RESEARCH STATEMENT

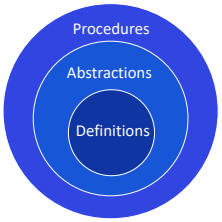
Teodora Baluta
Ph.D. Candidate at National University of Singapore (NUS), Singapore
https://teobaluta.github.io

I work in systems security. My research is on **rigorous security analysis in (ML) systems**. My approach is to design algorithmic tools with provable guarantees for security analyses and defenses, and show their utility in addressing practical concerns in ML security.

ML security is a new sub-area in security. It is bringing out many concerns with commercially deployed ML systems. But which of these concerns are actual vulnerabilities, as opposed to just expected outcomes of generalizing from samples to unknown distributions? Could ML creators prove that they have trained with a particular dataset and that their models are not violating certain desirable properties?

My research tackles these questions from first principles. I work on foundational aspects of rigorous security: precise definitions of the security vulnerabilities, frameworks of reasoning about the workings of the system, and sound procedures to verify security, or the lack thereof.

- **Security Vulnerabilities** [1]: We formulate the first precise security definitions for solving intellectual property disputes over training data. Under these precise definitions, our main result is that the de-facto training process in ML, called **stochastic gradient descent (SGD), is collision-resistant** under mild checkable conditions. We provide a sound procedure to check these conditions for large models with millions of parameters. We show empirically why relaxed security definitions are insufficient and lead to contradicting conclusions [1].



- **Abstractions** [2, 3]: We develop the **first causal models for stochastic gradient descent**. Our causal models uncover a new connection: the bias and variance components of generalization also separately affect memorization [2]. This explains why even when larger models generalize better, they also memorize more. Causal models are also useful in explaining satisfiability procedures, i.e., SAT solvers, on particular benchmarks, and drawing insights into solving heuristics and distributions of input formulae [3].

- **Sound procedures** [4, 5, 6, 7]: We give the **first sound procedures** that analyze security properties with probably approximately correct **(PAC)-style guarantees** in neural networks. We instantiate these procedures for adversarial robustness, fairness, and susceptibility to training data poisoning [5, 4]. I also worked on provable defenses, using differentially-private training for graph learning [6, 7].

Broader Research Interests [8, 9, 10]. Beyond my thesis research, I enjoy working on problems that are both algorithmic and practical. I worked on program synthesis with generalization guarantees [8], which is within the larger scope of algorithms with provable guarantees. I also worked on the learnability of rules for cross-language code transpilation [9], and inferring data flow rules for taint analysis of binaries [10].

Thrust I. Rigorous Security Analysis for Machine Learning Systems

Security is foremost about defining vulnerabilities under a threat model. Security of ML systems, i.e., systems that incorporate an ML model, is a new area with a deluge of concerns. For instance, we are currently witnessing several class-action lawsuits for copyright infringement that have been filed by artists against ML creators [15, 20]. When taken to court, what is the right procedure to decide if the ML creator used the copyrighted data samples, *beyond reasonable doubt*? If ML creators train models with private data, is it possible for models to leak the training data?

One might hope that traditional security approaches work to think about the security of ML systems, and answer such questions. But this is not true. The paradigm of learning from data poses many challenges compared to traditional software security. ML systems are stochastically trained on data coming from an unknown data distribution. It is thus unclear what exactly is the intended vs. unintended behavior (vulnerability) of the ML model or the training process. To make matters worse, the de-facto training process has many knobs that affect the outcome, and due to its complexity evades purely theoretical analysis. Thus, one of the key problems in analyzing machine learning systems is the lack of systematic tools and formal definitions of its key security properties.

Definitions of Security [1]. Several lawsuits have been filed claiming copyright infringement recently against ML models that could cost billions of dollars. To address both intellectual property concerns and ensure non-repudiation properties of training data, we require proofs of model ownership and creation using a

particular dataset beyond reasonable doubt. Such proofs are feasible if SGD executions can be recorded and proven to be unforgeable, i.e., there exists a unique set of samples that correspond to a gradient training step. We give the *first provable procedure for checking collision-resistance in SGD execution traces* [1]. The key to this result are precise definitions of forgery in SGD execution traces. We experimentally find that traces are unforgeable for all considered experimental setups. This is surprising because our results contradict conjectures made in prior work [24, 22]. The contrast stems from definitions of exact vs. approximate equal training step update. We show that if one replaces a training step with an approximately forged one, the difference between the forged and original traces *diverges* with subsequent training. Because of this, forgeries that result in approximately equal updates are detectable. We argue that forgery needs to be precisely defined with algebraic properties, for which we give sound procedure.

Our result points to a bigger conceptual contribution: SGD training updates in an execution trace can be checked to be collision-resistant. This opens up several interesting avenues of connecting cryptography and SGD, e.g., can we show one-wayness of SGD updates? To make this more practical is my research goal, and several challenges need to be addressed.

Abstractions for Reasoning about the Training Process [2]. What is the distinction and overlap between (intended) generalization and (unintended) memorization? In ML, the intended behavior is to generalize from training on a dataset, and not memorize properties of specific samples, revealing private information about individuals' data in the dataset. This question has been handled by many statistical approaches but has resulted in more unexpected overlaps than clear-cut dichotomies between generalization and memorization. We study this question through the lens of **causality**. Specifically, we consider membership inference attacks [21] that test whether ML models have memorized training examples. While many membership inference attacks have been proposed, it is difficult to understand why such tests are successful in ML only from experiments. We show that claims about *causes of memorization* via membership inference attacks are refutable and can lead to **paradoxes** because of incomplete characterization of the learning process [2]. Our work is the first to propose causal graphs to model the stochastic training and attack procedures. The causal graphs allow encoding known mathematical facts along with data-induced facts. We encode that the generalization error can be decomposed as bias and variance, and find that these components *individually* influence memorization. In particular, we explain why even when the generalization error goes down as the size of the model increases, memorization still increases up to a point: Because the variance component plays a role by itself in the accuracy of membership inference. In addition to these new connections, we check 18 hypothesized causes stated in prior works for membership inference attacks via causal reasoning on the graphs, refuting 7/18. We find that well-known membership inference tests have similar causes as poor generalization. We have also shown that causality helps understand benchmark-dependent performance of non-stochastic processes, for example common SAT solvers [3].

Sound Procedures for Statistical Verifiability [4, 5]. Having the right definitions and abstractions, security properties of ML models can naturally be specified as properties over distributions. Prior work on verification of these properties, however, is qualitative [19, 23]. It is concerned with the worst-case example in the distribution. We propose **quantitative ML verification frameworks** that measure how often the property is true on average with PAC-style [26] guarantees.

We have investigated both white-box and black-box models of access to the verifier. In the case of white-box, we instantiated a sound procedure for binarized neural networks for which there exist encodings to propositional logic. Our work shows that such encodings are sound in that counting the number of satisfying assignments over a set of projected variables of the formulae returns the estimate of how often the property is true. We show the utility of quantitative verification to adversarial robustness, fairness and susceptibility to trojan attacks as well [4]. In the case of black-box access to models, we propose a formulation that ascertains whether a property holds for less than a user-specified threshold or more than a user-specified threshold [5] with PAC-style guarantees. We show that when our algorithm Provero terminates, it returns with the desired confidence and error tolerance. Our empirical results demonstrate that Provero can statistically verify robustness for large deep neural networks such as VGG16, VGG19, ResNet50, DenseNet121 and InceptionV3. We find that such average-case adversarial robustness correlates highly with specialized attacks [5]. This shows that quantitative estimates are a good predictor of true robustness while not being attack-specific.

Our benchmarks have been used in SAT competitions since 2020, in the **model counting track**. Several works have improved quantitative verification for ML, and extended our ideas for both white-box and black-box approaches. We have released our code and benchmarks as open-source tools [12, 13].

Thrust II. Learnability of Analysis Rules

Beyond my main research focus, I am excited about combining statistical and symbolic techniques to enable generalizable symbolic systems. I have worked on several problems in this space.

Inferring Rules for Cross-Architecture Taint Analysis [10]. One of the difficulties in analyzing security properties of binaries is that it requires knowledge of architecture-specific semantics. In particular, taint analysis has been one of the cornerstones of binary analysis, but suffers from oversimplified, error-prone hand-written taint rules, and even undefined specifications. As a result, taint analysis engines often over-taint or under-taint. In our work [10], we learn taint rules with minimal architectural knowledge from executions behavior on for 4 widely-used architectures (x86, ARM, MIPS, AArch64). The inferred rules have superhuman accuracy, and can be used to reliably analyze vulnerabilities in binaries.

Inferring Rules for Cross-Language Code Translation [9]. In cross-language code translation we face the issue of deterministic hand-written translation systems that do not offer user-customizability per program. This customizability is important for many application-specific goals such as enhancing performance, readability, maintainability, and so on. We propose a paradigm shift from these static rules to mostly automatic and incremental, user-guided inference of translation rules for the given program [9]. We show that such translators are feasible for translating Python to Javascript programs, on popular code interviewing benchmarks, and on additional benchmarks of more challenging and longer programs. Our approach, DuoGlot, achieves 90% translation accuracy and so it outperforms all existing translators (both handcrafted and neural-based), while it produces readable code.

Thrust III. Provable Defenses

Another dimension in my research is on provable defenses in ML security. I have worked on differentially private algorithms for learning graphs [6, 7]. Learning structural information from graphs helps in achieving better accuracy in many applications such as social networks, computer vision and traffic prediction. However, graph edges often encode sensitive information, e.g., social or financial transactions between people represented as nodes in the graph. We show that we can achieve comparable or better accuracy to non-private baselines, while still protecting the privacy of graph edges via guarantees of differential privacy.

Private Hierarchical Clustering [6]. In federated social networks, users do not have personalized recommendations or online advertising. These services require answering questions regarding the structural properties of the social graph, such as "Which users are in the community of the target user?" but users are reluctant to share their contacts with an untrusted service provider. To answer such queries in the federated setup, we present the first work to learn hierarchical cluster trees using local differential privacy. Our algorithms for computing them come with theoretical bounds on the quality of the trees learned. The private trees are of comparable quality (with at most about 10% utility loss) to those obtained from the non-private algorithms, while having reasonable privacy parameters. We show the utility of such queries by redesigning two state-of-the-art social recommendation algorithms for the federated social network setup. Our recommendation algorithms significantly outperform the baselines that do not use social contacts.

Link Private Graph Networks [7]. Graph convolutional networks are increasingly used for node classification on graphs, wherein nodes with similar features have to be given the same label. Link-stealing attacks that infer whether an edge is present in the training set of a graph convolutional network even when given black-box access to the model are a concern for using these models on sensitive graph data. Our goal is thus to preserve the privacy of edges. We propose a novel neural network architecture for training on graphs called LPGNet. The key to this construction is to use coarser graph structural information called cluster degree vectors, which can be made differentially private by adding noise. LPGNet stacks layers of multilayer perceptrons trained on node features and differentially private cluster degree vectors. LPGNet models empirically achieve better privacy-utility tradeoffs compared to the state-of-the-art existing approach, which is a mechanism for retrofitting differential privacy into conventional graph convolutional network architecture. LPGNet models have better link-stealing attack resilience than non-private graph convolutional networks but also offer better utility than models that are not trained with graph data.

Future Research

With a combination of theory and systems, I aim pursue my vision of rigorous security analysis and built-in security protections for ML systems. The two directions of research I have been working on have interesting

synergies in automatically inferring or synthesizing analysis rules for systems, including ML systems. Below are some more concrete themes of near-term research topics.

Building ML Systems with Security. There are currently practical limitations to certifying training integrity via recording the execution traces of the training process. The training is distributed, which means we need to ensure atomicity of the training logs, as well as improving the demands on storage, since currently the procedure requires saving the model parameters at every training step. In order to build a system that is deployable for these large models, we also require more scalable procedures to check the collision-resistance property of training updates. I will seek academic and industrial collaborations to develop such a system in the publishing model of Hugging Face, specifically in open-sourcing large language models for code.

Deepening Foundations for Security. There is some evidence that it is highly unlikely that a perfectly robust model with high accuracy can be learned efficiently, stemming from both empirical and theoretical work [17, 18, 16, 25]. Hence, finding robustness counterexamples might not be surprising at this point, though humans seem to be robust classfiers with non-zero error rates that learn efficiently. More importantly, does this imply that there is some intrinsic hardness in the learning process that could be useful for constructing cryptographic primitives? To this end, I am exploring the *cryptographic properties* of SGD. Our existing and on-going work suggests that there are *geometric* properties of the gradients that allow us to obtain certain hardness results intrinsically in SGD, and enable cryptographic constructions from the process of SGD. Concretely, I want to explore these results for more meaningful definitions of security properties for ML systems, and whether it can lead to different ways of learning with guarantees for practical applications.

Extending Analysis for Defenses. Concerns of security do not exist in isolation in ML systems, and most often, the root causes of security vulnerabilities in ML systems are connected. Effective defenses should aim to eliminate the root causes rather than to fix one vulnerability at the potential cost of introducing more in the ML systems. I would focus on the following three research objectives: (1) developing sound procedures to estimate the effectiveness of defenses in the presence of entangled vulnerabilities, (2) proposing new defense objectives informed by root cause analysis of the security vulnerabilities, and (3) automated secure ML systems construction and rigorous testing. I have started investigating the first two objectives in the context of the right-to-be-forgotten requests, i.e., individuals have the right to request their data to be removed from a database or training dataset. Our ongoing work identifies gaps in existing memorization metrics when used to evaluate defenses such as unlearning via gradient-based methods. We find that existing security definitions of memorization are static, not adapted to batches of unlearning requests. This causes disparate impact on other samples, damaging the utility of the ML models to other samples. We aim is to provide a more robust definition for removing training samples, informed by this observation.

Publications

[1] **Teodora Baluta**, Ivica Nikolić, Racchit Jain, Divesh Aggarwal, and Prateek Saxena. Unforgeability in stochastic gradient descent. In *CCS*, 2023. https://dl.acm.org/doi/abs/10.1145/3576915.3623093.

[2] **Teodora Baluta**, Shiqi Shen, S Hitarth, Shruti Tople, and Prateek Saxena. Membership inference attacks and generalization: A causal perspective. In *CCS*, 2022. https://arxiv.org/abs/2209.08615.

[3] Jiong Yang, Arijit Shaw, **Teodora Baluta**, Mate Soos, and Kuldeep S. Meel. Explaining SAT solving using causal reasoning. In *SAT*, 2023. https://arxiv.org/abs/2306.06294.

[4] **Teodora Baluta**, Shiqi Shen, Shweta Shinde, Kuldeep S Meel, and Prateek Saxena. Quantitative verification of neural networks and its security applications. In *CCS*, 2019. https://arxiv.org/abs/1906.10395.

[5] **Teodora Baluta**, Zheng Leong Chua, Kuldeep S Meel, and Prateek Saxena. Scalable quantitative verification for deep neural networks. In *ICSE*, 2021. https://arxiv.org/abs/2002.06864.

[6] Aashish Kolluri, **Teodora Baluta**, and Prateek Saxena. Private hierarchical clustering in federated networks. In *CCS*, 2021. https://arxiv.org/abs/2105.09057.

[7] Aashish Kolluri, **Teodora Baluta**, Bryan Hooi, and Prateek Saxena. LPGNet: Link private graph networks for node classification. In *CCS*, 2022. https://arxiv.org/abs/2205.03105.

[8] Bo Wang, **Teodora Baluta**, Aashish Kolluri, and Prateek Saxena. SynGuar: guaranteeing generalization in programming by example. In *ESEC/FSE*, 2021. https://arxiv.org/abs/2301.11220.

[9] Bo Wang, Aashish Kolluri, Ivica Nikolić, **Teodora Baluta**, and Prateek Saxena. User-customizable transpilation of scripting languages. In *OOPSLA*, 2023.

[10] Zheng Leong Chua, Yanhao Wang, **Teodora Baluta**, Prateek Saxena, Zhenkai Liang, and Purui Su. One engine to serve'em all: Inferring taint rules without architectural semantics. In *NDSS*, 2019. https://www.ndss-symposium.org/ndss-paper/one-engine-to-serve-em-all-inferring-taint-rules-without-architectural-semantics/.

Software Releases

[11] Etio. https://github.com/teobaluta/etio.

[12] NPAQ. https://github.com/teobaluta/npaq.

[13] Provero. https://github.com/teobaluta/provero.

[14] Unforgeability in SGD. https://github.com/teobaluta/unforgeability-SGD.

References

[15] Ella Creamer. Authors file a lawsuit against OpenAI for unlawfully 'ingesting' their books. https://www.theguardian.com/books/2023/jul/05/authors-file-a-lawsuit-against-openai-for-unlawfully-ingesting-their-books, 5 July 2023. Accessed: 2023-08-13.

[16] Akshay Degwekar, Preetum Nakkiran, and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. In *COLT*, 2019.

[17] Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. In *ICML*, 2019.

[18] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. In *ICLR*, 2018.

[19] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *CAV*, 2017.

[20] Jack Queen. Sarah Silverman sues Meta, OpenAI for copyright infringement. hhttps://www.reuters.com/legal/sarah-silverman-sues-meta-openai-copyright-infringement-2023-07-09/, 9 July 2023. Accessed: 2023-08-13.

[21] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *S&P*, 2017.

[22] Ilia Shumailov, Zakhar Shumaylov, Dmitry Kazhdan, Yiren Zhao, Nicolas Papernot, Murat A Erdogdu, and Ross J Anderson. Manipulating SGD with data ordering attacks. In *NeurIPS*, 2021.

[23] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. In *POPL*, 2019.

[24] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *USENIX Security*, 2022.

[25] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

[26] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11), 1984.