6-2017

Multi-Agent Simulation of the Battle of Ankara, 1402

Ruili Tang
Union College - Schenectady, NY

Multi-Agent Simulation of The Battle of Ankara 1402

Ruili Tang

March 16, 2017

Abstract

In 1402, at the north of city Ankara, Turkey, a battle between Ottoman Empire and Tamerlane Empire decided the fate of Europe and Asia. Although historians largely agree on the general battle procedure, the details are still open to dispute. Several factors may have contributed to the Ottoman defeat, such as the overwhelming size of Tamerlanes army, poisoned water, the tactical formations of the military units, and betrayal by the Tartar cavalry in the Ottoman left wing. The approach is divided into two stages: the simulation stage, which provides data to analyze the complex interactions of autonomous agents, and the analysis stage, which uses data mining to examine the battle outcomes. The simulation is built on a finite state machine to evaluate the current situation of each agent and then choose the most appropriate action. To achieve historical accuracy, the simulation takes into account the topography of the battlefield, line-of-sight issues, period-specific combat tactics, and the armor and weapons used by the various military units at that time. The analysis stage uses WEKAs AttributeSelection Classifier to evaluate the association strength between the battle outcome and the various factors that historians consider crucial to the outcome.

Contents

List of Figures

List of Tables

1 Introduction

In 1402 a battle between two prominent conquerors took place which helped determine the fate of Europe and Asia. The Mongol leader Tamerlane won a decisive victory over and captured the Ottoman Sultan, Bayezid I, whose previous campaign in the Balkans had brought him the splendid appellation "the Thunderbolt"[9]. Tamerlane (Timur the Lame) had built a vast empire that stretched from Delhi to Moscow. Although the Sultan and the Khan had been exchanging letters for years prior to the battle, neither of them had stopped expanding their power over vassal states on the periphery of their domains. When Bayezid requested tribute from an emir loyal to Tamerlane, the latter finally gained an excuse to invade Anatolia [8].

The details of this battle are open to dispute[4]]. Historians are still debating what the most important factors were in the Ottoman defeat[8]. Spense C. Tucker argues that the betrayal of Tatar cavalry and Anatolian auxiliaries in the Ottoman left wing reduced the Ottoman army by a quarter and thus decided the battle. Stephen Turnbull speculates that either exhaustion caused by the forced march from Constantinople or a lack of water might have been the most important factor. The estimated size of both armies also varies a lot. Johann Schiltbergers contemporary account swells the numbers to "sixteen hundred thousand" for Tamerlane and "fourteen hundred thousand" under Bayezid[5]. Modern historian David Nicolle insists that the overwhelming difference in size (Tamerlane 140,000 and Ottoman 86,000) was a key factor, and that the betrayal of Tartar and Anatolian soldiers in Ottoman wings might also have contributed a lot [3].

Answering the question, "what was the key factor in the Ottoman defeat?" requires more than the understanding of the participants (or soldiers) in this particular battle. It also requires a knowledge of how agents communicate, cooperate and engage in combat with other agents. Agent-based modeling and simulation (ABMS) is well suited for this objective[2]. In ABMS, one agent represents a military unit of a certain number of soldiers, which analyzes and reacts to its surroundings autonomously. Each agent is also modeled individually with simple rules dictating their behaviors: *Move*, *Shoot*, *Retreat*, *Hand-to-hand Combat*, *Flee*, *Betray*, and *Fight to the Death*. The simulation starts with rigorously specified assumptions about the initial values of the agents and environment. The data for those values are mostly derived from the historical records of armor, weapon, and training of soldiers in the fifteenth and sixteenth centuries. The program is then iterated over four thousand times with several controlled experiments to generate data for the analysis stage, which uses WEKAs AttributeSelection Classifier to evaluate the strength of association between the battle outcome and the various factors that historians consider crucial to the outcome.

The remainder of this paper is structured as follows: section 2 briefly introduce the revelant works in the field of ABMS and applying computer science to the problems in social science and humanities. Section

3 explains the machinism of the simulation, the design of each agent, and illustrates the correctness of the model. Section 4 uses *WEKA* to evaluate the strength of association between each attribute and the result, on the dataset generated by my own simulation. I conclude with a summary and a vision as to how the ABMS can be further developed.

2 Related Works

Agent-Based Modeling and Simulation can be applied for problems which need to analyze the interaction of many autonomous individuals. To facilitate the use of Agent-Based Modeling across multiple disciplines, many frameworks and tools have been delevoped. For example, there are tools specifically made for modeling populations of disease vectors such as DTK (Disease Transmission Kernerl) and AGiLESim, which investigates the life cycle of anopheles gambiae mosquito, one of the most important vectors of malaria in Africa [10]. To analyze the evolution of the mosqito population, SAMPO(Scalable Agent-Based Mosquito POint model) was developed in OpenCL. In particular, the parallelism inside the OpenCL implementation has been maximized so that an overall simulation time speedup of is up to 576 and provides bettwe scalability as the agent-population size increases than AGiLESim[2]. Other generic agent-based modeling system, like RePast and MANSON(Multi-Agent Simulator Networks)[10] can also be extended to implement model-specific behavior.

3 Method

This paper intends to find answer to the question: **What is the most important factor in Ottoman's defeat?** History, like other disciplines in social science, seeks to understand how the interactions of each individual participant and the surrounding lead to an outcome. Answering the question above requires more than the understanding of participants (or soldiers) in this particular battle, it also requires the knowledge of how participants communicate, cooperate and engage in combat with others. Using Agent-based modeling and simulation (ABMS) can satisfy those requirements well.

In ABMS, each member of the population is modeled individually with simple rules dedicating their behavior[2]. In this particular battle, one agent represents a military unit of a certain number of soldiers, which can analyze and react to its surroundings autonomously. Theoretically, ABMS is also a combination of *induction* and *deduction* [2]. *Induction* is the process of inferring the a general law or principle from observation of particular instances, and *deduction*, on the other hand, uses general rules to infer conclusions

[6]. Simulation starts with rigorously specified assumptions of agents and environment like *deduction*, and generates data from controlled experiments for later analysis by induction.

The approach to the question above is divided into two stages: simulation stage and analysis stage. I implemented the agent-based simulation in *C++* without using any third party software, though a diverse group of simulation tools have been developed. Game engines like Unity3D can generate great graphics and demonstrate the interaction of agents more vividly, yet they leave little room for users to manipulate the combat mechanism and generally do not support a large number of sprites. Simulation tools like SimPy could be exceptional. Yet since the modeling of this battle does not entail the duration of combat or triggers of particular events at an exact time of the simulation (the aspect of SimPy), a program simply written in *C++* is satisfactory.

The analysis stage will use the data generated by multiple controlled experiments from the simulation stage. In each controlled experiment, the simulation will be rerun several times to more accurately characterize the results. It also uses WEKAs *AttributeSelection* Classifier to evaluate the strength of association between the battle outcome and various factors that historians consider crucial. For more detailed information, source code is publicly available on **https://github.com/thriller6767/battleofankara**.

4 Simulation

The simulation will read two input files: *battlefield.txt* contains the altitudes of the battlefield, and *basicdata.txt* provides the approximated initial values for each agent. After mapping the initial values and calculating the initial position for each agent, the program will compute the **Sight Range** and **Shoot Range** based on the altitude of current position and the direction this agent is facing.

Algorithm 1 Main Loop

```
1: procedure COMBAT
2:     while round ≤ 300 AND alive Ottoman or Tamerlane soldiers in battlefield do
3:         Update each agent
4:         Build Range Search Tree
5:         Do range search for the neighbors AND enemies in sight for each agent
6:         Each agent chooses and executes action
7:         Clear enemies in sight, enemies in shoot and neighbors vectors of each agent
8:         Delete Range Search Tree
9:     Print Result
```

The simulation is turn-based as in Algorithm 1. During each round, a range tree will search for the neighbors in **Neighbor Range** and enemies in **Sight Range**. 2D Range Tree is a binary tree containing the position of each agent. At odd level of the tree, the program will compare x_{pos}, and at even levelcompare y_{pos}: If pos is less than the current node's pos, add a new boundary node to left; Otherwise, add to right. In the ideal case, the tree is supposed to be perfectly balanced, thus the runtime of adding all agents to the tree will be $O(nlogn)$, where $n =$ number of agents. The major objective of using a Range Tree instead of a nested for-loop is to reduce the runtime of range queries. While a nested for-loop requires $O(n^2)$, using Range Tree to find nodes that lie in the desired interval $[a_1, a_2]$ can reach $O(logn)$. The runtime of doing n queries can thus reduce from quadratic time to linearithmic time. The pseudo-code for range query is in Algorithm 2.

Algorithm 2 Range Query using 2D Ranch Tree

```
1:  procedure RANGE QUERY(target, range)
2:      RANGE QUERY RECUR(root, target, range)

3:  procedure RANGE QUERY RECUR(node, target, range)
4:      if node ≠ null then
5:          if odd level then
6:              dis ← distance btw node and target
7:              if dis ≤ rangeANDtarget facing node then
8:                  add node to neighbor OR enemies          ▷ depend on which range is searching for
9:              else if targent + range is within left boundary of node then
10:                 RANGE QUERY RECUR(left, target, range)
11:             else if within right boundary then
12:                 RANGE QUERY RECUR(right, target, range)
13:             else if intersect then
14:                 RANGE QUERY RECUR(left, target, range)
15:                 RANGE QUERY RECUR(right, target, range)
16:             else
17:                 prune
18:         else                                             ▷ Even Level
19:             do the same as odd level
```

The flows of actions and states depends on a finite state machine in Figure 4. The core of the combat system is to **choose the weakest enemy within the line of sight**, and the three standards of choosing will be discussed in the following subsections. Each run of the simulation will write to two files: *result.csv* and *battle_index_detailed_result.csv* for later analysis.

4.1 Battlefield, Contour Lines, and Line of Sight Issue

I use **http://contourmapcreator.urgr8.ch/** to generate contour map and data. Its output includes a rectangular grid containing the height of each sample point of the selected area. However, this website only accepts 25 sample points at most at N-S axis and 20 sample points at W-E axis. In order to beter cooperate with the contour sample points, the size of battlefield must be carefully selected.

At the beginning of the battle, Bayezid took up a strong defensive position behind a stream and along low hills[3]. His camp was located at the village of Meliksah (the red petagon in Figure 1), which was defended by the Bogrek Hill and the Hamamtepsi Hill in the south as in Figure 1. Tamerlane, on the other hand, located his camp near the city of Ankara, and march north in search for enemies. At the end of the battle, Ottoman Sultan, Bayezid I, made his last stand at the Cataltepe Hill, and the rest of this troops retreated to the mountains in the north. Therefore, the selected battlefield for modeling must include those hills, with Cataltepe as the northernmost point and Ankara as southernmost point. The size of the battlefield chosen is around $12,500m$ (North to South) $\times 10,000m$ (West to East), so the distance between each sample point is $500m$.

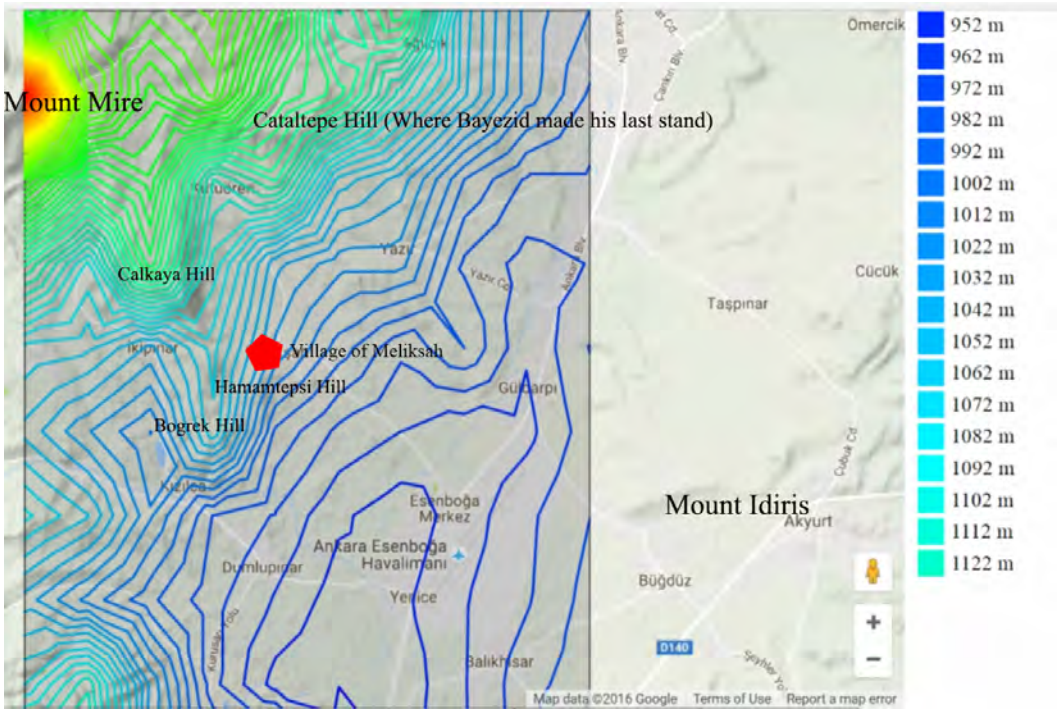
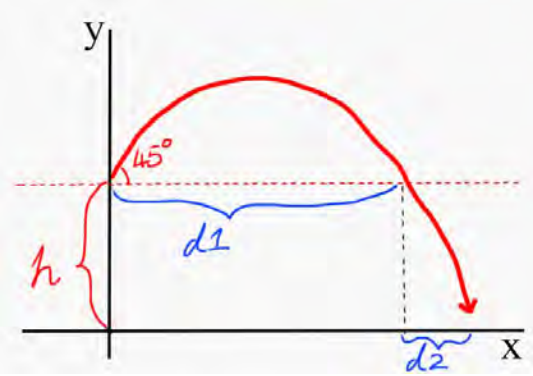
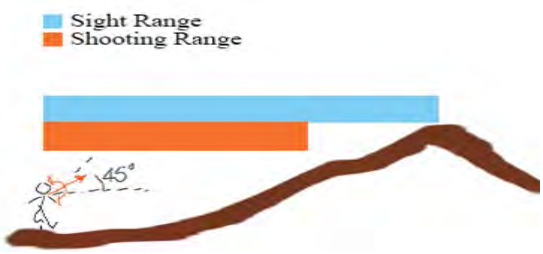


Figure 1: Battlefield

Generally speaking, people who stands on a higher ground can see further and shoot arrow at a larger



(a) Calculation of shooting range



(b) How far people can shoot and see

Figure 2: How altitude will influence shooting range and sight range

distance as in Figure 2. The simulation also takes care of this issue. The formula to calculate shooting range is demonstrated asin Figure 3 , and Figure 2a. The *shoot speed* of a Mongolian bow is $159fps = 48m/s$[7], the *shoot degree* in ancient time is usually $45°$ to maximize effective shooting range, *gravity acceleration* is $9.8m/s^2$, and the height of *shooter's shoulder* is generalized to $1.5m$.

The calculation of sight range is much simpler. As shown in Figure 2b, the line of sight is the distance between agent and a point of a curve at which a change in the direction of curvature occurs.

$$d_1 = \frac{\text{shoot speed}^2 \times \sin(2 \times \text{shoot degree})}{\text{gravity acceleration}}$$

$$h = \text{altitude} - \text{altitude in front} + \text{shooter's shoulder}$$

$$d_2 = h \times \frac{1}{\tan(\text{shoot degree})}$$

$$\text{range} = d_1 + d_2$$

Figure 3: Calculation of Shooting Range

4.2 Agent

The simulation uses a builder pattern to create new agents, because there are a few neccessary properties to initiate:

- inBattle, isAlive = **true**
- isAttacked, betray = **false**
- Name, Side, Size, Morale, Fatigue, Category
- Attack Damage, Armor Defense, Shoot Accuracy, Shoot Damage

- Position, Direction
- Agent State = *READY*
- Neighbors, Enemies In Sight, Enemies in Shoot Range
- Agent Index, Current Enemy Index $= -1$

In the ideal case, every soldier should be modeled as an autonomous agent. However, there might have been more than $230,000$ soldiers in the battlefield, modeling every soldier would dramatically increase runtime. An alternative way is to consider a group of soldiers as one entity, in which all soldiers share the same characteristics in initial values and choice making. In history, such entity is called a military unit, and soldiers who are organized into a military unit usually come from the same clan, receive the same military training, and carry the same type of weapons and armors. Each military agent can be categorized into five groups based on the way it moves and the way it engages in combat, as in Table 1.

Each agent will be given 1500 soldiers, except *TamurHorseArche* (3000) and *TamerReserves* (2000). The initial values of each agent can be inferred in Table 2 and Table 3. As mentioned above, the position of each Ottoman agent is along the small hills near village Maliksa, while Tamerlane agents, who started off marching from Ankara, are mapped at the southern border of the battlefield. For more information of where the number of soldiers in each type comes from, see Appendeix C.

Category	Explanation
Heavy Cavalry	On Horse, Melee
Light Cavalry	On Horse, Shooter
Heavy Infantry	On Foot, Melee
Light Infantry	On Foot, Shooter
War Elephant	Can be used as both melee and shooter

Table 1: The Definition of each troop category

Ottoman										
Name	**Category**	**size**	**Morale**	**Fatigue**	**Direction**	**AD***	**MD***	**MR***	**Accuracy**	**Defend**
Anatolian	Light Cavalry	15000	20	0	South	50	40	1	0.5	40
Azaps	Light Infantry	3000	40	0	South	40	40	1.2	0.6	30
Janissary	Heavy Infantry	20000	60	0	South	70	70	1.2	0.7	70
Rumelian	Light Cavalry	25000	40	0	South	60	60	1	0.5	50
SerbianInfantry	Heavy Infantry	1000	60	0	South	0	0	0	70	80
SerbianCavalry	Heavy Cavalry	2000	60	0	South	0	0	0	90	90
Kapikulu	Heavy Cavalry	5000	60	0	South	80	80	1.2	0.7	70
Tartar	Light Cavalry	7000	20	0	South	30	40	0.8	0.3	20
OttomanReserves	LightCavalry	7000	40	0	S	50	50	1	0.4	50
*AD = Attack Damage, MD = Missile Damage, MR = Missile Range										

Table 2: The initial values mapped to Ottoman Agent

Tamerlane										
Name	**Category**	**size**	**Morale**	**Fatigue**	**Direction**	**AD***	**MD***	**MR***	**Accuracy**	**Defend**
TamurHorseArcher	Light Cavalry	105000	40	0	North	60	70	1.2	0.6	50
Guards	Heavy Cavalry	10000	60	0	North	80	90	1.6	0.8	70
Tamurlane	Light Cavalry	15000	60	0	North	70	80	1.2	0.6	60
TamerReserves	Light Cavalry	20000	40	0	North	60	70	1.2	0.4	50
*AD = Attack Damage, MD = Missile Damage, MR = Missile Range										

Table 3: The initial values mapped to Tamerlane Agent

4.3 Choose the Weakest Enemy

There are three criteria by which to choose the weakest enemy, and with equal probability the program will throw a random number from $[0, 2]$ to pick one standard . The three standards are:

1. **Closest** It means the agent will choose the closest enemy in its line of sight. Preferably, the enemy is in the neighbor range so that this agent will be able to attack the enemy.
2. **Highest Height Bonus** It means the agent will choose an enemy which stands on a much lower ground. Generally speaking, standing on a higher ground will provide soldier a better line of sight, higher morale (because he can see what is coming forward), and a better degree to attack.
3. **Greatest Special Bonus**

   (a) *Category Bonus*

   As shown in Table 4. In history, *Heavy Cavalry(HC)* and *Heavy Infantry(HI)* wore thicker steel or leather armors to better engage in hand-to-hand combat, whilist *Light Cavalry(LC)* and *Light Intantry(LI)* were organized in a looser formation and wore lighter armor to move faster and shoot faster to perfrom "Hit-and-Run tactics". Also because they are lightly armored, LC and LI are vulnerable to melee fighting. The advantage is that they have greater mobility.

   As the first line of Table 4, HC has superiority in attacking LI. In close combat with a man on foot, mounted soldier enjoyed a number of advantages, including a higher position and a cosequent ability to strike down at his opponent. The horse itself was often a valuable ally, coupled with the high speed of charging [1]. LI, lightly armored and loosely organized, does not have the collision to resist HC. Furthermore, the formation of LI usually has little depth, since it is better to organize shooters in a single line to ensure everyone can have maximum missile range. The thinness of the formation and the lightness of the bowmen's equipment will not even provide them the opportunity to run away because HC can easily overtake them.

   However, as in the second line, HI has superiority in attacking LC. By giving his undivided

attention to his shooting, the foot archer has a greater rate of fire and, ever independent of the benefits of his steady platform, greater accuracy [1]. Compared to HC, LC wears lighter armor, which makes the soldier and his horse more vulnerable to arrows.

Against HI, thoroughly armored, protected by large shields and formed multiple ranks deep, HC has no chance to prevail. However, LC is a useful weapon against HI. Its mobility and "hit-and-run" tactics can keep HI in a safe distance, but in its shooting range. The tight formation makes HI a large and vulnerable target.

Category	**has Special Bonus against**
Heavy Cavalry	Light Infantry
Light Infantry	Light Cavalry
Light Cavalry	Heavy Infantry
Heavy Infantry	Heavy Cavalry
War Elephant	reduce the morale of all the other

Table 4: The relationship among different categories

(b) *Fleeing Enemy*

If there is an enemy who is running for its life, the enemy has lost all the ability to defend itself and attack others. Therefore, chasing after the fleeing enemy might be a good choice.

(c) *Stab in the back*

If there is an enemy in the line of sight but enaged in combat already, attacking this enemy and helping its comrade-in-arms sound beneficial. Even the strongest soldier cannot deal with attacks from multiple directions at the same time.

4.4 Formulas

The combat system has three parts: *Attack Damage to Deliver* (ADD), *Charge Bonus* (CB), and *Height Bonus* (HB).The action *Attack* has two categories: *keep attacking* and *charge*. *Keep attacking* means that the agent is already engaged in combat with a enemy, so this agent cannot quit combat until either it or its enemy dies or run for life. *Charge* means that the agent has made its enemy choice and rushes aggressively forward to attack. *HB* is 20 if the agent is attacking the enemy which is standing on a lower ground, otherwise, *HB* will be -20. If the agent is actively charging, *CB* will be 40. The basic formula for total damage delivered to the enemy is:

$$\text{ADD} = \frac{\text{size}}{\text{initial size}} \times (\text{AD} - \text{Enemy's armor} + \text{Category Bonus against enemy} - \text{fatigue})$$

$$\text{damage} = \text{ADD} + \text{CB (if there is one)} \pm \text{HB}$$

The value of *morale* depends on the casualty rate, which will only be influenced by the *damage* delivered from the enemy.

Algorithm 3 Morale

procedure UPDATE MORALE
  morale = initial morale $\times \frac{size}{initialsize}$
  if neighbor of its own side is dead **then**
    morale$-= 2$
  if neighbor of its own side is broken **then**
    morale$-= 3$
  if It is a Ottoman agent AND there is betrayal **then**
    morale$-= 5$
  if is surrounded by enemies **then**
    morale$-= 2$
  if is standing on a hill **then**
    morale$+= 10$

  if $60 \leq$ fatigue ≤ 100 **then**
    morale$-= 1$
  else if $100 \leq$ fatigue **then**
    morale$-= 2$

4.5 Correctness

4.5.1 Each Attribute Explained

1. **Ottoman-Marched-From-Constantinople.**

   This attribute is a binary number. If Ottoman troops march from Constantinople(1), the initial fatigue of all Ottoman agents will increase by 25. Note that later in the simulation, fatigue will only be influence by doing *Move*, *Withdraw* and *Attack* actions, and fatigue will impact the amount of damage delivered against the enemy.

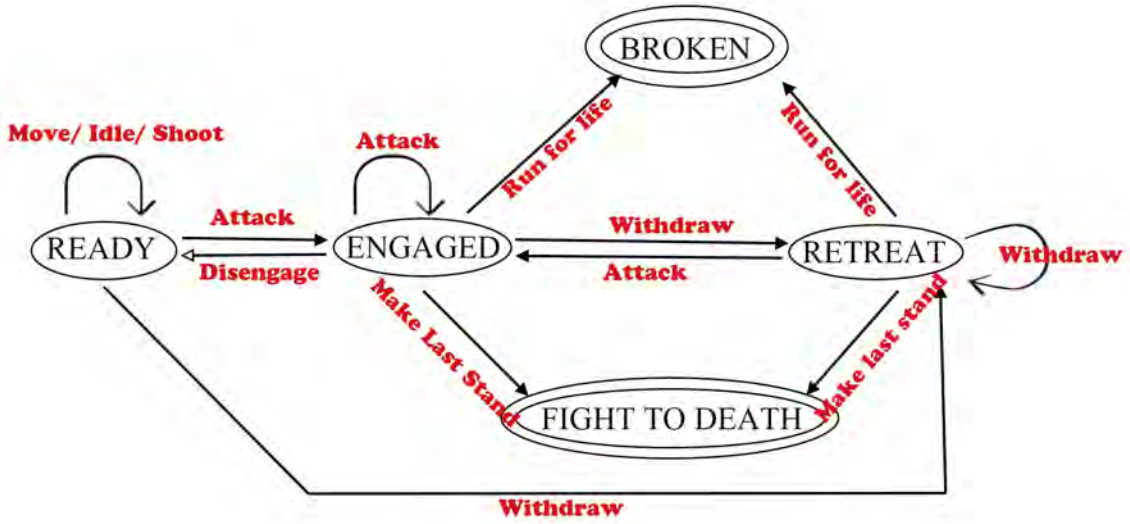


Figure 4: Finite State Machine

2. **Ottoman-Drank-poisoned-water.**

   This attribute is a binary number. If Ottoman troops drrink from a poisoned well(1), the initial morale of all Ottoman agents will decrease by 30, and the initial size will decrease by 100 soldiers per agent. Note that morale and size will determine the next state of one agent.

3. **Tartar-or-Anatolian-betrayal-in-Ottoman.**

   This attribute is a binary number. If this attribute is turned on, agents belong to Tartar or Anatolian probably will betray to Tamerlane's side. The trigger of "Betraying" has strict standards on Morale, Size and State. It will only take place when current state is *READY* or *ENGAGED*, morale < 20 OR casualty rate $> 50\%$.

4. **Ottoman-took-defensive-position.**

   This attribute has three values $0, 1, 2$. *Defensive* means that the agent of this side will remain idle until enemies enter the line of sight. *Offensive* means that the agent of this side will actively search for enemies and promptly attack them. If Ottoman is defensive and Tamerlane is on the offensive (0) as what happened in history, Ottoman agents will remain idle until there is at least one enemy enters the line of sight. Some Ottoman agents, like *Serbian Infantry*, *Serbian Cavalry*,*Tartar*, and *Rumelian*, are possible to perform the flanking tactic or become pugnacious to actively search for enemies. 1 means the opposite of history: Ottoman will be on the offensive, while Tamerlane agents will remain idle. 2 will make both sides on the offensive.

5. **Increase-Ottoman-Size.**

   This attribute has six values with bounds $[0, 5]$. It will add extra $13,500 \times i$, for $i \in [0, 5]$, soldiers to the Ottoman baseline total size $(87,000)$. The number $13,500$ is the total number of soldiers added if increasing every type of Ottoman agent by 1. There are 9 different types, each type will have one more agents, which has 1500 soldiers, so the total number of soldiers added is $9 \times 1500 = 13500$.

Type	# of Agents in this Type	# of soldiers in one agent	Total # of soldiers of this tyeo
Anatolian	10	1,500	15,000
Azaps	2	1,500	3,000
Janissary	14	1,500	21,000
Rumelian	16	1,500	24,000
Serb Infantry	1	1,500	1,500
Serb Cavalry	2	1,500	3,000
Kapikulu	3	1,500	4,500
Tartar	5	1,500	7,500
Otto_Reserves	5	1,500	7,500
Total	28		8.7000

Figure 5: Different types of Ottoman Agents, and number of agents in each type

4.5.2 Historical Baseline For Attributes

Several things can be approximated from historical sources. First, it was very likely that Ottoman soldiers suffered from forced march. When heard the news of Tamerlane invasion, Bayezid I suspended his eight-year siege of Constantinople, and marched to Ankara in nearly 15 days. Yet Tamerlane was at the city of Sivas then, when Ottoman troops reach Sivas, Tamerlane had already headed south and march toward Ankara. In order to save his city, Bayezid decided to return to Ankara from Sivas as soon as possible. For more detail of battle prelude in Appendix B.

David Nicolle is one of most famous military historian of medieval period. In his books, he approximated that a reasonale number of soldiers for Ottoman would be around $86,500$, and two third of them were infantry brought directly from the siege of Constantinople. Other soldiers include a large number of horse-archers from Ottoman vassal states or allied clans in Anatolia, and few Christian knights from Serbian kingdom. And he believes that **??**.

It is unlikely that Ottoman troops drank the poisoned water which was a strategy of Tamerlane. There is a small river running from south to north across the battlefield, and this river might be the source of water for both sides. If might be possible for Tamerlane to poison the river since his camp was located at upstream, yet it was hard to poison a flowing river. How much amount of poison he needed? Based on the information from primary sources and secondary sources, it is reasonable to believe that

March from Constantinople	Drink Poisoned Water	Betray?	*Who is on the Offensive*	*Ottoman Size*	Tamerlane Size	Result
YES	NO	YES	Ottoman	86,500	150,000	Bayezid LOST (10/10)

Figure 6: The historical assumption of this battle

5 Analysis

5.1 Data

The dataset is generated by the simulation. It contains 4320 instances, 5 numeric attribute explained above, and 1 class attribute: $\{$*Tamerlane Won*, *Ottoman Won*, *Draw*$\}$. For each combination of attributes, the simulation has 30 iterations(and therefore 30 instances) to better characterize the results.

5.2 Method

This section will explain the machine learning techniques used to analyze the dataset. The first step is to compare and choose the classifier with relatively high accuracy, the second step proceeds as to compare different combinations of evaluators and rankers. And then I use the chosen classifier, evaluator and ranker for attribute selection.

5.2.1 Classifier

We choose one or two classifier(s) which has a good reputation of having high accuracy results for each classifier category in *WEKA*. The classifiers picked are: (1) *NaiveBayes*, (2)*J48*, (3)*JRip*, (4)*KStar*, (5) *RandomForest*, and (6)*RandomTree*. All the classification will use cross validation with 10 folds,

In order to compare the performance on the same datasets, I use *WEKA*'s *Experimenter* interface to evaluate whether one classifier is statistically significant than the other with 95% confidence level. As shown in Figure 7, *NaiveBayes* is significantly less accurate than the other classifiers on this specific dataset. Exempt from *NaiveBayes*, I compare *J48* with other classifiers. As shown in Figure 8, they do not have satistical significance from *J48*. Therefore I will use *J48* for attribute slection.

Classifier	NaiveBayes	J48	JRip	KStar	RandomTree	RandomForest
Results	94.51	97.02 v	97.25 v	97.22 v	97.40 v	v 97.46 v
Compare	NaiveBayes is significantly less accurate than the other.					

Figure 7: Compare NaiveBayes with other Classifier

Classifier	J48	JRip	KStar	RandomTree	RandomForest
Results	97.02	97.25	97.22	97.40	97.46
Compare	No statistical significant between J48 and the other				

Figure 8: Compare J48 with other Classifier

5.2.2 Ranking

In order to know which ranking is more accurate, I perform a statistical test on the same dataset to compare different combinations of evaluators and rankers using J48. As shown in Figure 9, the combination of *CfsSubsetEval* as the evaluator and *BestFirst* as the ranking method is significantly better than *Wrapper + GreedyStepWise*. However, it does not have any statistical significance to the other three. Since cannot tell the ranking results of which method are more accurate and reliable, the following part of analysis stage will neglect the result of *Wrapper + GreedyStepWise* and focus on differentiating the other four methods.

Eval+Ranker	SubsetEval + BestFirst	SubsetEval + Greedy	Wrapper + Greedy	InfoGain + Ranker	GainRatio + Ranker
Result	95.66	95.66	82.06 *	97.02	97.02
Compare	Though better than Wrapper + Greedy , SubsetEval + BestFirst has no statistical significance with the rest.				

Figure 9: Compare SubsetEval + BestFirst with other evaluators and rankers

In Figure 10, from all of the four ranking results (regardless of *Wrapper + GreedyStepWise*) of the evaluators and rankers I compared, it is very clear that the attribute *Constantinople* does not have much influence over the result. This attribute is ranked as last one with 0 importance. The four results agree with the fact that *Size* and *Betrayal* have stronger strength of association than *Offensive* and *Poisoned*, but I cannot tell the specific ranking: in the results of *SubsetEval + BestFirst* and *SubsetEval + Greedy*, *Size* and *Betrayal* are both

Number of folds(%)	Attribute
10(100%)	Betrayal
10(100%)	Size
8 (80%)	Offensive
8 (80%)	Poisoned
0(0%)	Constantinople

SubsetEval + BestFrist

Number of folds(%)	Attribute
10(100%)	Betrayal
10(100%)	Size
8 (80%)	Offensive
8 (80%)	Poisoned
0(0%)	Constantinople

SubsetEval + Greedy

Number of folds(%)	Attribute
0(0%)	Betrayal
0(0%)	Size
0(0%)	Offensive
0(0%)	Poisoned
0(0%)	Constantinople

Wrapper + Greedy

Average Importance	Attribute
0.219 +- 0.003	Size
0.218 +- 0.002	Betrayal
0.016 +- 0.001	Offensive
0.001 +- 0.001	Poisoned
0 +- 0	Constantinople

InfoGain + Ranker

Average Importance	Attribute
0.218 +- 0.002	Betrayal
0.097 +- 0.001	Size
0.01 +- 0.001	Offensive
0.001 +- 0.001	Poisoned
0.001 +- 0.001	Constantinople

GainRatio + Ranker

Figure 10: Ranking Results of the five Evaluators and Rankers

correctly classified in all of the 10 folds. While those two method designate the two attributes at the same level of importance, *InfoGain + Ranker* does not agree with the outcome of *GainRatio + Ranker*. The former method prioritize *Size* with a very small ascendancy (0.001), in spite of the fact that the latter emphasizes the predominant position of Betrayal over all the other attributes.

	After Remove *Betrayal*	After Remove *Size*
Result	80.56	82.20 v
Compare	The first data set is significantly **worse** than the second	

Figure 11: Compare the accuracy after removing *Size* and *Betrayal*

To resolve this conflict, I need another statistical test to compare the classification accuracies of *J48* on two different data sets: one removes *Size*, another removes *Betrayal*. The baseline accuracy without removing anything is 96.90%. The larger the difference between the accuracy of removing and the baseline is, the stronger association strength this attribute might have. As shown in Figure 11, deleting *Betrayal* can result in a significanly worse accuracy than removing *Size* with $p = 0.0435$, which means removing *Betrayal* has more influence over the dataset. Therefore, attribute I can conclude that based on the simulation and its data, *Betrayal* is the dominating factor to the result amongst all the five attributes.

The statistic test in Figure 12 and Figure 13 confirms the ranking results. The *p-value* between *Remove Size*(82.20) and *Remove Poison*(97.43) < 0.0001, and *p-value* between *Remove Size*(82.20) and *Remove Offensive*

	After Remove Size	After Remove Poison	After Remove Offensive
Result	82.20	97.43v	94.70v
Compare	The first is significantly worse than the other two.		

Figure 12: Compare the accuracy after removing *Size*, *Poison* and *Offensive*

	After Remove Offensive	After Remove Poison
Result	94.70	97.43 v
Compare	The first data set is significantly **worse** than the second	

Figure 13: Compare the accuracy after removing *Poison* and *Offensive*

< 0.0001. In Figure 13, $p - value < 0.0001$. Removing *Size* lead to a significantly worse classification accuracy than *Poisoned* and *Offensive*, which means *Size* is the second most important factor to the outcome. Similarly, *Offensive* can be ranked as the third important attribute, followed by *Poisoned* and *Constantinople*.

5.3 Result

5.3.1 Statistical Analysis on Graphs

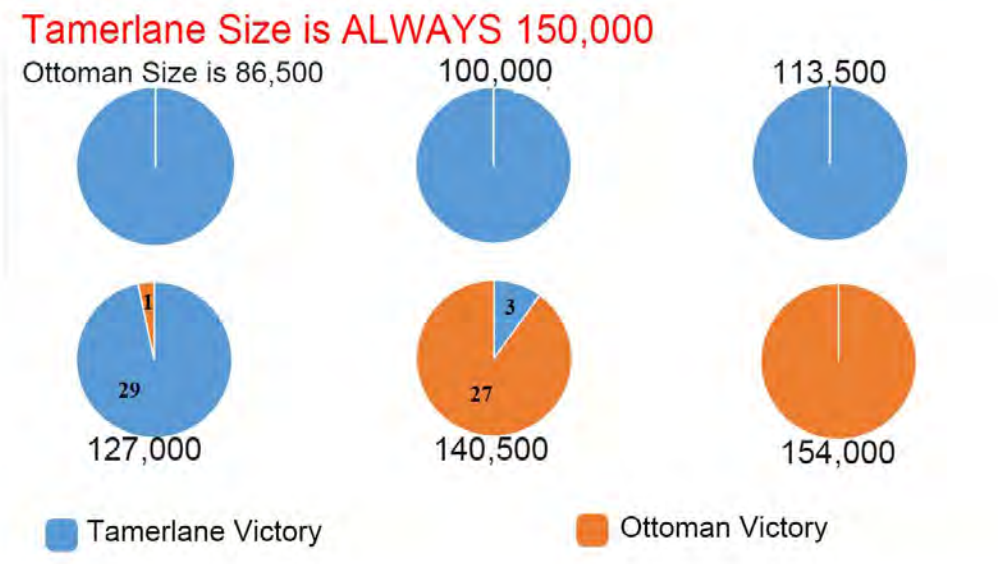


Figure 14: The influence of Ottoman size on the result

The Figures 14 and Figure 16 demonstrate that the overwhelming size of Tamerlanes forces and the betrayal by the Tartar cavalry contribut a lot to Ottomans' defeat. In Figure 14, adding extra Ottoman soldiers does not have much influence on the result until Ottoman size reaches $127,000$. It raises the chance

Figure 15: The influence of tactics on the result

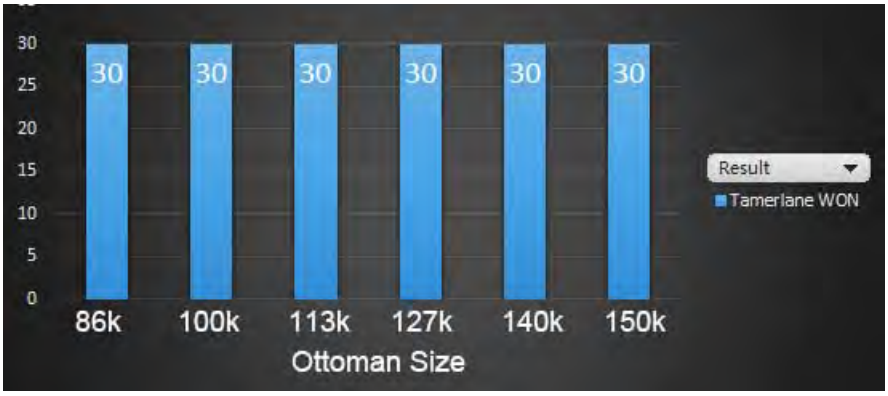


Figure 16: The influence of betrayal on the result

of an Ottoman victory from 0 to 10%, while an increase of 54,000 extra soldiers boosts the probability to 90%.

Nevertheless, in Figure 16, if the Tartar cavalry betrays their Ottoman allies, the chance of an Ottoman victory again falls to zero, no matter how many extra soldiers added to the Ottoman side. As shown in Figure 15, being on the offensive can also bring tactical advantage. While other attributes are controlled, with the tactics of Ottoman being on the offensive and Tamerlane being on the defensive (the opposite tactic to history), the chance of Ottoman winning rises from 0 to 43%. Nevertheless, Tamerlane troops regain the advantage when both sides are on the offensive, the chance of Ottoman victory drops to 17%.

5.3.2 Association Strength

Based on the graphs, *Betrayal* seems to be the most important factor that determines the battle outcome. All of the four ranking methods agree with the preeminence of *Betrayal* and *Size*, yet cannot differentiate those two attribute. The further statistical test on two datasets, which removes *Betrayal* and *Size* perspectively, shows that while using same classifier $J48$, deleting *Betrayal* will result in a significantly lower classification

accuracy. This means that *Betrayal* has more impact over the result than *Size*. With descending importance, the overall ranking is *Betrayal*, *Size*, *Offensive*, *Poisoned*, and *Constaninople*. Most of the ranking method that I compared agree with this result, and the statistic tests of the datasets which remove those attributes also confirm the correctness.

6 Conclusion and Future Work

In this work I have designed and implemented a multi-agent simulation of military units to analyze the reason for the Ottoman defeat at the hands of Tamerlanes army in the Battle of Ankara, 1402. The main goal of this work was to investigate what was the most crucial to the Ottoman defeat. Based on the information from primary sources and secondary sources, several reason may have contributed to the defeat, such as the overwhelming size of Tamerlanes army, poisoned water, the tactical formations of the military units, and betrayal by the Tartar cavalry in the Ottoman left wing.

The The approach is divided into two stages: the simulation stage, which provides data to analyze the complex interactions of autonomous agents, and the analysis stage, which uses data mining to examine the battle outcomes. The simulation is built on a finite state machine to evaluate the current situation of each agent and then choose the most appropriate action. To achieve historical accuracy, the simulation takes into account the topography of the battlefield, line-of-sight issues, period-specific combat tactics, and the armor and weapons used by the various military units at that time. The analysis stage uses WEKAs AttributeSelection Classifier to evaluate the association strength between the battle outcome and the various factors that historians consider crucial to the outcome.

The result shows that the betrayal by the Tartar cavalry was probably the most important factor. The overwhelming size of Tamerlanes forces also contributed a lot. Increasing the number of Ottoman soldiers by 40,500 raises the chance of Ottoman victory from 0 to 10%, while an increase of 54,000 soldiers boosts the probability to 90%. Nevertheless, if the Tartar cavalry betrays their Ottoman allies, the chance of an Ottoman victory falls back to zero. The third most important reason is the tactics. Being on the offensive can bring tactical advantages. While other attributes are controlled, with the tactics of Ottoman being on the offensive and Tamerlane being on the defensive (opposite to history), the chance of Ottoman winning rises from 0 to 43%. Yet when both sides are on the offensive, Ottoman victory reduces to 17%. Attribute *Poisoned* and *Constantinople* are ranked as No.4 and No.5 respectively.

In the future, I expect to extend the functionality of the simulation incorporating features to simulate the effects of commanders in the battlefield. In the fifteenth century, although it was hard for commanders to

give directions to military units far away, the commander could still have some control over the soldiers near him. This commander agent should be intelligent enough to judge on what is going on in the battlefield and give directions to soldiers near it. Also, agents are supposed to perform complicated tactics, such as flanking and charging from the back. Also, I plan to modify the three enemy-choosing criteria so that the simulation can find one enemy that maximizes all three criteria. That enemy is the true weakest enemy.

So far I cannot prove that my assumption of initial values of agents and the battle mechanism is 100% correct. I plan to deviate the fixed numbers of the simulation to see how those approximated numbers will influence the result. And then I can better prove the historical accuracy of this simulation.

Appendices

A Chronology

	Chronology
1227	Death of Chengis Khan
1336	Tamerlane was born in nowadays Uzbekistan
1359	Tamerlane rose to the head of Barlas clan by swearing loyalty to the Mogol Khan
1360	Bayezid was born
1384	Tamerlane's campaign in Iran and Georgia. Captured Tabriz
1389	Ottoman's victory at Kosovo. Bayezid succeeded the throne of Sultan as his father died in the battlefied
1390	Tamerlane's campaign on Golden Horde(Russia)
1391	Bayezid sieged Constantinople
1393	Tamerlane sacked Baghdad
1396	Bayezid defeated the last Crusaders in the battle of Nicopolis
1397	Bayezid united Anatolia under one rule
1398	Tamerlane's campaign in India.
1400	Tamerlane captured Ottoman city Sivas in his Anatolian campaign
1402	Battle of Ankara. Bayezid was captured
1403	Bayezid lost his life as a prisoner.
1405	Tamerlane died at Kazakhstan
1413	Ottoman finally ended ten-year civil war after Bayezid's death.
	His son Mehmet I ends the power struggle, and Ottomans revived as a growing strength

Table 5: The chronology of two empires since the death of Chengis Khan

B Battle Procedure

B.1 The Prelude

Although the detail of this battle is open to discussion, the major procedure is recognized by most historians. Before the crush of two empires in 1402, Tamerlane's influences had already penetrated Ottoman's eastern frontier. In 1400, Tamerlane captured the strategetic city Sivas (a city in the center of Asia Minor), whose history could be traced back to the Roman period. During his Anatolian campaigns, Tamerlane continued diplomatic exchanges to the Christian lands nearby, who feared the Ottomans more than Tamerlane, and thus hurried to send tribute in order to gain protection . Particularly, Tamerlane sought the "return from Bayezid's protection of certain dignitaries who had escaped the fall of Baghdad [in which Tamerlane slaughtered every inhabitants]"[9].

By that time, Bayezid, the Ottoman Sultan, had gained his reputation as a military genius. In the battle of Nicopolis in 1396, Bayezid brilliantly defeated the last Crusaders, and thus gained control of the Balkan

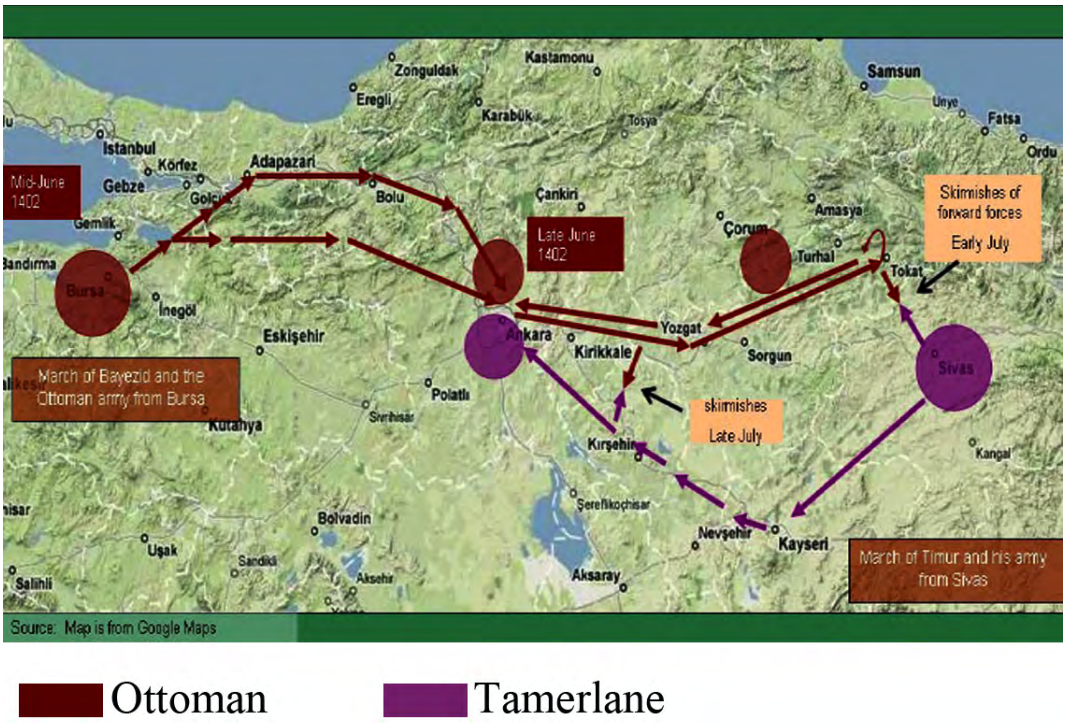


Figure 17: Manoeuvres of Tamerlane and Bayezid before they met in Ankara

Pennisula. The city of Constantinople became a Christian island. When Bayezid received the news that Tamerlane had conquered Sivas, he suspended his long siege of Constantinople and moved as fast as possible to prevent further penertration of his territory by Tamerlane, who was infamous for sacking cities. In order to speed up, Bayezid even left heavy equipment and packages in Ankara and forced march to Sivas.

Kept informed of Ottoman movements, Tamerlane ordered a forced march from Sivas to siege Ankara (the distance between nowadays Ankara and Sivas is around 275 miles). During the fourteen days of marching, Tamerlane purportedly supposed that Bayezid would came from south east, so he laid siege at the south of Ankara until his messagers informed him that Ottoman troops were only two days away from him. Not only was Tamerlane shocked by the speed of Ottoman troops, but also the direction they came from. Now Tamerlane found himself in a wrong and passive position.

However, the state of Ottoman troops was even worse. Some historians believe that Tamerlane had poisoned the water source [4], some suggest that Tamerlane diverted the upper stream of Cubuk River, which run through Ankara from south to north [9]. Besides those assumptions, it could be confirmed that Ottoman troops were exhausted from forced marching. In the early morning of July 28 1402, Ottoman troops prepared the formation at the north-east of Ankara. Standing on the small hills around village of Maliksa, Bayezid and his soldiers had been waiting for Tamerlane's arrival on the horizon.

B.2 The Key Events

This section will discuss the positioning and general procedures of the battle. As shown in Figure 18, Ottomans stood on the hills at the north, while Tamerlane came from south. At least one third of Ottoman force were infantry, the remaining part fought in the same Central-Asia horse-archery tradition as Tamerlane's troops. The size of the two armies was reliably estimated by David Nicolle at 140,000 on Tamerlane's side and no more than 80,000 on the side of Bayezid.[3]

The battle proceed as in Figure 18:

1. Tamerlane noticed the tactical weakness of Ottoman left wing for the lack of natural obstacles. Thus, he first ordered the attack against Ottoman left wing.
2. Tartar Cavalry betrayed and also attacked Ottoman left wing. Historians assume that the Tartar had secret terms with Tamerlane, who himself was a Tartar.
3. Tamerlane then ordered attack against Ottoman right wing. This was a classic tactic which utilized the size advantage and flank the enemy. However, the attack was driven back by the Serbs, who also lost cohesion and abandoned their original position.
4. The newly-conquered Anatolian emirates, forming the bulk of Ottoman right wing, now also deserted to Tamerlane, thus leaving the Janissaries and Azaps exposed on both flanks.
5. Attacked from both back and front, Rumelians slowly gave ground.
6. Judging the battle lost, Ottoman Reserves retreated under the order of Sultan's heir.
7. Bayezid and the remaining troops, including six squads of Anatolians who remained loyal, retreated to Catal hill to make a last stand.

B.3 The Postlude

The battle lasted over twelve hours: it began at morning, Bayezid moved to Catal hill aroud 5pm, and he broke out the siege at night. During Bayezid's last stand at Catal hill, Tamerlane's overwhelming troops surrounded the hill but all of their assualts were beaten back. [3] When night fell, Bayezid and other three hundred cavalries finally broke out eastwards, but the sultan's hourse fell and he was captured.

C Initial Value Estimations

This section explains upon what standard I determine the numberic values assigned for each attributes.

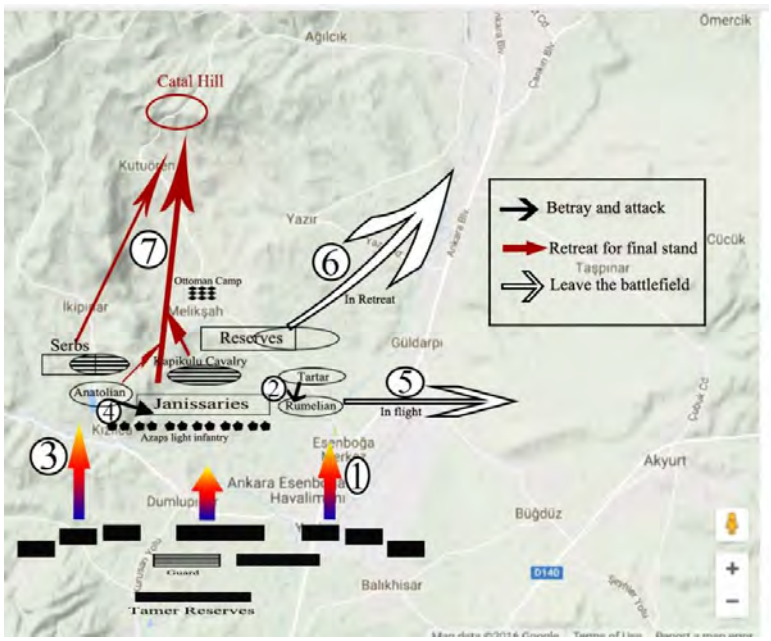


Figure 18: Battle procedures according to David Nicolle

- Attack Damage (AD), Missile Damage(MD) and Accuracy: are based on how much training this troop has received. For example, professional royal troops like Janissaries or elite cavalry like Kapikulu will have better AD, MD and Accuracy than militia like Azaps or Tartar. Initial AD and MD are measured from 0 to 100, while Accuracy is measured from 0 to 1.0 for their shooting ability. There are two exceptions: Serbian infantry and Serbian cavalry. Although Serbian Kingdom was a vassal state to Ottoman, they fought in Mediaval European style rather than Central Asia horse-archery form. They were usually equipped with spear, shield and long swords, and did not use arrows and bows. Therefore, Serbian Cavalry and Serbian Infantry do not have MD and Accuracy.
- Defense: is based on how much armor this troop wore. It is measured from 0 to 100. Mongolian cavalries and Turkish cavalries who followed the same tradition usually wore much lighter armor than the Crusaders. Therefore Serbian Inf and Serbian Cavalry will have the highest Defense value (90). Elite Heavy cavalris like Kapikulu and Tamerlane Guards will also have high Defense value than light cavalry and light infantry.

C.1 Where Does SIZE come from?

- Janissary: numbered 20,000 during A.D. 1520-66, and in this battle, David Nicolle underlined that at least a third (around 28,000) of Ottoman army are infantry.

- Azaps: The number of Azaps is the total number of infantry $(28,000) -$ number of Janissary $= 3000$.
- Kapikulu Cavalry: In the 16th century, Kapikulu was around 6,000.
- Rumelian: Sipahis (regular) cavalry numbered around 40,000 men in the 15th and 16th centuries **??**, over half of whom came from Rumelia.So the size of Rumelian is 25,000
- Serbian Cavalry: At the battle of Nicopolis (1396), Serbs provided 1500 knights.

References

[1] Archer Jones. *The Art of War In The Western World*. Urbana and Chicago: University of Illinois Press, 1987. ISBN: 0-252-06966-8.

[2] Klaus Kofler, Gregory Davis, and Sandra Gesing. "SAMPO: An Agent-based Mosquito Point Model in OpenCL". In: *Proceedings of the 2014 Symposium on Agent Directed Simulation*. ADS '14. Tampa, Florida: Society for Computer Simulation International, 2014, 5:1–5:10. URL: http://dl.acm.org/citation.cfm?id=2665049.2665054.

[3] David Nicolle. *Armies of the Ottoman Turks, 13001774*. London: Osprey Publishing, 1983. ISBN: 0850455111.

[4] David Nicolle. *The Age of Tamerlane*. London: Osprey Publishing, 1990. ISBN: 978-0850459494.

[5] Johann Schiltberger. *Bondage and Travels of Johann Schiltberger*. London: Hakluyt Society, 2007. ISBN: 978-0548286999.

[6] Sanjit A. Seshia. "Sciduction: Combining Induction, Deduction, and Structure for Verification and Synthesis". In: *Proceedings of the 49th Annual Design Automation Conference*. DAC '12. San Francisco, California: ACM, 2012, pp. 356–365. ISBN: 978-1-4503-1199-1. DOI: 10.1145/2228360.2228425. URL: http://doi.acm.org/10.1145/2228360.2228425.

[7] Varin smith. *Testing and comparing bows by measuring arrow speeds in feet per second*. 2007. URL: http://www.greenmanlongbows.co.uk/SPEED%20TESTING%20Measuring%20the%20arrow%20speed%20of%20bows%20and%20longbows%20using%20a%20chronometer.htm.

[8] Spencer C Tucker. *Battles That Changes History: An Encyclopedia of World Conflict*. ABC-CLIO, 2010. ISBN: B006Q7EWNC.

[9] Stephen Turnbull. *The Ottoman Empire 1326-1699*. London: Osprey Publishing, 2003. ISBN: 978-1841765693.

[10] Ying Zhou et al. "An Agent-based Model of the Anopheles Gambiae Mosquito Life Cycle". In: *Proceedings of the 2010 Summer Computer Simulation Conference*. SCSC '10. Ottawa, Ontario, Canada: Society for Computer Simulation International, 2010, pp. 201–208. URL: http://dl.acm.org/citation.cfm?id=1999416.1999441.