Research Statement – A. Pinar Ozisik

The foundation of my research focuses on the robustness of models that incorporate randomness, either naturally or artificially. Specifically, my goal is to analyze the robustness of such models, and create robust systems with practical applications.

Robustness is an overloaded term with different definitions in statistics, artificial intelligence (AI), and systems. For estimation problems using data, robust statistics aim to minimize errors due to outliers or minor deviations from distributional assumptions [11]. AI expanded this definition to define robust models as those that can withstand uncertainty introduced during data generation or at any stage in the data processing pipeline. In the context of systems security, robustness can refer to computer programs that are designed to offer protection against an attacker model. Recently, adversarial machine learning has become an emerging subfield of AI, and studies learning in the presence of an attacker.

I have worked on one or more of the following items in reinforcement learning (RL) and blockchain systems: 1) what a system must be robust against; 2) how we can ensure robustness; and 3) how robustness can be measured. In my research so far, due to the random nature of the models at hand, I have leveraged an important tool: concentration inequalities (CIs). Also known as tail inequalities or tail bounds, CIs bound the probability that a random variable deviates from some value, typically its expectation or median.

1 Robustness in Safe/Seldonian RL

RL algorithms have been proposed for many high-risk applications, such as improving type 1 diabetes and sepsis treatments [12, 23]. One type of safe RL algorithm [21, 22], also referred to as safe and/or Seldonian RL [23], enables these high-risk applications by ensuring safety, which provides high-confidence guarantees that the application will not cause undesirable behavior like increasing the frequency of dangerous patient outcomes. A specific component, called the safety test, is largely responsible for ensuring safety. This test can output a new policy—the mechanism for selecting actions within an agent—using training data collected from a baseline policy. It uses CIs, most recently the Chernoff-Hoeffding (CH) [10] bound, to lower bound the performance of the new policy. If this lower bound is higher than the performance of the baseline policy, the new policy is outputted. Our analysis of the safety test answers to what extent the safety guarantee holds when some of the samples in the training data are not random.

In real world applications, such non-random samples caused by anomalies are common when data comes from a pipeline that includes human interactions, natural language processing, device malfunctions, etc. For e.g., the recent application of RL to sepsis treatment in the intensive care unit (ICU) used training data generated from hand-written doctors’ notes [12]. In a high-stress ICU environment, missing records and poorly written notes are difficult to automatically parse [1].

Our formulation of the problem represents anomalies as samples artificially created by a malicious attacker. The incorporation of an adversary provides a worst-case analysis to understand the robustness of safety tests to anomalies in data; because any algorithm robust to an adversary is also robust to non-adversarial anomalies in data. First, we introduce a new measure of security to quantify the susceptibility of training data to corruptions. Second, we show that a couple of safe RL methods are extremely sensitive to even a few data corruptions, completely breaking the probability bounds guaranteed by CIs. Then we introduce a new algorithm, called Panacea [20], which provides a user-specified level of robustness when the number of non-random samples in the dataset is upper bounded. We also demonstrate Panacea’s usage in practice on a diabetes treatment simulation.
2 Robust Blockchain Systems

The blockchain [15] concept was originally designed to be the backbone of the Bitcoin distributed cryptographic currency. Over the last decade, Bitcoin has been adopted more widely for e-commerce than any previous digital currency. Therefore, just like any credit or debit card, Bitcoin must be robust against theft and scams.

In Bitcoin, transactions (txns), similar to those generated by financial institutions, are recorded on a public ledger called a blockchain. Txns are broadcast by users on Bitcoin’s peer-to-peer network. A subset of users, known as miners, independently agglomerate a set of txns into a candidate block and attempt to solve a predefined cryptographic puzzle as proof of work. The first miner to solve the problem broadcasts his solution to the network, and is able to add the block to the ever-growing blockchain as a child of the prior block. The miners then start over, using the newly appended blockchain and the set of remaining txns. When txns appear in a block, they are confirmed, and each subsequent block provides additional confirmation.

2.1 Estimating Mining Power

Bitcoin’s cryptographic puzzle consists of the miners’ applying a hash algorithm [9], which takes as input key identifiers of a block and a random number, called a nonce. If the resulting value, i.e., the hash, is not less than the known target that is set by the network, then a new nonce is selected to compute a new hash. This process repeats until some miner finds a solution. Each time a hash is computed, a miner samples a value from a discrete uniform distribution; hence, the likelihood of discovering a block increases with a miner’s hash rate or mining power, i.e., the number of the hashes computed per second. With the published block and nonce, any miner can verify that the resulting hash is less than the target.

A double-spend attack [15] occurs when an evil miner, acting as a customer, tries to steal goods from a merchant by purposely creating a fork on the chain. He releases a txn, $x$, that transfers money to the merchant and waits for the blockchain, including $x$, to grow; but then he attempts to rewrite the chain, excluding $x$, once he receives the goods. The evil miner can succeed only if he has enough mining power: all remaining miners will switch over to the new chain if it is longer. Therefore, the status of txns and blocks is not immutable—a fork of blocks supported by greater mining power can emerge at any time. To create a transparent system that thwarts double-spend attacks, we quantify the consensus of a blockchain towards its blocks and the txns they hold.

First, we design a method of accurate hash rate estimation based on compact status reports [18] issued by miners. These reports require each miner to periodically report the block and nonce resulting in the minimum hash value since the last block broadcast on the chain or their last report. Second, we show how hash rates can be estimated from only blocks that are published to the blockchain. For this estimator, we treat the entire network as a single miner and a block as a status report that can only be observed at certain intervals. This approach is less accurate than status reports because we have less information about each miner’s progress. To find a lower and upper bound on the number of hashes computed per miner, we use a well-known CI, the Chernoff bound [6], for our first method and calculate empirical bounds based on bootstrapping [5, 7] for our second method.
2.2 Minimizing Communication Cost

In Bitcoin, newly created txns get propagated in the network, and often times a user already possesses most or all of the txns when they appear in a block. This realization allows us to minimize the size of blocks. This minimization helps us achieve robustness against forks: smaller encodings of blocks allow for miners to receive blocks more rapidly, enabling them to quickly discard current work and start creating new blocks. Hence, to avoid forks, we answer the question of how to quickly relay information, i.e., a block of txns, from a sender to a receiver if the receiver already possesses all or some of the txns.

Our proposed solution, instead of sending all txns directly, uses a novel combination of two probabilistic data structures: a Bloom filter [4] and an Invertible Bloom Lookup Table (IBLT) [8]. These data structures use hash functions to compactly represent a set of items. The widely held assumption that a good hash function appears random [14] means that these data structures can fail. This failure can occur when a receiver, who obtains a Bloom filter or IBLT representation of a set, tests whether an item at hand is a part of the set represented by the data structure. If the membership test for that item returns negative when, in fact, it is actually a part of the set, errors occur. This error rate is tunable with a tradeoff: a smaller error rate means that the data structure becomes bigger. Our protocol, Graphene [17, 19], minimizes the total size of both data structures sent between sender and receiver given a low error rate. We use Chernoff bounds to set the size of our probabilistic data structures in order to guarantee Graphene’s performance with high probability.

3 Future Directions

Three years ago, I created a curriculum for first year students and taught a course, called Ethical Issues in Technology, where we discussed the ethical implications of computing. This course fueled my already existent interest in social justice, shifting my focus to the creation and analysis of algorithms with social implications. Recently, there has been abundant work on identifying bias and discrimination in AI systems, and creating robust models [2, 3, 16]. Safe RL has also been proposed for creating fair algorithms that reduce discriminative behavior in intelligent tutoring systems and loan approvals [13].

For my work on blockchains, robustness implies mathematical safety, which requires CIs to ensure that training data does not include too many outliers and represents the true underlying distribution of the data. For safe RL, on top of this already existent requirement, I argue that training data must also be secure, i.e., robust against data points that do not come from the true underlying distribution. However, note that these safety and security requirements arise from domain-specific needs: cryptocurrencies must be robust against scams and attacks, and inferences made from medical data must be robust against anomalies. Overall, in both areas of my work, robustness has included concepts that are both generalizable and domain-specific, and has become interchangeable with safety and security.

In addition to developing domain-specific and generalizable mathematical definitions of robustness, in the future, I would also like to work with data, thinking about its limitations. By developing metrics that evaluate the “quality” of training data, which often reflect systemic biases and human error, I would like to create AI algorithms that can be applied to problems with real world impact.
References


