Mathias Lécuyer – Research Statement

The recent machine learning (ML) revolution, powered by new systems supporting massive amounts of data and advanced algorithms to analyse them, is transforming a wide range of applications. These include search, personalized recommendations, and smart assistants, as well as security and safety-critical applications, such as self-driving cars, face recognition based access control, and fraud detection. However, ML-driven applications also introduce disruptive workloads and atypical semantics that do not fit existing system paradigms. Adversaries can manipulate model predictions without explicitly breaking into host machines. Traditional protection abstractions from operating systems are ill-equipped to handle the new data access patterns of ML workloads, hence being ineffective at safeguarding the vast troves of collected data. And ML models are often too opaque for end-users to understand and trust, and too brittle for system designers security requirements. These challenges pose serious barriers to reaching ML's full potential.

My research addresses these problems by developing new software systems, abstractions, and tools that enhance the security, robustness, and transparency of ML-powered systems. For instance, I developed a new defense strategy against adversarial examples, an ML-specific class of attacks. This defense not only introduces scalable algorithms for learning more robust models, but it also enables a firewall-like security architecture, where a small model is prepended to an existing, already trained one to make it more robust. Such an architecture is common in traditional software systems but unique for ML workloads. As another example, I built new data protection abstractions better suited to ML workloads than traditional ones. These abstractions cleanly separate the historical data that is summarized in protected feature models, from the currently used data that is minimized in size and time span. This rigorous data protection approach minimizes the exposure of sensitive data to hackers and malicious employees. Finally, I developed transparency tools providing a new visibility into how ML-driven web services use end-users' data for targeting.

Common across all this work is my research approach *combining systems and theory*. I design, implement, and evaluate rigorous, theory-backed systems that are both practical and provide provable guarantees of security, protection, and statistical soundness. Three steps are involved. First, I separate problem components requiring strong guarantees from those that can do with best-effort, heuristic operation. This separation lets me design practical systems that still enforce clear semantics. Second, for each component requiring rigorous guarantees, I identify the theoretical field best suited as a foundation for a solution, either within theoretical computer science or from other domains. For example, my defense against adversarial example attacks leverages differential privacy, a seemingly unrelated theory from the privacy domain; my web transparency tools apply methods from statistics and econometrics. Third, I assemble the building blocks into a coherent architecture, implement a prototype that I subsequently release publicly, and evaluate it with multiple workloads and at scale to understand its strengths and limitations.

Following are examples from my thesis research that illustrate how I apply my methodology combining systems and theory to address challenges raised by ML workloads. I then describe new directions I want to pursue as faculty.

Thesis Research

Rigorous and Flexible Defense Against Adversarial Examples. Adversarial examples are a new class of attacks opened by ML deployments making predictions based on user-generated inputs. The adversary finds a small perturbation to an otherwise correctly classified input, which results in a misclassification and triggers a faulty action. For example, an adversary wearing makeup may fool a face recognition model into classifying him or her as someone else, enabling access to a phone or building if the model's predictions are used for authentication. Moreover, malicious perturbations can be found for almost all inputs, bringing the accuracy of ML-driven systems under attack close to zero! Numerous defenses have been proposed, but most were best-effort approaches. While they improved accuracy under contemporary attacks, subsequent attack versions erased the improvements. Recent rigorous defenses, called *certified defenses*, come with a guaranteed level of robustness against arbitrary attack implementations. However, they either do not scale to large models or are not sufficiently flexible to apply to all relevant model structures.

I therefore developed *PixelDP*, the first certified defense that both gives a guaranteed level of robustness against norm-bounded adversarial example attacks and applies to large models with arbitrary structures [2]. PixelDP builds on a novel connection I formally established between differential privacy, a theory from the privacy domain, and the definition of robustness against norm-bounded adversarial perturbations. Briefly, the expected output of a differentially private mechanism can be shown to be bounded under small changes in its input. I use this fact to assess whether any adversarial attack bellow a given size can change the prediction of a PixelDP model on a given input. If it cannot, the prediction is certifiably robust against attacks up to that size. This *robustness certificate* for an individual prediction can be used in two ways. First, a building authentication system could use it to decide whether a prediction is robust enough to rely on a face recognition model's prediction and make an automatic decision, or whether a human should be consulted. Second, a model designer can use robustness certificates for predictions on a test set to assess a lower bound of the accuracy under attack, which holds for any future attack.

In addition to providing robustness certificates, PixelDP introduces a *firewall security architecture* that is brand-new for ML workloads. A core property of a differentially private mechanism, called the post-processing guarantee, is that any computation based on its output remains differentially private. To defend any predictive model, or set of models, that are running on a given dataset, it is possible to develop a PixelDP autoencoder, a deep neural network (DNN) that learns the identity function. This autoencoder is then stacked in front of the undefended models and, through the post-processing guarantee, PixelDP's robustness certificates apply to the predictions of the undefended models. The autoencoder thus acts as a firewall between attackers and undefended models, separating the concern of designing predictive models from that of defending them. This separation lets our defense scale to large models with arbitrary structures while enforcing clear defense semantics.

Using the PixelDP firewall architecture, I produced the very first version of Google's Inception DNN for ImageNet that has non-trivial guaranteed accuracy under arbitrary, norm-bounded adversarial example attacks. This network is orders of magnitude larger than what previous certified defenses handled. The guaranteed accuracy is reasonable for small attacks (60% for 2-norm attacks of size 0.1), but it is much lower for larger attacks (15% for 2-norm attacks of size 0.5). I am now developing new methods to increase guaranteed accuracy by reducing the impact of the differentially private noise added during training and prediction.

New Protection Abstractions for ML Ecosystems. A second looming security threat posed by ML workloads is the vast troves of personal information that organizations routinely collect to fuel them. Improperly protected, this data can be grabbed by hackers or unethical employees – and frequently is, as countless data breach reports demonstrate. I believe that the core reason why organizations are unable to properly protect the data within their ML ecosystems is the lack of protection abstractions suitable for these workloads. Traditional protection abstractions, such as files or database tables, were designed for functionality with well-specified data requirements. For example, an in-app product purchase service may need access to credit card information but not to social graph or account information. In contrast, ML applications have blurry data requirements: data collected to improve a news recommendation service may also be relevant for ad targeting, a smart assistant, or fraud detection. Because data requirements are unclear many organizations adopt wide-access policies, such as the "data lake" policy: all user information collected from the company's various products is integrated into one central repository, to which all employees and services get access.

I developed *protected feature models*, a new data protection abstraction tailored to ML workloads that creates a clean separation between data needed in raw form, and data that can be replaced by *summaries*, or *features*. Intuitively features such as movie embeddings, already learned over historical data to represent relevant characteristics, may be as beneficial, if not more, to recommendation algorithms than raw histories of viewing and browsing activity. Leveraging this fact, feature models can be used to summarize historical data, and to enhance a small amount of current raw data to train ML models. To enable sharing and provide clear protection semantics for historical data, these feature models are protected using differential privacy. Using this abstraction, organizations can adopt a much more principled approach to data protection, such as enabling raw data access on a needs basis only.

As a concrete example, I developed Pyramid based on *count featurization*, a feature model commonly used (non privately) in ad targeting and personalization [3]. Count featurization summarizes past data with a set of count tables keeping the number of times a given feature value (e.g., a $userId$) was observed with an outcome of interest (e.g., liking a movie). These count tables are then used to replace the features of raw examples with the probability of the example's label, conditioned on the feature values (e.g., $userId$ becomes $P(rating|userId)$) reducing the amount of raw data needed for training. My study shows the potential of count fearization to be used as a protected unit of access to historical data. The historical counts are scalably stored in a differentially private count sketch I developed, and I show that predictive models leveraging these private counts need to access just 1% of the raw data to reach similar accuracy compared to models trained on the full raw data. These results stem from an evaluation on three ML workloads, including from a production news recommendation engine. My current work generalizes the abstraction to arbitrary feature models, and develops a differentially private model management system that continuously trains, enforces the privacy of, and makes available protected feature models on streams of user information.

Transparency Tools for Targeting Services. Today's web services are black boxes giving little visibility into the data they collect and little control over how this data is used. Are people being targeted because their emails or browsing patterns suggest that they might be vulnerable (e.g., depressed, in financial difficulty)? Are such inferences being used to place potentially damaging products, such as risky mortgage deals?

To answer these questions, I built scalable tools to detect the causal effect of user data on targeting. The key insight in my first system, XRay [6], is to infer targeting by correlating user inputs (such as searches, emails, or locations) to service outputs (such as ads, recommendations, or prices) based on observations obtained from user profiles populated with different subsets of the inputs. A major contribution in XRay is its design that scales to a large number of inputs, enabling large-scale studies of targeting. Indeed, XRay's algorithms need a number of profiles that grows only logarithmically with the number of inputs to track, which I showed both theoretically and experimentally. However, the lack of tests of statistical validity for XRay's predictions made conclusions hard to interpret. My second system, Sunlight [5], offers rigorous statistical justification for its inferences. Two key insights allow Sunlight to perform causal targeting detection at scale. First, Sunlight decouples the formulation and the statistical testing of targeting hypotheses. This allows the use of scalable ML models to formulate hypotheses on a training set, and the assessment of statistical significance on a testing set. Second, statistical methods to control for multiple hypothesis testing heavily penalize incorrect assumptions. Sunlight thus favors high-precision algorithms to generate hypotheses which, counter-intuitively, results in better recall after p-value correction. As important, Sunlight shows that the correlations I detect have a causal interpretation because input values are randomly assigned for each user profile.

I used my tools to run large-scale studies of online ad targeting. Among other findings, I identified strong evidence of targeting on sensitive personal information – such as religion and sexual orientation – and sensitive financial information that should not be targeted according to Google's own privacy FAQ. This work also spawned interest from the Federal Trade Commission. Next, I plan to leverage recent advances in causal inference from observational data to formulate and test causal targeting hypotheses on real user accounts. As results from randomized experiments are valid only for the studied population, this approach would be more broadly applicable.

Counterfactual Evaluation for Systems. ML and statistics also present new opportunities to better solve traditional systems challenges. For example, deploying efficient load balancing or resource allocation policies often requires answering what would have happened with a different algorithm. Many existing methodologies for answering this question, such as simulation or trace-driven models, exhibit bias and can require heavy, system-specific engineering. Other methods, like A/B testing, can be costly, especially under an increased amount of automated policy exploration.

I developed Sayer [4, 1], a generic tool for counterfactual evaluation in systems that enables rigorous evaluation of arbitrary policies without actually running them. Sayer leverages theory from a domain in statistics called counterfactual evaluation; it incorporates techniques from reinforcement learning to account for the long-term impact that current policy decisions may have on future performance. Intuitively, Sayer uses randomization inherent in typical systems, or introduced through light changes, to estimate the long-term performance of policies that are not directly evaluated. In addition to using counterfactual evaluation theory and reinforcement learning methods, I also devised novel techniques to make Sayer practical and efficient in production systems, leveraging the fact that systems with conservative defaults (e.g., in timeouts) naturally offer more information about counterfactual outcomes, or that large distributed systems often have natural baselines (e.g., a similar machine) that can help reduce the variance of counterfactual estimates.

Sayer provides accurate counterfactual evaluations on three applications that I evaluated: a replica selector prototype that routes requests to replicas in a data store, and data from two production applications, a health monitor that reboots unresponsive machines and a geo-distributed proxy that routes user requests to cloud services.

Future Directions

Going forward, I plan to continue applying my methodology combining systems and theory to other problems that are impactful, have a strong practical component, and require systems-level thinking. This includes further addressing challenges that hinder important applications of ML ecosystems, as well as opportunities for better understanding and optimizing system performance.

Improving System Understanding and Optimization through Causal Reasoning. Modern systems consist of numerous components, including storage, caches, application-level services, or ML components. The current approach to handle such ecosystems is to isolate components with stable APIs and service level agreements. While generally good, this approach can miss performance interdependencies. Indeed, changing one component often involves trade-offs: e.g., measures to reduce tail latency can slightly increase average latency, or a new caching policy may improve overall performance but degrade performance on some requests. These trade-offs can affect other components in unpredictable and cascading ways. Compounding this problem is a recent push to deploy online, data-driven optimization into systems. There, small changes can disproportionately impact learning and predictions, and ripple through the entire ecosystem, a behavior often described as "Changing Anything Changes Everything." I believe that these problems are symptomatic of a *lack of understanding of the causal impact* of each component on the ecosystem.

To address this problem and give more visibility into systems' performances, I plan to leverage structural causal models, a well developed theory with intuitive tools to hypothesize causal relations, and algorithmic approaches to evaluate these hypotheses and learn causal models based on them. These techniques rely on extensive measurements and must often run controlled experiments, which will require a tight integration with systems' instrumentation and experimentation components. Such causal models can then help forecast the impact of changes in individual components and perform root-cause analysis of performance issues. I believe that a causal model can also serve as the basis for optimization, coupled with techniques such as model-based reinforcement learning, which optimizes decisions over long time horizons and maps particularly well to systems optimization. To offer clear semantics, the main requirement will be the development of optimization methods with convergence, and hopefully performance, guarantees.

Both approaches can leverage typical systems characteristics. Levels of indirection are natural places to perform clean interventions to identify causal interactions. Some systems are amenable to safe exploration, trading off resources for information: for instance, to know how a different load balancing decision would perform without decreasing latency, one can imagine querying both alternative choices. Finally, systems rely on common patterns, such as caching, for which I plan to develop efficient featurizations to include in causal and reinforcement learning algorithms.

Access Control for Machine Learning Ecosystems. Traditional access control mechanisms rely on well-defined data ownership and siloed access. ML workloads exhibit new access patterns defying these assumptions. First, model training, evaluation, and trouble-shooting access all available data. Second, trained models are broadly exposed: they serve predictions to all users, are shipped on mobile devices, and are reused across teams. Third, models and their predictions leak information about the training data, allowing membership queries or reconstruction attacks.

I plan to design access control mechanisms that complement traditional ones to support ML workloads. To prevent data leakage from the models, differential privacy has been shown to be necessary and sufficient. For guarantees to intuitively map to traditional access control semantics, protection should be at the user granularity and enforced over the entire life cycle of ML ecosystems, including model development, training, evaluation, periodic retraining, and troubleshooting. Supporting such semantics raises multiple challenges. For instance, a large number of models and summary statistics need to be supported with frequent access to historical data, requiring new privacy budget management and optimization techniques. I believe that developing differentially private generative models is a promising approach. Such models could enable efficient differential privacy mechanisms – such as multiplicative weights – to scale to high dimensional data, and serve as surrogates for historical data with exhausted privacy budget. Another challenge will be to support personalization, an important use case that is inherently user specific, and thus cannot be protected at the user level. I will develop techniques to systematically split models into a global part, generally accessible but with user-level protection, and a local part, without such protection but accessible only to relevant users. This approach cleanly separates the state following traditional access patterns from the state following ML ones.

Principled access control with clear guarantees will be an important enabler of new use cases. This includes use cases requiring multiple companies to co-train models without sharing their data, such as personalized medical diagnosis, fraud detection, and epidemiology.

References

[1] M. Lécuyer, M. Nanavati, J. Jiang, A. Mobasher, A. Savelieva, S. Sen, A. Sharma, and A. Slivkins. Sayer: Counterfactual evaluation of systems. *Under Submission.*

[2] M. Lécuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *Proc. of IEEE Symposium on Security and Privacy (Oakland)*, 2019.

[3] M. Lécuyer, R. Spahn, R. Geambasu, T.-K. Huang, and S. Sen. Pyramid: Enhancing selectivity in big data protection with count featurization. In *Proc. of IEEE Symposium on Security and Privacy (Oakland)*, 2017.

[4] M. Lécuyer, J. Lockerman, L. Nelson, S. Sen, A. Sharma, and A. Slivkins. Harvesting randomness to optimize distributed systems. In *Proc. of the Workshop on Hot Topics in Networks (HotNets)*, 2017.

[5] M. Lécuyer, R. Spahn, Y. Spiliopoulos, A. Chaintreau, R. Geambasu, and D. Hsu. Sunlight: Fine-grained targeting detection at scale with statistical confidence. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*, 2015.

[6] M. Lécuyer, G. Ducoffe, F. Lan, A. Papancea, T. Petsios, R. Spahn, A. Chaintreau, and R. Geambasu. XRay: Increasing the web's transparency with differential correlation. In *Proc. of USENIX Security*, 2014.