Processing Systems*, pages 4675–4684, 2017.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, and Aaron Courville. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

References VIII

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems*, pages 3909–3917, 2016.

Marius Kloft and Pavel Laskov. Online anomaly detection under adversarial impact. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 405–412, 2010.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

References IX

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, pages 4069–4079, 2017.

Pierre Lison and Vasileios Mavroeidis. Automatic detection of malware-generated domains with recurrent neural models. *arXiv preprint arXiv:1709.07102*, 2017.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158. ACM, 2012.

David J.C. MacKay. *Information theory, inference, and learning algorithms*, volume 7. Cambridge University Press, 2003.

References X

Pratyusa K Manadhata, Sandeep Yadav, Prasad Rao, and William Horne. Detecting malicious domains via graph inference. In *European Symposium on Research in Computer Security*, pages 1–18. Springer, 2014.

Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(Feb):955–984, 2010.

Shike Mei and Xiaojin Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI*, pages 2871–2877, 2015.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

References XI

Mehran Mozaffari-Kermani, Susmita Sur-Kolay, Anand Raghunathan, and Niraj K Jha. Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE journal of biomedical and health informatics*, 19(6):1893–1905, 2015.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Roberto Perdisci, David Dagon, Wenke Lee, Prahlad Fogla, and Monirul Sharif. Misleading worm signature generators using deliberate noise injection. In *Security and Privacy, 2006 IEEE Symposium on*, pages 15–pp. IEEE, 2006.

References XII

Roberto Perdisci, Igino Corona, and Giorgio Giacinto. Early detection of malicious flux networks via large-scale passive dns traffic analysis. *IEEE Transactions on Dependable and Secure Computing*, 9(5):714–726, 2012.

Babak Rahbarinia, Roberto Perdisci, Andrea Lanzi, and Kang Li. Peerrush: Mining for unwanted p2p traffic. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 62–82. Springer, 2013.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

References XIII

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment–an introduction. In *Guided Self-Organization: Inception*, pages 67–114. Springer, 2014.

William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. *arXiv preprint arXiv:1707.05173*, 2017.

Stefano Schiavoni, Federico Maggi, Lorenzo Cavallaro, and Stefano Zanero. Phoenix: Dga-based botnet tracking and intelligence. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 192–211. Springer, 2014.

References XIV

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. *arXiv preprint arXiv:1706.03691*, 2017.

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: introduction to covariate shift adaptation*. MIT Press, 2012.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

References XV

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX Security Symposium*, pages 601–618, 2016.

Alfredo Vellido, José David Martín-Guerrero, and Paulo JG Lisboa. Making machine learning models interpretable. In *ESANN*, volume 12, pages 163–172. Citeseer, 2012.

David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997.

References XVI

Ting-Fang Yen and Michael K Reiter. Traffic aggregation for malware detection. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 207–227. Springer, 2008.

Junjie Zhang, Roberto Perdisci, Wenke Lee, Xiapu Luo, and Unum Sarfraz. Building a scalable system for stealthy p2p-botnet detection. *IEEE transactions on information forensics and security*, 9(1):27–38, 2014.