

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická

Draftování v MOBA hrách s pomocí strojového učení

Petr Šiška

Školitel: Ing. Ondřej Hubáček

Obor: Otevřená informatika

Zaměření: Základy umělé inteligence a počítačových věd

Květen 2021

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šiška** Jméno: **Petr** Osobní číslo: **483649**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra kybernetiky**
Studijní program: **Otevřená informatika**
Specializace: **Základy umělé inteligence a počítačových věd**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Draftování v MOBA hrách s pomocí strojového učení

Název bakalářské práce anglicky:

Drafting in MOBA Games using Machine Learning

Pokyny pro vypracování:

Profesionální hraní počítačových her se v posledních letech těší vzrůstající oblibě, které byla v nedávných časech posílena nedostupností klasických sportů. Mezi nejhranější herní žánry patří tzv. MOBA hry. V MOBA hrách soupeří dva týmy hráčů, z nichž každý ovládá jeden virtuální charakter. Charaktery pro určitý zápas jsou vybírány (draftovány) ze společné nadmnožiny. Tento výběr je podobný výběru sestavy v klasických týmových sportech a je zcela zásadní pro průběh hry. Cílem této práce bude navrhnout a implementovat modely predikující vítěze zápasu a s jejich pomocí optimalizovat draftování.

1. Představte a vysvětlete důležitost draftování v MOBA hrách.
2. Prozkoumejte state-of-the-art v draftování s pomocí strojového učení.
3. Vyberte vhodnou MOBA hru a sesbírejte relevantní data.
4. Navrhněte, implementujte a vyhodnoťte modely pro predikci vítěze zápasu ve fázi před/po draftu.
5. Navrhněte a implementujte systém pro asistenci při draftu.

Seznam doporučené literatury:

- [1] Chen, Z., Nguyen, T.H.D., Xu, Y., Amato, C., Cooper, S., Sun, Y. and El-Nasr, M.S., 2018, September. The art of drafting: a team-oriented hero recommendation system for multiplayer online battle arena games. In Proceedings of the 12th ACM Conference on Recommender Systems (pp. 200-208).
- [2] Summerville, A., Cook, M. and Steenhuisen, B., 2016, September. Draft-analysis of the ancients: predicting draft picks in dota 2 using machine learning. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 12, No. 1).
- [3] Makarov, I., Savostyanov, D., Litvyakov, B. and Ignatov, D.I., 2017, July. Predicting winning team and probabilistic ratings in "Dota 2" and "Counter-Strike: Global Offensive" video games. In International Conference on Analysis of Images, Social Networks and Texts (pp. 183-196). Springer, Cham.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Ondřej Hubáček, Intelligent Data Analysis FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **07.01.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Ondřej Hubáček
podpis vedoucí(ho) práce

prof. Ing. Tomáš Svoboda, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu této bakalářské práce panu Ing. Ondřeji Hubáčkovi za cenné rady, ochotu a předané odborné znalosti, které pro tuto práci byli neocenitelné.

Také bych rád poděkoval rodině a přátelům za dlouhodobou podporu mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškerou použitou literaturu v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 21. května 2021

Abstrakt

Tato bakalářská práce se věnuje problému draftování v MOBA hrách. Zabýváme se predikcí vítěze zápasu MOBA hry s pomocí neuronových sítí.

Navrhli a implementovali jsme embedding modely, které využívají informace o vybraných šampionech a příznacích hráčů. U embedding modelů jsme dosáhli přesnosti až 54 % a překonali tak apriorní pravděpodobnost výhry modrého týmu 50,43 %. Tyto modely jsme následně použili při draftu v asistenčním algoritmu doporučující vhodné šampiony. Systém jsme aplikovali na draftování ve hře League of Legends, jakožto nejhranější a nejpopulárnější MOBA hru dnešní doby. Náš asistenční algoritmus dokázal doporučit šampiony na základě již zvolených šampionů a příznaků hráčů podle našeho očekávání.

Klíčová slova: MOBA hry, League of Legends, Strojové učení, Neuronové sítě, Asistence u draftu, MCTS

Školitel: Ing. Ondřej Hubáček
Intelligent Data Analysis FEL

Abstract

This bachelor thesis focuses on the problem of drafting in MOBA games. We predict the winner of the MOBA game match with the help of neural networks.

We designed and implemented embedding models that use information about selected champions and player features. For embedding models, we achieved an accuracy of up to 54 % and thus exceeded the a priori probability of blue team winning 50,43 %. We then used these models in a draft assistance algorithm recommending suitable champions. We applied the system to drafting in the game League of Legends, the most played and most popular MOBA game nowadays. Our assistance algorithm was able to recommend champions based on already selected champions and player features according to our expectations.

Keywords: MOBA Games, League of Legends, Machine Learning, Neural Networks, Draft assistance, MCTS

Title translation: Drafting in MOBA Games using Machine Learning

Obsah

1 Úvod	1		
1.1 MOBA hry	1		
2 League of legends	3		
2.1 Průběh hry	3		
2.2 Mapa	3		
2.3 Šampioni	4		
2.4 Hráč	5		
2.5 Motivace výběru League of Legends	5		
2.5.1 Prize pool	6		
2.5.2 Počty aktivních hráčů	6		
2.5.3 Počet diváků na streamovacích platformách	6		
2.6 Předzápasová fáze	7		
2.6.1 Průběh draftu	7		
2.6.2 Pořadí výběru a zákazu šampionů	8		
3 Přehled literatury	9		
3.1 Draft	9		
3.2 Spojitosti mezi šampiony	10		
3.3 Před hrou	10		
3.4 Související ML	11		
4 Data	13		
4.1 Analýza dat	14		
5 Modely pro predikce výsledku zápasu	17		
5.1 Architektura embedding modelů	17		
5.2 Předzpracování dat	18		
5.3 Implementace modelů	18		
5.4 Optimalizace hyperparametrů	19		
6 Asistence při draftu	23		
6.1 Monte Carlo Tree Search UCT	23		
6.2 Draftování jako hra s neúplnou informací	26		
6.3 Redukce možných voleb	27		
7 Experimenty	29		
7.1 Experiment predikce před draftem	29		
7.2 Výběr modelů	29		
7.3 Experimenty embedding modelů	31		
7.3.1 Výsledky modelů na testovací množině	32		
7.4 Příklady využití asistenčního systému	33		
7.4.1 Obsazení chybějících rolí	33		
7.4.2 Doporučení šampiona na roli proti soupeři	34		
7.4.3 Reakce algoritmu na příznaky hráčů	35		
8 Závěr	39		
8.1 Možná rozšíření	39		
Literatura	41		

Obrázky

2.1 Schématický náčrt mapy MOBA hry League of Legends.	4
2.2 Prize pool historie od roku 2016 do roku 2020 pro hry League of Legends, DOTA 2 a CS:GO.	6
2.3 Počet diváků nejsledovanějších her na platformě Twitch v miliardách.	7
2.4 Pořadí výběru šampionů v hodnocených zápasech.	8
4.1 Prvních dvacet pět šampionů s nejlepší poměrem výher ku odehraným zápasům.	15
4.2 Prvních dvacet pět nejčastějších šampionů v zápasech.	15
4.3 Prvních dvacet pět nejvíce zakázaných šampionů.	16
4.4 Procentuální vyjádření počtu hráčů hrajících více rolí.	16
5.1 Graf průměrných přesností u jednotlivých hyperparametrů pro <i>champ_model</i> na validační množině.	20
5.2 Graf průměrných přesností u jednotlivých hyperparametrů pro <i>feat_model</i> na validační množině.	21
6.1 Náčrt stromu reprezentující postup vybírání šampionů v draftovací fázi.	24
6.2 Schéma algoritmu MCTS.	26
7.1 Graf maximálních přesností u jednotlivých hyperparametrů pro <i>champ_model</i> na validační množině.	30
7.2 Graf maximálních přesností u jednotlivých hyperparametrů pro <i>feat_model</i> na validační množině.	31

Tabulky

4.1 Vybrané příznaky ze záznamu odehraného zápasu.	13
5.1 Tabulka rozsahu hodnot k jednotlivým hyperparametrům	19
7.1 Tabulka nejlepší kombinace hyperparametrů na validační množině pro <i>champ_model</i>	30
7.2 Tabulka nejlepších kombinace hyperparametrů na validační množině pro <i>feat_model</i>	31
7.3 Šampioni doporučení našim systémem na potřebnou roli <i>top</i>	33
7.4 Šampioni doporučení našim systémem na potřebné role <i>top</i> a <i>jungle</i>	34
7.5 Šampioni doporučení našim systémem na roli <i>mid</i>	35
7.6 Šampioni doporučení našim systémem na roli <i>mid</i> proti soupeři.	35
7.7 Šampioni doporučení našim systémem na roli <i>top</i> s vysokou hodnotou <i>přijatého poškození</i> u hráče.	36
7.8 Šampioni doporučení našim systémem na roli <i>top</i> s vysokou hodnotou <i>způsobeného poškození</i> u hráče.	37

Kapitola 1

Úvod

Esport je elektronický sport, který spočívá v hraní videoher na profesionální úrovni. V této práci se zaměříme na herní žánr MOBA (multiplayer online battle arena), což je podžánr strategických her. V MOBA hrách hrají proti sobě dva týmy o pěti hráčích, z nichž každý ovládá jeden virtuální charakter. Základem zápasů je výběr (draft) charakterů ze společné nadmnožiny. Cílem této práce je navrhnout a implementovat modely predikující vítěze zápasu a s jejich pomocí vytvořit asistenci k draftování.

1.1 MOBA hry

V dnešní době jsou MOBA hry jedny z nejpoblárnějších počítačových her. Je jich celá řada a hrají se nejen na počítačích (League of Legends, Dota 2, Heroes of the Storm), ale i na konzolích (Smite, Battleborn) a mobilních telefonech (Honor of Kings, League of Legends Wild Rift).

MOBA hry mají své společné rysy, díky kterým se těší takové oblíbě u hráčů. Základním rysem je výběr vhodného šampiona (virtuální charakter), který má své unikátní schopnosti. Hráč pak za něj hraje po zbytek celého zápasu. Další důležitou částí je od začátku zápasu získávat zdroje pro svého šampiona, aby byl silnější. Nakonec společně se svými spoluhráči strategicky obehřát soupeře, postupně získat výhodu a na závěr zničit nepřátelskou základnu a vyhrát tak celý zápas.

V raných začátcích MOBA her hráči byli svědky malých rozdílů mezi jednotlivými hrami. To se časem změnilo a dnes už má každá MOBA hra své unikátní vlastnosti, kterými se kromě společných rysů velmi liší. Proto MOBA hry budeme v této práci zkoumat zvlášť. Zaměříme se na hru League of Legends z důvodu expertních znalostí, které v této hře máme.

Kapitola 2

League of legends

League of Legends (LoL) je jedna z aktuálně nejhranějších i nejsledovanějších počítačových her. Je to hra žánru MOBA vydána na podzim roku 2009 společností RIOT GAMES, přičemž se vývojáři nechali inspirovat podobným herním titulem Defense of the Ancients (Dota) od společnosti Blizzard Entertainment.

Jak jsme již zmínili výše, tak LoL má spoustu svých unikátních vlastností, kterými se odlišuje od zbylých MOBA her. Pojďme si podrobněji představit důležité znaky LoL, přičemž podstatnou částí jsou sdílené rysy MOBA her.

2.1 Průběh hry

V této práci se budeme zabírat hodnocenými zápasy, kde každý hráč je zařazen do divize podle *Elo* ratingu (hodnocení hráče). *Elo* je rating, který zastupuje zkušenost hráče a jeho výsledky zápasů proti hráčům se stejným *Elo* ratingem.

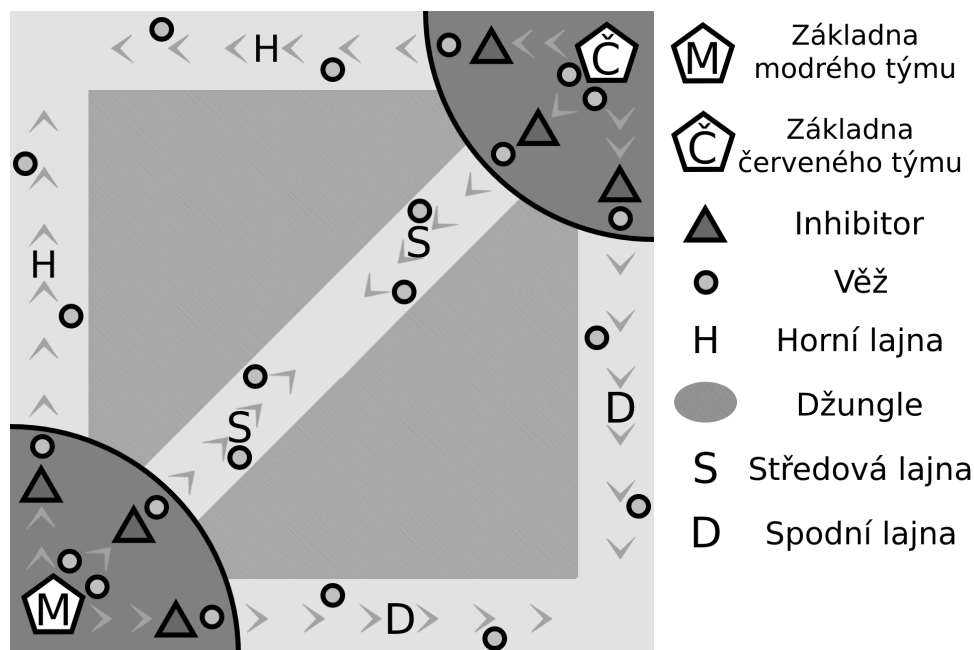
Na začátku se vytvoří dva co nejvíce vyrovnané týmy (nejpodobnější divize či *Elo*) a každý hráč si vybere šampiona, za kterého pak po zbytek hry hraje. Cílem hry je zničit nepřátelskou základnu, ke které se tým dostane až po postupném pročištění alespoň jedné ze tří linií. Každou půlminutu se rodí po skupinkách jednotky ovládané počítačem, které pomáhají svému týmu k vítězství a pohybují se po každé lajně směrem k nepřátelské základně. Na každé lajně stojí v cestě tři věže, které působí poškození nepřátelským jednotkám a šampionům. Za třetí věží je budova zvaná Inhibitor, která po zničení poskytuje týmu, který ji zničil, silnější jednotky po určitý čas v dané linii a poté se budova sama opraví. Samotnou základnu pak stráží dvě věže.

2.2 Mapa

Mapa League of Legends je symetrická podle středu a společně s hrou DOTA 2, jako dvě nejpopulárnější MOBA hry, mají mapu čtvercovou 2.1. V levém dolním rohu je základna modrého týmu a v protějším rohu je základna červe-

ného týmu. Základny spojují horní lajna, středová lajna a spodní lajna. Mezi lajnami se nachází džungle, ve které jsou neutrální příšery a rostliny. Některé po zabití dávají hráčům unikátní bonusy (vylepšení atributů šampiona) buď na celou hru, nebo jen na určitou časovou dobu.

Hráči hrající za šampiona vidí pouze na určitou vzdálenost kolem něj, stejně tak je to i na mapě, kterou hráč při hře vidí po celou dobu zmenšenou v rohu obrazovky. Zároveň ještě vidí omezenou vizi okolo přátelských věží, spoluhráčů a jednotek.



Obrázek 2.1: Schématický náčrt mapy MOBA hry League of Legends.

2.3 Šampioni

Každý šampion má své unikátní pasivní a aktivní schopnosti a kouzla. Každá schopnost po použití potřebuje určitou dobu, než bude znovu k dispozici. Šampioni v průběhu hry získávají zkušenosti a zlatáky za zabíjení nepřátelských šampionů, jednotek, neutrálních příšer v džungli a za ničení věží protivníka. Získanými zkušenostmi se zvyšuje úroveň šampiona, čímž mu nabydou základní atributy (získá tím například bonusové základní body do zdraví, poškození a obrany). Za každou novou úroveň se na šampionovi může vylepšit jedna schopnost. U většiny šampionů je jedna schopnost ultimátní, zpravidla je nejsilnější a má mnohem vyšší dobu obnovy než zbylé schopnosti. Ultimátní schopnost je dostupná až od vyšší úrovně, většinou od 6. úrovně. Na začátku zápasu mají všichni šampioni úroveň 1 a každý může dosáhnout stejné maximální úrovně. Za získané zlatáky se dají koupit různé předměty v obchodě, který se nachází za základnou. Předměty přidávají

šampionovi určité atributy, některé se postupně tvoří z jiných předmětů a ty pak zpravidla mají speciální efekty, které zvýhodňují svého vlastníka.

2.4 Hráč

Hráč je uživatel, který si v předzápasové části zvolí šampiona, za kterého odehraje celý zápas. Před začátkem samotného vyhledávání zápasu, při kterém matchmaking systém hledá deset hráčů na stejné úrovni pro složení dvou vyrovnaných týmů, si hráč volí svou preferovanou a jednu záložní roli. Rolí je myšleno: *top* (horní linie), *jungle* (džungle), *mid* (středová linie), *ADC* a *support* (spodní linie). Je typické, že si hráč oblíbí jednu roli, která jej nejvíce baví hrát nebo na které podává nejlepší výkon. Zároveň na roli, kterou si hráč volí jako záložní, je zběhlý a obecně můžeme říct, že většina hráčů převážně hraje na jedné až dvou rolích.

Volba šampiona, který má své unikátní schopnosti, je často úzce spjata s rolí, a také je ovlivněna již vybranými šampiony spoluhráčů či protivníků. Hráč má často na své primární roli naučených několik šampionů, se kterými má nejvíce zkušeností. Pokud má příležitost, tak většinou volí toho, se kterým je v aktuální verzi hry nejsilnější. Ne vždy je to tak jednoduché. Pokud hráč nevybírá jako první, musí reagovat na výběr ostatních hráčů. Jelikož se všichni šampioni nehrají na každé roli, může hráč poznat, pro jaké role protivník šampiony zvolil a pokusit se tak vybrat šampiona, který je proti šampionovi protivníka na dané roli silnější. Hráč má také možnost místo svého oblíbeného šampiona vybrat jiného, se kterým si sice není tak jistý, ale s kombinací šampionů jeho spoluhráčů budou mít výhodu jako celek.

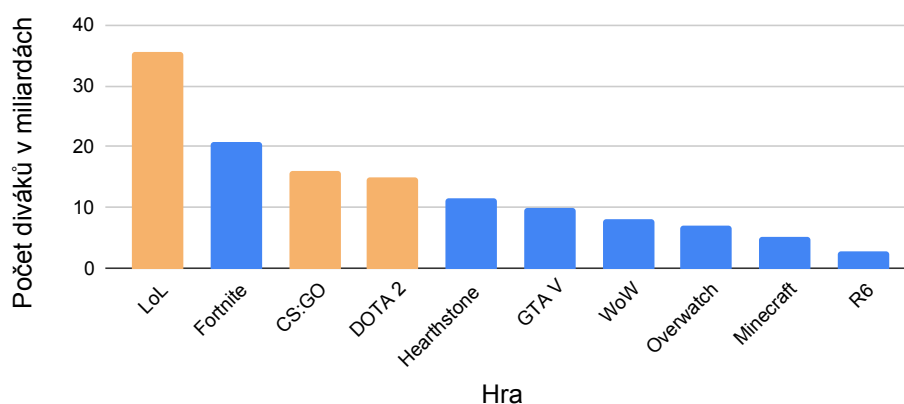
Hráč si tedy v předzápasové fázi vybírá šampiona podle toho, na které roli bude hrát, jak bude týmu přínosný a jak dokáže získat výhodu nad soupeři. Dále si hráč ze stejného výběru pro každého zvolí dvě Vyvolávačova kouzla a runy. Vyvolávačovo kouzlo je schopnost s unikátními efekty a runy přidávají šampionovi bonusové atributy nebo jiná zvýhodnění do zápasu. Vyvolávačova kouzla i runy jsou voleny v závislosti na šampionovi a preferencích hráče.

2.5 Motivace výběru League of Legends

League of Legends je velice populární hra, kterou denně hrají miliony hráčů z celého světa. Také má rozsáhlou profesionální scénu, ve které soutěží mnoho velkých známých eSportových týmů, které každý rok bojují o titul mistra světa na světovém šampionátu. O možnost zahrát si na nejprestižnějším turnaji roku bojují desítky týmů v průběhu roku v regionálních ligách.

Profesionální týmy mají celou řadu analytiků, aby přinejmenším drželi krok s ostatními týmy. V této práci se zabýváme především draftem (předzápasovou

Legends v jeden okamžik sledovalo na platformách Twitch a YouTube téměř 3,9 milionů diváku.



Obrázek 2.3: Počet diváků nejsledovanějších her na platformě Twitch v miliardách. Zvýrazněné hry jsou ty, které se objevily již v grafu 2.2.

2.6 Předzápasová fáze

Před zahájením samotného zápasu proběhne předzápasová fáze (draft), jejímž cílem je pro hráče si vybrat šampiona, Vyvolávačova kouzla a runy. Je to velice důležitá součást hry, která může značně ovlivnit následující průběh zápasu.

V této práci se touto fází zabýváme. Naším cílem je vytvořit asistenční algoritmus pro výběr šampionů a dokázat předpovědět vítěze podle informací z proběhlého draftu

2.6.1 Průběh draftu

Před samotným zahájením draftu se vytvoří dva týmy po pěti hráčích, kde jeden tým je na modré straně a druhý na červené. Pořadí hráčů je od začátku dané. Předzápasová fáze se skládá ze zakázání pěti šampionů (zakázaného šampiona již nebude možné vybrat) a vybrání pěti šampionů pro oba týmy. V zápase se žádný šampion nemůže objevit více jak jednou (pokud hráč vybere šampiona, pro ostatní tato volba nebude již možná).

Zakázání a vybrání šampiona probíhá střídavě mezi týmy a tyto volby vidí i druhý tým ihned. Spoluhráči mezi sebou mohou komunikovat pomocí chatu nebo hlasového hovoru. Každý hráč hraje za jednoho šampiona. Jelikož je pořadí hráčů ve vybírání stanovené hrou, je zde strategická možnost, že si spoluhráči mohou vybrané šampiony prohodit. Tento tah se často využívá v případech, kdy hráč role hlavního článku týmu by musel vybírat jako jeden

Kapitola 3

Přehled literatury

V poslední době přibývá prací zabývajících se eSportem. Je zde mnoho zajímavých, které se zaměřují na MOBA hry, na téma draftu a predikce před zápasem. Dále jsou zde práce, které používají zajímavé metody strojového učení, které určitě stojí za zmínku.

3.1 Draft

Předpovědi průběhu draftu v profesionálních zápasech se zabývala práce [20] pomocí metod strojového učení bayesovské sítě a LSTM. Obě metody si při výběru prvních šampionů vedli podobně, ale s postupem draftu je LSTM až 2,4x lepší. Dosáhli v 11 % úspěšné predikce a následně provedli porovnání s experty, kteří si vedli lépe s více než 30% přesností.

Predikce, jaký šampion bude vybráný nebo zakázaný v závislosti na již zvolených šampionech, je velice užitečné jak pro aktuální volbu, tak pro budoucí volby spoluhráčů a protivníků. Tímto se zabývala práce [26] s použitím metody Bi-LTSM a dospěla k přesnosti 84 %. Zjištěním bylo, že čím více šampionů zbývá vybrat, tím se procentuální úspěch zmenšuje.

Práce [8] pro doporučení šampionů v určitou fázi draftu použila metodu minimax Alpha-Beta prořezávání společně s lineární regresí, kde vstupními daty jsou pouze *id* šampionů. Podobnou metodu MCTS UCT použili v [4] se třemi klasifikačními modely gradient boosted decision trees, neuronové sítě a logistická regrese. Nejlepším klasifikačním modelem byly neuronové sítě s 65% přesností, kde opět vstupními daty jsou pouze *id* šampionů.

Automatický draftovací systém v práci [10] vytvořili za pomoci neuronové sítě. V zhruba 50 % případů se objevil uživatelem zvolený šampion v nabídce systému deseti nejlepších šampionů.

Na řešení problému doporučení šampionů při draftu, pouze nad informacemi *id* šampionů, si práce [6] vybrala metodu KNN společně s logistickou regresí a dosáhli tak přesnosti 67 %. Bi-LSTM model pro doporučení sestavy hrdinů s použitím modelu continuous bag Of words v modelu Word2vec

pro vygenerování vektorů hrdinů použila práce [27]. Cílem bylo co nejlépe doplnit šampiona do sestavy k již čtyřem vybraným, uspěli s přesností 67 %.

3.2 Spojitosti mezi šampiony

Užitečnými informacemi pro draft jsou, zda šampion A má určitou výhodu nad šampionem B podle pořadí výběru šampionů. Poté co je zvolen šampion A, tak nepřátelský tým se na stejnou roli snaží vybrat šampiona B, který má v určité míře výhodu nad šampionem A. Pozoruhodná je také závislost výběru šampiona na již vybraných šampionech spoluhráčů, tedy že jistí šampioni společně tvoří silnější celek.

V práci [23] se tomu věnovali pomocí logistické regrese a rozhodovacího stromu. Porovnávali jednotlivé statistiky šampionů a podařilo se jim nalézt spojitosti mezi šampiony, zmíněné v předchozím odstavci. Použitím embedding modelů v práci [5] zachytili synergické a opoziční vztahy mezi šampiony. Tyto poznatky úzce odpovídaly názorům hráčů.

3.3 Před hrou

Mnoho prací se zabývá předpovědí výsledků daného zápasu v MOBA hrách podle informací z proběhlého draftu a také podle předzápasových informací získaných z dat již odehraných zápasů. Nejdůležitější údaje z draftu jsou vybraní a zakázaní šampioni, zvolené runy a Vyvolávačova kouzla.

S daty z 1500 zápasů a metodami adaboost, gradient boosting, extreme gradient boosting a random forest se tímto zabývá práce [2]. Dosáhli 95% přesnosti s pomocí random forest metody, zbylé tři metody se pohybovaly kolem 60 %.

V práci [18] porovnali nejčastější algoritmy strojového učení použité pro predikci vítěze zápasu z dat draftu. Pro data zápasů na nejvyšší úrovni byl s 65 % nejpřesnější algoritmus gradient boosted decision trees. Naivní bayesovský klasifikátor společně s logistickou regresí byly o procento horší. Porovnáním tří metod: lineární regrese, KNN a naivní bayesovský klasifikátor, v práci [21] došli k výsledkům, že logistická regrese je s 53% přesností nejvhodnější, přitom KNN a naivní bayesovský klasifikátor na tom jsou jen nepatrně hůře. Práce [22] pracovala s metodami strojového učení logistická regrese, SVM a random forests. Jejich přesnost se pohybovala kolem 63 % a SVM uspělo nejlépe.

K datům o draftu přidali v práci [24] informaci o délce historických zápasů vstupující do metody lineární regrese a multi-layer feedforward neuronových sítí. Tímto obohacením získali trochu lepší výsledky a lineární regrese překonala NN. Práce [1] porovnávala pro logistickou regresi více možností vstupních

informací. Kombinací dat o šampionech, jejich zprůměrováním nebo seřazením vychází, že neupravená data jsou s 62% přesností o 5 % lepší než upravovaná.

■ 3.4 Související ML

Dále zmíníme zajímavé práce, které se zabývají MOBA hrami, používají strojové učení a mohou být užitečné pro tuto práci.

S metodou hlubokého učení se sdílenými váhami za účelem předpovědi smrti šampiona v pětisekundovém okně se zabývala práce [13]. Se zájmem o predikci výsledku zápasu v danou chvíli a porovnáním rozdílných rekurentních neuronových sítí, došla práce [19] k výsledku, že jednoduchá RNN je v jejich případě nejlepší. Také práce [12] se zajímala živou předpovědí výsledku s použitím standardních modelů strojového učení s přesností 85 % po uplynutí prvních pěti minut zápasu.

Zajímavá práce [17] s cíli určit vlivy na výkon hráče a možnost doporučení spoluhráče k sebezlepšení s použitím hlubokého učení dospěla k tomu, že z dlouhodobého hlediska tato doporučení zvyšují výkon a dovednosti hráče.

Kapitola 4

Data

Pro náš případ použijí datábázi z internetové stránky Kaggle [9], která obsahuje data z 60 tisíc zápasů ve vysoké *Elo* úrovni, ve které hrají jedni z nejzkušenějších hráčů, což je méně než 0.1 % nejlepších. To je pro účel této práce velice přínosné, protože vysoká zkušenost hráčů otevírá všechny možnosti hry.

Hry jsou odehrané v regionech Západní Evropa, Severní Amerika a Korea. Všechny zápasy jsou hrány na stejné verzi hry 10.16, což je vyhovující, protože hra, šampioni i předměty mají stejné vlastnosti. Na této verzi hry je možné hrát za 150 šampionů. Každý zápas obsahuje informace o délce hry, jejím vítězi a podrobné informace o každém hráči daného zápasu. Několik nejzajímavějších dat je znázorněno v následující tabulce 4.1.

Název sloupce	Popis
champId	<i>Id</i> vybraného šampiona
ban_champId	<i>Id</i> zakázaného šampiona
item(0 6)	Jaké itemy hráč zakoupil
kills	Počet zabití
deaths	Počet úmrtí
assists	Počet asistencí
totalDamageDealt	Celkové způsobené poškození
magicDamageDealt	Celkové způsobené magické poškození
physicalDamageDealt	Celkové způsobené fyzického poškození
totalDamageDealtToChampions	Celkové způsobené poškození do šampionů
magicDamageDealtToChampions	Celkové způsobené magické poškození do šampionů
physicalDamageDealtToChampions	Celkové způsobené fyzického poškození do šampionů
totalDamageTaken	Celkové přijaté poškození
visionScore	Skóre vidění
goldEarned	Počet vydělaných zlatáků
turretKills	Počet zničených věží
champLevel	Úroveň šampiona

Tabulka 4.1: Vybrané příznaky ze záznamu odehraného zápasu.

Jelikož chceme brát v potaz co nejreálnější situaci, nejprve jsme data všech zápasů chronologicky seřadili. Následně jsme data rozdělili na trénovací množinu (prvních 35 tisíc zápasů), validační množinu (následujících 10 tisíc zápasů) a testovací množinu (zbývajících 15 tisíc zápasů). Na trénovací mno-

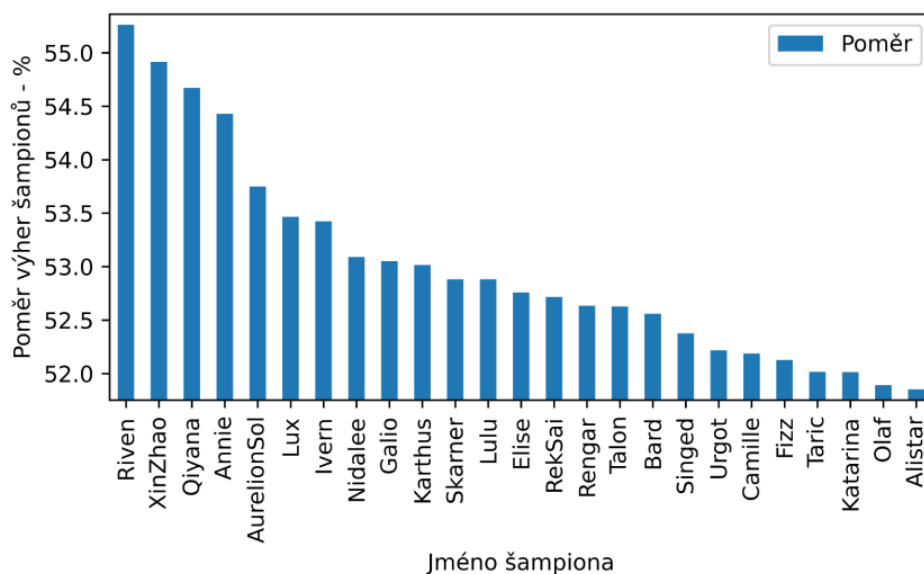
žině budeme provádět veškeré trénování modelů. S pomocí validační množiny vybereme nejvhodnější modely. Testovací množina je určena pro nezávislé testování natrénovaných modelů a pro experimenty.

4.1 Analýza dat

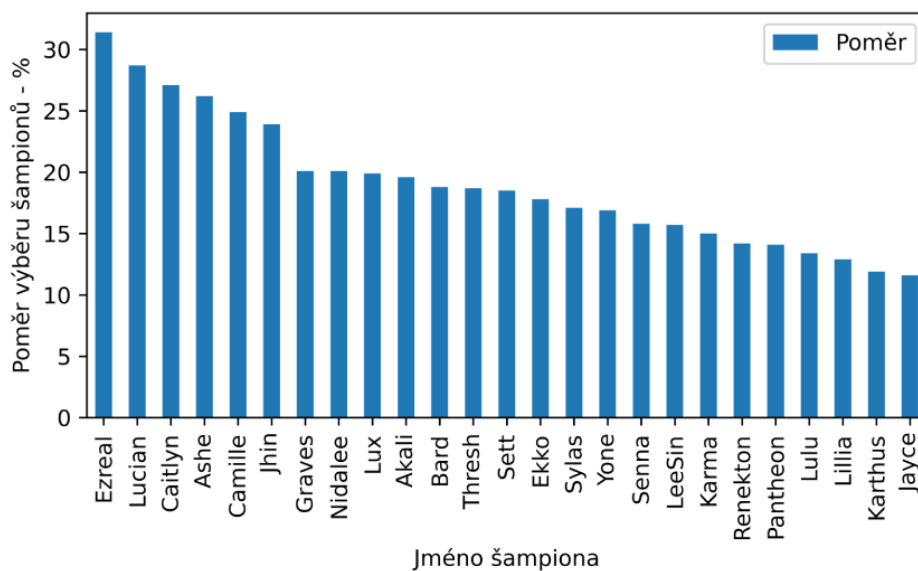
Nad trénovací a validační množinou dat jsme provedli analýzu, abychom se s daty lépe seznámili a všimli si nějakých zajímavostí. Nejprve jsme si zjistili, že modrý tým vyhrál v 50,43 % zápasů a červený potom v 49,57 %. Pravděpodobnost výhry týmu nezáleží na straně a naše data tomu s nepatrnou odchylkou odpovídají.

Dalším krokem bylo seznámit se s daty o šampionech. Nejvíce výher v poměru s počtem odehraných zápasů má šampionka Riven, která vyhrála ve více jak 55 % odehraných zápasů, následují šampioni Xin Zhao, Qiyana a další nejlepší šampiony lze vidět na grafu 4.1. Dále zajímavým faktorem je celkový počet vybraných (graf 4.2) a zakázaných (graf 4.3) jednotlivých šampionů. Ezreal a Lucian jsou těmi nejhranějšími a objevili se ve 30 % her. Na druhou stranu nejvícekrát zakázaná byla Caitlyn v neskutečných 2/3 her a šampioni Yone s Karthussem ve skoro 50 % zápasů. Caitlin ve verzi hry našich dat byla nejsilnějším šampionem na roli ADC a jediným šampionem, který se jí dokázal vyrovnat byl Ezreal. Yone byl přidán jako nový šampion společně s verzí hry a Karthus byl několikrát zahráný na profesionální scéně, kde všem hráčům ukázali jak dokáže být silný.

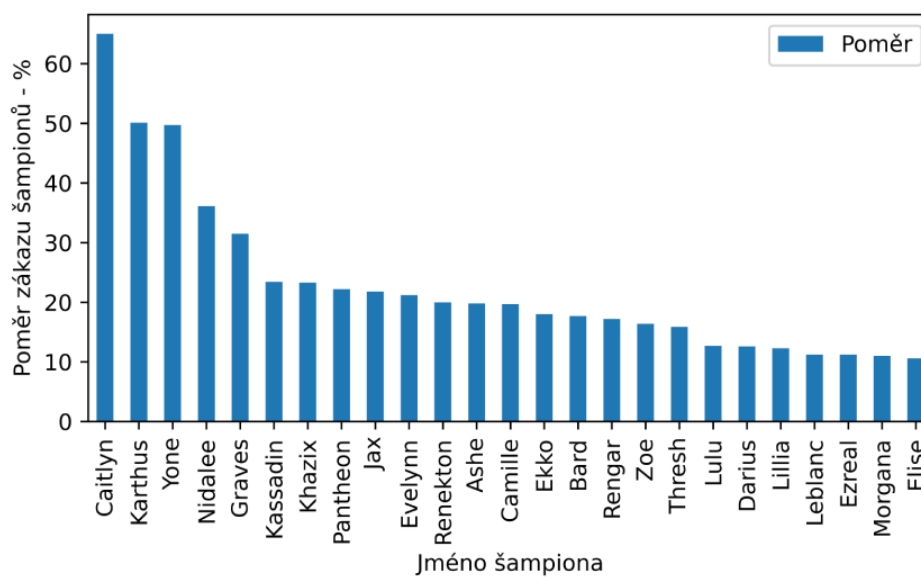
Dalším zajímavým krokem bylo zjistit, kolik hráčů hraje více rolí. Hráč má možnost si před začátkem zápasu vybrat jednu preferovanou roli a druhou jako záložní. Občas se stává, že mu je nakonec přiřazena jiná role, než si vybral. V grafu 4.4 je vidět, že více jak polovina hráčů hrála maximálně dvě role. Více jak 20 % hráčů hrálo čtyři nebo pět rolí, což mohl zapříčinit matchmaking systém, ale najde se i dost hráčů, kteří jsou schopni na stejné úrovni zahrát všechny role.



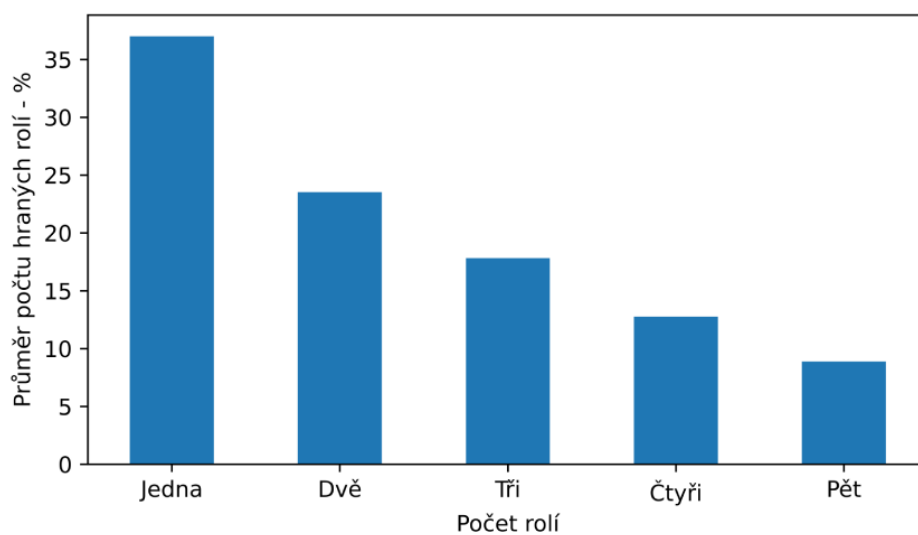
Obrázek 4.1: Prvních dvacet pět šampionů s nejlepší poměrem výher ku odehraným zápasům.



Obrázek 4.2: Prvních dvacet pět nejčastějších šampionů v zápasech.



Obrázek 4.3: Prvních dvacet pět nejvíce zakázaných šampionů.



Obrázek 4.4: Procentuální vyjádření počtu hráčů hrajících více rolí.

Kapitola 5

Modely pro predikce výsledku zápasu

Začneme jednoduchým modelem předpovídající výsledky zápasu před draftem a následně vytvoříme dva modely predikující výsledek s informacemi z proběhlého draftu.

Prvotní přepověď bude mít vstupní informace průměrného Ela pro oba týmy. Naším dalším krokem bude představit model (*champ_model*), do kterého jako vstupní data budou vstupovat informace o *id* deseti šampionů. Vybraní šampioni by měli tvořit podstatnou část predikce výsledů v MOBA hrách. Proto i druhý model (*feat_model*) používá data o *id* šampionech, ke kterým jsme přidali pět nových příznaků, týkajících se hráčů. Příznaky jsme z dat pro *feat_model* vybrali ty nejzajímavější a nejdůležitější s vizí, že se nové budou dát jednoduše přidat.

5.1 Architektura embedding modelů

Modely zmíněné v předchozím odstavci zaměřené na predikci po draftu jsou embedding modely neuronových sítí (NN). Jsou vhodné pro naši práci, protože NN se dokáže naučit spojitost mezi šampiony, jejich rolemi a příznaky hráče nebo přiřadit optimální váhy a ovlivnit tak každou vstupní informaci.

Pro *champ_model* jsme jako vstupní vrstvu použili embedding vrstvu vhodnou pro vstupní parametry kategoričkových hodnot jako *id* šampionů. Navázali jsme poté skrytými vrstvami lineární transformace a nakonec použili výstupní vrstvu také lineární transformace. Za vstupní vrstvou a každou skrytou vrstvou jsme použili aktivační funkci. Po výstupní vrstvě jsme použili LogSoftmax funkci, jejíž formulace je:

$$\text{LogSoftmax}(x_i) = \log \left(\frac{\exp(x_i)}{\sum_j \exp(x_j)} \right)$$

Výstupem modelů jsou zlogaritmované pravděpodobnosti výhry modrého a červeného týmu, kde nám nezlogaritmovaná část funkce LogSoftmaxu zaručuje, že součet těchto pravděpodobností je 100 %.

Použité vrstvy ve *feat_modelu* jsou stejné jako u předchozího *champ_modelu*. Do *feat_modelu* však vstupují i data pěti příznaků o hráčích, které se napojí na výstup ze vstupní embedding vrstvy, kterou prošly data *id* šampionů. Dále se budou lišit vstupní parametry u skryté nebo případně u výstupní vrstvy, aby seděly na větší data.

5.2 Předzpracování dat

Z předchozí sekce víme, že embedding modely potřebují jen některé informace z dat. pro *champ_model* jsme tedy z dat vyfiltrovali jen informace o šampionech, výherci a *id* zápasu. Data jsou seřazené podle rolí pro oba týmy stejně: s1, a2, m3, j4, t5, S1, A2, M3, J4, T5 (malé písmeno značí roli u modrého týmu, velké písmeno roli u červeného týmu a číslice zastupují stejnou roli). Jelikož obě strany týmů jsou vyrovnané, rozšíříme data tak, že otočíme výherce a pořadí týmů následovně: S1, A2, M3, J4, T5, s1, a2, m3, j4, t5. Takto zdvojnásobíme počet dat a zároveň zaručíme nezávislost pravděpodobnosti výhry se stranou týmu.

Feat_model použije data připravené pro *champ_model* a k nim bude potřeba získat odpovídající data pěti příznaků. Z dat jsme vyfiltrovali *id* zápasu a k tomu vybrali následující informace o hráčích: *Elo* hráče, počet her na dané roli, průměrný zisk zlatáků, průměrné *způsobené poškození* a průměrné *přijaté poškození*. *Elo* hráče můžeme použít z informací aktuálního zápasu, ale další čtyři příznaky jsou závislé na již odehraných hrách. Máme-li tedy chronologicky seřazené zápasy, tak v prvním nebudeme mít žádné (kromě *Elo* hráče) bonusové informace oproti *champ_modelu*. Jak časově postupujeme dopředu, tak máme více odehraných zápasů, z kterých získáme lepší informace k aktuálnímu datu. Data na validační množině tedy mají bonusové informace z méně zápasů než data na následné testovací množině.

Pět příznaků o hráčích znají jen spoluhráči. Aby tyto nové informace pro *feat_model* seděly na data z *champ_modelu*, propojíme je pomocí *id* zápasu. Jeden zápas je ale v datech 2x díky otočení pořadí týmu, a proto dosadíme data o příznacích modrého týmu k těm datům z *champ_modelu*, kde je první modrý tým, a podobně i pro příznaky u červeného týmu.

5.3 Implementace modelů

Pro práci s neuronovými sítěmi jsme zvolili programovací jazyk Python a open source knihovnu PyTorch [15] vhodnou pro strojové učení. Velmi nám ulehčila práci s již implementovanými funkcemi pro NN, jako transformační funkce embeddingu a lineární funkce nebo aktivační funkce tanh, sigmoid a logsoftmax. Pro čtení dat, které jsme měli ve formátu csv, jsme použili knihovnu Pandas [25], na pomoc při práci s daty jsme přidali knihovnu NumPy [11].

Kriteriální funkci jsme zvolili negative log likelihood (NLLLoss). Vstupní data očekává ve formě zlogaritmovaných pravděpodobností jednotlivých tříd (pro náš případ jsou dvě třídy - modrý a červený tým). Vstupní data pro NLLLoss jsou zároveň výstupní data embedding modelů. Ztrátu funkce NLLLoss lze popsat vzorcem:

$$NLLLoss(x_i, y) = -x_{i_y}$$

kde x_i jsou výstupní data embedding modelů, y je binární příznak výherní týmu.

Přidali jsme další funkce a metody, které zlepšují a zkvalitňují průběh učení NN. Nejprve jsme implementovali zamíchání dat pro trénovací množinu před každou epochou, aby se NN nevázalo na pořadí. Dále jsme implementovali early stopping, který dokáže při trénování rozeznat overfitting a zastavit tak učení včas. Lepších výsledků jsme dosáhli přidáním plánovače, který sníží learning rate pokaždé, když přestane klesat ztráta na trénovací množině.

5.4 Optimalizce hyperparametrů

Neuronové sítě samy optimalizují spoustu parametrů funkcí, které jsme v modelech použili. K fungování NN jsou ještě zapotřebí hyperparametry, které si neuronové sítě nedokážou optimalizovat samy. Výběr správných hyperparametrů je důležitý, a pro nalezení těch nejlepších jsme provedli grid search. Použité hyperparametry a jejich rozsahy jsou popsány tabulkou 5.1.

Learning rate se váže na typ optimalizátoru, proto jsme pro optimalizátor SGD a Adam zvažovali pokaždé jiný rozsah (viz. tabulka 5.1). Volba nejlepších hyperparametrů je pro *champ_model* i *feat_model* ze stejné tabulky 5.1. Počet kombinací hyperparametrů je 864 a je pro oba modely stejný. Všechny operace jsme paralelně spustili s využitím výpočetní kapacity virtuální organizace MetaCentrum VO [3].

Hyperparametr	Hodnoty
Dimenze embeddingu	1, 2, 3, 4, 5, 6
Aktivační funkce	tanh, sigmoid
Počet skrytých vrstev	0, 1, 2, 3
Vstupní a výstupní parametry skrytých vrstev	10, 20, 30
Optimalizátor	SGD, Adam
Learning rate	<u>0.1</u> , <u>0.01</u> , <u>0.001</u> , <i>1e-4</i> , <i>1e-5</i>

Tabulka 5.1: Tabulka rozsahu hodnot k jednotlivým hyperparametrům. Podtržené hodnoty u learning rate patří k optimalizátoru SGD a hodnoty v kurzívě patří k optimalizátoru Adam.

Výběr rozsahů hodnot hyperparametrů v tabulce 5.1 popíšeme s pomocí grafů 5.1 a 5.2 představující získanou průměrnou přesnost na validační množině pro jednotlivé hyperparametry. U dimenze embeddingu lze vidět (zřetelněji na grafu 5.1), že průměrná přesnost s vyšší dimenzí roste až do dimenze 4, kdy začíná stagnovat. Pro aktivační vrstvu jsme vybrali funkci Tanh a Sigmoid:

$$\text{Tanh}(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

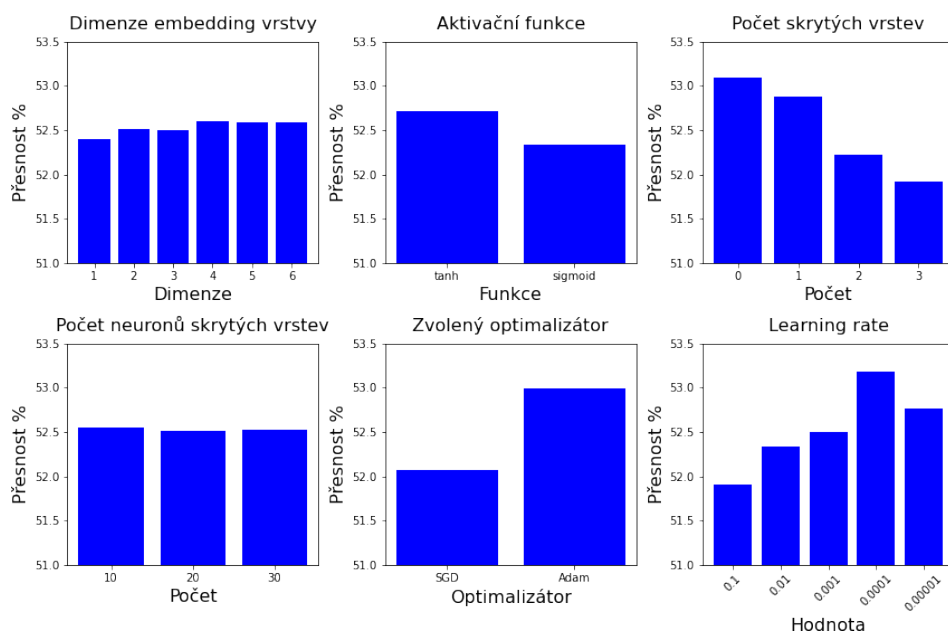
$$\text{Sigmoid}(x) = \frac{1}{1 + \exp(-x)}$$

Aktivační funkce Tanh a Sigmoid jsme si vybrali, protože si jsou funkčně dost podobné. Známa aktivační funkce ReLU:

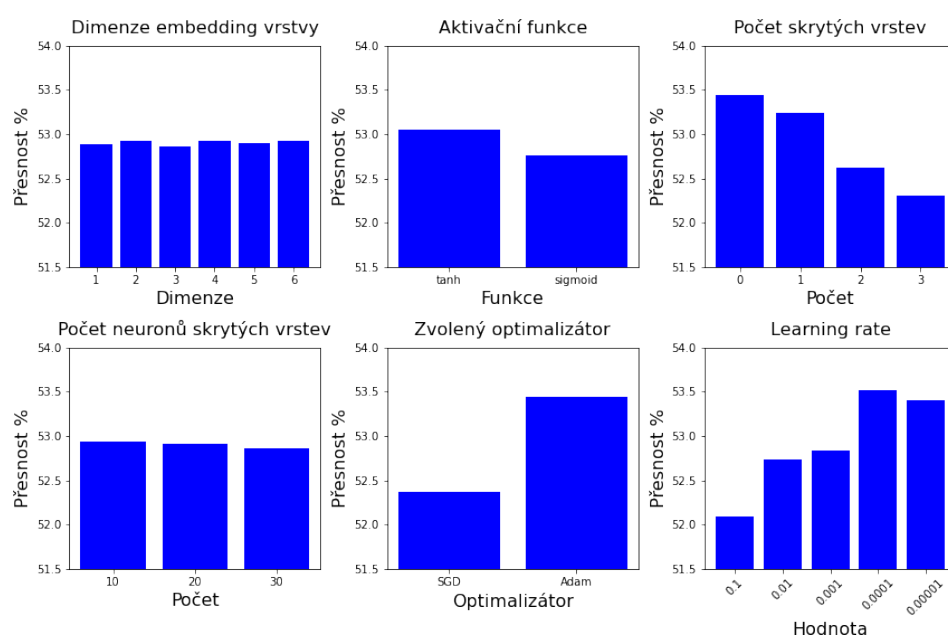
$$\text{ReLU}(x) = \max(0, x)$$

si při testování nevedla dobře, a tak jsme ji do konečného výběru nezařadili.

Na grafech lze vidět, že funkce Tanh je pro oba modely lepší než funkce Sigmoid. Průměrná přesnost s vyšším počtem skrytých vrstev pomalu klesá, proto jsme vyšší jak 3 nebrali v úvahu. Počet vstupních a výstupních neuronů skrytých vrstev jsme zvolili v rozumném poměru k dimenzím embeddingu a nijak výraznou výchylku jsme nepozorovali. Dále můžeme sledovat průměrnou přesnost pro optimalizátory, kde Adam je značně lepší než SGD. Z tabulky 5.1 víme, že learning rate se váže na typ optimalizátoru, čemuž odpovídají i hodnoty learning rate v grafech 5.1 a 5.2.



Obrázek 5.1: Graf průměrných přesností u jednotlivých hyperparametrů pro *champ_model* na validační množině.



Obrázek 5.2: Graf průměrných přesností u jednotlivých hyperparametrů pro *feat_model* na validační množině.

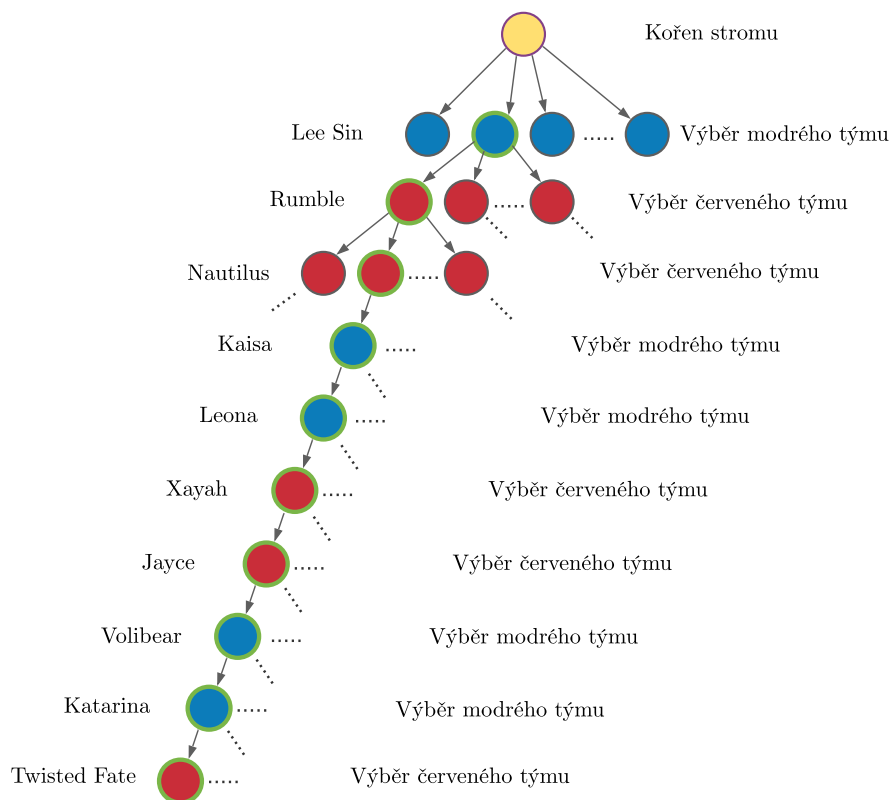
Kapitola 6

Asistence při draftu

Je obecně známo, že výběr toho nejlepšího šampiona ve 30 vteřinách, které jsou na to dány, je občas náročné a hráč nestihne promyslet všechny možnosti. Proto chceme dosáhnout asistence doporučením nejvhodnějších šampionů pro hráče pomocí Monte Carlo Tree Search (MCTS) s použitím UCT [14] algoritmu.

6.1 Monte Carlo Tree Search UCT

V průběhu předzápasové fáze chceme být schopni pro každý výběr nabídnout aktuálně ty nejlepší možnosti šampionů. Nejdříve je třeba si vytvořit vyhledávací strom, kde se v každé vrstvě vybírá šampion právě pro jeden z týmů (znázorněno na obrázku 6.1).



Obrázek 6.1: Náčrt stromu reprezentující postup vybírání šampionů v draftovací fázi. Pořadí výběrů a střídání týmů je popsáno tabulkou 2.4. Prvím vybraným šampionem modrého týmu je Lee Sin, následně vybírá dva šampiony po sobě červený tým, v naší ukázce tedy Rumble a Nautiluse. Postupuje se dál a ve finále pro modrý tým jsou vybraní šampioni: Lee Sin, Kaisa, Leona, Volibear a Katarina, pro červený tým jsou vybraní šampioni: Rumble, Nautilus, Xayah, Jayce a Twisted Fate.

Tento problém by mohl řešit algoritmus Minimax Alpha-Beta prořezávání, ale ten by byl pro výběry šampionů hned ze začátku pomalý, z důvodu procházení až mnoho větví. Proto jsme se rozhodli pro MCTS UCT. Dokáže v relativně krátkém čase rozumně projít všechny možnosti a celkově konverguje k výsledkům Minimaxu.

Algoritmus MCTS v naší práci použijeme následovně: Začínáme v kořenu stromu s informacemi o již vybraných a zakázaných šampionech u obou týmů. Jelikož je postup výběru šampionů vždy stejný, dokážeme podle počtu již vybraných šampionů přesně určit, jaký tým je právě na řadě. Postupujeme hlouběji stromem dokud nenarazíme na jeden z listů, které lze rozeznat podle již deseti vybraných šampionů. V listu máme všechny potřebné vstupní data pro použití modelů, z kterých dostaneme ohodnocení stavu.

UCT algoritmus vybírá následující uzly tak, že volí ty s nejvyšším ohod-

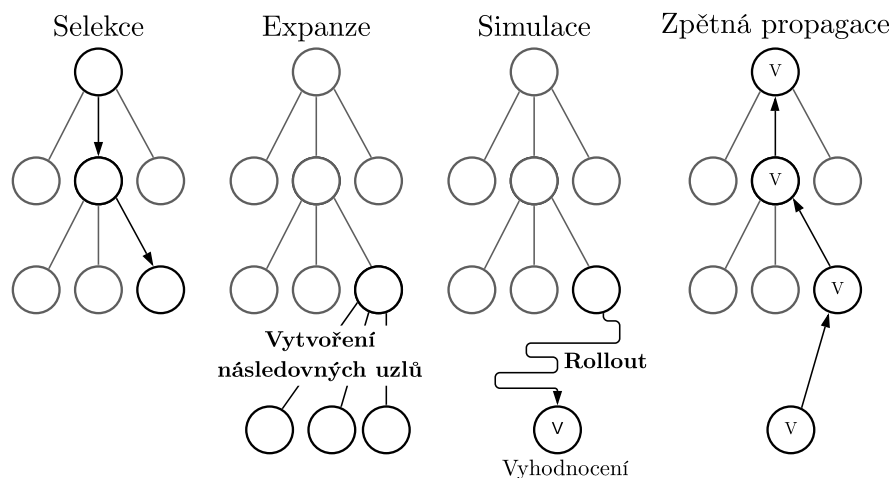
nocením a zároveň nezapomíná i na ostatní méně navštěvované uzly. Řídí se vzorcem:

$$\operatorname{argmax}_a \left(\frac{h_i}{n_i} + c * \sqrt{\frac{\ln(p_i)}{n_i}} \right)$$

kde a jsou akce, které vedou z aktuálního uzlu, h_i je hodnota pravděpodobnosti pro daného šampiona, n_i je počet navštívení uzlu, p_i je počet navštívení rodičovského uzlu, a nakonec konstanta c . Ze vzorce jde vidět, že UCT se snaží prozkoumávat ty nejslibnější uzly (část $\frac{h_i}{n_i}$) a zároveň stále rozumně prozkoumávat do šířky zbylé možnosti (část $c * \sqrt{\frac{\ln(p_i)}{n_i}}$). Hodnotu konstanty c jsme zvolili $c = 1$, inspirovali jsme se v práci [4].

Algoritmus MCTS použitý pro náš případ se dá popsat čtyřmi kroky (také znázorněné grafem 6.2):

- **Selekce:** Začínáme v kořenu stromu a vybíráme následné uzly podle UCT vzorce dokud nenarazíme na list stromu nebo na uzel, který ještě nemá rozšířené následovné uzly. Kořen stromu zastupuje aktuální fázi draftu a list stromu je uzel, ve kterém je již vybráno deset šampionů. Po nalezení listu následuje krok simulace, po nalezení uzlu bez rozšířených následovných uzlů následuje krok expanze.
- **Expanze:** Pokud se nacházíme v uzlu, který nemá žádné následovné uzly a byl již navštívený, vytvoříme pro tento uzel následovné uzly a vrátíme se pro tento uzel ke kroku selekce. Pokud tento uzel ještě navštívený nebyl, provedeme následující krok.
- **Simulace:** Provedeme pro aktuální uzel *rollout* (náhodné doplnění *id* šampionů k již určeným v daném uzlu), díky kterému budeme schopni provést vyhodnocení modelů. V listu *rollout* není potřeba, má všechny informace potřebné k vyhodnocení modelů.
- **Zpětná propagace:** Hodnotu z *rolloutu* zpropagujeme postupně do každého uzlu postupem ke kořenu. Dostaneme se tak zpět do kořene a provedeme znovu krok selekce.



Obrázek 6.2: Schéma algoritmu MCTS.

6.2 Draftování jako hra s neúplnou informací

Základ, bez kterého se MCTS neobejde, je získání hodnoty pravděpodobnosti v listu, která se pak propaguje až ke kořenu. Chceme pracovat s daty o šampionech a informacemi o hráčích. Bohužel problémem je, že v průběhu draftu jsou vzájemně mezi týmy sdíleni pouze vybraní šampioni. Jak už bylo zmíněno výše, informace o hráčích z již odehraných zápasů znají pouze spoluhráči. V listu tedy bude zapotřebí vyhodnotit *champ_model*, *feat_model* z modré strany (*feat_model_blue*) a *feat_model* z červené strany (*feat_model_red*).

Ukážeme, jak se využijí tyto tři vyhodnocení: je začátek předzápasové fáze a ještě nebyl vybrán žádný šampion. Začneme tedy v kořenu stromu a podle vstupních dat již zvolených šampionů (v našem příkladu není zatím žádný) zjistíme, že je na řadě modrý tým. Jelikož se snažíme získat nejvhodnější šampiony pro první výběr modrého týmu, tak známe data o hráčích modrého týmu z předchozích zápasů. Hodnotu h_i budeme brát z *feat_model_blue* pokaždé, když v nějaké vrstvě ve stromu bude výběr na straně modrého týmu. Pokud je ale řada na červeném týmu, nelze brát hodnotu h_i z *feat_model_blue*, protože červený tým nemůže znát a ani použít informace o hráčích modrého týmu. Nemůžeme ani použít *feat_model_red*, protože právě volíme za modrou stranu, která nemá informace o hráčích červeného týmu. Vybereme tedy třetí možnost použít hodnoty z *champ_modelu*, který potřebuje informace pouze o šampionech, o čemž mají oba týmy stejný přehled. Hodnoty z *champ_modelu* nám nedají to, co by pro červený tým byla nejlepší volba, reprezentují totiž pravděpodobnost výhry průměrného soupeře.

6.3 Redukce možných voleb

Z popisu dat víme, že pracujeme se 150 různými šampiony. Když začneme procházet strom od kořenu, známe jen vybrané šampiony v obou týmech a pokud nevíme, kterou roli má aktuální hráč na řadě v plánu hrát, musíme počítat se všemi rolemi, což nám počet možností až zpětinásobí. Naštěstí se nehraje každý jeden šampion na všech rolích a můžeme tak rozsah výběru omezit. Na validační množině jsme pro každou roli vzali v úvahu ty šampiony, kteří byli obsazeni alespoň v 0,5 % zápasů. Tímto jsme omezili počet šampionů pro jednotlivé role následovně: *support* - 34, *ADC* - 29, *mid* - 58, *jungle* - 35, *top* - 52. Přidali jsme také možnost pro uživatele, aby si před začátkem výběru určil, zda chce dostat doporučení šampionů na konkrétní roli, čímž se také může zúžit výběr a zvýšit přesnost.

Kapitola 7

Experimenty

Nejprve se budeme věnovat jednoduchému modelu, který predikuje před draftem. Následně se zaměříme na naučení a nalezení nejlepších embedding modelů na validační množině. Dalším krokem jsou konečné testy a experimenty na testovací množině dat pro nejlepší embedding modely, které zároveň použijeme i pro experimenty asistenčního algoritmu.

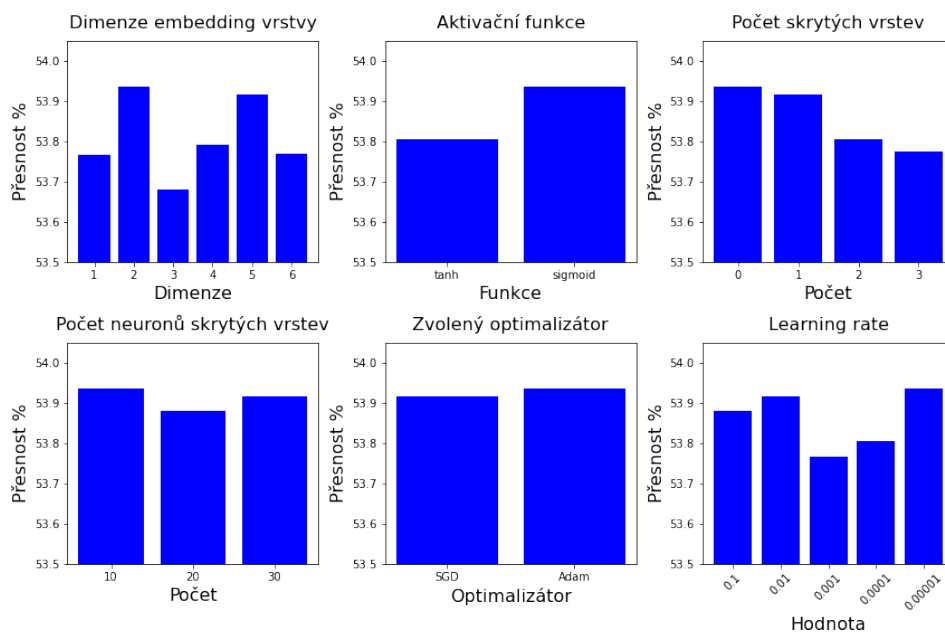
7.1 Experiment predikce před draftem

Vytvořili jsme model (*pre_draft_model*) predikující výherce zápasu s použitím informací průměrného Ela hráčů v týmu, které jsou známé před draftem. Tento model jsme otestovali na testovací množině dat a dosáhli jsme přesnosti necelých 68 %, což se dá považovat za dobrý výsledek. Nutno ale podotknout, že vstupními informacemi byly data z obou týmů, které při draftu nejsou dostupná, protože tým A nemá takovou informaci o týmu B. Naše dosažená přesnost *pre_draft_modelu* je z pohledu matchmaking systému, a proto je také tak vysoká. Můžeme zároveň říct, že si matchmaking systém v tvorbě týmů pro naše data nevedl dobře, protože jeho úkolem je sestavit dva co nejvíce vyrovnané týmy, což náš *pre_draft_model* spíše vyvrátil.

7.2 Výběr modelů

Pomocí grid searche jsme na validační množině dat prošli všechny hyperparametry (zmíněné v tabulce 5.1) pro *champ_model* a *feat_model*. Zhodnotili jsme výsledky a pro *champ_model* i *feat_model* jsme našli tu nejúspěšnější kombinaci hyperparametrů.

Porovnání výsledků hyperparametrů pro *champ_model* si lze prohlédnout na grafu 7.1, který pro každou hodnotu hyperparametru zobrazuje její maximální dosaženou přesnost.



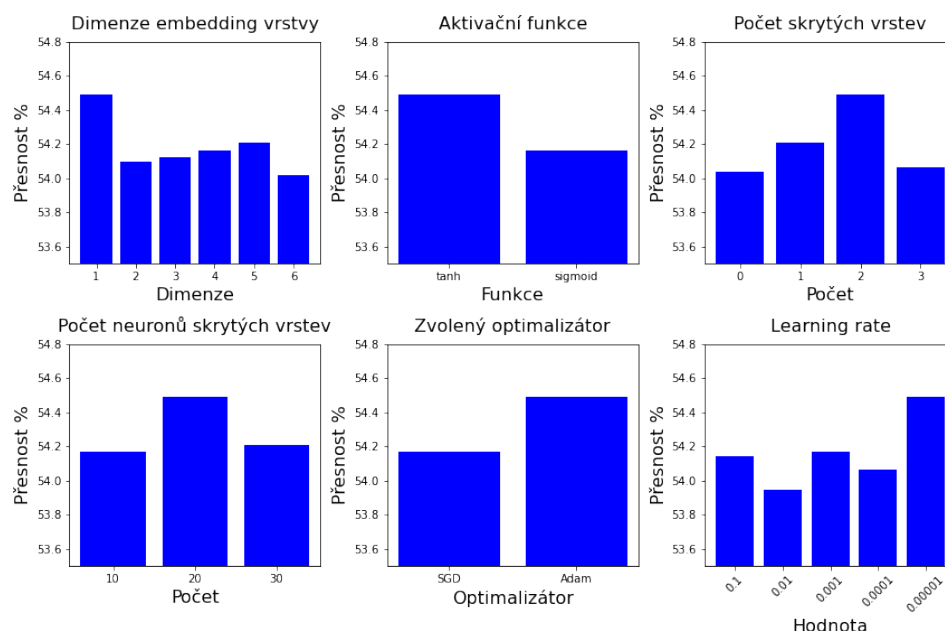
Obrázek 7.1: Graf maximálních přesností u jednotlivých hyperparametrů pro *champ_model* na validační množině.

Kombinace hyperparametru s nejvyšší přesností 53.94 % pro *champ_model* můžeme sledovat v tabulce 7.1.

Dimenze Embed.	Aktiv. funkce	Počet skryt. v.	Poč. Neur. skryt. v.	Optima - lizátor	Learning rate	Přesnost
2	Sigmoid	0	10	Adam	1e-5	53.94 %

Tabulka 7.1: Tabulka nejlepší kombinace hyperparametrů na validační množině pro *champ_model*.

Výběr nejlepších hyperparametrů pro *feat_model* probíhal stejně jako u *champ_modelu*. Graf 7.2 nám pro lepší představu opět zobrazí maximální přesnost u jednotlivých hodnot hyperparametrů.



Obrázek 7.2: Graf maximálních přesností u jednotlivých hyperparametrů pro *feat_model* na validační množině.

Znovu jsme zvolili model s hyperparametry (zobrazené tabulkou 7.2), kterým se na validační množině vedlo nejlépe.

Dimenze Embed.	Aktiv. funkce	Počet skryt. v.	Poč. Neur. skryt. v.	Optima - lizátor	Learning rate	Přesnost
1	Tanh	2	20	Adam	1e-5	54.49 %

Tabulka 7.2: Tabulka nejlepších kombinace hyperparametrů na validační množině pro *feat_model*.

Nejlepší *champ_model* dosáhl na validační množině přesnosti 53.94 % a nejlepší *feat_model* dosáhl přesnosti 54.49 %. Nejlepším optimalizátorem se u obou modelů stal Adam s learning rate 1e-5. Zbylé nejlepší hyperparametry se mezi modely liší. Tyto nejlepší modely jsme následně otestovali na testovací množině dat a použili pro asistenční algoritmus.

7.3 Experimenty embedding modelů

Tabulky 7.1 a 7.2 nám znázornili nejpřesnější kombinace hyperparametrů na validační množině pro *champ_model* a *feat_model*. Důležitými experimenty bude otestovat tyto nejspěšnější modely na testovací množině, tedy jak

by si modely s těmito hyperparametry vedly v budoucích zápasech.

Nejprve na testovací množině vyhodnotíme *champ_model*, který dosáhl přesnosti 54,06 %. To je o 0,12 % vyšší než na validační množině a můžeme tak předpokládat, že pro *champ_model* při učení nenastal overfitting a použité hyperparametry jsou vhodné.

Následně jsme vyhodnotili *feat_model* na testovací množině. Tento *feat_model* dosáhl přesnosti 53,45 %, což je o více než jedno procento méně oproti výsledkům na validační množině.

7.3.1 Výsledky modelů na testovací množině

Feat_model na validační množině dosáhl vyšší přesnosti než *champ_model*, na testovací množině byl již ale o 0,5 % horší. Rozdíly nejsou tak vysoké a daly by se vysvětlit například tím, že se styl hry v časovém horizontu dat změnil. Zápasy na stejné verzi hry jsou ve většině případů za stejných podmínek, občas je ale vydaná malá úprava herní verze. Dalším faktorem, který mohl ovlivnit výsledek *feat_modelu* na testovacích datech, jsou vstupní data příznaků hráčů, které závisí na předchozích odehraných zápasech. Data na validační množině, jakožto chronologicky první data, čerpají tyto informace z výrazně menšího počtu zápasů, než data na testovací množině. U *champ_modelu* tento případ nemohl nastat, protože jeho vstupní data nejsou vázány na již odehrané zápasy.

To, že si nejlepší *feat_model* z validační množiny vedl na testovací množině o něco hůře, můžeme vysvětlit také tím, že přidané příznaky oproti *champ_modelu* mohou zapříčinit jednodušší overfitování. Možná *feat_model* potřebuje pro trénování data z více zápasů, nebo lépe upravená data, anebo lepší implementaci modelu. V celých datech jsme měli k dispozici dalších alespoň 50 příznaků ke každému hráči, které by značně rozšířily oblast vstupních informací pro *feat_model* a zvýšily tak úspěch na validační a testovací množině. My jsme se zaměřili na pět nejvýznamějších příznaků.

Výsledky experimentů pro embedding modely na testovací množině dopadly relativně dobře, pokud předpokládáme, že matchmaking systém vytvořil dva co nejvíce vyrovnané týmy. Výsledek *pre_draft_modelu* nám ukázal, že pro naše data tomu tak není. Nicméně z pohledu týmu tuto informaci nemáme a nezbyvá nám nic jiného, než věřit, že protivníci jsou pro nás nejvhodnějším soupeřem. Z pohledu týmu lze před draftem odhadovat pravěpodobnost výhry na 50 %. Tuto hranici jsme překonali a na testovací množině jsme u *champ_modelu* získali přesnost přes 54 % a u *feat_modelu* přes 53,5 %. Naše modely používají pouze takové informace, které jsou v reálné situaci a momentu pro hráče dostupné. Proto jsou také přesnosti embedding modelů značně horší, než *pre_draft_model* z pohledu matchmaking systému, který zná zároveň informace o obou týmech.

7.4 Příklady využití asistenčního systému

Cílem této práce bylo vytvořit asistenční algoritmus pro hráče, kterému doporučí nejlepší šampiony pro aktuální výběr. S pomocí natrénovaných embedding modelů jsme sestavili vyhledávací strom, který se řídil algoritmem MCTS UCT. Vstupními daty jsou doposud vybraní a zakázání šampioni a uživatel má možnost si zvolit, pro kterou roli chce dostat doporučení.

Testování našeho asistenčního algoritmu bylo limitované daty, které jsme po celou dobu používali. Data neobsahují informaci, v jakém pořadí hráči vybírali šampiony. Ukážeme tedy, jak asistenční systém funguje, jak dokáže doplnit šampiona na chybějící roli, vybrat vhodného šampiona tak, aby obstál proti protivníkovi nebo zapadal do sestavy šampionů spoluhráčů. Pro experimenty použijeme data o hráčích pro jeden ze zápasů z testovací množiny dat.

7.4.1 Obsazení chybějících rolí

Začneme od nejjednoduššího případu, kdy stačí vybrat posledního šampiona. Udělali jsme simulaci, kde modrý tým má již vybraných pět šampionů, červený čtyři šampiony a právě probíhá poslední výběr. Náš algoritmus si sám přiřadí šampiony na určitou roli, proto jsme vybrali takové šampiony, aby pro systém byly jednodušší na přiřazení. Šampiony jsme zvolili tak, že nám za červený tým zbývá vybrat šampiona na roli *top*. Náš asistenční algoritmus přijímá jako vstupní informaci také roli, na kterou chceme získat doporučení šampionů. Tuto možnost jsme nevyužili, a nechali jsme tak náš systém, aby potřebnou roli našel sám. Výstup našeho algoritmu je zobrazen v tabulce 7.3, kde lze vidět, že potřebnou roli *top* našel a doporučil nám na ni šampiony.

Jméno šampiona	Role	Pravděpodobnost výhry
Riven	Top	65.94 %
Kled	Top	65.51 %
Shen	Top	65.32 %
Cassiopeia	Top	64.93 %
Singed	Top	64.90 %
Hecarim	Top	64.84 %

Tabulka 7.3: Šampiony doporučení našim systémem na potřebnou roli. Algoritmus doplňoval páteho šampiona k již čtyřem zvoleným a správně volil šampiony na roli *top*.

Přiřadit posledního šampiona ke čtyřem předchozím v případě, že již vybraní šampiony jdou s velkou jistotou přiřadit k jedné roli, náš systém zvládá,

jak by uživatel očekával.

Zkusíme podobný experiment, ale s menším počtem již vybraných šampionů. Pro tento případ bude týmu zbývat vybrat dva šampiony na roli *jungle* a *top*. Algoritmus nám vrátí šampiony, které nejvíce doporučuje. Rozpoznal, že je zapotřebí doplnit role *top* a *jungle* a vrátil nám následující doporučení zobrazené tabulkou 7.4.

Jméno šampiona	Role	Pravděpodobnost výhry
Riven	Top	64.01 %
Kled	Top	63.53 %
Shen	Top	63.41 %
Elise	Jungle	63.08 %
Vi	Jungle	62.95 %
Cassiopeia	Top	62.83 %

Tabulka 7.4: Šampioni doporučení našim systémem na potřebné role. Algoritmus doplňoval dva šampiony k již třem zvoleným a správně volil šampiony na role *top* a *jungle*.

Doporučit šampiona k již vybraným šampionům se našemu systému daří, proto další experiment bude o tom, jak dokáže doporučit šampiona na roli, na kterou již soupeř šampiona zvolil.

7.4.2 Doporučení šampiona na roli proti soupeři

Pro jednoduchost zvolíme jednoho šampiona na *mid* za modrý tým a následně si necháme od algoritmu poradit jakého šampiona na roli *mid* zvolit za červený tým. Výběr na roli *mid* za modrý tým jsme nechali zvolit naším algoritmem, který nám s největší pravděpodobností doporučil šampiona Kassadin (tabulka 7.5). Pro první výběr modrého týmu by se měl vybrat takový šampion, proti kterému se těžce volí silnější šampioni. V League of Legends jsou ale pro každého šampiona účinní protivníci, které hráči většinou dopředu zakáží. V této ukázce jsme se zakázanými šampiony nepočítali. U doporučení algoritmu za červený tým tak alespoň můžeme sledovat, zda dokáže správně poradit.

Jméno šampiona	Role	Pravděpodobnost výhry
Kassadin	Mid	69.70 %
Lux	Mid	69.47 %
Katarina	Mid	69.23 %
Galio	Mid	69.06 %
Qiyana	Mid	68.71 %
Cassiopeia	Mid	68.66 %

Tabulka 7.5: Šampioni doporučení našim systémem na potřebnou roli. Algoritmus vybíral šampiona na roli *mid*.

Následně jsme spustili algoritmus pro doporučení šampiona na *mid* za červený tým. Jako uživatel předpokládáme šampiony, kteří dokáží mít nad Kassaden od začátku výhodu nebo s ním alespoň držet krok. K šampionům, které nám algoritmus doporučil, jsme v tabulce 7.6 přidali expertní názor potvrzený externím zdrojem CounterStats [7], kterým komentujeme jak daný šampion dokáže proti Kassadinovi obstát.

Jméno šampiona	Role	Pravděpodobnost výhry	Expertní názor na doporučení šampiona proti Kassadinovi
Lux	Mid	66.94 %	Lux má nad Kassadinem malou výhodu.
Katarina	Mid	66.13 %	Souboj je skoro vyrovnaný, velmi záleží na zkušenostech hráče.
Talon	Mid	65.89 %	Už od začátku zápasu má Talon proti Kassadinovi podstatnou výhodu.
AurelionSol	Mid	65.74%	Kassadin by měl bez problému roli vyhrát.
Qiyana	Mid	65.68 %	Qiyana by měla být od začátku ve značné výhodě.
Galio	Mid	65.67 %	Galio je začátkem zápasů mírně horší, ale postupem času Kassadina nakonec přemůže.

Tabulka 7.6: Šampioni doporučení našim systémem na potřebnou roli. Algoritmus vybíral šampiona na roli *mid* a přitom šampion na *mid* soupeře byl již zvolen.

Náš algoritmus nám doporučil z prvních šesti šampionů čtyři, kteří by podle našich expertních znalostí měli mít výhodu nad Kassadinem. Zbylé dva výběry, které nejsou tak dobré, mohly zapříčinit data o hráčích, které jsme doposud nijak neměnili.

7.4.3 Reakce algoritmu na příznaky hráčů

Abychom otestovali, jak náš algoritmus dokáže reagovat na data o hráčích, upravíme tyto data, které momentálně používáme tak, že u hráče modrého týmu, který bude hrát roli *top*, značně zvýšíme hodnotu u *přijatého poškození*. Z expertního pohledu bychom předpokládali, že hráč s vysokou hodnotou

u *přijatého poškození* hraje šampiony, kteří mají hodně životů a vysokou odolnost proti poškození (*tank*) nebo šampiony, kteří se velmi často učásastího bojů v průběhu hry (*fighter*). Náš algoritmus doporučil šampiony zobrazené tabulkou 7.7.

Jméno šampiona	Role	Pravděpodobnost výhry	Expertní názor na doporučení šampiona
Shen	Top	69.66 %	Shen odpovídá popisu <i>tanka</i> , v bojích je štítem pro své spoluhráče.
Riven	Top	69.49 %	Riven je <i>fighter</i> , primárně ale způsobuje poškození.
Kled	Top	69.48 %	Kled je <i>fighter</i> , hlavně uděluje poškození, ale také dokáže přijmout dost poškození za tým.
Lulu	Top	69.22 %	Neodpovídá ani jednomu popisu výše.
Singed	Top	68.91 %	Odpovídá popisu <i>tanka</i> .
Cassiopeia	Top	68.91 %	Cassiopeia neodpovídá ani jednomu popisu výše.

Tabulka 7.7: Šampioni doporučení našim systémem na potřebnou roli. Algoritmus vybíral šampiona na roli *top* a přitom hodnota informace *přijatého poškození* o hráči na této roli byla značně zvýšena.

Algoritmus nám na prvním místě doporučil Shena. Z expertního pohledu můžeme říct, že přesně odpovídá našemu předpokladu. Další doporučení šampioni, které by šlo spojit s vysokým *přijatým poškozením* jsou Riven, Kled a Singed.

Abychom si ověřili, že tyto výběry záleží na změněné informaci o hráči, upravíme pro následující experiment opět pro modrý tým u role *top* hodnotu *způsobeného poškození*. Nastavíme ji vysokou a budeme očekávat šampiony takové, kteří udělují vysoké poškození, což může být již dříve zmíněný *fighter*. Také šampioni (*carry*), jejichž cílem je během bojů v průběhu zápasu způsobit největší poškození, jsou očekávanou volbou.

Jméno šampiona	Role	Pravděpodobnost výhry	Expertní názor na doporučení šampiona
Riven	Top	70.18 %	Riven odpovídá popisu <i>fightera</i> a primárně způsobuje vysoké poškození.
Kled	Top	69.53 %	Kled je <i>fighter</i> , hlavně uděluje poškození, ale také dokáže přijmout dost poškození za tým.
Shen	Top	69.52 %	Shen odpovídá popisu <i>tanka</i> a není šampionem udělující vysoké poškození .
Cassiopeia	Top	69.49 %	Cassiopeia je <i>carry</i> , její hlavním úkolem je udělovat poškození.
Hecarim	Top	68.89 %	Hecarim se řadí mezi <i>fightery</i> , zastupuje podobnou funkci jako Kled.
Lulu	Top	68.76 %	Lulu neodpovídá ani jednomu popisu výše.

Tabulka 7.8: Šampioni doporučení našim systémem na potřebnou roli. Algoritmus vybíral šampiona na roli *top* a přitom hodnota informace *způsobeného poškození* o hráči na této roli byla značně zvýšena.

Tím, že jsme dostali rozdílné výsledky (tabulka 7.8) oproti předchozímu experimentu, ukazujeme, že příznaky hráče ovlivňují doporučení algoritmu. Jako nejvhodnějšího šampiona vybral náš algoritmus Riven. Doporučené šampiony jsme následně zhodnotili z expertního pohledu podložené externím zdrojem stránek League of Legends [16]. Opět čtyři šampioni z šesti odpovídali našemu očekávání. Riven primárně způsobuje vysoké poškození stejně tak i Kled, Hecarim a Cassiopeia.

Kapitola 8

Závěr

Draftování šampionů v MOBA hrách je klíčovým faktorem celého zápasu, od kterého se odvíjí průběh celé hry. Výběr správného šampiona závisí na několika faktorech, které jsou v první řadě ovlivněny sestavou týmu a výběry soupeřů. Hráči v limitovaném čase musí zvážit všechny důležité faktory a také své vlastní zkušenosti. Provést správný draft je náročné nejen pro zkušené hráče, ale dokonce i pro profesionální týmy, které čas od času udělají nějakou chybu.

Cílem této bakalářské práce bylo navržení a implementace vhodných modelů predikující vítěze zápasu před/po draftu společně s asistenčním algoritmem pro draft v MOBA hrách.

Měli jsme k dispozici data o 60 tisících zápasech, obsahující informace o daném zápase a hlavně o každém hráči zápasu. Implementovali jsme *champ_model*, který využívá informace o *id* šampionů a také *feat_model*, který k těmto datům navíc pracuje s příznaky hráčů. Pro oba modely jsme s pomocí grid searche našli nejlepší hyperparametry. Ty nejúspěšnější modely jsme poté použili v asistenčním algoritmu. Pro hledání optimální strategie při draftu jsme použili Monte Carlo Tree Search, kde jsme čelili problému asymetrických informací.

Na závěr jsme provedli experimenty s vybranými modely na testovací množině dat. *Champ_model* uspěl s přesností 54,06 % a *feat_model* dosáhl 53,45 %. Následně jsme prokázali funkčnost asistenčního algoritmu, který dokáže přiřadit šampiona na správnou roli, doporučit vhodného šampiona na základě výběrů spoluhráčů a soupeřů, a také vhodně doporučit šampiona v závislosti na příznacích hráče. Oproti stávající literatuře pro asistenční algoritmus jsme v naší práci použili asymetrické informace a přidali jsme uživateli možnost si zvolit roli, na kterou získá doporučení.

8.1 Možná rozšíření

Během řešení problematiky této bakalářské práce jsme narazili na mnoho věcí, které jsme z časových důvodů nestihli prozkoumat. Přesto, že naše

modely neuronových sítí dosáhly uspokojujících výsledků, věříme, že je zde prostor pro jejich zdokonalení. Pro *feat_model* jsme vyzkoušeli pět významných příznaků a rozšíření v tomto směru by mohlo znamenat zlepšení výkonu. Například by mohlo pomoci zvážení skupiny šampionů, které daný hráč hraje. Také přidání některých z dalších 50 příznaků, které naše data obsahovaly, by určitě pozitivně rozšířilo vstupní data. Pro *champ_model*, jehož vstupní data jsou pouze *id* šampionů, by mohlo být přínosné vyzkoušet data zápasu z nižší *Elo* úrovně.

Náš asistenční algoritmus lze obohatit o doporučení, jaké šampiony zakazovat. Pro uživatele by také bylo užitečné zdůvodnění volby algoritmu, například k doporučení přidat informaci, ke které sestavě šampionů se daný šampion hodí, či proti kterým šampionům je dominantní.



Literatura

- [1] Atish Agarwala and Michael Pearce. Learning dota 2 team compositions. *Sl: sn*, 2014.
- [2] R Ani, Vishnu Harikumar, Arjun K Devan, and OS Deepa. Victory prediction in league of legends using feature selection and ensemble methods. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 74–77. IEEE, 2019.
- [3] CESNET. Metacentrum virtual organization, 2021. URL: <https://metavo.metacentrum.cz/>.
- [4] Zhengxing Chen, Truong-Huy D Nguyen, Yuyu Xu, Christopher Amato, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. The art of drafting: a team-oriented hero recommendation system for multiplayer online battle arena games. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 200–208, 2018.
- [5] Zhengxing Chen, Yuyu Xu, Truong-Huy D Nguyen, Yizhou Sun, and Magy Seif El-Nasr. Modeling game avatar synergy and opposition through embedding in multiplayer online battle arena games. *arXiv preprint arXiv:1803.10402*, 2018.
- [6] Kevin Conley and Daniel Perry. How does he saw me? a recommendation engine for picking heroes in dota 2. *Np, nd Web*, 7, 2013.
- [7] CounterStats. Counter stats, 2019. URL: <https://www.counterstats.net/>.
- [8] Vinicios da Costa Oliveira, Bruno José Placides, Matheus de Freitas Oliveira Baffa, and Alex Fernandes da Veiga Machado. A hybrid approach to build automatic team composition in league of legends. *Proceedings of SBGames*, 2017.
- [9] Fernando Rubio Garcia. League of legends high elo patch 10.16, 2020. URL: <https://www.kaggle.com/fernandorubiogarcia/league-of-legends-high-elo-patch-1016>. Accessed: 2021-05-12.

- [10] Daniel Gourdeau and Louis Archambault. Discriminative neural network for hero selection in professional heroes of the storm and dota 2. *IEEE Transactions on Games*, 2020.
- [11] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [12] Victoria J Hodge, Sam Michael Devlin, Nicholas John Sephton, Florian Oliver Block, Peter Ivan Cowling, and Anders Drachen. Win prediction in multi-player esports: Live professional match prediction. *IEEE Transactions on Games*, 2019.
- [13] Adam Katona, Ryan Spick, Victoria J Hodge, Simon Demediuk, Florian Block, Anders Drachen, and James Alfred Walker. Time to die: Death prediction in dota 2 using deep learning. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2019.
- [14] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European Conference on machine learning*, pages 282–293. Springer, 2006.
- [15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [16] Inc. Riot Games. League of legends, 2021. URL: <https://eune.leagueoflegends.com/cs-cz/>.
- [17] Anna Sapienza, Palash Goyal, and Emilio Ferrara. Deep neural networks for optimal team composition. *Frontiers in Big Data*, 2:14, 2019.
- [18] Aleksandr Semenov, Peter Romov, Sergey Korolev, Daniil Yashkov, and Kirill Neklyudov. Performance of machine learning algorithms in predicting game outcome from drafts in dota 2. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 26–37. Springer, 2016.

- [19] Antonio Luis Cardoso Silva, Gisele Lobo Pappa, and Luiz Chaimowicz. Continuous outcome prediction of league of legends competitive matches using recurrent neural networks. In *SBC-Proceedings of SBCGames*, pages 2179–2259, 2018.
- [20] Adam Summerville, Michael Cook, and Ben Steenhuisen. Draft-analysis of the ancients: predicting draft picks in dota 2 using machine learning. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.
- [21] Luwei Wang, Yan Tang, and Jie Liu. Wpqa: A gaming support system based on machine learning and knowledge graph. In *Joint International Semantic Technology Conference*, pages 191–204. Springer, 2019.
- [22] Nanzhi Wang, Lin Li, Linlong Xiao, Guocai Yang, and Yue Zhou. Outcome prediction of dota2 using machine learning methods. In *Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence*, pages 61–67, 2018.
- [23] Tian Wang. Predictive analysis on esports games: A case study on league of legends (lol) esports tournaments. 2018.
- [24] Weiqi Wang. *Predicting multiplayer online battle arena (moba) game outcome based on hero draft data*. PhD thesis, Dublin, National College of Ireland, 2016.
- [25] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.
- [26] Cheng Yu, Wan-ning Zhu, and Yu-meng Sun. E-sports ban/pick prediction based on bi-lstm meta learning network. In *International Conference on Artificial Intelligence and Security*, pages 97–105. Springer, 2019.
- [27] Lei Zhang, Chenbo Xu, Yihua Gao, Yi Han, Xiaojiang Du, and Zhihong Tian. Improved dota2 lineup recommendation model based on a bidirectional lstm. *Tsinghua Science and Technology*, 25(6):712–720, 2020.