# Co-evolution of strategic decision making: DOTA 2 as a proxy for cybersecurity environments

**A. L. Sallaska[1,*], D. Biro[2], S. Duran[3], and M. Stuart[4]**

[1]The MITRE Corporation, McLean, VA, 22102, USA
[2]Albert Einstein College of Medicine, Bronx, NY, 10461, USA
[3]Pompeu Fabra University, Barcelona, 08002, Spain
[4]University of California, Berkeley, School of Social Welfare, Berkeley, CA, 94720, USA
[*]asallaska@mitre.org

## ABSTRACT

Computer hackers use specific strategies to penetrate systems. These strategies evolve over time, usually in response to the defense mechanisms employed by the system administrators. Being able to identify the strategies and when they change is of paramount importance to ensure the safety of the systems. Because data to help this effort is scarce, this paper explores the possibility of using competitive, strategic video game data as a proxy to identify strategies and their change points.

Keywords: Cybersecurity, intrusion detection algorithms, adaptive strategies, video game data

## Introduction

Existing performance metrics of cyber intrusion detection algorithms, such as false positive or negative rates, are unable to capture the level of granularity necessary to significantly improve the algorithms, especially when a single, definitive outcome is rarely the case for this domain. Traditional static cyber defense systems also require long lead times to install patch updates, and staving off damage in real time is unfeasible using these metrics. Therefore, metrics must evolve to provide real-time evaluations of adaptive adversaries whose strategies may change depending on the protective mechanisms they encounter, as well as characterizing sequences of detections.

Usable data from within the cyber domain that may help to strengthen detection algorithm scoring is nearly nonexistent. If actual attacks are documented in the real world, this data is rarely made available and may be network specific (hence, not generalizable). Thus, a proxy for the data is necessary. An online battle arena video game called Defense of the Ancients 2 (DOTA 2) is a rich data source of real-time adaptive adversaries. DOTA 2 is a strategic, competitive, multiplayer game where two teams of five individuals each compete against each other to complete objectives and to destroy the other team's base in a time frame of $\sim$ 20 to $\sim$ 90 minutes. The players deploy various in-game and between-game tactics and procedures to achieve a specific measurable objective. Professional players vie for tens of millions of dollars in prize pools each year, and over 2 billion games have been played. The results in this paper are using data from a cache of 500 GB of aggregated game data ($\sim$ 2.5 million games played over one year) and raw, event-by-event data of $\sim$ 100 Mb per game. Our goal is to use this data to shed light on how we can 1) detect and 2) quantify the rate of change of strategies in co-evolutionary systems.

This is akin to the classic 'Red Queen hypothesis' in evolutionary biology, but this case involves human behaviors where a strategy can be considered most generally as a "meme" of sorts. Assumptions include: 1) a mapping exists between game observables and a set of strategies, 2) a measurable signal can be extracted from which to ascertain the adoption, stabilization, and decay of specific strategy traits, 3) changes in strategies occur over time, and 4) strategy changes are driven (at least in part) from behaviors of the opposing team. We postulate that testing these hypotheses will require an understanding of the co-evolutionary dynamics of the overall environment. In particular, the human behavioral components underlying the adversary/defense team actions will need to be assessed. The use of data from a multiplayer online game for this purpose assumes there is a valid mapping between the "game-space" to "cyber-space" behaviors from which useful inferences can be made. Through this analysis, one goal is to understand what types of data (if any) are useful for this purpose and how one might develop proxies to characterize strategies. The ultimate goal would be to apply these methods against realistic data specific to a cyber intrusion.

## The Game: DOTA 2

An aerial view of the DOTA game board is shown in Fig. 1. The purpose of the game is similar to "capture the flag" in which each team seeks to destroy the opposing team's base (which is defined by their "Ancient"), located either in the lower left or upper right corner of the figure.

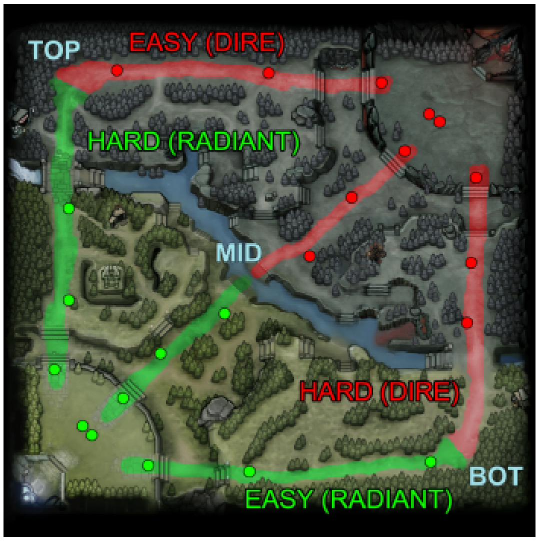

**Figure 1.** Game board of DOTA 2.

Ten human players are divided into two teams of five (the Dire and the Radiant teams), each controlling a 'hero' character, which is chosen via a drafting process in professional games (discussed in more detail in the following section). Each hero has its own set of unique abilities and is generally divided into one of three categories: intelligence, strength, or agility. Throughout the game, the heroes amass 1) experience (XP) via fighting in order to become more powerful and unlock special abilities, and 2) gold in order to buy items which also increase power.

There are three main 'lanes' to reach each Ancient, along the outer edges and down the diagonal of the board. There is also a jungle landscape between the lanes. Various computer-controlled characters called 'creeps' are deployed throughout the game which also allow heroes to gain experience. In addition, defensive towers line the lanes and protect each Ancient against enemy heroes.

### The Draft

Heroes are chosen from a pool of $\sim$ 120 characters. In professional games, there are twenty choices for ten human players–five hero picks and five hero bans for each team–chosen in the following order:

B1 B2 B1 B2 P1 P2 P2 P1 B1 B2 B1 B2 P2 P1 P2 P1 B2 B1 P2 P1

where B indicates a ban, P indicates a pick, and the number refers to the team. The team, character, order, and whether the selection was a pick or a ban is recorded, as well as the team that won the match.

## Analysis and Results

### Characterizing the complexity of the DOTA 2 draft

An important aspect to consider when comparing proxy data to real cybersecurity data is the inherent complexity of the system, i.e., how large and non-trivial the solution space for the game is. In order to gain some insights, a simple analysis was performed on team composition in terms of hero picks and their effect on win rates for the teams, without considering strategies that might be at play during the game. 84509 league games were analyzed, which spanned three DOTA 2 patches introduced by the game developers. Interestingly, it was discovered that even if such balance changes radically modify what heroes and strategies are objectively better, statistical regularities in hero usage can be found that survive these events (see Fig. 2). Additionally, hero usage was found to be not correlated to win rate (data not shown) suggesting that other processes (e.g., copying popular strategies) might underlie player's choices during the draft.

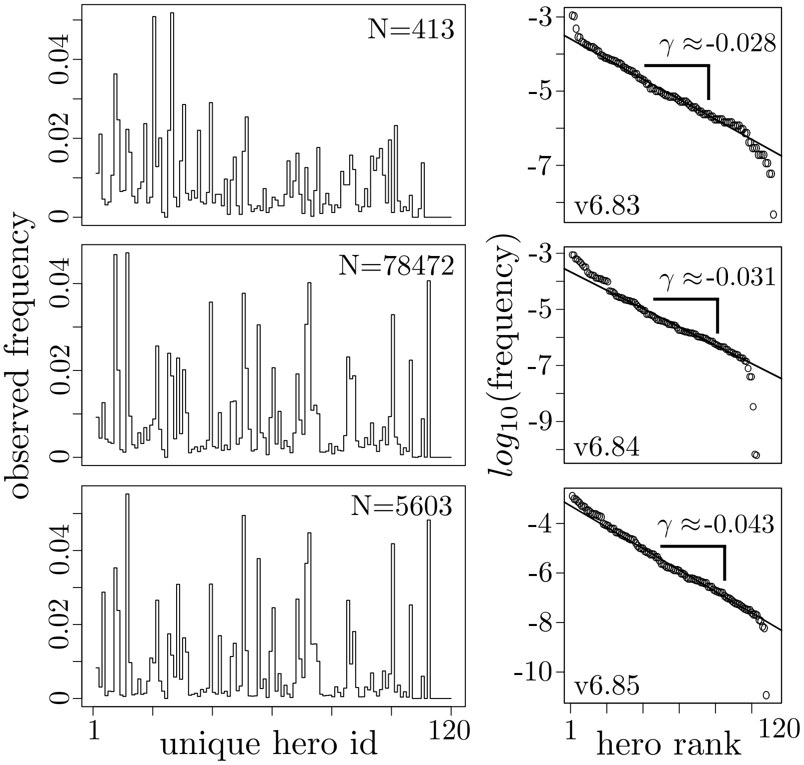

**Figure 2.** Frequency of picks in different patches of DOTA 2. Even if different heroes cease to be used or gain notoriety due to balance changes put forward by the developers of the game (plots on the left), statistical regularities can be observed that survive such changes (right side of the panel). For different patches (from 6.83 at the top to 6.85 at the bottom) we show the hero frequency rank distribution, fitted to an exponential distribution with similar exponents ($\gamma$) across various versions of the game.

## Characterizing the draft with a hidden Markov model

### *Model*

Initially, the draft was modeled as a Markov Decision Tree. However, due to the massive number of possible states, e.g. $3^{112}$, corresponding to the 112 characters and the option for unused, picked, or banned for each one, and the 69,464 transitions corresponding to 3,493 games for the pro game data set, analyzing the data in this manner was prohibitive.

In order to circumvent this challenge, the heroes were coarse grained into groups using character descriptions from the game itself, which separates the characters into classes of strength, agility, and intelligence. Additionally, each character fills one of nine roles, leading to possibly 27, 9, or 3 coarse-grained groups. This leads to sufficient state density to perform hidden Markov Model type analyses.

### *Further Analysis*

Once the data is coarse grained into the described groups, a hidden Markov Model could be generated to describe the underlying strategy of the players in determining their next pick or ban in the draft phase of the game. Additionally, further work into the optimal coarse graining of the characters may yield insights into how they could be grouped for this analysis. Possible unsupervised learning algorithms may be useful for this portion of the analysis.

## Raw Game Data

Event-by-event data throughout the entire duration of each game was extracted and analyzed in order to determine underlying strategies of the players. A hidden Markov model was applied and is discussed below after an overview of the required data processing.

### *Event-by-Event Data*

The raw game data was initially downloaded from a DOTA 2 repository in a binary form. A java parser was written by a colleague (Ellen Badgley) to convert the data into a JSON format, with each game event generating a name-value pair. An example event for a hero X changing locations is shown below:

{“tick”:24516,“time”:825,“type”:“DT_DOTA_Unit_Hero_X”, “team”:3,“x”:99,“y”:171}

where ‘tick’ is a subunit of ‘time’ (on the order of milliseconds), and team indicates either 2 or 3 (which must be paired with Dire or Radiant, see below). The type of event includes position, items purchased, items used, abilities used, gold collected, experience gained, damage taken or given, healing, and death, with each type triggering different tags following the type. This event-by-event list data, which can include a myriad of name-value pairs per time step, was transformed into an observation of a single time step, with columns for each event occurring at that step for each hero on each team. Aggregated statistics for the game as a whole, such as which team won and draft order, was folded in with this event data. The draft order from the aggregated statistics allows the hero names and team number (2 or 3) to be correlated with each team, Dire or Radiant, as the Radiant are denoted as team 0 in the draft data. This is important as the win is denoted as a boolean value for if the Radiant team won or lost, not if team 2 or 3 won or lost. Fig. 3 shows an example of the time evolution of one game metric, experience XP. This was a fairly unbalanced, faster-paced game in which the Dire team won handily.

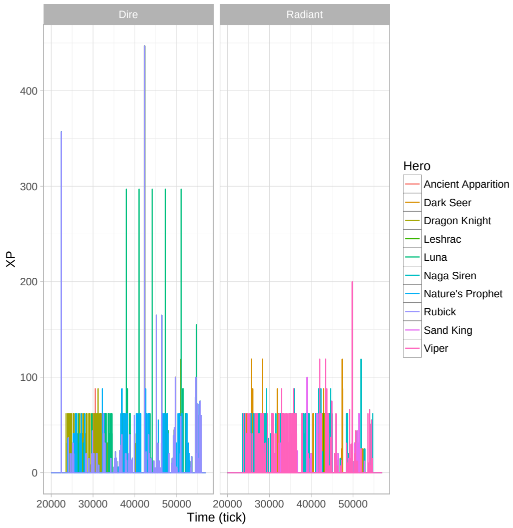

**Figure 3.** Evolution of experience points, XP, throughout a sample game for each team and hero.

### *Model*

A hidden Markov model (HMM) was used to explore latent strategies within DOTA games. The observed signals were assumed to be observables from the game itself, and the hidden signal was assumed to be the desired strategy of the players. One of the most relevant and revealing observables as a function of time step are the relative player positions. Team fights were also explored (e.g., the number of players fighting as a function of time). However, if two opposing heroes are in close proximity with each other, the probability they will fight is high (because of their objective to win the game). Fighting is therefore highly correlated with relative distance. Experience for each hero is accumulated through fighting and, hence, is also correlated with relative player distance.

Two methods were used to transform clusters of player distances into observable “states”: 1) deterministic based on a set distance threshold, and 2) unsupervised machine learning. The first method coarse grained the relative distances among heroes on a given team. The states were defined as follows, where the number indicates the number of heroes that are considered “close” together in a group:

- 1-1-1-1-1
- 1-1-1-2
- 1-1-3
- 1-2-2
- 2-3
- 1-4

- 5

For example, the state “1-1-1-1-1” indicates all heroes are far apart, whereas “5” denotes heroes are clustered closely together. “Closeness” is defined by a relative distance threshold: if the relative distance between hero A and hero B is below the set threshold, then A and B are considered “close”. This threshold was set to be 5 units, with the range of the game board being $\sim 100$ units $\times \sim 100$ units.

The second method is based on kmeans clustering of the relative positions of all players on both teams simultaneously. First, the absolute positions of each player on the game board was centered and scaled such that the the mean of positions was zero in both coordinates. Then the clustering algorithm was run as a function of time on the normalized, relative positions of each player. In order to determine the optimal number of clusters, the explained variance of the system was determined as a function of number of clusters. This was calculated as the inter-cluster distance relative to the total variance of the system. Ideally, the optimal cluster number is chosen such that this value is maximized, meaning most of the variance observed in the system can be explained by the grouping of the data. Fig. 4 shows the explained variance as a function of cluster number for the final set of data used in this analysis. A discussion of the optimal cluster number for this data set follows a brief description of the data shown.

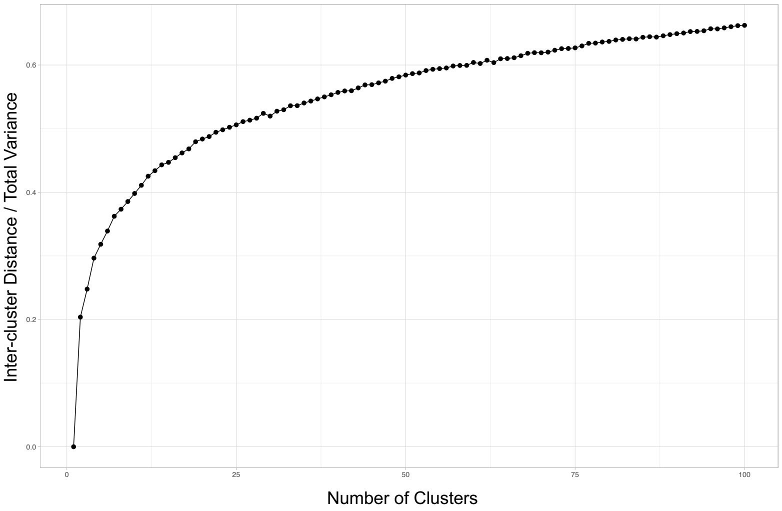

**Figure 4.** Explained variance as a function of number of clusters.

Although probing single games separately is an option for this work, it became more interesting to analyze multiple games simultaneously. This was accomplished simply by stacking the matrix-formatted time series of single games together and clustering on the combined time series (each row is a time step and each column is a normalized player position). This ordering of columns in this stacking of time series matrices becomes tricky because hero characters and players change from game to game. Several permutations of the order of player columns were tested, ranging from random ordering of columns of players to more methodical strategies. For example, the strategy that was ultimately chosen used an ordering the first 5 player columns from the winning team followed by the losing them. Then within each team’s set of columns, ordering those columns by amount of total gold accumulated, as relative gold can be an indication of a hero’s role within the game. No appreciable differences were observed for any of these permutations. The clustering optimization shown in Fig. 4 uses this method for $\sim 50$ games in a single tournament. Because of the mammoth size of the data sets, the position data was averaged over a 60 second time window for each hero to make the computation time of the model more tractable.

Instead of an obvious maximum, the explained variance does not stop increasing as additional clusters were added. This is not ideal. For future work, a more robust clustering method such as a Gaussian mixture model may produce more satisfying results. For this work, because the slope of this curve begins to change more slowly around 25 clusters, this number was chosen to produce the observed states for the HMM.

The time evolution of observable states and the results from the HMM are shown in Fig. 5. For each time step in the game, an observed state was assigned via the clustering algorithm described above. This sequence served as an input to the HMM. The dashed lines in the figure denote the breaks between stacked games. The code to estimate the model parameters and the most likely hidden state sequence was provided by Simon DeDeo[1]. The code uses the Akaike information criterion (AIC) in order to estimate the optimal number of hidden states in the model by balancing the log-likelihood of the model against the number of hidden states (i.e., there is a penalty for increasing the complexity of the model). For the example game above, the optimal number of hidden states is 18, as compared to 25 observable states.

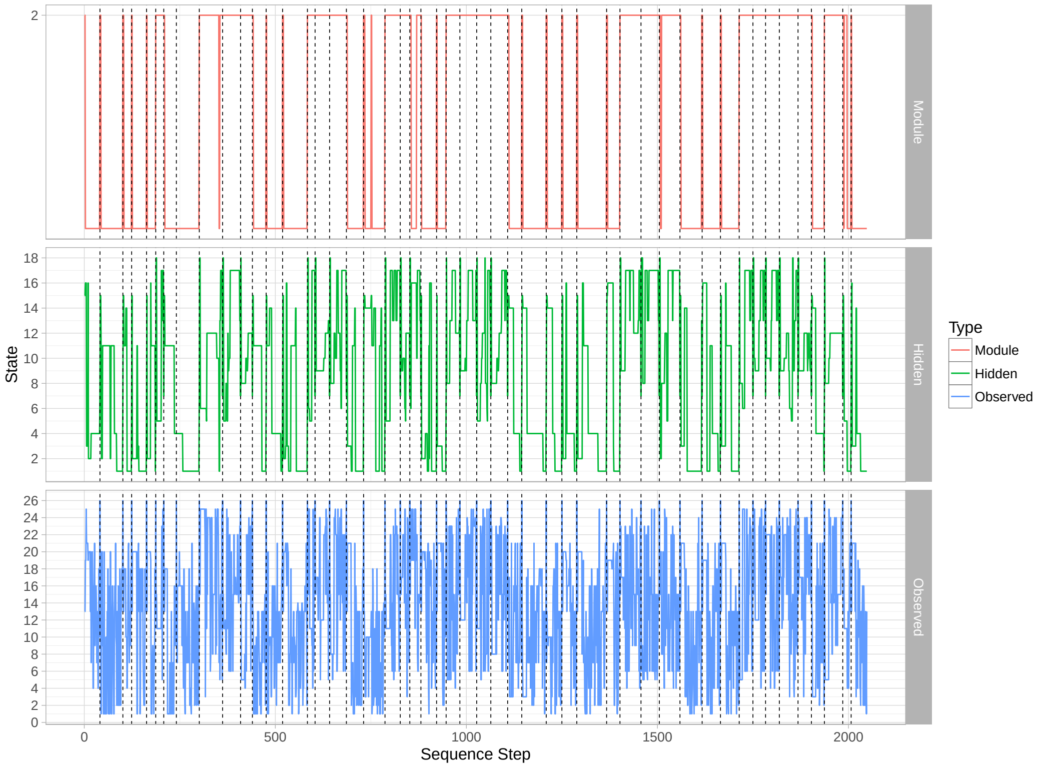

**Figure 5.** Evolution of hidden and observed states for a sample of 50 games from a single tournament. Note, the hidden state X may not correspond directly to the observed state X. These numerical labels are arbitrary and used only for plotting purposes. Dashed lines indicate the start of new games. The module states show the dynamic flow between the possible stationary state and its longest lasting perturbation.

Also included in the figure is perhaps the most interesting aspect: the modules. These represent the stationary state that the system eventually evolves to and its longest lasting perturbation. As shown in the figure, for this set of data, a given game *will* be in either module 1 or 2 throughout the entirety of the game. Very little oscillation between these modules is observed within games. This is indicative of some underlying structure.

If this behavior is predictive, the HMM has the potential to be used as a continuously running tool for cyber security analysts to be alerted in real time as to when these overarching strategies change in order to adapt their defensive algorithms. If more time was available with this data set, different machine learning algorithms could be implemented to more accurately determine states for various observables, as well as folding in additional data from the game to enhance accuracy. Because this framework is general and simply extracts hidden structure from sequences of observables, it is widely applicable to many dynamic domains. Attempts to correlate this finding with the aggregated game statistics have not yet revealed strong, consistent relationships between games in one module with the various statistics.

The model that produces the hidden states for the Radiant is shown in part in Fig. 6. Some states with low transition probabilities have been removed from the diagram for clarity. The highest non-self transition probabilities originate from hidden state 7 to hidden states 15 and 18. There is also a relatively strong transition from state 18 to state 8 and from state 13 to 17. This indicates that a handful of states dominate the hidden transitions. The median transition probability for states in which there is a non-zero transition probability is 0.055 (mean of 0.20). Even up to the third quartile, the transition probability is only 0.165. Clearly, this is a skewed system in which few states dominate the transitions, with the rest to having zero or near-zero transition probabilities (only 28% of the 324 possible transitions among the 18 hidden states are non-zero).

Future work for the event-by-event data could involve adding additional features in addition to distances such as XP or gold and using another clustering method to determine the observed states. A more in depth analysis could include only following certain teams throughout the professional tournaments to see if non-oscillatory in-game pattern holds and deeper analyses of correlations could be performed.

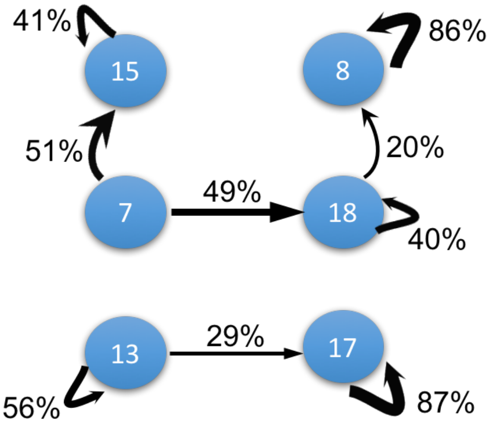

**Figure 6.** Hidden Markov model for 50 sample games. Selected hidden states and their transition probabilities are shown. Transitions below a probability of 20% are not shown in order to increase clarity. Self transitions among states in the diagram are shown. Self transitions among states *not shown* above were almost all above 85%.

## Conclusions

Data from the hero drafting process and in-game events of the multiplayer game Defense of the Ancients 2 (DOTA 2) were examined in order to inform new methods to interdict criminal cyber activity. Hidden Markov models were applied, which enables observables in the data to be associated with system "states". The fluctuations among these states can potentially be correlated with hidden states, revealing underlying structure and strategies in the system. The initial results are promising but require a much deeper analysis in order to produce actionable consequences for the cyber realm.

## References

1. DeDeo, S., SFIHMM, `http://bit.ly/sfihmm`, 2016

## Acknowledgments

We would like to acknowledge the originator of this idea, David Slater. Domain knowledge is key. We would also like to acknowledge Sanith Wijesinghe, Matt Koehler, Ellen Badgley, and Shaun Michel of the MITRE Corporation; and Simon DeDeo of the Santa Fe Institute for their input and advice.