Adversarial Teaching Approach to Cybersecurity: A Mathematical Model Explains Why It Works Well

Christian Servin, Olga Kosheleva, and Vladik Kreinovich

Abstract Teaching cybersecurity means teaching all possible ways how software can be attacked – and how to fight such attacks. From the usual pedagogical viewpoint, a natural idea seems to be to teach all these ways one by one. Surprisingly, a completely different approach works even better: when the class is divided into sparring mini-teams that try their best to attack each other and defend from each other. In spite of the lack of thoroughness, this approach generates good specialists – but why? In this paper, by analyzing a simple mathematical model of this situation, we explain why this approach work – and, moreover, we show that it is optimal in some reasonable sense.

1 Formulation of the Problem

Cybersecurity is important. In the modern world, everything relies on computers – even more so with the current COVID'19 pandemic. Computers run our communications, computers control our utilities, computers largely control our planes, cars, etc. For our civilization to continue to function, it is important to protect all these computer systems from malicious attacks.

Teaching cybersecurity is important. Whatever automatic tools we place in to prevent cyber-attacks, smart adversaries learn to overcome. The only way to maintain cybersecurity is to train a large corpus of specialists who would protect us from all the newly appearing threats.

Christian Servin
Computer Science and Information Technology Systems Department
El Paso Community College (EPCC), 919 Hunter Dr., El Paso, TX 79915-1908, USA
cservin1@epcc.edu

Olga Kosheleva and Vladik Kreinovich
University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA
e-mail: olgak@utep.edu, vladik@utep.edu

Traditional way to teaching – in particular, to teaching cybersecurity. The usual way of teaching any material is to present, to the students, the needed information and skills. With respect to cybersecurity, this means explaining, to the students, the main types of cyber-attacks and the main ways to defend against these attacks. After that, we can let the students show their creativity, but usually, teaching the basics is a must.

Adversarial teaching: a successful alternative approach. Interesting, lately, a different approach has been very popular and very successful, in which, instead of teaching students the usual way, the instructor divides the class into one or more pairs of sparring mini-teams. In each pair, the teams interchangingly try to attack each other and to defend their team from a partner's attacks; see, e.g., [2, 3].

This works, but why? The above strategy works, which is somewhat surprising. In the absence of a thorough coverage of all possible topics, one would expect gaps in the ability of students who have been taught this way – but there are usually no such gaps. So, the first question is: why this approach works?

A natural second question: is this approach close to optimal or we can drastically further improve it – and if yes, how?

What we do in this paper. In this paper, we answer both questions: we explain why the adversarial teaching approach works, and we show that this approach is – in some reasonable sense – optimal.

Comment. This paper is a revised and extended version of our shorter conference paper [8].

2 Analysis of the Problem

Similar approach works in design. For teaching, this approach may be somewhat new, but a similar approach works in military engineering. For example, according to [7], new fighter planes are designed as follows (by using a program that simulates dogfights between different planes):

The first stage is natural: we consider several possible designs, and for each of them, we simulate how this design will perform in a possible confrontation with the fighter planes used by the existing adversaries. We continue doing this until we find a design that can beat all the possible opponents.

At first glance, this may seem to be sufficient, but, on second thought, it is not: it is not enough for a future plane to be better that what the opponent has now, we need to have a design that will be better than what the opponent will have in the future.

To design such a plane, we perform the second stage of the design process: namely, we design a plane that will be better than not only the current planes, but also better than our first-stage design.

Then, we design a plane that will be better than the second-stage design, etc. At the end, we get an almost perfect future plane – and this is what is then implemented and tested.

What can we conclude from this fact. The fact that a similar idea works successfully in such completely different application areas as teaching cybersecurity and designing fighter planes makes us confident that these successes are not due to any specific features of these areas, they are due to the general structure of this approach. Let us therefore describe a simple mathematical model that would capture this structure.

Comment. We are not specialists in plane design, but, as educators, we are clearly more familiar with educational applications. So, while the model will be potentially general, we will illustrate it on the example of teaching – namely, on the example of teaching cybersecurity.

Towards a model. We want the students to be able to handle all possible attack situations. Of course, different situations are all somewhat different, but ideally, what we want is to make sure that whatever new situation surfaces, the students should have some experience successfully fighting a similar attack in the past, experience that would help the student fight the new attack as well.

In mathematics, a natural way to describe similarity is by assuming that there is a some metric $d(a, b)$ on the set S of possible situations, a metric that describes to what extent situations a and b are different from each other – or similar to each other. The smaller the distance $d(a, b)$, the more similar are situations a and b.

In these terms, "similar" means that the distance $d(a, b)$ is smaller than or equal to some small threshold value $\varepsilon > 0$.

Therefore, we arrive at the following model.

3 Resulting Model: What We Have and What We Want

Resulting model. The above requirement can be formulated as follows. We have a set S of possible situations. On this set, we have a metric $d(a, b)$.

We want the student to experience situations $s_1, \ldots, s_n$ such that every situation s from the set S is ε-close to one of such situations.

Comments.

- In mathematics, such a set is known as an *ε-net*; see, e.g., [5, 6].
- The exact value of the threshold is determined by our resources: the smaller ε, the better – but a drastic decrease in ε would mean a drastic increase in situations experienced during teaching, and the teaching time is limited.

How do we compare quality of different teaching schemes. In view of the previous comment, once we fix $\varepsilon > 0$, a natural measure of quality is the number of experiences situations n: the smaller n, the faster we can train.

Alternatively, we can fix n – and thus, the training time – and try to find the situations $s_1, \ldots, s_n$ that lead to the smallest possible ε.

Comment. For each metric space, the smallest possible number of elements in an ε-net is called *ε-entropy*; to be more precise, usually the logarithm of this smallest number is called the ε-entropy; see, e.g., [5, 6].

The corresponding optimization problem is known to be NP-hard. It is known that problem of finding the smallest ε-net is, in general, NP-hard; see, e.g., [1]. This means, crudely speaking, that unless P = NP (which most computer scientists believe to be false), no feasible algorithm is possible that would always find the optimal ε-net.

4 Adversarial Teaching Reformulated in Terms of the Model: Formulation and Analysis

Let us reformulate adversarial teaching in these terms. The first team starts with some attack situation s_1. Then, the sparring team learns how to defend against this attack. So, next time, the attacking team will try to find a new way of attacking that has the most chances of success – i.e., the situation s_2 which is as far away from the original situation s_1 as possible:

$$d(s_2, s_1) = \max_{s \in S} d(s, s_1). \tag{1}$$

Once the sparring team learns how to deal with the situation s_2 as well, the next attacking situation s_3 will be as far away from both s_1 and s_2 as possible, i.e., for which the distance

$$d(s, \{s_1, s_2\}) \stackrel{\text{def}}{=} \min(d(s, s_1), d(s, s_2)) \tag{2}$$

is the smallest possible:

$$\min(d(s_3, s_1), d(s_3, s_2)) = \max_{s \in S} \left(\min(d(s, s_1), d(s, s_2))\right). \tag{3}$$

In general, once we have experiences the situations $s_1, \ldots, s_k$, we select the next situation s_{k+1} for which

$$\min(d(s_k, s_1), \ldots, d(s_k, s_{k-1})) = \max_{s \in S} \left(\min(d(s, s_1), \ldots, d(s, s_{k-1}))\right). \tag{4}$$

We continue until this way, we can find a situation which is different from all the previous ones – i.e., for which $d(s_k, s_i) > \varepsilon$ for all $i < k$.

When this is no longer possible, i.e., when we have

$$\max_{s \in S} \left(\min(d(s, s_1), \ldots, d(s, s_n))\right) \le \varepsilon, \tag{5}$$

we stop.

This strategy works: a proof. There are only finitely many possible situation – e.g., since each situation has to be described in a reasonable time and thus, contain a reasonable number of characters N to describe, and for each N and for each set of possible symbols, we have a finite number of strings of this (or smaller) length.

At each iteration, we generate a situation which different from all the previous once. Thus, eventually, the above process will stop.

Let us show that the resulting set of situations $s_1, \ldots, s_n$ indeed satisfies the desired property – that every situation $s \in S$ is ε-close to one of the situations s_i.

Indeed, the formula (5) means that for every situation $s \in S$, we have

$$\min(d(s, s_1), \ldots, d(s, s_n)) \le \varepsilon, \tag{6}$$

which, in its turn, means that for every situation $s \in S$, there exists a situation i for which $d(s, s_i) \le \varepsilon$.

This strategy is asymptotically optimal: formulation. Let n be the number of situations that the students have experienced by following this strategy. As we have mentioned earlier, because the strategy is feasible and the problem is NP-hard, we cannot expect that for this number n, the threshold ε is optimal. It is thus possible that, in principle, with the same number n, we can reach a smaller value ε'.

What we *can* prove, however, is that this decrease cannot be too drastic: namely, we will prove that even for one fewer $(n-1)$ situation, the corresponding optimal value ε' is at best twice smaller, i.e., that $\varepsilon' \ge \varepsilon/2$.

This strategy is asymptotically optimal: a proof. Let us prove this optimality result by contradiction.

Indeed, by our construction, we have

$$d(s_i, s_j) \ge \varepsilon \tag{7}$$

for all $i \ne j$. Suppose that we have a ε'-net $s'_1, \ldots, s'_{n-1}$. By definition a ε'-net, each element s_i is ε'-close to some element $s'_{e(i)}$.

For $i \ne j$, we cannot have $e(i) = e(j)$: otherwise, we will have

$$d(s_i, s_j) \le d(s_i, s_{e_i}) + d(s_j, s_{e_i}) \le 2\varepsilon' < \varepsilon, \tag{8}$$

which contradicts (7). Thus, to each of the n elements s_i, we assign a different element s'_j – but this is impossible, since we assumed that we only have $n-1$ elements e'_j.

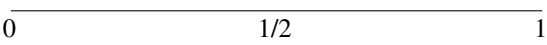
The optimality is thus proven.

5 Graphical illustration

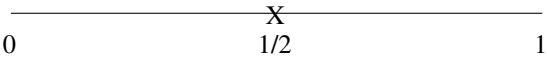
To make it easier to understand, let us give two simple geometric illustrations of the above idea. These examples are similar to examples provided in [4] for a different

application of a similar idea – to selecting benchmarks for testing different numerical algorithms.

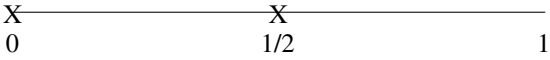
1D example. Let us start with the simplest example of a metric space S – namely, the interval $[0,1]$:



It is reasonable to select the midpoint 1/2 as s_1:

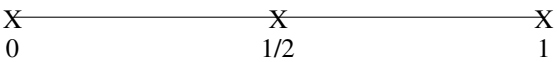


There are two points that are the farthest from s_1: the left endpoint 0 and the right endpoint 1. Without losing generality, let us select $s_2 = 0$:

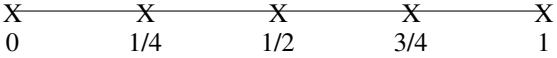


Now, $s_3 = 1$ is the point with the largest value of

$$d(s, \{s_1, s_2\}) = \min(d(s, s_1), d(s, s_2)) :$$



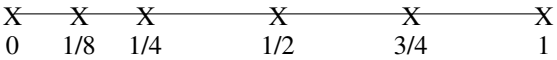
At this stage, the midpoints between 0 and 1/2 and between 1/2 and 1 are the farthest from the set $\{s_1, s_2, s_3\} = \{0, 1/2, 1\}$, so, after two stages, we add them both:



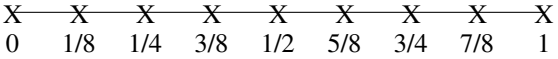
Now, the largest possible value of

$$d(s, \{s_1, s_2, s_3, s_4, s_5\}) = d(s, \{0, 1/4, 1/2, 3/4, 1\})$$

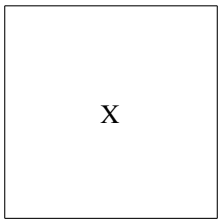
is 1/8. So, at the next stage, we add one of the points in between the existing ones, e.g., the first one (1/8):



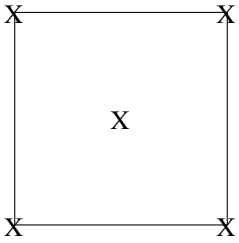
After three more stages, we add all midpoints, so we arrive at the following configuration:



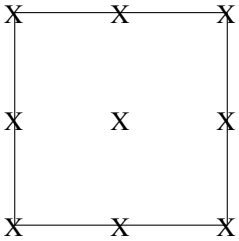
2D example: square. For a unit square, we get a similar situation. First, let us pick the midpoint as s_1:



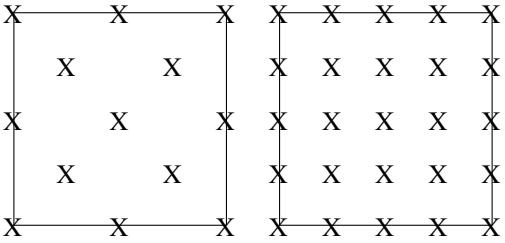
Then, the next four selections s_i are the vertices:



After this, the next four selected points s_i are he midpoints of the four edges:



Here, we have, in effect, four sub-squares. On the next stage, the same procedure is repeated for each sub-square, etc.



6 Competition Is Good: General Idea

Competition is good: a conclusion. The above text shows that a healthy and productive competition between students is good for education.

How to best arrange the competition? In the previous text, we simply explained why the competition-related arrangement works well. Now that we know that it works well, a natural next question is: how to best arrange this competition so as to make the education most effective?

Is it better to divide students into two competing teams – the case we analyzed in the previous text – or is it better to divide them into three or more teams? To answer this question, let us formulate it in mathematical terms – by constructing yet another (simple) mathematical model of the corresponding phenomenon.

It is important to take student feelings into account. By using our first mathematical model, we have showed that a competition between two teams leads to successful learning, and, moreover, that this way of learning is asymptotically optimal. Similar results can be proven for the case when we divide the students into three or more competing teams. So, if we only take into account teaching effectiveness, there is no difference between dividing students into two, three, or more teams.

Does this mean that it does not matter into how many competing teams we divide the students? Not really. As all teachers know, effectiveness is not the only criterion that we use when selecting a teaching strategy. We also need to make sure that students feel good about this way of teaching. For example, if we force the students to study 14 hours a day, they will probably learn more – but they will feel stressed and upset.

From the viewpoint of student feelings, which way of dividing the students into teams works better? Let us describe this in precise mathematical terms.

Towards a mathematical model of student feelings. If we divide students into competing teams, then naturally students have:

- positive feeling towards their teammates, but
- somewhat negative feeling towards their competitors – i.e., students from competing teams.

Vice versa, a student gets:

- positive feedback from other students from his/her team and
- negative feedback feeling from students from competing teams.

Since a priori, we do not have any reason to believe that some of these feedbacks are more important than others, it is reasonable to view all these positive and negative feedbacks as equally important. From this viewpoint, for each student:

- each member of his/her team (and this student himself) brings to this student 1 positive feedback unit, and
- each member of each of the competing teams bring to this student 1 negative feedback unit.

If we add up all these units, we conclude that as a natural measure of the student's feelings we can take the difference between the number of the students in this student's team (who bring positive feedback) and the number of students in all competing teams (who bring negative feedback).

Let us describe this in precise terms.

Definitions and the first result.

Definition 1. *Let a positive integer N be given. This integer will be called the* number of students in the class. *The set $S = \{1, \ldots, N\}$ will be called the* set of students*; integers from this set will be called* students.

- *By a* division *into competing teams, we mean a tuple $D = (T_1, \ldots, T_k)$ of non-empty subsets of the set of all students such that the sets T_i are disjoint ($T_j \cap T_{j'} = \emptyset$ when $j \neq j'$), and their union is equal to the whole set S: $T_1 \cup \ldots \cup T_k = S$.*
- *The sets T_j are called* teams.
- *For each student i, let us denote, by $D(i)$, the number j of the team that contains this student: $i \in T_j = T_{D(i)}$.*
- *For each division D and for each student i, by the* feeling $f_i(D)$ *of the i-th student, we mean the value*

$$f_i(D) = |T_{D(i)}| - (|T_1| + \ldots + |T_{D(i)-1}| + |T_{D(i)+1}| + \ldots + |T_k|), \qquad (9)$$

  where $|T|$ denotes the number of elements in a set T.
- *We say that a division D' is* better *than a division D if for all students i, we have $f_i(D) \leq f_i(D')$, and for some students i, we have $f_i(D) < f_i(D')$.*
- *We say that the division D is* optimal *if no other division is better than D.*

Proposition 1. *A division D is optimal if and only if it divides students into two teams $k = 2$.*

Comment. This result explains that, from the viewpoint of student feelings, the best way to organize competition is to divide students into two competing teams.

Proof.

1°. Let us first prove that if any division $D = (T_1, T_2, T_3, \ldots, T_k)$ that divides students into $k > 2$ teams is not optimal.

To prove this, we will show that the division $D' \stackrel{\text{def}}{=} (T_1 \cup T_2, T_3, \ldots, T_k)$ is better than D. Indeed, for all students from the teams T_3 through T_k, the feeling does not change when we go from D to D', but for students from the teams T_1 and T_2 the feelings become better:

- students from team T_1 used to get negative feelings from students from the team T_2 (term $-|T_2|$ in $f_i(D)$), now this feeling is positive, while all other feelings remain unchanged;
- similarly, students from team T_2 used to get negative feelings from students from the team T_1 (term $-|T_1|$ in $f_i(D)$), now this feeling is positive, while all other feelings remain unchanged.

2°. In Part 1, we showed that only divisions into two teams can be optimal. To complete the proof, let us show that every division into two teams is optimal.

To prove this, we need to show that no division into two teams can be better that another division into two teams.

Indeed, let us consider a division $D = (T_1, T_2)$ into two teams. Without losing generality, let us assume that the size of the first team is smaller than or equal to the size of the second team. So, this division divides students into a team $n \stackrel{\text{def}}{=} |T_1|$ of size $n \le N/2$ and the remaining team T_2 of size $N - n$. In this division:

- students from the first team get the feeling $f_i(D) = n - (N - n) = 2n - N \le 0$, while
- students from the second team get the feeling $f_i(D) = (N - n) - n = N - 2n \ge 0$.

Let us consider three possible cases.

2.1°. If two divisions D and D' have the same first-team size n, then they have the same feeling for the same number of students, so the sum of all the feelings is the same – but if D' was better than D, we would have $\sum\limits_{i=1} f_i(D) < \sum\limits_{i=1} f_i(D')$.

2.2°. Let us now consider the case when the division $D' = (T_1', T_2')$ has more students in its first team than D, i.e., when $n' = |T_1'| > n = |T_1|$. In this case, $N - 2n' < N - 2n$. So, every students who had a non-negative feeling $N - 2n$ in the division D can only get smaller positive feeling in the division D'. So, such D' cannot be better than D.

2.3°. Finally, if D' has fewer students in its first team, i.e.,if $n' < n$, then we have $2n' - N < 2n - N$. So in the division D', some students will get lower negative feelings than in the division D.

2.4°. In all three cases, D' is not better than D.

The statement is proven, and so is the proposition.

Comment. This proposition makes sense: even if we divide the class into more than 2 teams, since the bigger team has an advantage, some team will naturally start cooperating – to increase their chances of succeeding, and we will end up with fewer and fewer competing teams – unitl it gets to two.

So which division into two teams should we select? A natural idea is to select a division in which the worst-case feeling

$$w(D) \stackrel{\text{def}}{=} \min_i f_i(D). \tag{10}$$

is the largest possible. This is described by the following result.

Proposition 2.

- *For even N, the division with the largest possible value of $w(D)$ is the division into two equal size teams.*
- *For odd $N = 2m + 1$, the division with the largest possible value of $w(D)$ is the division into teams of size m and $m + 1$.*

Proof. According to the formulas that we obtained when proving Proposition 1, for any division into teams of size n and $N-n$, the value $w(D)$ is equal to $w(D) = \min(2n-N, N-2n)$. Since $N-2n = -(2n-N)$, this means that $w(D) = -|2n-N|$. Thus, the largest possible $w(D)$ corresponds to the smallest possible value of $|2n-N|$.

For even N, the smallest possible value is clearly 0, when $n = N/2$. For $N = 2m+1$, we cannot have 0, but we can have 1, when $n = m$. The proposition is proven.

Competition is good in general: one more argument. Competition may be good not just for adversarial learning, but in genera.

Indeed, competition means that if one team is somewhat head, the other team tries to catch up, and vice versa. Without a competition, the folks may stagnate and not progress too much. Theoretically, this can also happen in a competitive environment, when two teams's performance is exactly the same. However, there are usually many random factors affecting each team's performance, these factors are independent, and the probability that two independent random variables are exactly equal is 0.

Thus, one of the teams will be always slightly ahead, thus providing an incentive for the other team to catch up – and this way, stagnation will be avoided.

7 What Next?

What we did. In this paper, we provided a *simplified* mathematical model that explains why adversarial teaching works – and show that, in some reasonable sense, adversarial teaching is indeed a close-to-optimal teaching strategy.

The existence of such an explanation made us (and will hopefully make others) more confident that this method is a right one.

Can we do better? Teaching with more confidence is good, but it would nice to have a model that helps us teach *better*.

For this, we need a more realistic model. Such model should take into account that some attacks are more difficult to defend against, while others are easier. Such models should take into account that in adversarial teaching, it is often not individual against individual, but rather a team against a team – so we need to take into account team dynamics, etc.

We hope that our simplified model will provide a starting point for developing such more realistic models.

How to motivate? This paper presents a mathematical demonstration of how adversarial learning *can be* beneficial for teaching cybersecurity topics. But how to make sure that adversarial learning *is* beneficial?

In this paper, we concentrated on the technical part, on *what* to teach – implicitly assuming that students have the needed motivation (and, of course, the needed background).

In reality, while some students are always eager to learn, but for other students, it is important to keep them motivated. In our experience, when properly organized, competitive environments like hackathons are great motivators – but, on the other

hand, pedagogy teaches us that many students do not perform well in competitive environments.

How to best motivate students remains an important open problem.

Acknowledgments

This work was supported in part by the US National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).

The authors are greatly thankful to Boris Kovalerchuk for his encouragement.

References

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, and Mc-Graw Hill Co., New York, 2009.
2. S. T. Hamman and K. M. Hopkinson, "Teaching adversarial thinking for cybersecurity", *Journal of the Colloquium for Information Systems Education (CISSE)*, September 2016, Vol. 4, No. 1.
3. F. Katz, "Adversarial thinking: teaching students to think like a hacker", *Proceedings of the 2019 Kennesaw State University Conference on Cybersecurity Education, Theory and Practice*, Kennesaw, Georgia, October 11–12, 2019.
4. V. Kreinovich and S. A. Starks, "Why benchmarking is an (asymptotically) optimal approach to numerical methods: a geombinatoric proof", *Geombinatorics*, 2004, Vol. 13, No. 3, pp. 131–138.
5. G. G. Lorentz, *Approximation of Functions*, Halt, Reinhart, and Winston, New York, 1966.
6. G. G. Lorentz, "The 13-th problem of Hilbert", in: F. E. Browder (ed.), *Mathematical Developments Arising from Hilbert's Problems*, American Math. Society, Providence, Rhode Island, 1976, Part 2, pp. 419–430.
7. N. Moiseev, *Elements of the Theory of Optimal Systems*, Moscow, Nauka, 1975 (in Russian)
8. C. Servin, O. Kosheleva, and V. Kreinovich, "Adversarial Teaching Approach to Cybersecurity: A Mathematical Model Explains Why It Works Well", *Proceedings of the 24th International Conference on Information Visualisation IV'2020*, Vienna and Melbourne, September 7–11, 2020, pp. 313–316.