TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Andrei Hirvi 183173IABM

Reinforcement Learning for Maximizing Gaming Experience:

An Application in Sports Betting

Master's thesis

Supervisor: Innar Liiv

Ph.D

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Andrei Hirvi 183173IABM

Mängukogemuse maksimeerimine spordipanustamises kasutades stiimulõppe meetodit

Magistritöö

Juhendaja: Innar Liiv

Ph.D

Tallinn 2021

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Andrei Hirvi

Abstract

Enjoyment of the gaming process is crucial for the success of every product in the gaming field. Some gambling companies combine betting odds from multiple odds providers, and in some cases tune odds for managing risks and do it manually. From the author's experience, the 'maximizing customer experience' part is done just by combining maximum odds from different providers and some gambling companies even have a special trading department, where traders manually "tune" odds.

Here author sees an opportunity to apply some Artificial Intelligence solution, which could analyze the behavior of the customers in real-time, tune odds not only for managing risks but for maximizing customers' experience as well by applying previous scientific findings of why people gamble. This thesis gives an exploratory overview of the state of the art about what motivates gambling behavior and how Reinforcement Learning is used in the gambling field.

The main contributions of this thesis are proposing and validating a novel method for tuning betting odds in the sports betting field. For validating it the author has developed an artificial environment using the Reinforcement Learning area of machine learning for emulating the collaboration of gambling companies and gamblers.

The proposed environment consists of two business-agents that provide events and odds for placing bets, and gambler-agents that emulate the behavior of the human being gambler. While one business agent's odds are regular, the other one tunes odds in respect of the found in the state-of-the-art reasons why people gamble. Results show that tuned odds provide a longer and roller-coaster-like journey to the customers.

The author has validated the proposed hypothesis that it is possible to apply AI solution for automatization of tuning odds process and use it not only for managing risks but for maximizing customers' enjoyment as well. Moreover, using the proposed solution enables to use of odds from only one odds provider, which could dramatically reduce costs for gambling company and increase the loyalty of its customers.

But, everything has its own flip side: results show that using the solution brings more risks to the gambling company, for providing longer journey companies cannot just maximize income but should give an opportunity to make more stakes and in some cases increase the number of won stakes.

As further steps, the author proposes to implement a feedback loop, stabilize its policy, and evaluate the proposed solution in a real-life scenario.

The author has concluded that on the one hand, the proposed novel method of tuning betting odds may bring many advantages for gambling companies (automatization and optimization of the process and loyalty of the customers). On the other hand, the solution brings more risks for gambling companies, but all risks are manageable.

This thesis is written in English and is 49 pages long, including 6 chapters, 7 figures and 2 tables.

Annotatsioon

Mänguri saadav rahulolutunne on üks olulisemaid protsessi omadusi tagamaks teenust pakkuva meelelahutusettevõtte edu. Mõned panustamisettevõtted kombineerivad selle saavutamiseks panuste koefitsente mitmetelt koefitsentide teenuse pakkujatelt ning lisaks muudetakse koefitsente ka käsitsi riskide haldamiseks. Autori kogemusest on erinevate teenuste pakutud koefitsentide kombineerimine ainus tegevus, mida kliendi mängukogemuse parandamiseks tehakse. Mõnel panustamisettevõttel on selle jaoks isegi osakond, mis koefitsente manuaalselt haldab.

Töö autor nägi selles olukorras võimalust automatiseerida protsess kasutades masinõppe algoritmi, mis analüüsiks reaalajas klientide käitumist ning muudaks panuseid mitte ainult riskide kontrollimiseks, vaid ka kliendikogemuse maksimeerimiseks, kasutades seejuures varasemate teadustööde tulemusi, mis on selgitanud põhjusi, miks inimesed panustavad. Käesolevas lõputöös on ülevaade uusimatest leidudest mänguri motivatsioonilisest käitumises ning kuidas on siiani stiimulõppe meetodeid panustamisvaldkonnas rakendatud.

Lõputöö peamine eesmärk on pakkuda välja ja valideerida uudne meetod spordipanuste koefitsentide muutmiseks. Välja pakutud meetodi valideerimiseks on autor välja töötanud tehiskeskkonna, milles kasutatakse stiimulõppe meetodit jäljendamaks mängurite ja ettevõtete vahelist koostööd.

Kavandatud tehiskeskkond koosneb kahest äriagendist, kes pakuvad sündmusi ja panuste koefitsiente ning mänguragentidest, kes jäljendavad mänguri käitumist. Ühe äriagendi koefitsiendid on tavapärased ning teine muudab koefitsente väljapakutud meetodi alusel. Tulemused näitavad, et häälestatud koefitsiendid pakuvad klientidele pikemat ja kaasahaaravamat mängukogemust.

Autor on oma töös valideerinud hüpoteesi, et masinõppe algoritmi on võimalik rakendada panuste koefitsentide optimiseerimiseks nii riskide haldamiseks kui kliendikogemuse parandamiseks, kasutades selleks võimalikult reaalse elu sarnast simulatsiooni. Seejuures

võimaldab pakutud lahendus ettevõtte kulusid vähendada kasutades ainult ühe teenusepakkuja koefitsente ning lisaks, pakkudes atraktiivseimaid koefitsente, suurendab klientide lojaalsust ettevõtte suhtes.

Välja pakutud lahendusel on ka oma negatiivne külg. Simulatsiooni tulemused näitavad, et luua klientidele rohkem mängunaudingut peab ettevõte olema valmis ka natuke suuremaks riskiks.

Autor toob välja ideed, kuidas võiks tehtud tööga edasi liikuda. Järgmiste võimalike sammudena tuuakse välja tagasiside tsükli implementeerimine, selle stabiliseerimine ning lahenduse rakendamine ja hindamine reaalses stsenaariumis.

Töö tulemusena järeldab autor, et masinõppe algortimi rakendamine panustamise koefitsentide muutmiseks on võimalik ja uudne lahendus, mis võimaldaks protsessi automatiseerida. Seejuures, heade tulemuste saavutamiseks, peaks ettevõte olema valmis suurematekс riskideks. Riskihaldus on pakutud lahenduses samuti hallatav ja automatiseeritud.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 49 leheküljel, 6 peatükki, 7 joonist, 2 tabelit.

List of abbreviations and terms

AI	Artificial Intelligence
ANN	Artificial Neural Network
DA	Mesolimbic dopamine
PG	Pathologic Gambler
MARL	Multi-Agent Reinforcement Learning
MDP	Markov Decision Process
MSE	Mean Square Error
PPO2	Proximal Policy Optimization 2

Table of contents

List of figures

List of tables

1 Introduction

1.1 Background

The theory of games is a theory of decision making. It concerns how you should make decisions. Your decisions are linked to your goals—if you know the consequences of each of your options, the solution is easy. Goals give you a reward (=enjoyment).

According to the research paper [1] which describes what motivates gambling behavior, there are several motivators: One of them is money, it is known to enhance mesolimbic dopamine (DA) levels in the human striatum during gambling episodes, suggesting that money is what motivates gamblers. Also, according to the compensatory hypothesis losses are also very important in motivating human gamblers: without the opportunity of receiving no reward, gains become predictable and hence most games become dull. Based on this assumption, Zack and Poulos note that several pay-off schedules (slot machines, roulette, and dice game of craps) have a probability of winning close to 50%, so that they are expected to elicit maximal DA release and, therefore, reinforce the act of gambling.[1]

Enjoyment of the gaming process is crucial for the success of every product in the gaming field. For gambling companies maximizing enjoyment in sports betting gives advantages over competitors; for gamblers, it gives more satisfaction for the same amount of money (in some way even reduces costs for the client).

1.2 Problem

Companies like Sportradar, Betradar, etc. who provide odds for sports events for betting companies use machine-learning mathematical calculation models with information and liability-driven odds [2], but these models are universal for all clients (gambling companies) and not optimized nor personalized for the concrete client (gambling companies).

Moreover, some gambling companies don’t use only one odds provider but combine events and odds suggestions from multiple providers in order to personalize and enrich events and markets selection for their customers (bookmakers’ clients) which means there could be room for optimizing betting odds. This situation is illustrated in Figure 1. From

the author's personal experience, some gambling companies have a special trading department, where traders manually “tune” odds for managing risks.

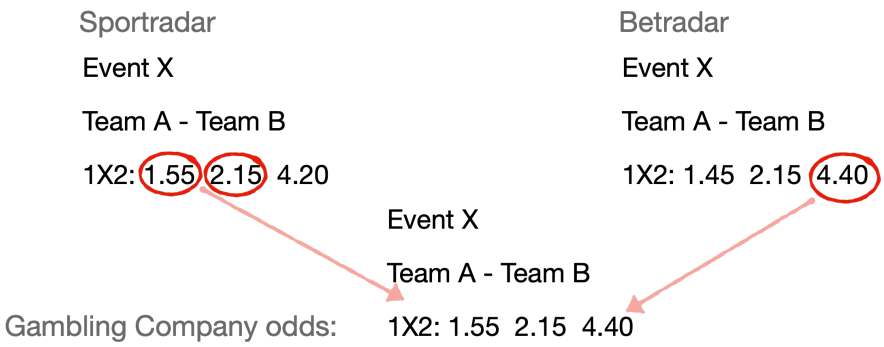

Figure 1 Example of combining Betting Odds

1.3 Purpose

The main goal is to understand if it is possible to maximize the enjoyment of the gambling process of the customer by tuning betting odds (by creating a roller-coaster experience, 50/50 win/lose, length of the game episode, and other factors that increase the gaming experience; but the company should still be profitable).

The main research questions of this thesis are: (1) Is it possible to tune odds proposed by gambling companies (bet365, betbrain.com, etc.) in such way that it will increase customer gaming enjoyment, and (2) will gambling company be still profitable if the algorithm will try to maximize customer gaming enjoyment?

In the beginning, the author will answer the research question by creating such an ML model, which will be personalized for the concrete gambling company and its clients (synthetic data will be generated: AI agents will place bets).

Reinforcement Learning (RL) field of machine learning will be used for developing a model since in the author’s opinion it is the most suitable area of machine learning for this kind of problem.

For validating the results author will create an artificial environment where there will be multiple trained RL agents in parallel: two business-agents (a gambling companies)

which will provide events and odds for placing bets (first agent will provide tuned odds, the second - original odds), and multiple customer agents: the first type of agents odds will not be changed, for the second type RL policy will tune odds for maximizing experience.

The Design science research method will be used in this thesis using simulation: executing artifact with artificial data for validating proposed hypothesis.

2 State-of-the-art literature review

2.1 Gambling theory and gambling behavior

The theory of gambling is strongly related to two concepts - decision making and risk. “The act of making a decision consists of selecting one course of action, or strategy, from among the set of admissible strategies. A particular decision might indicate the card to be played, a horse to be backed. [..] If each specific strategy leads to one of a set of possible specific outcomes with a known probability distribution, we are in the realm of decision making under risk.” [3]

Gambling is not an exception. People are predictable in this field as well and moreover, we can manipulate them.

As one example of predicting humans’ behavior is The Theory of Gambler’s Ruin which was used in [4] research paper: “The most original meaning of Gambler’s Ruin is that a gambler who raises his bet to a fixed fraction of their bankroll when he wins but does not reduce it when he loses, will inevitably go broke—even if he has a positive expected value on each bet.” Other interesting examples of gambler predictability are described in the book [5]: “There is a positive skewness of the frequency distribution of total gains and losses is desirable. [..] We tend to wager more conservatively when losing moderately and more liberally when winning moderately”. A Second example from the book is part of Prospect Theory: "People are inherently (and irrationally) less inclined to gamble with profits than with a bankroll reduced by losses.” [5]

Very exciting research is done in “What motivates gambling behavior? Insight into dopamine’s role” paper by Patrick Anselme and Mike J. F. Robinson. Researchers show us alternative reasons why people gamble. In the beginning, they introduce us to the traditional view, where money is the main reason why people gamble: “Common sense suggests that if gambling at casinos is attractive for many people, it is because it offers an opportunity to win money. […] Money is known to enhance mesolimbic DA levels in the human striatum during gambling episodes, suggesting that money is what motivates gamblers". Then, they challenge this view by questioning why people often describe gambling as a pleasant activity rather than as an opportunity to gain money. Researchers explain to

us, that there are two more reasons why people gamble: first is that losses motivate gambling more than wins: “During gambling episodes, PG report euphoric feelings comparable to those experienced by drug users, and the more PG lose money, the more they tend to persevere in this activity— a phenomenon referred to as loss-chasing". One more reason why people gamble is the attractiveness of reward uncertainty: “without the opportunity of receiving no reward, gains become predictable and hence most games become dull”. Based on this reason, some casino games (slot machines, roulette, and dice game of craps) have a probability of winning close to 50%, so that they are expected to elicit maximal DA release and, therefore, reinforce the act of gambling. [6]

2.2 Using Reinforcement Learning in Gambling and Gaming fields

Using Machine Learning (and Reinforcement Learning field) in the gambling field is not a breakthrough idea, but most research papers are concentrated on different perspectives: predicting outcomes of the sports events, gambling addiction studies, discover betting strategy, or predicting customer’s next gamble.

Researchers from Nelson Marlborough Institute of Technology (Auckland, New Zealand) are focusing on the application of Artificial Neural Network (ANN) to sports results prediction, they proposed a novel sports prediction framework for the Machine Learning field for predicting outcomes of sports events. They have built a Machine Learning model, which average performance in predicting results was around 67.5%. [7]

Researchers from another research paper are trying to predict the next gamble of the customer to improve recommendation systems for gambling platforms. They propose a machine learning model namely psychological factorization machine (PsychFM), which achieves a Mean Squared Error (MSE) of 0.0736, on an average for one prediction there is an error of 0.27 (there is a 27% error in the predicted probability). [8]

Researchers from DeepMind have applied Reinforcement Learning for asymmetric games: in this research paper they "examine how two intelligent systems behave and respond in a particular type of situation known as an asymmetric game, which include Leduc poker and various board games such as Scotland Yard.” [9]

One of the recent research papers propose to use Reinforcement Learning for maximizing gamers' entropy in computer games: they proposed their own exploration approach called Maximum Entropy Explore (MEE) for finding maximum entropy. To evaluate the performance of their approach, they constructed environments in the Grid World and StarCraft II games. [10]

2.3 Using Reinforcement Learning for tuning betting odds

After doing some research about the problem introduced in this thesis the author has concluded, that there are no research papers or other studies that are trying to solve it (=tuning odds) by using the Reinforcement Learning field. It is possible (and sounds very likely) that some odds provider companies or gambling companies use RL for automatizing odds tuning or for managing risks by tuning odds, but there is no any evidence (source code, article, etc.) of using it.

3 Theoretical Framework

3.1 Sports betting

Sports betting (or gambling) is very closely related to math, probability, and the theory of games. This is one of the reasons why the author has decided to apply reinforcement learning in the sports betting field: sports betting is measurable.

Next, the author will briefly introduce the math behind sports betting and challenges of gambling companies.

3.1.1 Sports betting odds

In sports betting, odds are some kind of coefficient of risk, odd shows the probability of this outcome of the event. The lower the odd is, the lower risk to lose, the smaller amount of potential win.

There are different formats for presenting the odds and usage of them depend on the geographical location and also on the betting market. In Northern Europe, the most common type is decimal odds, where the odds are the inverse of the offered probability of the outcome. [11]

Example

Let's say that there is some fictional football match where there are two teams playing: Barcelona vs Manchester United. There is probability of 70% (0.7), that Barcelona will win: then odd will be: 1 + (1 - 0.7) = 1.30.

Let's suppose that someone just made a stake of 10 EUR for Barcelona.

Then, possible win = 10 EUR * 1.30 = 13 EUR,

possible income = 13 EUR - 10 EUR (stake amount) = 3 EUR

The general formula is: Possible win = Stake amount * Odd

In sports betting there are different outcomes (markets) to place bets: who will win the match, correct score, odd/even score, who will score the next goal, etc. The proposed solution concentrates only on one market - who will win the match (short "1X2").

3.1.2 Gambling business and odds

During the last few decades sports betting business has grew exponentially: The Internet and smart devices have caused globalization of the business. A lot of gambling companies have appeared on the market, which caused very strong competition, which is ongoing at the moment as well. Betting companies need to offer the best (=highest) betting odds to keep up with the competition. In 2019 sports betting market was valued at US$85.047 billion. [12]

Most of the gambling companies do not generate nor manage sports events on their own, but buy events data and odds from 3rd party services, such as sportradar, betradar, etc. These companies provide sports events information, markets, and betting odds, and all updates related to this data.

As it was mentioned earlier, some gambling companies combine events and odds suggestions from multiple providers in order to personalize and enrich events and markets selection for their customers (bookmakers' clients). Some gambling companies "tune" original odds for manipulating customers' behavior (for managing risks).

3.2 Reinforcement Learning

For brief introduction of RL author will refer to his favorite book about Reinforcement Learning: "Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal. [..] The most important feature distinguishing reinforcement learning from other types of learning is that it uses training information that evaluates the actions taken rather than instructs by giving correct actions." [13] [14]

Reinforcement learning is strongly related to Markov decision process (Figure 2), to be more precise, it is the core of reinforcement learning: "MDPs are meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The learner and decision maker is called the agent. The thing it interacts with, comprising everything outside the agent, is called the environment. These interact continually, the agent selecting actions and the environment responding to these actions and presenting new situations to the agent." [15]

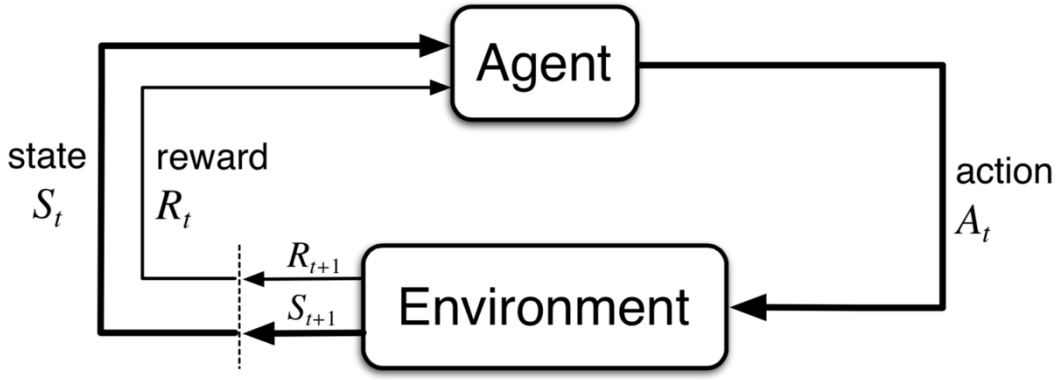

Figure 2 The agent–environment interaction in a Markov decision process. [15]

Agent represents some active role or party (such as gambler, of a betting company) that has some policy according to which it acts (depending on observations).

Policy is some set of rules that controls the agent's behavior. [16]

Actions are acts/operations/moves, etc. that agent is able to do in the environment. Actions can be any set of something (acts/operations/moves) we want to learn, and the **states** can be anything we can know that might be useful in making them. [17]

The **environment** is everything outside the agent, let's call it the universe. It is a 'place' where agents act and the environment reflects agents' acts by giving rewards to their actions.

Observation is something that an agent 'sees'. Observations are pieces of information that the environment provides the agent with that say what's going on around the agent. [18]

Every agent has its own observation. The policy is some set of rules that controls the agent's behavior. [19]

3.2.1 OpenAI Gym

OpenAI Gym is a "life simplifier" and it is used for the proposed solution. OpenAI is an Artificial Intelligence research company, and the gym is a toolkit for developing reinforcement learning algorithms, developed by OpenAI. [20] [21]

The gym is a toolkit for developing and comparing reinforcement learning algorithms. The main goal of Gym is to provide a rich collection of environments for RL experiments using a unified interface. [22]

OpenAI Gym provides its Environment (it's called Env) with its action-space and observation-space, which can be discrete, continuous, or a combination of the two.

The actions that an agent can execute can be discrete, continuous. Discrete actions are a fixed set of something (acts, moves, etc.) that an agent can do, for example, wager or not to place (0 or 1). On the other hand, continuous action has some value, and this value can be anything between some range. For example, odds change can be any number in a range from -1 to 1, such as 1, or -0.2222. [23]

The same logic with observation-space: observations can be discrete, for example, did agent place bet or not, and continuous: what stake did agent place (amount of money).

3.2.2 Multi-agent environment

Multi-agent RL (sometimes abbreviated to MARL) is a very young and promising field. A multi-agent system is a group of autonomous, interacting agents sharing a common dynamic environment, agents learn by interacting with its environment. At each time step, the agent perceives the state of the environment and takes an action, which causes the environment to transit into a new state. [24]

There are multiple types of relationships between agents: cooperative, competitive, or independent (when agents use their observations and own hidden memory state). [25]

In case of cooperative relationships, agents have shared observation. The diagram below (Figure 3) shows an example of a multi-agent environment, where agents have cooperative relationships and common action:

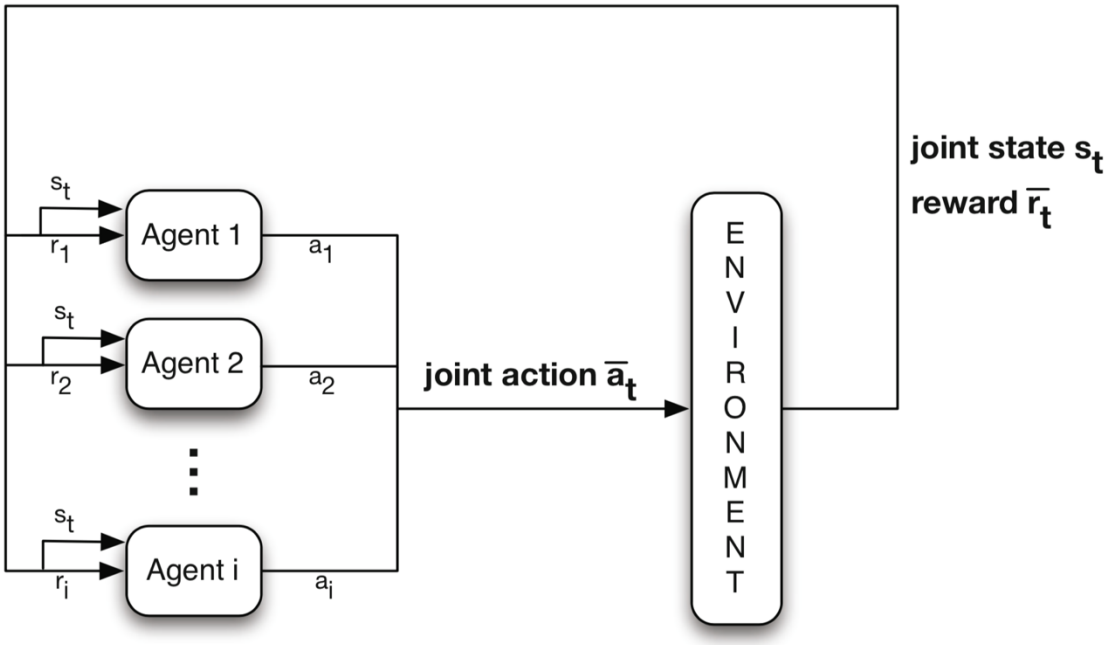

Figure 3 Multiple agents acting in the same environment [26]

3.2.3 Proximal Policy Optimization (PPO2)

While choosing the right algorithm, the author had two criteria, which algorithm had to complete: the first one is good performance and the second is the complexity of implementation (the author did not want to spend months to implement an algorithm). After some research author has found a Proximal Policy Optimization (PPO) algorithm.

According to OpenAI, Proximal Policy Optimization algorithm "performs comparably or better than state-of-the-art approaches while being much simpler to implement and tune. PPO has become the default reinforcement learning algorithm at OpenAI because of its ease of use and good performance. [..] PPO strikes a balance between ease of implementation, sample complexity, and ease of tuning, trying to compute an update at each step that minimizes the cost function while ensuring the deviation from the previous policy is relatively small." [27]

4 Methodology

4.1 The proposed method for tuning betting odds in the sports betting

As it was mentioned earlier, some gambling companies combine betting odds from multiple odds providers, and in some cases tune odds for managing risks and do it manually.

From the author's experience, the 'maximizing customer experience' part is done just by combining maximum odds from different providers and some gambling companies even have a special trading department, where traders manually "tune" odds.

Here author sees an opportunity to transform this process by tuning odds not only for managing risks but for maximizing customers' experience as well by applying previous scientific findings of why people gamble.

The author suggests applying a Reinforcement Learning solution for transforming this process.

The solution could take into account and analyze the behavior of the customers in real-time: for example, what odds customers prefer, how tuning odds impact customers' behavior and whole gaming journey, and what odds (and odds changes) are the most profitable for gambling companies.

Customers' behavior, company risks and liabilities, and original odds for some market (in this thesis 1X2 market with 3 odds) could be input for the solution. Then, based on input data and past experience, the algorithm 'decides' how to modify original odds.

As an output of RL solution returns tuned odds (3 odds).

Below is the architecture diagram (Figure 4 and Appendix 1) of the proposed solution. Grey dashed elements are not realized in the author's artificial environment since it is needed only for a real-life scenario, not for emulation. Other elements (black) are part of developed artificial environment.

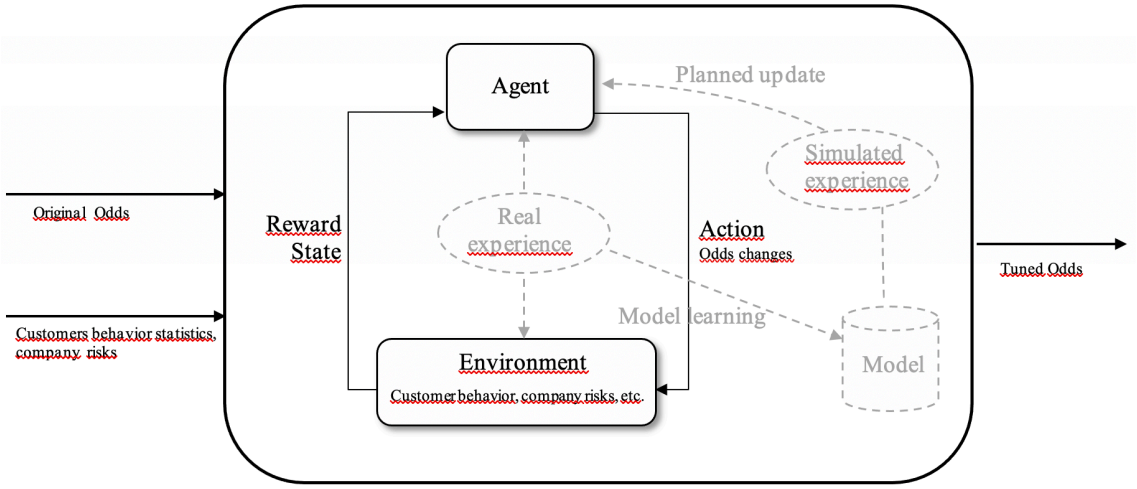

Figure 4 Proposed solution architecture

4.2 RL multiagent environment design

The environment is based on OpenAI Multi-Agent Particle Environment: "A simple multi-agent particle world with a continuous observation and discrete action space, along with some basic simulated physics", which is described in Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments research paper. [28]

The author chose this because of multiple reasons: the multi-agent RL environment matches best with a real-life scenario and it fully covers the needs for building an artificial environment for emulating the collaboration of gambling companies and gamblers.

The proposed solution is a system with multiple Reinforcement Learning autonomous agents sharing a common environment. There are multiple types of relationships between agents: cooperative, competitive, or independent (when agents use their observations and own hidden memory state). [25]

Some Agents have shared observation, which means they have the same information about the environment based on which they are acting. Moreover, agents can share their observations and states, which makes them cooperative.

The environment consists of the agents, which are in the system, agents' policies, actions and observations, info about the world. The environment is responsible for executing each

step for each agent, getting a reward for it, and for watching for when the episode is done. In such environment each agent has its possible actions, own observation, and policy, according to which it choices how to behave. All agents in the environment are simultaneous, which means that neither of the agents proceed to the next step before all agents in the environment had finished the previous step and got some reward (according to its policy). Agents learn by interacting with the environment, which gives some reward according to its policy.

Proposed by the author solution is a competitive environment, where there are two parties with contra-posed interests: Gambling company which provides a possibility for its customers to wager and second party - gambler, who is interested in placing bets. Of course, both sides are interested in getting income.

For emulating a real-life scenario and validating the proposed hypothesis there are needed only two parties, which were described above: there are two types of agents in the environment, Gambler and Gambling company (=Business agent).

4.2.1 Gambler-agent

After doing some research the author has found that building a smart gambling bot for getting some income is a very well-known problem and there are a lot of open-source solutions for doing that. Most of the solutions are based on Reinforcement Learning.

Gambler-agent is based on a Sports odds betting environment by Ory Jonay (https://github.com/OryJonay/Odds-Gym). This solution was chosen because of several reasons: first of all, it is based on RL, uses OpenAI environment, state-of-the-art algorithm (PPO2) for its' policy, and shows good performance.

This RL agent chooses the best possible outcome(s) of the proposed market, stake size, and places bet(s). The main goal of this agent is to maximize its income. As a reward agent multiplies stake size by the winning odds and subtracts the initial bet and any losses. Agent possible actions are: place a bet to this selection or not. In case of a market with 3 possible selections agent can place 1, 2 or 3 bets, or not wager to this market at all.

The agent has an initial balance of 25 euro. This amount has been chosen empirically and it is enough for the agent to "teach" how to play in order to start making income.

One step of the agent is placing a bet on a single market. After placing a bet agent gets results of stake and according to it gets some reward. The episode ends when the agent loses all money, the agent has placed bets to all markets, or when the liability of the Gambling Company has reached the set limit (see more in Business-agent chapter). [25] [29]

For evaluating the results, the solution needs to emulate not only the gaming journey of the customers who wager on tuned odds but emulate those customers, who use original odds, as well. It is needed in order to compare how the customer journey changes because of tuning odds.

There are two types of gambler agents: super gambler-agent and ordinary gambler-agent.

As it has been mentioned below (in the section about OpenAI), the OpenAI environment supports three different action and observation spaces: Discrete, Continuous (Box), and mixed. Discrete actions are a fixed set of something (acts, moves, etc.), continuous action can be any value between some range.

Gambler-agent has two types of actions: the first one is discrete - place a bet or not, the second one is stake amount. According to a 'gambling math' - a reward of gambler-agent is the multiplication of stake size by the winning odds and subtraction of the initial bet and any losses.

Proximal Policy Optimization (PPO2) algorithm is used by an agent for learning and predicting the next best action. The reasons why the author has chosen PPO2 are written in the Theoretical Framework chapter.

4.2.2 Business agent

Business agent represents Gambling Company party, which provides the possibility to place bets for its customers. A business agent is able to tune original betting odds provided by sports events provider (in proposed solution events and odds are stored in a historical data set). The business-agent main goal is to follow the gaming journey of its customers and influence customers' experience by changing (tuning) betting odds, which customers use for placing bets. To be more precise, tuning betting odds influence the decision process of the customer. This maximizes the release of the DA in order to maximize the enjoyment of the gaming process. The author of the solution takes into account previous

scientific studies about gambling, what maximizes the release of DA, and uses it. Unlike the gambler agent, the business agent has 3-dimensional Box(-1…1) action-space, which means that action value can be three any decimal number from -1 till 1.

As it was mentioned earlier (in the Theoretical Framework chapter), there are two types of actions: continuous action and discrete action. In the case of continuous action-space, it is needed to define boundaries of possible action, other case action would be any number (integer and decimal) in the world, which is not practical. Here author decided to limit possible action (=odds change value) to range from -1 to 1. From the author's experience, this range should be enough to dramatically impact on customer's decision since odds are considered as a probability or chance that something will not happen or will happen (but not exactly), and probability with a value of -1 means 100% that it will happen (since odds are the inverse of the offered probability of the outcome), and in case of 1 vice versa.

Action of the business-agent is three numbers that are the change of each odd in the market (each market has three original odds).

Example:

```
Event: Barcelona - Manchester United
Market: 1X2 (1 - Barcelona, 2 - Manchester United, X - draw)
Original odds: 1.52, 3.30, 5.70
Business agent action: [0.25, -0.2, -0.5]
Tuned odds: 1.77, 3.10, 5.20
```

One step of the agent is tuning one market for one event. The reward for this step is calculated according to the customer's experience from this game.

Agent observation contains such data as original odds of the market, company liability, company profit, customer stakes amount, and other data.

The business agent uses the same algorithm for learning and predicting the next best action - Proximal Policy Optimization (PPO2).

Since in the emulated environment each gambling company (=business-agent) has only two customers and each customer has a balance of 25 EUR (see more in Gambler-agent chapter), then the author has decided to set a liability limit of 50 EUR. Reaching the limit

would mean that company has lost the same amount of money as its' possible maximum profit.

4.3 Data

Two types of data is needed to build proposed solution: customer behavior data and betting odds data.

4.3.1 Customer behavior data

For optimizing the gaming experience there must be customer's gambling behavior data: what bets customer places, what odds customer prefers, and how he/she responds to odds changes. And finally, how customer's gaming journey changes from tuning odds. Since the author does not have the opportunity to evaluate his solution on a real customer in a real-life environment (as well as it would be expensive and risky), he proposed to build a whole environment with gambler agents, who generate synthetic behavior data and respond to tuning odds in real-time.

There are two examples of the stake objects in the system (Table 1), which could give better understanding of what RL system takes into account in order to make decisions.

Table 1 Examples of stakes objects in the system

Won stake	Lost stake
{ 'agent_type': 1, 'action': 'home', 'selection': 0, 'odd': 1.96, 'amount': 1.0, 'win_amount': 1.96, 'company_risk': 0.96,	{ 'agent_type': 1, 'action': 'away', 'selection': 2, 'odd': 3.5, 'amount': 1.0, 'win_amount': 0.0, 'company_risk': -1.0,

'position_in_array': 0 }	'position_in_array': 35 }

'agent_type' – Gambler-agent's type. Possible values: 1 – 'super' agent, who uses tuned odds; 2 – ordinary agent, who uses original odds.

'action' – Selected action for this step (=selected odd for placing a bet). Possible values: 'home', 'draw', 'away'.

'selection' – Number interpretation of the selected action. Possible values: 0, 1, 2.

'odd' – Odds of the stake (if 'agent_type', then it is tuned odd, else original odd).

'amount' – Stake amount.

'win_amount' – How much money the customer got back in respect of the event results.

'company_risk' – Company liability. How much money the company has lost (minus value means company revenue).

'position_in_array' – System's internal value.

4.3.2 Betting Odds Data

For building agents who would place bets on proposed events the author uses Historical Football Results and Betting Odds Data.

In history data, there are fulltime and halftime results for up to 22 European league divisions from 25 seasons, closing match odds (best and average market price) from multiple betting platforms. [30]

Using this data environment proposes gambling agents to place bets using historical odds. For ordinary gambler-agents environment proposes original odds from the historical dataset, for super gambler-agents environment proposes tuned odds.

Below (Table 2) is an example of aggregated betting odds data:

Table 2 Examples of aggregated betting odds data

	home_team	away_team	home	draw	away	result
1	Aston Villa	West Ham	1.96	3.3	4.03	0
2	Blackburn	Everton	2.92	3.25	2.44	0
3	Bolton	Fulham	2.2	3.26	3.32	1
4	Chelsea	West Brom	1.16	6.9	17.47	0
5	Wigan	Blackpool	1.82	3.45	4.5	2

Betting odds data contains information about teams, 1X2 (who will win the match) market odds, and the actual result of the match. There are 3 possible values of the result: 0, 1, or 2: ‘home’, ‘draw’ or ‘away’.

5 Empirical Analysis and Results

5.1 Experiment design

There are two separate datasets for gambler-agents: the first dataset is used for training and contains events from English Premier Football League, the second dataset is used for evaluation of the model and contains different events from German leagues from different years.

Customer behavior data for training business agent generates real-time during training.

In total there are six agents in the environment: two business agents (=gambling companies) and four gambler-agents. As it was mentioned earlier, gambling company should provide same odds for all customers, that is why it is needed to take into account gaming journey of multiple customers simultaneously, but not only one customer's journey. In proposed solution each gambling company has 2 customers and based on experience of both of them algorithm tunes odds.

Like in real-life scenario, business agents (=gambling companies) are independent of each other, each of them has its model, observation, and state. One business agent provides original odds to its ‘customers’ and it is related to two gambler-agents (customers), which use original odds for placing bets. The second business agent ‘has’ two customers and provides them tuned (modified by the algorithm) odds. For evaluating the results, the solution needs to emulate not only the gaming journey of the customers who wager on tuned odds but emulate these customers, who use original odds, as well. It is needed in order to compare how the customer journey changes because of tuning odds.

All agents simultaneously act and learn: gambler-agents learn how to place bets to get the maximum income and second business agent learns how to tune betting odds to maximize the enjoyment of the customer.

The author has tried different reward policies and observation datasets and proposed in this thesis solution is the most logical (according to the author’s opinion) and it has the best results.

Like in most Machine Learning models, the solution has two stages: training and evaluation stages. Results, which are provided in section 4.3 are from the evaluation stage.

The author has tried different amounts of training iterations: 1, 10, 20, 50, 100. The most significant evolution of the results is between 1, 10, and 20 iterations. There is some improvement between 20 and 50, 50, and 100 training iterations, but improvements are not significant.

Since building and training a model with the best performance is not part of this thesis, the author has decided to stop experimenting after 100 iterations. After each training set, the author made a few evaluation iterations. Episodes, which illustrate achieved results the best (to validate the proposed hypothesis), the author describes further in this thesis.

5.2 Evaluation and analysis of the results

During evaluation there are six agents in the environment who act simultaneously:

- two ordinary gambler-agents who place bets using original odds,
- ordinary business-agent who provides original odds for gambler-agents mentioned earlier,
- two (super) gambler-agents who wager using tuned odds,
- (super) business-agent who provides tuned odds.

Results of three episodes are provided for evaluating the proposed solution: one evaluation episode after 20 episodes of training, and two episodes after 50 training episodes. The first two episodes illustrate very well, that it is possible to significantly improve the gaming experience of the customer by tuning odds. Moreover, these two episodes illustrate the evolution of the algorithm performance and good ability to learn: the difference of the results of the algorithm between 20 and 50 training iterations is significant. The third episode illustrates the instability of the solution.

As it was already mentioned, there are two different datasets with different events for waging: one for training and one for evaluation.

In the beginning, each of four gambling-agents has 25 € for waging, both business agents (=gambling companies) have 0 € profit at step 1 with 50 € limit of maximum possible company liability. The episode ends in case if all gambling-agents have lost their money (with balance approx. 0 €) or if the liability of at least one gambling company is above 50€.

The first steps of both parties are clumsy, gambler-agents place bets to random odds and business agent randomly tunes odds (one of the tuned odds is <1, which is against logic):

```
Business agent current profit : -1.39

Customer agent 1: Home Team Wigan VS Away Team Blackpool. Odds for
customer: [[1.82 3.45 4.5 ]]. Bets placed: ['home', 'draw', 'away'].
Current balance at step 6: 23.49

Customer agent 2: Home Team Wigan VS Away Team Blackpool. Odds for
customer: [[1.82 3.45 4.5 ]]. Bets placed: ['home', 'draw', 'away'].
Current balance at step 6: 27.91


Super Business agent current profit : -2.28

Super Customer agent 1: Home Team Wigan VS Away Team Blackpool. Odds
for customer: [[0.82 2.82 5.5 ]]. Bets placed: ['draw']. Current balance
at step 6: 25.75

Super Customer agent 2: Home Team Wigan VS Away Team Blackpool. Odds
for customer: [[0.82 2.82 5.5 ]]. Bets placed: ['draw', 'away']. Current
balance at step 6: 26.54
```

In contrast, during the 20th iteration, we see the logic in agents' acts: gambling-agents prevailing choose not risky odds, but sometimes still a risk (experiment).

Business agents are much smarter, already on step 6, both agents have profit.

```
Business agent current profit : 3.87

Customer agent 2: Home Team Wigan VS Away Team Blackpool. Odds for
customer: [[1.82 3.45 4.5 ]]. Bets placed: ['home', 'away']. Current
balance at step 6: 22.96

Customer agent 3: Home Team Wigan VS Away Team Blackpool. Odds for
customer: [[1.82 3.45 4.5 ]]. Bets placed: ['home', 'away']. Current
balance at step 6: 23.16
```

```
Super Business agent current profit : 2.6

Super Customer agent 4: Home Team Wigan VS Away Team Blackpool. Odds
for customer: [[1.94 2.82 3.52 ]]. Bets placed: ['home', 'away']. Cur-
rent balance at step 6: 23.0

Super Customer agent 5: Home Team Wigan VS Away Team Blackpool. Odds
for customer: [[1.94 2.82 3.52 ]]. Bets placed: ['draw', 'away']. Cur-
rent balance at step 6: 24.40
```

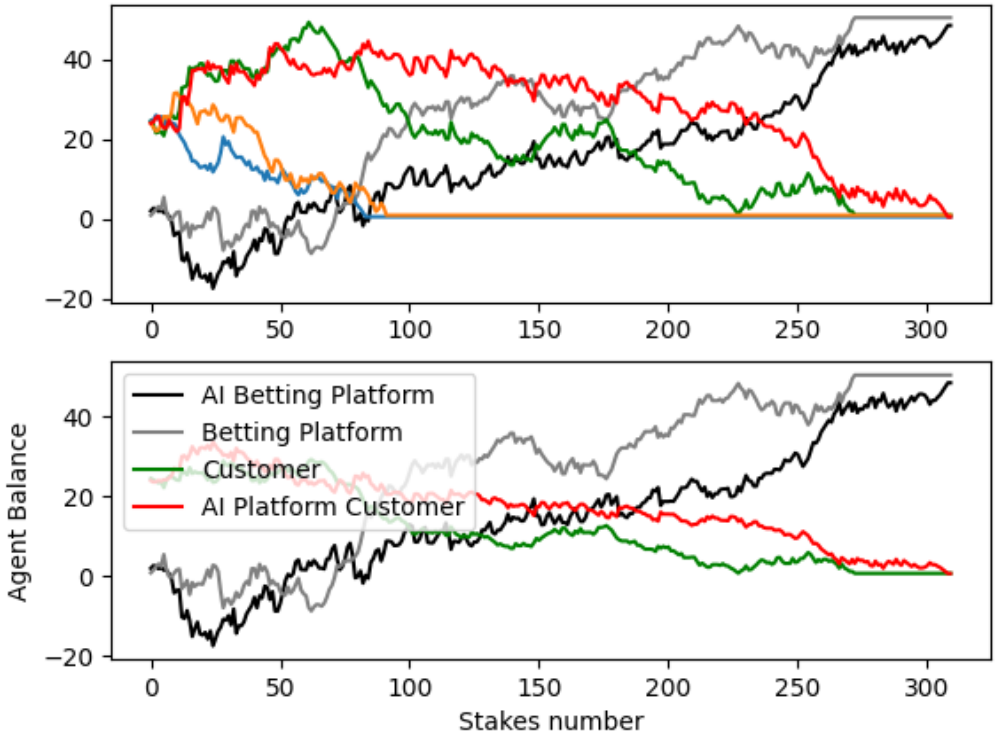

Figure 5 Solution evaluation diagram after 20 iterations

Figure 5 illustrates one evaluation episode after 20 training iterations. The upper diagram shows the progress of all agents during the whole episode: two gambler-agents (green and blue lines on the diagram) who wager using business-agent providing original odds (grey line), two (super) gambler-agents (orange and red colors) who place bets using tuned odds providing by (super) business-agent (black line on the diagram).

The diagram below shows the average values each of two gambler-agents and values of business-agents at the current step.

The whole episode picture very clearly evaluates the behavior of business agent:

In the beginning gambling company has some liability, since gambler-agents learn more quickly than business-agents.

The reason is that at the beginning business agents get the wrong ‘impression’ (data) of gambler-agents, and it takes time to re-think and re-evaluate the behavior of gambler-agents. Only after that moment, when gambler-agents have learned how to ‘intelligently' place bets (like a human being does) and start to make some income, the business agent’s true learning starts.

In the chart above it took about 75 stakes.

The episode ends on step 309, when (super) gambler-agent lost it’s all money.

Below are some statistics of the described episode:

```
-- Customer Agents:
Agent Max stakes amount in one episode: 273
Super Agent Max stakes amount in one episode: 309
Agent Average stakes amount in one episode: 178.5
Super Agent Average stakes amount in one episode: 200.5
-- Business Agents:
Super Agent Max liability in one episode: 18.55
Agent Max liability in one episode: 14.50
```

Below (Figure 6) is presented one of the evaluation episodes after 50 training episodes. This episode gives an unambiguous answer to the research question of this thesis “Is it possible to tune odds proposed by betting companies in such a way that it will increase customer gaming enjoyment?” and the answer is Yes, it can.

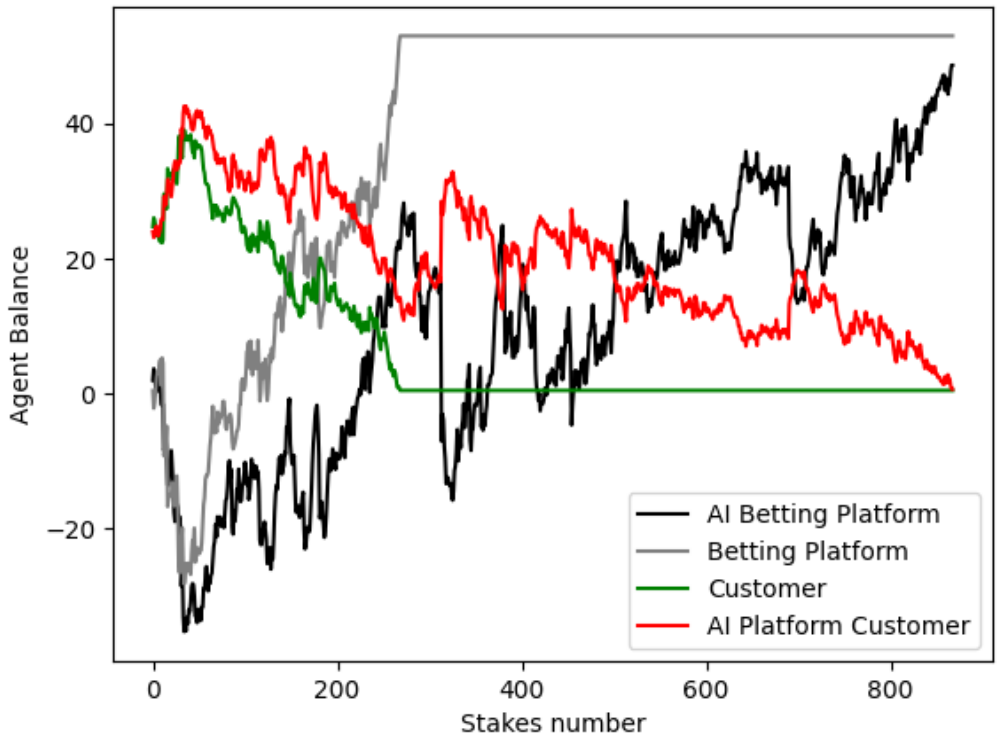

Figure 6 Solution evaluation diagram after 50 iterations

In Figure 6 we see the average balances of four gambler-agents: the red line represents an average of two (super) gambler-agents and the green line is the average balance of two ordinary gambling-agents. The given episode lasted for 867 steps and ended when (super) gambler-agent lost its money. We also see that the gaming journey of (super) gambler-agent was three times as long as the ordinary gambler-agent's and (super) gambler-agent's journey was more thrilling and had a lot of lows and highs: there were big wins and losses.

During this episode (super) gambling company had the maximum liability of 53.03 € and ordinary gambling company had the maximum liability of 48.68 €, which means that the liability difference is about 9% and it can actually be controlled by an algorithm.

Below are statistics of the described episode:

```
-- Customer Agents:
Agent Max stakes amount in one episode: 269
Super Agent Max stakes amount in one episode: 867
```

```
Agent Average stakes amount in one episode: 264.5
Super Agent Average stakes amount in one episode: 697.5
-- Business Agents:
Super Agent Max liability in one episode: 53.03
Agent Max liability in one episode: 48.68
```

In Figure 7 there is another example of an evaluation episode after 50 training iterations which shows us one of the vulnerable spots of the proposed solution: its' instability.

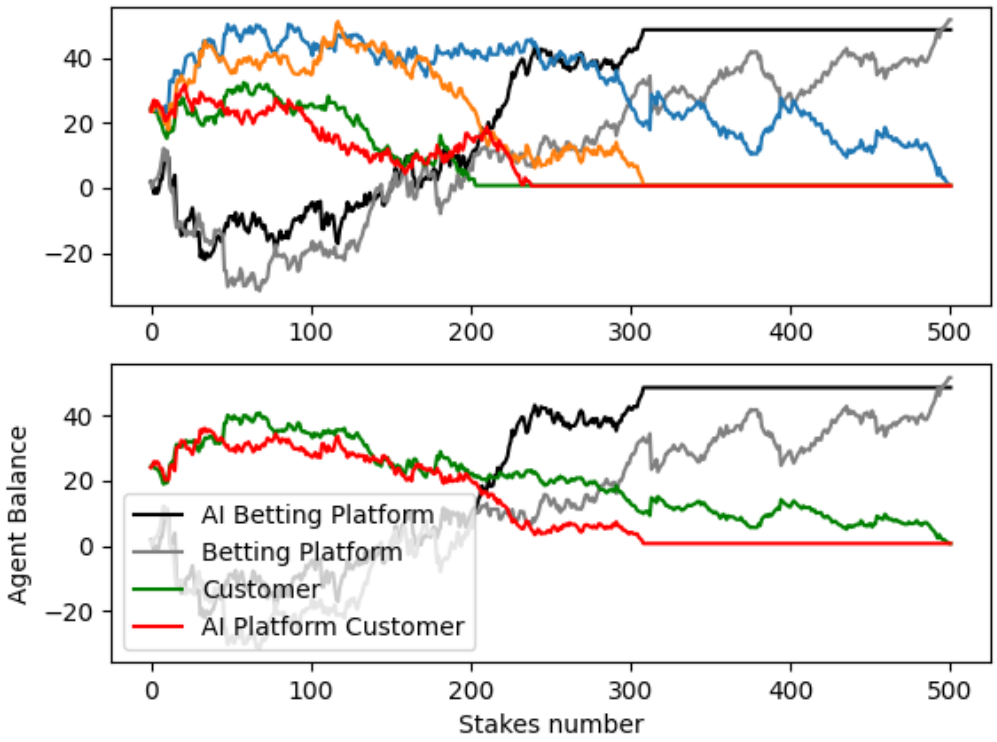

Figure 7 Solution evaluation diagram after 50 iterations: Instability case

In this example, there is the opposite picture: a journey of (super) gambler-agent which had maximum steps number is almost twice smaller than of ordinary gambler-agents maximum journey: 274 versus 500.

Below are statistics of the described episode:

```
-- Customer Agents:
Agent Max stakes amount in one episode: 500
Super Agent Max stakes amount in one episode: 309
Agent Average stakes amount in one episode: 352.0
Super Agent Average stakes amount in one episode: 274.0
-- Business Agents:
Super Agent Max liability in one episode: 20.62
Agent Max liability in one episode: 25.68
```

In order to answer research question 2 (will gambling company be still profitable if the algorithm will try to maximize customer gaming enjoyment) let's refer to the provided episodes' statistics earlier: maximum liabilities of business agents. After 20 training iterations (super) business-agent had a maximum liability of 18.55 EUR, ordinary business-agent had a maximum liability of 14.50 EUR (28% difference); after 50 training iterations agents had 9% maximum liability difference, (super) business-agent had larger liability; in case of second example (instability case), when ordinary gambler-agents had longer gaming journey, business-agents had 25% difference, this time ordinary business-agent had larger liability. Here we can make the following conclusion: for providing longer journey companies cannot just maximize income but should allow making more stakes and in some cases increase the number of won stakes, which means bigger risk and bigger liability, but there is a great chance that at the end of the day customers anyways will lose all their money. The second remark is that it is very difficult to unambiguously answer this question, because in this solution there are 'sterile' conditions where gamblers don't have any context knowledge about football teams, odds from other gambling companies, etc. In order to give an exact and confident answer solution should be tried in a real-world scenario.

5.3 Solution vulnerable spots

The proposed solution has a few known vulnerable spots which need to be improved to use this solution in real-life. The author will review some of them.

5.3.1 Risk for gambling company

The algorithm does not know anything about that Spain has had the best football team of all time and about the fact that almost everyone in Brazil plays football and it is the reason why Brazil has a very strong football team. The algorithm knows only the odds and behavior of the gamblers, and it can do absurd actions (because of our background knowledge mentioned above).

As one of such examples could be a (fictional) match between Brazil - Blackpool with original odds [1.01 - 29.0 - 50.0]. It is obvious who will be the winner here, but theoretically, the algorithm could propose something like [1.80 - 28 - 50], which could become easy meat for gamblers and a big loss for a gambling company.

5.3.2 Results instability

As it was shown earlier, some episodes fail success of the solution, but the author of this thesis believes, that it is not a problem - since there are two competitive parties, and both parties are learning with each iteration and since both agent types use the same algorithm, their capabilities are approximately the same. But, it still can be a problem, since it is hard to evaluate.

5.4 Additional risks for gambling business

As it has been discussed earlier, the proposed solution gave new insight to the author (seems self-evident fact, but it has been confirmed by algorithm analysis as well) while answering the second research question (will gambling company be still profitable if the algorithm will try to maximize customer gaming enjoyment): for providing longer journey companies cannot just maximize income but should allow making more stakes and in some cases increase the number of won stakes, which means bigger risk and bigger liability, but there is a great chance that at the end of the day customers anyways will lose all their money. The proposed solution brings more risks for gambling

companies, but it does not mean that companies cannot use the proposed algorithm as part of their business. All risks are manageable.

According to American Gaming Associate [31] gambling companies have a very good profit from their businesses and it means that there is a lot of room for playing. The author suggests experimenting, but keep in mind that all risks related to this algorithm should be managed and (at least in the beginning) its' behavior should be attentively monitored.

5.5 Solution possible benefits and value for gambling business

The main purpose of the proposed solution is to optimize odds in such a way that it would be more attractive for its' customers by automating the work of traders. The author sees that one side effect of this optimization could be the loyalty of the customers. Let's review these two benefits in more detail.

5.5.1 Automatization of tuning odds

Automatization traders'work will give huge value to the gambling business. First of all, the algorithm is able to react to odds changes, company risk, customers' behavior, etc. almost instantly, and tune odds more precisely than traders. Secondly, the proposed solution could reduce labor costs by partly automating the trading department. Finally, the proposed solution could provide unique odds for its' customers.

5.5.2 Customers Loyalty

Since the main purpose of tuning odds is to influence the decision process of the customer and maximize the enjoyment of the gaming process (by maximizing the release of the DA), the proposed solution will give gamblers a unique experience from the gambling process, which could increase the loyalty of customers.

The proposed solution is first of a kind (at least the author did not find any case of usage of similar solutions) and that is why the solution needs to be validated in a real-life scenario to validate the author's assumptions.

5.6 Further development

Since the author has empirically proved the hypothesis that it is possible to manipulate customers' behavior by tuning odds and by doing it maximize gaming process enjoyment, solution "as it is" has completed its purpose. As the next step solution has to be adjusted for a real-life scenario.

5.6.1 More stable model

As the first step author proposes to improve the stability of the model.

At the moment for the same odds model proposes different odds, and in some cases, the difference is multiple, which is unacceptable for the real-life scenario.

The author sees multiple possibilities:

- Observation-space review and update. Business-agent has to exactly “understand” what actions what observation values update and how action value influences observation values. Possible that some extra observation values are needed, or the opposite: may be there is some ‘noise’ which needed to be removed from the observation.

- Reward logic update for business-agent. Giving a reward to the agent for the stability: calculate average odds difference and give positive reward for following this average value.

- Applying “experience replay". Experience replay was introduced by Google's DeepMind in the “Human-level control through deep reinforcement learning” research paper [32]. Researchers propose that “instead of running Q-learning on state/action pairs as they occur during simulation or actual experience, the system stores the data discovered for [state, action, reward, next_state] - typically in a large table.[..] In DQN, the DeepMind team also maintained two networks and switched which one was learning and which one feeding in current action-value estimates as "bootstraps". This helped with stability of the algorithm”. [33]

5.6.2 Feedback loop

As the second step author proposes to implement a feedback loop for the algorithm.

Let's suppose there is a model that is trained on emulated gambler-agents in a 'sterile' world. Real-life gamblers have different gambling behavior, tactic and they could react to odds changes in a different way. The algorithm needs to be able to quickly adapt to a changing environment and the reaction of the agents (gamblers). For doing that some feedback loop should be implemented to the proposed solution. There is one challenge for doing that: the agent cannot receive an immediate reward (as it was in an emulated environment) since there will no results of the match at the moment of tuning odds by business-agent (match is still on at that moment). That is why some kind of 'delayed' training logic should be implemented.

As one of the possible solutions could be to apply model-based Reinforcement Learning. "Reinforcement learning systems can make decisions in one of two ways. In the model-based approach, a system uses a predictive model of the world to ask questions of the form "what will happen if I do x?" to choose the best x1. In the alternative model-free approach, the modeling step is bypassed altogether in favor of learning a control policy directly" [34]

By using this approach solution would have its own model of the environment for policy learning using the model data.

6 Conclusions

Enjoyment of the gaming process is crucial for the success of every product in the gaming field. Gambling companies combine odds events from multiple odds providers to enrich event selection and to provide the best odds for their customers. From the author's personal experience, some gambling companies have a special trading department, where traders manually “tune” odds for managing risks.

The main goal of the thesis is to understand if it is possible to maximize the enjoyment of the gambling process of the customer by tuning betting odds using the Reinforcement Learning field of Machine Learning. For validating the results author has created a system with multiple Reinforcement Learning autonomous agents sharing a common environment. In the environment, there are two business-agents (gambling companies) which provide events and odds for placing bets (one agent provides tuned odds, second – original odds), and multiple customer agents: for the first type of agents odds are not changed, for the second type RL policy tunes odds for maximizing their experience.

By analyzing the results of the proposed system author has confirmed the proposed hypothesis (yes, it is possible to manipulate customers’ behavior and maximize their enjoyment by using RL) by using simulation as close to a real-life scenario as possible. While answering the second research question (will gambling company be still profitable if the algorithm will try to maximize customer gaming enjoyment) author has confirmed that for providing longer journey companies cannot just maximize income but should allow making more stakes and in some cases increase the number of won stakes. The proposed solution brings more risks for gambling companies, but all risks are manageable.

The main contributions of this thesis are:

- a novel method of tuning betting odds in the sports betting field;
- Reinforcement Learning solution for automatization of the process of tuning betting odds.

Proposed in this thesis novel method could bring the original idea of sports betting back: primarily, it is entertainment and a way of getting enjoyment from the gaming process for the customers, but not a way for gambling companies for making money. In the best scenario, the proposed method could be a win-win game for both parties: customers could get much more enjoyment from the gaming process and companies could get the loyalty of the customers and reduced expenses in return.

References

[1] P. &. R. M. Anselme, "What motivates gambling behavior: Insight into dopamine's role.," 2013. [Online]. Available: https://www.researchgate.net/publication/259353550_What_motivates_gambling_behavior_Insight_into_dopamine%27s_role.

[2] "Managed Trading Services," [Online]. Available: https://mts.betradar.com.

[3] R. A. Epstein, "The Theory of Gambling and Statistical Logic, Second Edition," Academic Press is an imprint of Elsevier, 2009, p. 43.

[4] M. Robert M. Doroghazi, "The Theory of Gambler's Ruin and Your Investments," *EDITORIAL,* vol. 125, no. 4, 2020.

[5] R. A. Epstein, "The Theory of Gambling and Statistical Logic Second Edition," Academic Press is an imprint of Elsevier, 2009, pp. 47-52.

[6] M. J. F. R. Patrick Anselme, "What motivates gambling behavior? Insight into dopamine's role," *Frontiers in Behavioral Neuroscience,* vol. 7, 2013.

[7] F. T. Rony Bunker, "A machine learning framework for sport result prediction," *Applied Computing and Informatics,* vol. 15, no. 1, pp. 27-33, 2019.

[8] P. Rajan and K. P. Miyapuram, "PsychFM: Predicting your next gamble," in *International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom, 2020.

[9] M. L. J. P. Karl Tuyls, "Game-theory insights into asymmetric multi-agent games," Google DeepMind, 17 01 2018. [Online]. Available: https://deepmind.com/blog/article/game-theory-insights-asymmetric-multi-agent-games. [Accessed 06 12 2020].

[10] "An effective maximum entropy exploration approach for deceptive game in reinforcement learning," *Neurocomputing,* vol. 403, pp. 98-108, 2020.

[11] H. Lahtinen, "When do betting odds best represent the actual outcomes? Predicting NHL results based on moneyline odds movement," 2019. [Online]. Available: https://aaltodoc.aalto.fi/bitstream/handle/123456789/42669/master_Lahtinen_Henri_2019.pdf?sequence=1&isAllowed=y,%20p.11. [Accessed 06 12 2020].

[12] Knowledge Sourcing Intelligence LLP, "Sports Betting Market - Forecasts from 2020 to 2025," https://www.researchandmarkets.com/reports/5138739/sports-betting-market-forecasts-from-2020-to, Global, 2020.

[13] R. S. S. a. A. G. Barto, "Reinforcement Learning: An Introduction," p. 1.

[14] R. S. S. a. A. G. Barto, "Reinforcement Learning: An Introduction," p. 25.

[15] R. S. S. a. A. G. Barto, "Reinforcement Learning: An Introduction," pp. 47-48.

[16] M. Lapan, "Deep Reinforcement Learning Hands-On Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more, 2nd Edition," p. 23.

[17] R. S. S. a. A. G. Barto, "Reinforcement Learning: An Introduction," p. 50.

[18] M. Lapan, "Deep Reinforcement Learning Hands-On Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more, 2nd Edition," p. 9.

[19] M. Lapan, "Deep Reinforcement Learning Hands-On Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more, 2nd Edition," p. 23.

[20] OpenAI, "About OpenAI," [Online]. Available: https://openai.com/about/. [Accessed 06 12 2020].

[21] OpenAI, "Gym," [Online]. Available: https://gym.openai.com. [Accessed 06 12 2020].

[22] M. Lapan, "Deep Reinforcement Learning Hands-On Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more, 2nd Edition," p. 30.

[23] M. Lapan, "Deep Reinforcement Learning Hands-On Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more, 2nd Edition," pp. 30-31.

[24] R. B. a. B. D. S. L. Bus ̧oniu, "Multi-agent reinforcement learning: An overview. Chapter 7 in Innovations in Multi-Agent Systems and Applications – 1," pp. 183-221.

[25] OpenAI, "Emergent Tool Use from Multi-Agent Interaction," OpenAI, 17 09 2019. [Online]. Available: https://openai.com/blog/emergent-tool-use/. [Accessed 06 12 2020].

[26] M. v. O. Marco Wiering, "Reinforcement Learning: State-of-the-Art," p. 443.

[27] OpenAI, "Proximal Policy Optimization," 20 07 2017. [Online]. Available: https://openai.com/blog/openai-baselines-ppo/. [Accessed 06 12 2020].

[28] "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," [Online].

[29] O. Jonay, "Welcome to oddsgym's documentation!," [Online]. Available: https://oryjonay.github.io/Odds-Gym/. [Accessed 06 12 2020].

[30] "Historical Football Results and Betting Odds Data," [Online]. Available: https://www.football-data.co.uk/data.php. [Accessed 06 12 2020].

[31] The American Gaming Association, "97% of Expected $10 Billion Wagered on March Madness to be bet Illegally," 12 03 2018. [Online]. Available: https://www.americangaming.org/new/97-of-expected-10-billion-wagered-on-march-madness-to-be-bet-illegally/. [Accessed 06 12 2020].

[32] K. K. D. S. A. A. R. J. V. M. G. B. A. G. M. R. A. K. F. G. O. S. P. C. B. A. S. I. A. H. K. D. K. D. W. S. L. &. D. H. V. Mnih, "Human-level control through deep reinforcement learning," *NATURE,* vol. 518, 2015.

[33] fuglede, "What is "experience replay" and what are its benefits?," Stack Exchange Inc., 29 11 2018. [Online]. Available: https://datascience.stackexchange.com/questions/20535/what-is-experience-replay-and-what-are-its-benefits. [Accessed 06 12 2020].

[34] M. Janner, "Model-Based Reinforcement Learning: Theory and Practice," Berkey Artificial Intelligence Research, 12 12 2019. [Online]. Available: https://bair.berkeley.edu/blog/2019/12/12/mbpo/. [Accessed 06 12 2020].

[35] M. Lapan, "Deep Reinforcement Learning Hands-On Apply modern RL methods to practical prob-lems of chatbots, robotics, discrete optimization, web automation, and more, 2nd Edition," p. 765.

[36] O. Jonay, "Theory behind the environment," [Online]. Available: https://oryjonay.github.io/Odds-Gym/docsrc/theory.html. [Accessed 06 12 2020].

[37] OpenAI, "openai/multiagent-particle-envs," GitHub, Inc., [Online]. Available: https://github.com/openai/multiagent-particle-envs. [Accessed 06 12 2020].

[38] I. K. T. M. Y. W. G. P. B. M. I. M. Bowen Baker, "Emergent Tool Use From Multi-Agent Autocurricula," 17 09 2019. [Online]. Available: https://arxiv.org/abs/1909.07528. [Accessed 06 12 2020].

[39] Y. W. A. T. J. H. P. A. I. M. Ryan Lowe, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," 07 06 2017. [Online]. Available: https://arxiv.org/abs/1706.02275. [Accessed 06 12 2020].

[40] R. B. a. B. D. S. L. Bus ̧oniu, "Chapter 7 in Innovations in Multi-Agent Systems and Applications – 1 (D. Srinivasan and L.C. Jain, eds.), vol. 310 of Studies in Computational Intelligence," in *Multi-agent reinforcement learning: An overview*, Berlin, Germany, 2010, p. 183–221.

Appendix 1 – Proposed solution architecture