SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2210

DETEKCIJA KORISNIČKIH INTERAKCIJA U SVRHU STVARNO-VREMENSKE PROCJENE ISKUSTVENE KVALITETE USLUGE YOUTUBE TEMELJENA NA STROJNOM UČENJU

Filip Matić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2210

DETEKCIJA KORISNIČKIH INTERAKCIJA U SVRHU STVARNO-VREMENSKE PROCJENE ISKUSTVENE KVALITETE USLUGE YOUTUBE TEMELJENA NA STROJNOM UČENJU

Filip Matić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 13. ožujka 2020.

DIPLOMSKI ZADATAK br. 2210

Pristupnik:	**Filip Matić (0036490845)**
Studij:	Računarstvo
Profil:	Programsko inženjerstvo i informacijski sustavi
Mentor:	izv. prof. dr. sc. Lea Skorin-Kapov

Zadatak: **Detekcija korisničkih interakcija u svrhu stvarno-vremenske procjene iskustvene kvalitete usluge YouTube temeljena na strojnom učenju**

Opis zadatka:

Posljednjih godina svjedočimo konstantnom značajnom porastu globalnog Internet prometa, čemu najviše doprinosi masovno korištenje usluga strujanja videa. U tim uvjetima, davatelji mrežnih usluga su suočeni s izazovom efikasnog upravljanja ograničenim mrežnim resursima, uz pružanje zadovoljavajuće razine iskustvene kvalitete krajnjim korisnicima. Imajući u vidu enkripciju mrežnog prometa, čak i samo praćenje performansi usluga strujanja videa u mreži predstavlja izazov. U tom kontekstu, rješenja za praćenje iskustvene kvalitete temeljena na metodama strojnog učenja pokazala su se kao obećavajuća. Međutim, korisničko ponašanje, a naročito korisničke interakcije vezane za prikazivanje video sadržaja, utječu na uzorke u mrežnom prometu, a samim time i (najčešće negativno) na procjenu iskustvene kvalitete. Vaš zadatak je uspostaviti okruženje za automatizirano prikupljanje velike količine podataka vezanih uz prijenos YouTube video sadržaja na zahtjev te unutar tog okruženja implementirati različite korisničke interakcije koje će se programski aktivirati za vrijeme prikazivanja video sadržaja. Nakon uspostave okruženja potrebno je prikupiti veliku količinu podataka koji za svaki prikazani video sadržaj, sa ili bez uključenih korisničkih interakcija, uključuju značajke mrežnog prometa i podatke o performansama aplikacijske razine. Pomoću tehnika strojnog učenja potrebno je ispitati mogućnost otkrivanja korisničkih interakcija u stvarnom vremenu te analizirati kako iste utječu na promjenu prometnih uzoraka. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Rok za predaju rada: 30. lipnja 2020.

Sadržaj

Uvod

U posljednje vrijeme na globalnoj razini bilježimo sve veći porast broja korisnika Interneta. Za 2018. godinu taj broj iznosio je 3.9 milijardi, dok je u 2020. iznosio 4.6 milijardi [1]. Prema predviđanjima Cisca, gotovo dvije trećine svjetske populacije imati će pristup Internetu do 2023. godine [2].

To dovodi i do sve većeg porasta količine Internet prometa na globalnoj razini. Upravo sve učestalije korištenje usluga video strujanja pridonosi tom porastu. Prema dostupnim podacima, u 2020. strujanje video sadržaja činilo je 62% ukupnog Internet prometa. Najveći dio mobilnog Internet prometa odnosio se na YouTube [3].

Većina tog prometa šalje se isključivo putem protokola HTTPS (HTTP Secure) koji kriptira podatke. Upravo to predstavlja problem za davatelje mrežnih usluga koji nemaju uvid u performanse i parametre kvalitete video tokova koji prolaze njihovom mrežom. Budući da davatelji usluga imaju ograničene mrežne resurse, potrebno je u takvim uvjetima pružiti odgovarajuću razinu iskustvene kvalitete krajnjim korisnicima. U takvim su se okolnostima razvila rješenja za praćenje iskustvene kvalitete temeljena na metodama strojnog učenja koja su se pokazala kao potencijalno obećavajuća.

No, korisničke interakcije prilikomprikazivanja video sadržaja utječu na uzorke u mrežnom prometu i otežavaju procjenu iskustvene kvalitete. U sklopu ovog rada će se pomoću tehnika strojnog učenja ispitati mogućnost otkrivanja korisničkih interakcija u stvarnom vremenu na temelju analize mrežnog prometa. Radi prikupljanja što veće količine podataka, potrebno je automatizirati proces prikupljanja podataka vezanih uz prijenos YouTube video sadržaja na zahtjev na Android mobilnom uređaju. Kako bi se proces prikupljanja podataka automatizirao, potrebno je programski implementirati korisničke interakcije za vrijeme prikazivanja video sadržaja.

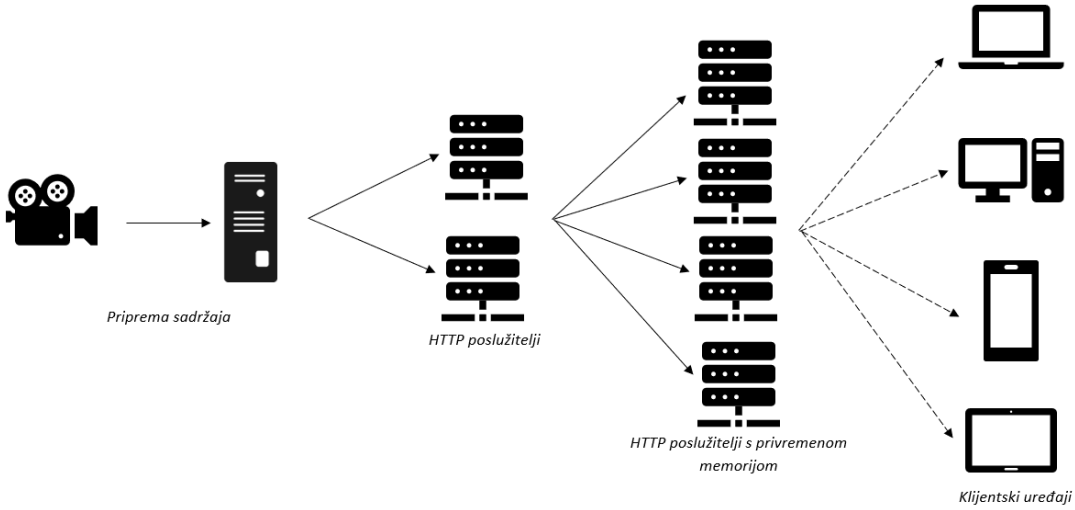
Ovaj rad sastoji se od sedam poglavlja. U prvom poglavlju dan je pregled standarda za dinamičko prilagodljivo strujanje putem protokola HTTP. U drugom poglavlju opisuje se pojam iskustvene kvalitete. Treće poglavlje

opisuje korisničke interakcije za vrijeme strujanja video sadržaja. Nadalje, u četvrtom poglavlju razrađuje se proces prikupljanja video ID-eva i pripadnih meta podataka. Peto poglavlje sadrži opis laboratorijskog okruženja u kojem su se vršila mjerenja i opisuje proces obrade podataka. U šestom poglavlju daje se pregled definicija pojmova strojnog učenja i mjera za vrednovanje modela strojnog učenja. Također, opisan je i postupak izgradnje modela strojnog učenja i analiza dobivenih rezultata. Na kraju se nalazi zaključak, popis literature, sažetak, popis slika, popis tablica.

1. Dinamičko prilagodljivo strujanje putem protokola HTTP

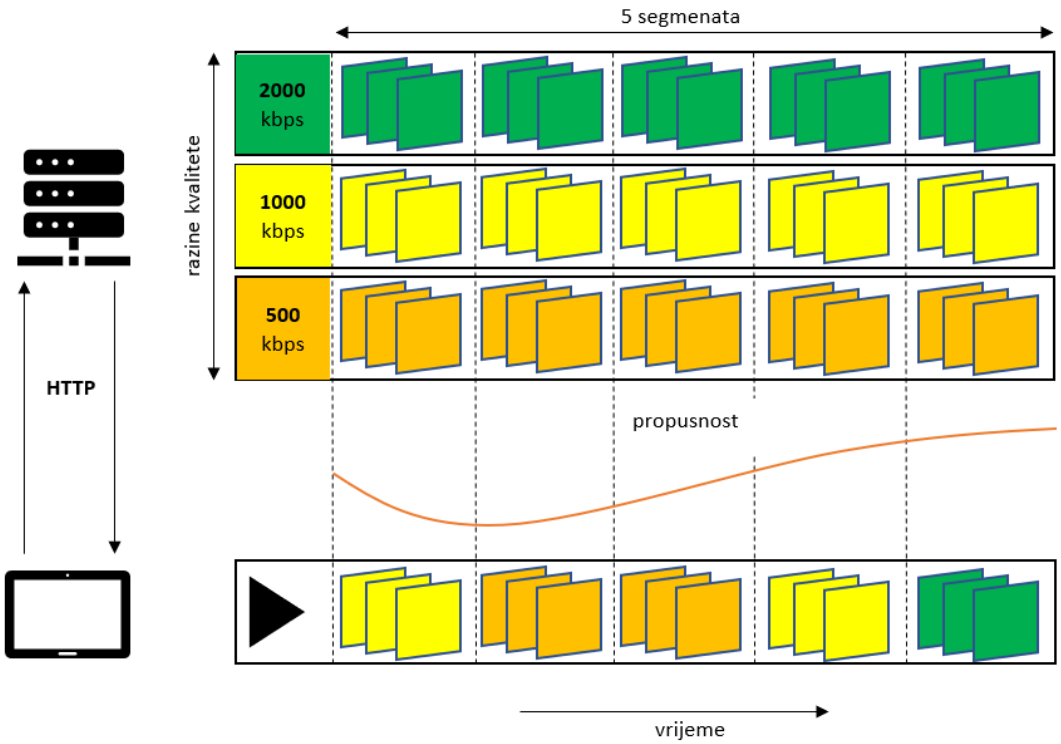
Veliki izazov za strujanje višemedijskog sadržaja na Internetu predstavljaju promjenjivi uvjeti u mreži kao što su promjene širine propusnog pojasa (engl. *bandwidth*), kašnjenja (engl. *delay*) i gubitak paketa (engl. *packet loss*). Dinamičko prilagodljivo strujanje putem protokola HTTP (engl. *Dynamic Adaptive Streaming over HTTP*, *DASH*) je MPEG standard koji omogućuje strujanje višemedijskog sadržaja u najvećoj mogućoj kvaliteti prilagođavanjem sadržaja trenutnim mrežnim uvjetima, mogućnostima uređaja korisnika i trenutnom opterećenju poslužitelja [34]. Osim toga pružateljima usluga olakšava upravljanje resursima i osigurava poboljšanje iskustvene kvalitete (engl. *Quality of Experience, QoE*) korištenjem povratnih informacija sa klijenta. Najprimjetnija prednost DASH-a nad strujanjem u stalnoj kvaliteti je kraće vrijeme učitavanja i manji broj prekida kod reprodukcije višemedijskog sadržaja [4].

Slika 1.1 prikazuje arhitekturu distribucije medija za prilagodljivo strujanje putem protokola HTTP. Proces pripreme medija (engl. *Media Preparation*) dijeli datoteku video sadržaja na male segmente duljine od dvije do deset sekundi koji mogu biti kodirani u više različitih formata, različitih rezolucija i brzina video kodiranja (engl. *bitrate*). Nakon toga se segmenti pohranjuju na jedan ili više HTTP poslužitelja (engl. *Media HTTP Origin Servers*) i na poslužitelje s privremenom memorijom (engl. *HTTP Cache Servers*) gdje je svaki segment adresiran HTTP URL-om kako bi bio dostupan klijentskim uređajima (engl. *Client Devices*). Zajedno sa segmentima pohranjena je i datoteka koja sadrži opis višemedijskog sadržaja (engl. *Media Presentation Description*, *MPD*). MPD sadrži informacije o svim dostupnim razinama kvalitete segmenata, URL-ove segmenata i metapodatke koji opisuju odnose između segmenata te na koji način segmenti formiraju čitav višemedijski sadržaj [5].



Slika 1.1 DASH arhitektura distribucije sadržaja.

Klijentska aplikacija od poslužitelja prvo dohvaća MPD datoteku iz koje dobiva listu dostupnih razina kvalitete, rezolucije i brzine kodiranja. Klijent zatim kontinuirano procjenjuje mrežne uvjete kao što je propusnost (engl. *throughput*) na temelju kojih odlučuje koje će segmente preuzeti. Preuzeti segmenti pohranjuju se u međuspremnik (engl. *buffer*). Ukoliko prilikom gledanja videa dođe do pogoršanja mrežnih uvjeta, video se neće prekinuti nego će se preuzimati segmenti manje kvalitete. Čim se uvjeti poboljšaju, povećat će se i kvaliteta segmenata koji se preuzimaju što uzrokuje povećanje ukupne iskustvene kvalitete korisnika [4], [6] (Slika 1.2).



Slika 1.2 Primjer segmenata različite kvalitete u međuspremniku

Jedan od najpopularnijih servisa koji koristi dinamičko prilagodljivo strujanje putem protokola HTTP je YouTube. Trenutno YouTube broji preko dvije milijarde prijavljenih korisnika mjesečno koji dnevno pregledaju milijardu sati video sadržaja [7]. YouTube-u se može pristupiti preko Internetskog preglednika ili nativne aplikacije za pametne telefone. Servis je dostupan na 80 različitih jezika, a preko 70 posto pregleda ostvareno je na mobilnim uređajima. Od raznih sadržaja na YouTube-u se mogu pronaći video na zahtjev (engl. *video on demand*, *VOD*), video sadržaj uživo (engl. *livestream*), 3D video i 360 video.

2. Iskustvena kvaliteta

Krajem prošlog stoljeća, prije nego što je definiran pojam iskustvene kvalitete, kvaliteta komunikacijskih usluga poistovjećivala se s pojmom kvalitete usluge. No, zbog povećanja zastupljenosti višemedijskog sadržaja na Internetu, pojam iskustvene kvalitete počinje dobivati sve veću pozornost [8]. U nastavku ovog poglavlja su detaljnije opisani pojmovi kvalitete usluge i iskustvene kvalitete, te razlike među njima. Osim toga opisana je i metoda mjerenja iskustvene kvalitete i parametri koji imaju utjecaj na razinu iskustvene kvalitete usluga strujanja video sadržaja.

2.1 Definicija iskustvene kvalitete

Kod kvalitete usluge (engl. *Quality of Service*, *QoS*) promatra se tehnička strana komunikacijskih usluga. Kvaliteta usluge definira se kao mjera koja opisuje performanse nekog sustava. Kod mjerenja QoS-a prilikom strujanja video sadržaja promatraju se različiti mrežni parametri poput veličine paketa, gubitka paketa (engl. *packet loss*), mrežne propusnosti i kašnjenja u prijenosu (engl. *transmission delay*). Za razliku od QoS-a, iskustvena kvaliteta (engl. *Quality of Experience*, QoE) usmjerena je prema zadovoljstvu korisnika kojemu se usluga isporučuje [9]. Formalno je QoE definiran kao stupanj zadovoljstva odnosno nezadovoljstva krajnjeg korisnika nekom uslugom [8]. Kod usluga prilagodljivog strujanja video sadržaja putem protokola HTTP, faktori koji najviše utječu na QoE su početno kašnjenje (engl. *initial delay*), zastajkivanje (engl. *stalling*) i reproducirana rezolucija [8].

Pružatelji mrežnih usluga (engl. *Internet Service Provider*, ISP) trude se krajnjim korisnicima osigurati što je moguće bolji QoE. Jako je važno razumjeti odnos između QoS-a i QoE-a. Dobar QoE ne može se osigurati ukoliko je QoS jako loš, ali se u slučaju pogoršanja QoS-a pomoću tehnologija kao što je DASH može smanjiti degradacija QoE-a [10].

2.2 Vrednovanje iskustvene kvalitete

Za vrednovanje iskustvene kvalitete postoje subjektivne i objektivne metode. Među najčešće korištenim mjerama u uporabi za subjektivno ispitivanje iskustvene kvalitete je mjera srednje ocjene korisničkog iskustva (engl. *Mean Opinion Score*, *MOS*). Jedan od problema subjektivnih metoda mjerenja QoE-a je činjenica da svaki korisnik različito percipira kvalitetu [11]. Međunarodna telekomunikacijska unija (engl. *International Telecommunication Union*, *ITU*) definira ljestvicu korisničkog iskustva kao [12]: „vrijednost na unaprijed određenoj skali koju ispitanik dodijeli usluzi prema svojem mišljenju“. Jedan od primjera je da svaki korisnik ocjenjuje svoj doživljaj usluge brojem između 1 i 5 (1 je oznaka za najgoru ocjenu, a 5 za najbolju). Naziv brojčane skale za određivanje QoE-a je ACR (engl. *Absolute Category Rating*) (Tablica 2.1). Konačno, MOS je prosjek ocjena svih ispitanih korisnika.

Tablica 2.1 MOS vrijednosti

Ocjena	MOS
5	Izvrsno
4	Dobro
3	Prihvatljivo
2	Loše
1	Vrlo loše

2.3 Iskustvena kvaliteta strujanja video sadržaja putem protokola HTTP

Kod strujanja video sadržaja putem protokola HTTP preuzimanje (engl. *download*) i prikazivanje (engl. *playback*) sadržaja odvija se istovremeno. Podaci se klijentu isporučuju putem protokola HTTP i pohranjuju se u

međuspremnik. Nakon što se preuzme dovoljna količina podataka, klijent može započeti s reprodukcijom sadržaja. Za vrijeme prijenosa podataka različiti problemi u mreži (npr. zagušenje, zbog kojeg dolazi do gubitka paketa, smanjene širine raspoloživog mrežnog pojasa, kašnjenja i kolebanja kašnjenja) mogu smanjiti propusnost zbog čega preuzimanje ne može pratiti prikazivanje sadržaja. Posljedica toga je pražnjenje međuspremnika i prekid prikazivanja videa sve dok se dio međuspremnika ponovo napuni [4].

Za strujanje video sadržaja putem protokola HTTP, faktori koji imaju najveći utjecaj na QoE su početno kašnjenje i zastajkivanje. Osim toga, kod prilagodljivog strujanja video sadržaja se pojavljuje novi faktor – promjena kvalitete sadržaja (Tablica 2.2) [4].

Tablica 2.2 Faktori koji utječu na QoE prilikom strujanja video sadržaja

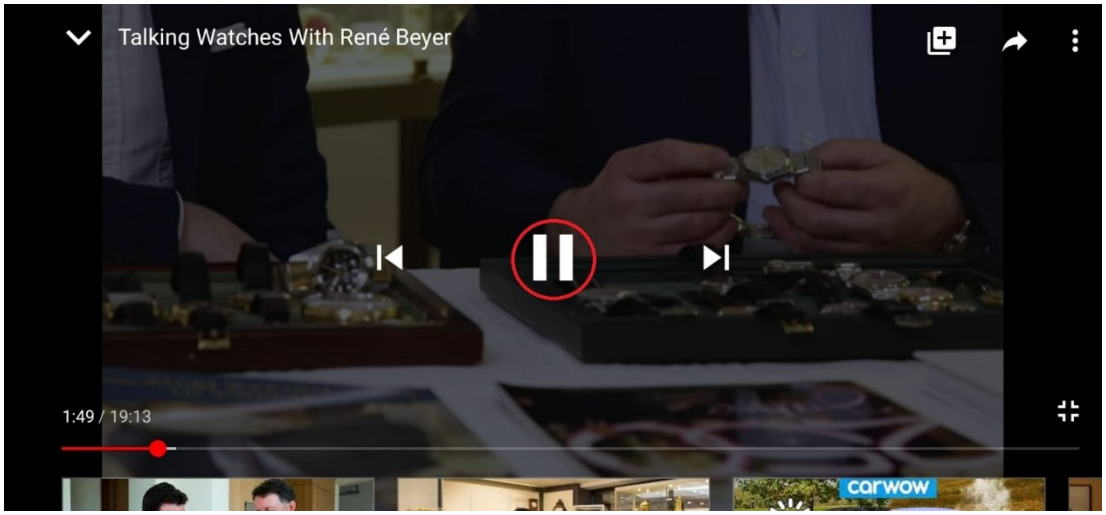
QoE faktor	engl.	opis
Početno kašnjenje	*initial delay*	Vrijeme početnog punjenja međuspremnika prije početka prikazivanja
Zastajkivanje	*stalling*	Prekid prikazivanja uzrokovan pražnjenjem međuspremnika
Prilagodba kvalitete	*adaptation*	Promjena kvalitete videa za vrijeme prikazivanja uzrokovana prilagođavanjem trenutnim uvjetima u mreži

Svaki korisnik ima različitu percepciju kvalitete, pa će tako netko više preferirati veću kvalitetu videa, dok će nekom drugom biti važnije da nema početnog kašnjenja i zastajkivanja. Osim navedenih parametara, na iskustvenu kvalitetu utječu i interakcije inicirane od strane korisnika koje su detaljnije opisane u Poglavlju 3.

3. Korisničke interakcije

Korisničke interakcije su akcije inicirane od strane korisnika za vrijeme strujanja i reprodukcije video sadržaja. Interakcije koje YouTube aplikacija omogućuje korisniku su: pauziranje videa, premotavanje videa unaprijed i unatrag, promjena kvalitete videa, prebacivanje na sljedeći video, promjena glasnoće, pisanje komentara, označavanje videa sa „sviđa mi se“, označavanje videa sa „ne sviđa mi se“, promjena brzine reprodukcije videa (engl. *playback speed*), uključivanje titlova (engl. *captions*), te dijeljenje videa (engl. *share*). Neke korisničke interakcije, poput označavanja videa sa „sviđa mi se“ ili „ne sviđa mi se“, nemaju nikakvog utjecaja na reprodukciju videa. U sklopu ovog rada stavljen je fokus na interakcije koje prekidaju reprodukciju videa i time mogu imati veliki utjecaj na QoE [13]: pauziranje, premotavanje videa unaprijed i prebacivanje na sljedeći video.

Interakcija pauze (engl. *pause interaction*) zaustavlja reprodukciju videa. Interakcija se postiže klikom na gumb sa ikonom koja predstavlja pauzu (Slika 3.1). Kada je video pauziran, klikom na gumb sa ikonom za pokretanje, video nastavlja s reprodukcijom od trenutka u kojem je bio zaustavljen.

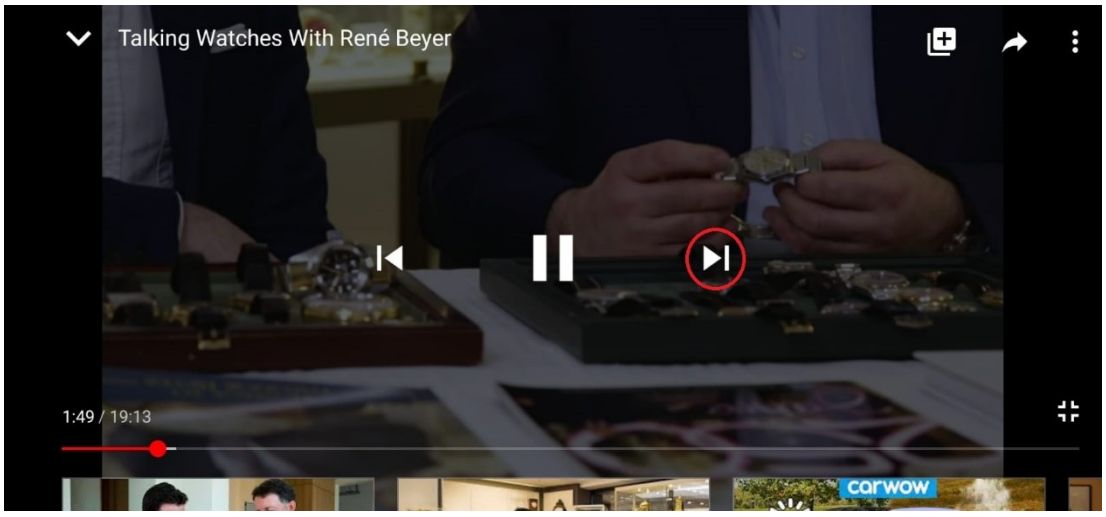


Slika 3.1 Interakcija pauze

Interakcija premotavanja videa unaprijed (engl. *seek forward*) postiže se tapkanjem po ekranu na dio videa koji se još nije prikazao (označeni dio na Slici 3.2).

Slika 3.2 Interakcija premotavanja videa unaprijed

Interakcija prebacivanje videa (engl. *skip video*) označava prekid reprodukcije trenutnog videa i otvaranje sljedećeg videa. Interakcija se postiže klikom na gumb za pokretanje sljedećeg videa (Slika 3.3).



Slika 3.3 Interakcija prebacivanja videa

4. Baza video ID-eva i metapodataka

U ovom je poglavlju detaljno opisan proces prikupljanja liste YouTube video ID-eva i pripadnih metapodataka. Navedeni podaci su potrebni da bi se proces prikupljanja podataka s aplikacijske razine mogao automatizirati. YouTube video ID je jedinstveni identifikator videa. Metapodaci služe da bi se lista video ID-eva mogla filtrirati po parametrima kao što su broj pregleda, broj oznaka „sviđa mi se“, datumu objave, dostupnim rezolucijama. Konačna svrha je dobiti listu relevantnih videa koji su dostupni u visokoj rezoluciji. Pri tome je nužno da prikupljeni skup podataka bude dovoljno velik da se iz njega može dobiti filtrirana lista videa nad kojima će se vršiti daljnje istraživanje.

4.1 Prikupljanje podataka

YouTubeVideoMetadata.java

Za prikupljanje općih podataka o videima napravljena je skripta u programskom jeziku Java (Isječak koda 4.1), a za pohranu podataka koristila se mysql baza podataka. Za dohvaćanje video ID-eva koristila se krajnja točka (engl. *endpoint*) *„/search/list?q={riječ}*“ YouTube Data API-a koja ima istu ulogu kao i pretraživanje videa na YouTube-ovoj stranici. Da bi se dobili relevantni rezultati pretraživanja, za generiranje engleskih riječi koristila se biblioteka (engl. *library*) random-word-generator [14]. Zatim se za dohvaćanje metapodataka o videu koristila krajnja točka *„/videos/list?id={videoID}*“ koja vraća podatke o videu kao što su datum objave, naslov, naziv kanala, trajanje, broj pregleda, broj ocjena „sviđa mi se“ (engl. *like*) i „ne sviđa mi se“ (engl. *dislike*). Dobiveni podaci se na kraju spremaju u tablicu *video*, a detaljniji opis svih podataka koji se spremaju za svaki video nalazi se u Tablici 4.1.

Isječak koda 4.1 Pseudokod skripte za prikupljanje video metapodataka

```
rijec = generiraj_rijec()
videoIdList = api_odgovor(/search/list/q=rijec)
Za svaki videoID iz videoIdList
{
        videoMetadata = api_odgovor(/video/list/id=videoID)
        spremi_u_bazu(videoMetadata)
}
```

Tablica 4.1 Opis metapodataka o videu koji su pohranjeni u bazi

Naziv	Opis
videoID	Jedinstveni identifikator YouTube videozapisa
publishedAt	Vremenska oznaka (engl. *timestamp*) kada je videozapis objavljen
channelID	Jedinstveni identifikator YouTube kanala na kojem je objavljen videozapis
title	Naziv videozapisa
channelTitle	Naziv kanala na kojem je objavljen videozapis
categoryId	Jedinstveni identifikator kategorije kojoj pripada videozapis
duration	Ukupno trajanje videozapisa u formatu ISO 8601
dimension	Dimenzija u kojoj je videozapis dostupan (npr. '*2d*')
definition	Definicija slike videozapisa (npr. '*hd*')
projection	Projekcija videozapisa (npr. '*rectangular*')
uploadStatus	Status učitavanja (engl. *upload*) videozapisa
privacyStatus	Status privatnosti videozapisa
license	Licenca videozapisa

madeForKids	Oznaka je li videozapis prikladan za djecu
viewCount	Ukupan broj pregleda
likeCount	Ukupan broj *like*-ova
dislikeCount	Ukupan broj *dislike*-ova
favoriteCount	Ukupan broj favorita
commentCount	Ukupan broj komentara
created	Vremenska oznaka trenutka kada je zapis dodan u bazu podataka

Besplatna uporaba YouTube Data API-a ima dnevno ograničenje broja prikupljenih podataka o videima od 10000 jedinica/dan (engl. *units/day*), odnosno za nešto više od 300 videa [15]. Tijekom travnja i svibnja 2020. prikupljeni su podaci o 105001 videu, pri čemu je korišten veći broj API ključeva. Kako je trajanje videa izraženo u ISO 8601 formatu (primjer: „PT5M15S“ je 315 sekundi), napravljena je SQL procedura koja pretvara ISO 8601 format u sekunde (Isječak koda 4.2) i rezultate pretvorbe sprema u novu tablicu *duration_in_seconds*.

Isječak koda 4.2 SQL: pretvorba ISO 8601 formata u sekunde

```
SELECT videoID, videoDuration,
        CASE
            WHEN videoDuration LIKE 'P%DT%H%M%S' THEN
TIME_TO_SEC(STR_TO_DATE(duration, 'P%dDT%HH%iM%sS'))
            WHEN videoDuration LIKE 'P%DT%M%S' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'P%dDT%iM%sS'))
            WHEN videoDuration LIKE 'P%DT%H%S' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'P%dDT%HH%sS'))
            WHEN videoDuration LIKE 'P%DT%H%M' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'P%dDT%HH%iM'))
            WHEN videoDuration LIKE 'P%DT%H' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'P%dDT%HH'))
            WHEN videoDuration LIKE 'P%DT%M' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'P%dDT%iM'))
            WHEN videoDuration LIKE 'P%DT%S' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'P%dDT%sS'))
            WHEN videoDuration LIKE 'P%DT' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'P%dDT'))
            WHEN videoDuration LIKE 'PT%H%M%S' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'PT%HH%iM%sS'))
            WHEN videoDuration LIKE 'PT%H' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'PT%HH'))
            WHEN videoDuration LIKE 'PT%H%M' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'PT%HH%iM'))
            WHEN videoDuration LIKE 'PT%H%S' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'PT%HH%sS'))
            WHEN videoDuration LIKE 'PT%M' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'PT%iM'))
            WHEN videoDuration LIKE 'PT%M%S' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'PT%iM%sS'))
            WHEN videoDuration LIKE 'PT%S' THEN TIME_TO_SEC(STR_TO_DATE(duration,
'PT%sS'))
        END AS durationInSeconds
FROM video
```

YouTubeDlMetadata.java

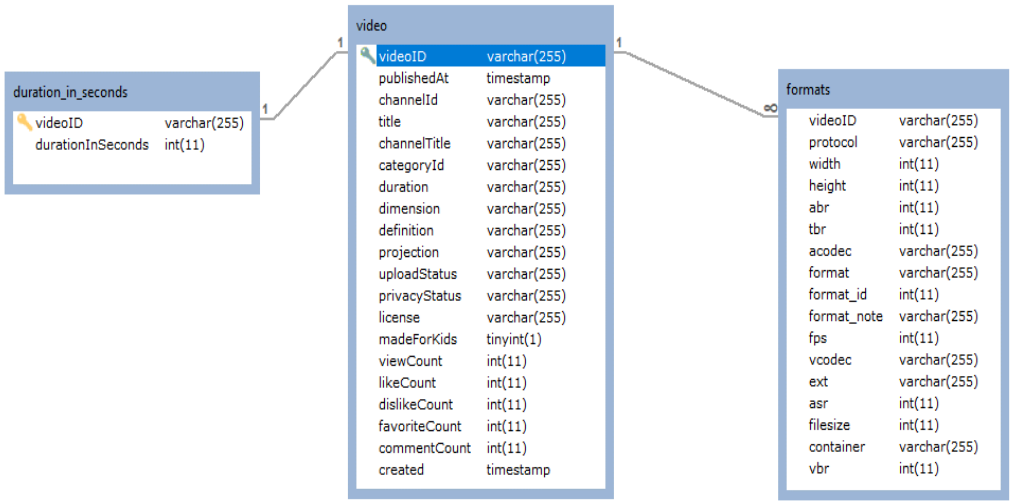
Druga skripta napravljena je za prikupljanje metapodataka o dostupnim verzijama video sadržaja, *YouTubeDlMetadata.java,* služi za izvlačenje informacija o svim formatima u kojima je video dostupan na YouTube-u. Navedeni podaci se prikupljaju pomoću *youtube-dl* alata [16]. Prikupljeni se podaci spremaju u bazu u tablicu *formats*, a detaljniji opis svih podataka koji se spremaju za svaki format videa nalazi se u Tablici 4.2.

Tablica 4.2 Opis metapodataka o video formatu [16]

Naziv	Opis
videoID	Jedinstveni identifikator YouTube videozapisa
protocol	Protokol koji se koristi za preuzimanje videozapisa
width	Širina videozapisa
height	Visina videozapisa
abr	Prosječna brzina audio kodiranja (engl. *bitrate*) u *KBit/s*
tbr	Prosječna brzina audio i video kodiranja (engl. *bitrate*) u *KBit/s*
acodec	Naziv korištenog audio kodeka
format	Opis formata
format_id	Jedinstveni identifikator formata
format_note	Dodatni opis formata
fps	Broj sličica u sekundi (engl. *frames per second*)
vcodec	Naziv korištenog video kodeka
ext	Ekstenzija video datoteke
asr	Frekvencija audio uzorkovanja (engl. *audio sampling rate*) u *Hz*
filesize	Veličina datoteke u *byte*-ovima
container	Naziv *container* formata
vbr	Prosječna brzina video kodiranja (engl. *bitrate*) u *KBit/s*

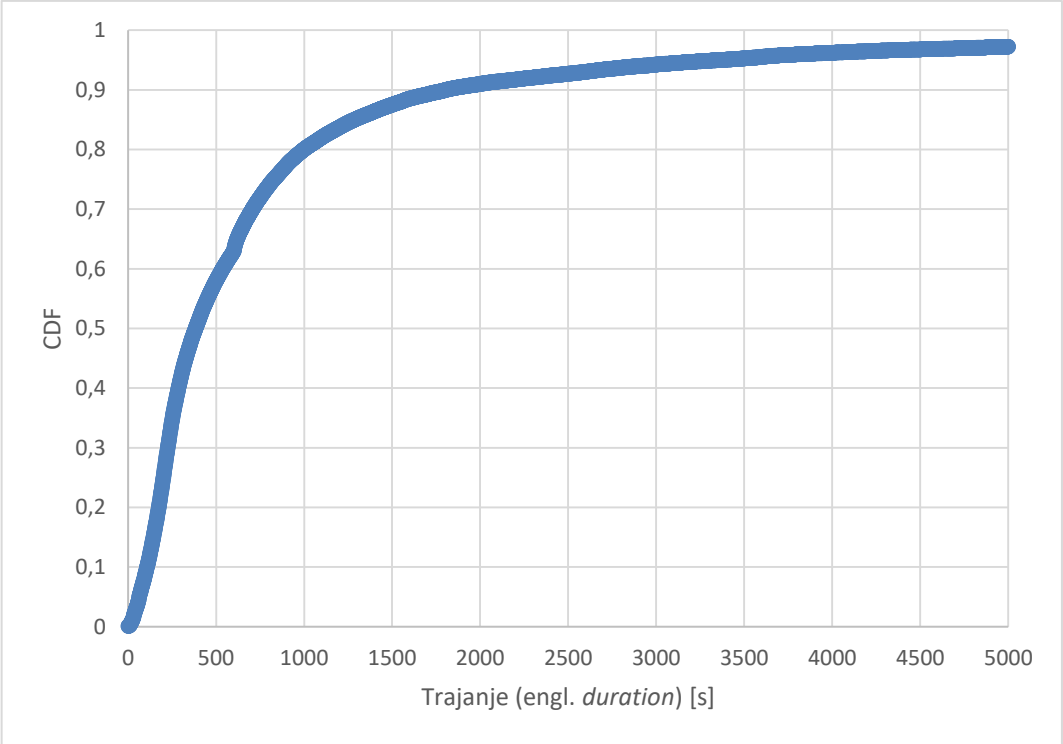
4.2 Analiza prikupljenih podataka

Nakon završetka procesa prikupljanja podataka o videima, u bazi se nalazi zapis o 105001 različitih videa. Struktura baze prikazana je na Slici 4.1.



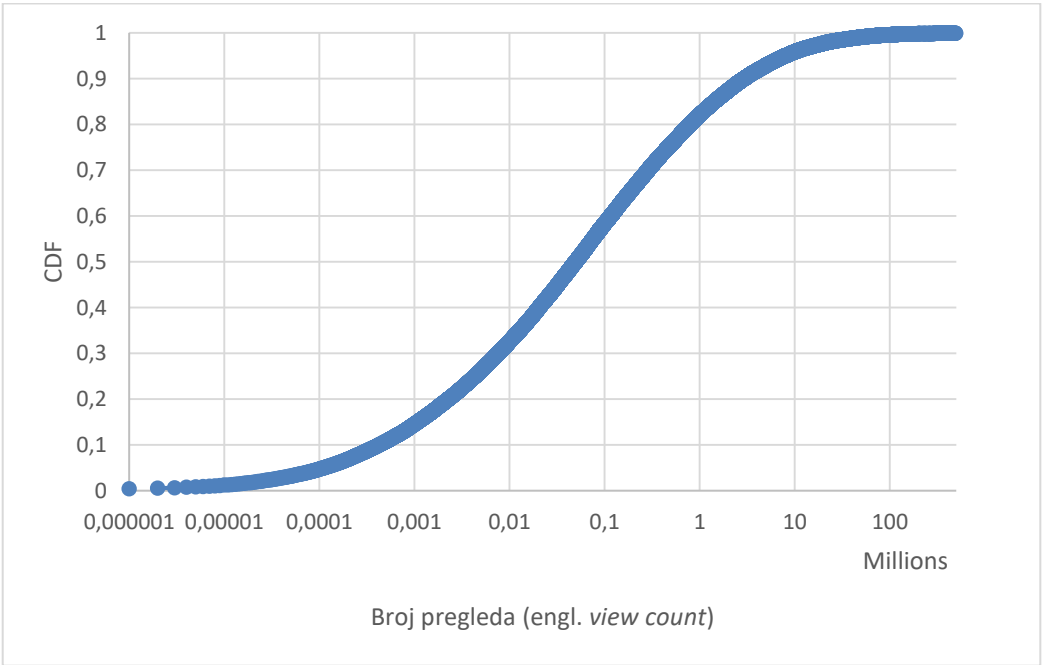
Slika 4.1 Shema baze youtube_metadata

Graf na Slici 4.2 prikazuje raspodjelu prikupljenih videa prema duljini trajanja. Iz grafa je vidljivo da oko 70% videa ne traje dulje od 750 sekundi. Prosječno trajanje videa je 941.5 sekundi, a medijan je 383 sekundi.



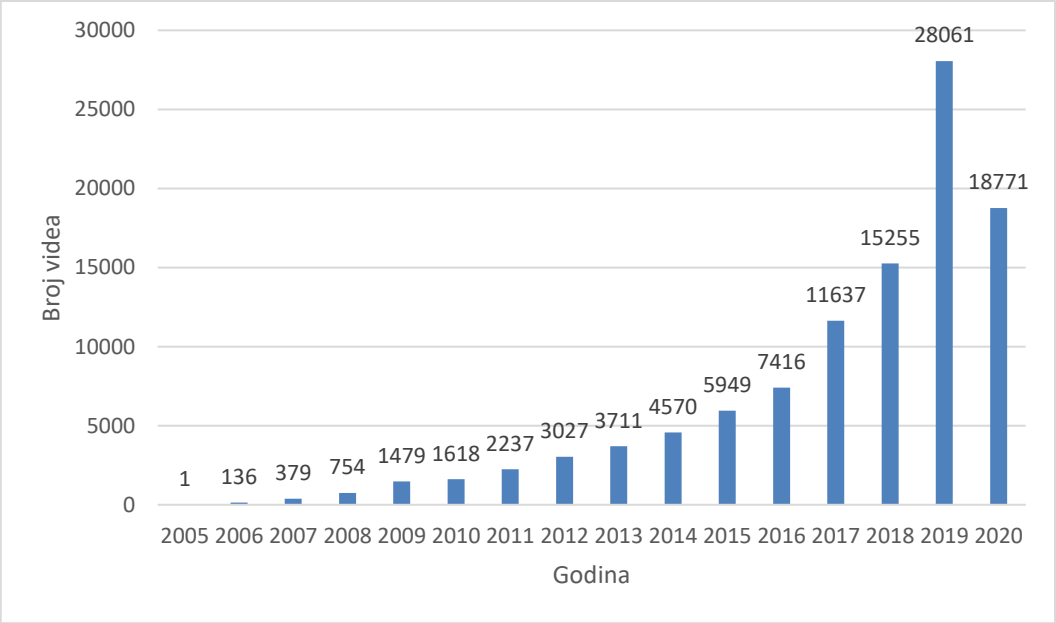
Slika 4.2 Distribucija videa prema trajanju

Graf distribucije videa prema broju pregleda se nalazi na Slici 4.3. Manje od 20% videa ima više od milijun pregleda. Prosječan broj pregleda po videu je 2759063, a medijan je 49650.



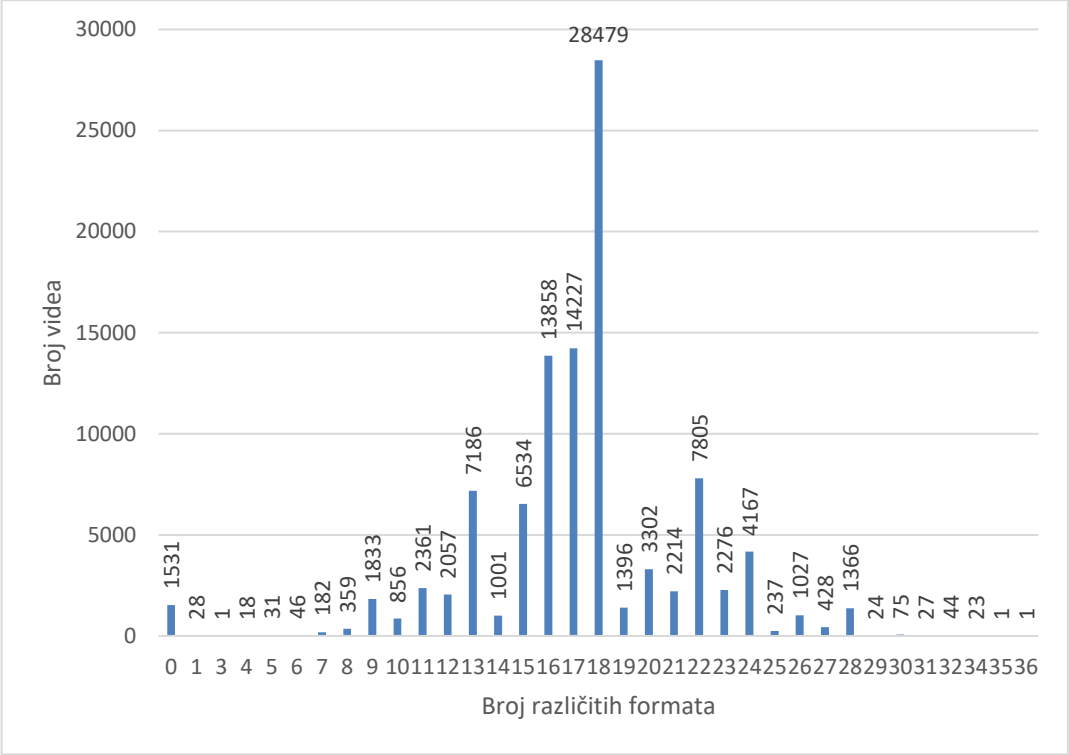
Slika 4.3 Distribucija videa prema broju pregleda

Prikupljeni videozapisi su objavljeni u periodu od 2005. do 2020. godine. Iz 2005. godine prikupljen je najmanji broj videa, a za svaku je sljedeću godinu prikupljen sve veći broj. 77% prikupljenih videa objavljeno je u zadnjih pet godina (Slika 4.4).

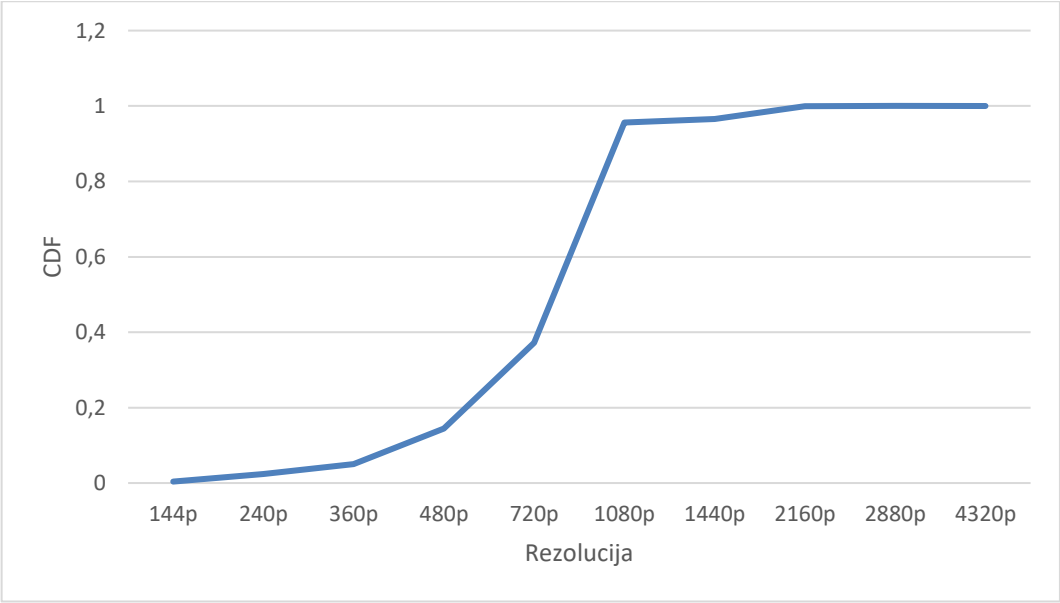


Slika 4.4 Distribucija videa prema godini objave

Graf na Slici 4.5 prikazuje raspodjelu videa prema broju dostupnih formata. YouTube za označavanje različitih formata koristi *itag* kodove [38]. To je brojčana vrijednost koja služi za identifikaciju i razlikovanje različitih razina kvalitete video sadržaja (npr. 136 označava 720p mp4 video format). 1531 video dostupan je u 0 formata – razlog tome je zato što su ti videozapisi u međuvremenu (u vremenu od pokretanja prve skripte do pokretanja druge skripte) uklonjeni s YouTube-a. Ostali videozapisi dostupni su u rasponu od 1 do 36 različitih formata. Prosječni broj dostupnih formata po videu je 17.27, odnosno 17.53 ako se izuzmu uklonjeni videozapisi, dok medijan u oba slučaja iznosi 18. Konačno, na Slici 4.6 se nalazi graf distribucije videa prema maksimalno dostupnoj rezoluciji.



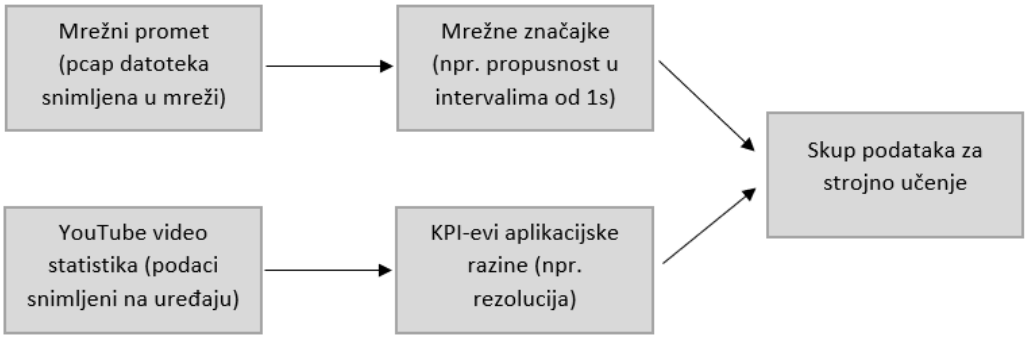
Slika 4.5 Distribucija videa prema broju dostupnih formata



Slika 4.6 Distribucija videa prema maksimalno dostupnoj rezoluciji

5. Metodologija rada

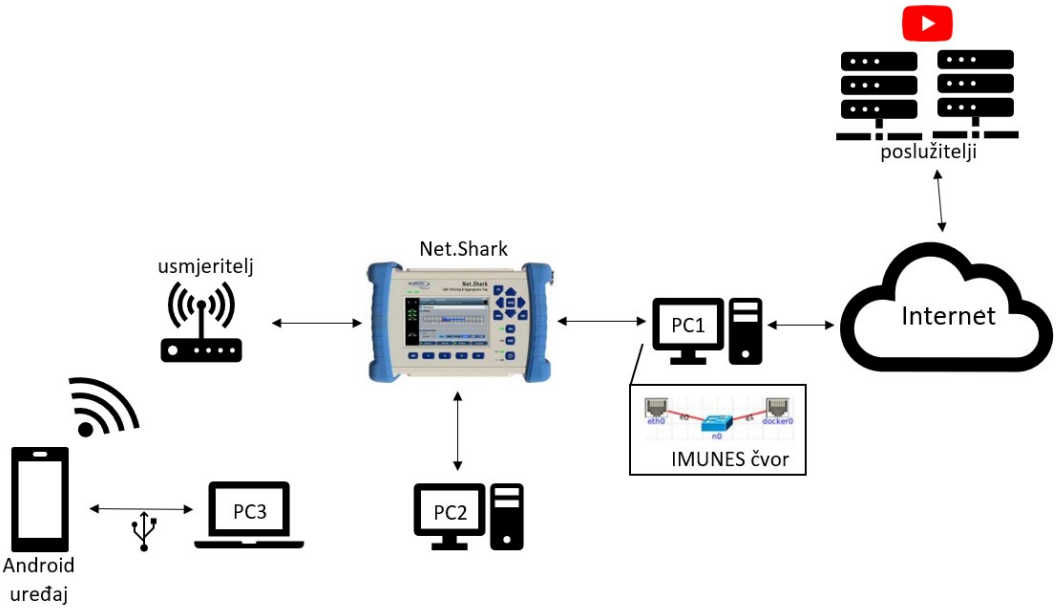
U ovom su poglavlju detaljno opisani svi elementi koji su sastavni dio metodologije istraživanja provedenog u sklopu ovog rada. Istraživanje se može podijeliti u dva dijela. Prvi dio odnosi se na postavljanje laboratorijskog okruženja i prikupljanje podataka, odnosno gledanje video sadržaja na nativnoj YouTube aplikaciji na mobilnom uređaju sa operacijskim sustavom Android, uz praćenje KPI-eva na aplikacijskoj razini i snimanje mrežnog prometa. Drugi dio istraživanja obuhvaća obradu prikupljenih podataka kako bi ti podaci mogli predstavljati ulazni skup podataka za treniranje i validaciju modela baziranih na strojnom učenju (Slika 5.1).



Slika 5.1 Shema postupka obrade podataka

5.1 Laboratorijsko okruženje

Svrha laboratorijskog okruženja je prikupljanje KPI-eva na aplikacijskoj razini i snimanje mrežnog prometa za vrijeme gledanja video sadržaja. Laboratorijsko okruženje u kojem se provodilo istraživanje prikazano na Slici 5.2. Svaka komponenta opisana je detaljnije u nastavku.



Slika 5.2 Shema laboratorijskog okruženja

PC1

PC1 je stolno (engl. *desktop*) računalo sa operacijskim sustavom FreeBSD [17]. PC1 ima ugrađene dvije mrežne kartice. Na prvu mrežnu karticu spojen je usmjeritelj (preko uređaja Net.Shark), dok je druga mrežna kartica spojena na Internet. Na računalu PC1 instaliran je alat IMUNES (Poglavlje 5.2.1.1) koji se koristi za manipulaciju mrežnih parametara.

PC2

Svrha računala PC2 je snimanje mrežnog prometa. PC2 je stolno računalo sa operacijskim sustavom Ubuntu [18]. Alat koji se koristio za snimanje mrežnog prometa je tcpdump (Poglavlje 5.2.1.2).

PC3

PC3 je prijenosno računalo sa operacijskim sustavom Windows 10. Računalo PC3 služi za upravljanje Android mobilnim uređajem i za prikupljanje podataka s aplikacijske razine (Poglavlje 5.2.2).

Net.Shark

Net.Shark [19] je uređaj koji se u ovom radu koristi za repliciranje mrežnog prometa. U laboratorijskom okruženju nalazi se između računala sa IMUNES-om (PC1) i usmjeritelja. Sav mrežni promet koji prolazi kroz njega replicira se i šalje do računala na kojem se snima mrežni promet (PC2).

Android uređaj

Android uređaj koji se koristi u ovom radu je Samsung Galaxy S8 (Tablica 5.1). Uređaj je USB kabelom povezan sa računalom PC3 i bežično je povezan sa usmjeriteljem. Glavna svrha Android uređaja je reproduciranje YouTube videa.

5.2 Prikupljanje podataka

Za vrijeme gledanja video sadržaja paralelno se prikupljaju podaci sa mrežne (mrežni promet) i aplikacijske (video statistika) razine. U nastavku slijedi detaljniji opis procedura prikupljanja podataka za obje razine.

5.2.1 Mrežna razina

Podaci mrežnog prometa se prikupljaju na računalu PC2 (Slika 5.2). Uređaj Net.Shark replicira mrežni promet do računala PC2, a alat tcpdump snima taj promet i pohranjuje ga u datoteku formata pcap (*packet capture*).

U laboratoriju je mrežna propusnost velika, dok u realnim uvjetima uglavnom nije. Da bi izgrađeni modeli mogli detektirati degradacije QoE-a emulirani su realni uvjeti mrežne propusnosti u pokretnim mobilnim mrežama. U procesu prikupljanja podataka emulirala su se tri različita realna scenarija mrežne propusnosti. Za prva dva scenarija koristile su se dvije skripte koje sadrže vrijednosti propusnosti na razini sekunde u pristupnim mrežama 3G [20] i 4G/LTE [21] mrežne propusnosti. Za treći scenarij se koristilo fiksno ograničenje mrežne propusnosti od 1.5 *Mbps* koje simulira uvjete opcije neograničenog strujanja sadržaja koju nude pružatelji mobilnih usluga [35].

5.2.1.1 IMUNES

IMUNES (engl*. Integrated Multiprotocol Network Emulator/Simulator*) je alat čija je glavna svrha emulacija odnosno simulacija raznih IP mrežnih topologija [22]. Temelji se na operacijskom sustavu FreeBSD i operacijskom sustavu Linux. Jedna od značajki alata je da omogućuje postavljanje ograničenja na propusnost (engl. *bandwidth*) mrežnog prometa, te je moguće koristiti vlastite skripte koje će automatski mijenjati ta ograničenja u realnom vremenu.

5.2.1.2 Tcpdump

Tcpdump je besplatan alat bez grafičkog sučelja distribuiran pod BSD licencom. Koristi se za analizu i nadgledanje mrežnog prometa koji se šalje preko mreže na koju je spojeno računalo na kojem je pokrenut alat [23].

5.2.2 Aplikacijska razina

Prikupljanje podataka s aplikacijske razine odvija se na mobilnom uređaju Samsung Galaxy S8 (Tablica 5.1). Proces prikupljanja video statistike i simulacije korisničkih interakcija u potpunosti je automatiziran koristeći radni okvir Appium [25]. Skripta za automatizaciju detaljno je opisana u Poglavlju 5.2.2.3.

Tablica 5.1 Specifikacije uređaja Samsung Galaxy S8 [24]

Naziv	Samsung Galaxy S8
Operacijski sustav	Android 9.0 (Pie)
Veličina zaslona	5.8“
Rezolucija zaslona	1440 x 2960 piksela
Radna memorija	4 GB
Bežična mrežna kartica (WLAN)	Wi-Fi 802.11 a/b/g/n/ac

5.2.2.1 YouTube statistika za štrebere

Statistika za štrebere (engl. *Stats for nerds*) opcija je unutar YouTube aplikacije koja korisniku omogućuje uvid u detaljnije informacije o video sadržaju prilikom strujanja. Na ekranu se prikazuju informacije o trenutnoj rezoluciji, audio i video kodecima, broju sličica po sekundi i jedinstveni identifikator videozapisa (Slika 5.3). Statistika se osvježava svake sekunde te postoji mogućnost da se kopira klikom na link *copy debug info*. *Debug info* je JSON struktura koja sadrži više informacija od onih prikazanih na ekranu (Isječak koda 5.1).

Slika 5.3 YouTube statistika za štrebere

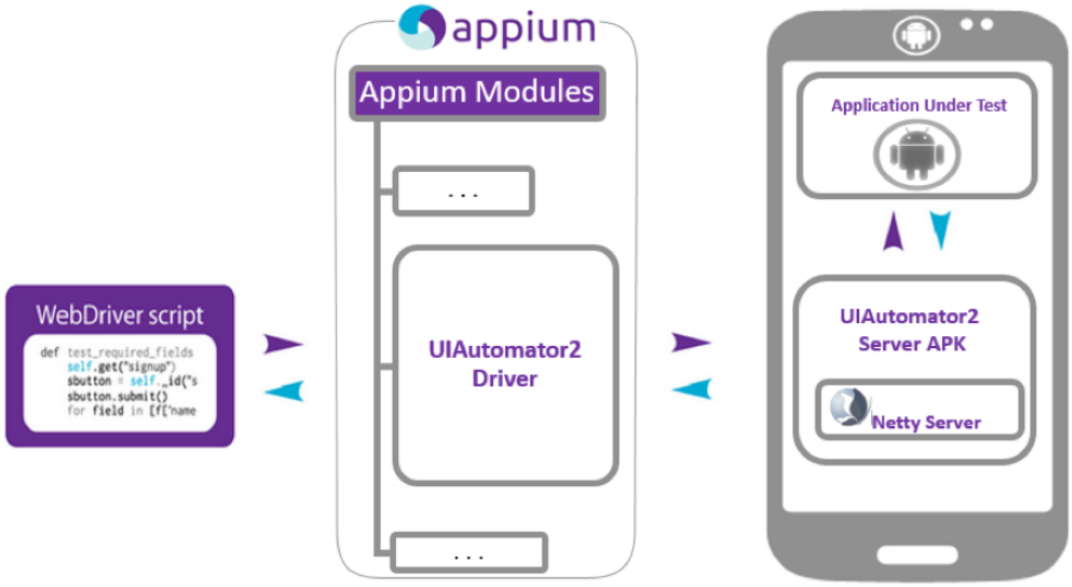
Isječak koda 5.1 YouTube *debug info*


```
{
    "cplatform": "mobile",
    "cbr": "com.google.android.youtube",
    "c": "android",
    "cplayer": "ANDROID_EXOPLAYER_V2",
    "cmodel": "ELE-L29",
    "cos": "Android",
    "csdk": "29",
    "cbrver": "15.17.38",
    "cver": "15.17.38",
    "cosver": "10.HUAWEIELE-L29.10.0.0.190C431",
    "cbrand": "HUAWEI",
    "videoid": "4G6xDyhau_0",
    "cpn": "sjcBP-b2eXE1EsIb",
    "fmt": "244  854x480",
    "afmt": "140 ",
    "bh": 33329,
    "conn": 3,
    "volume": 0,
    "loudness": "-4.306",
    "bat": "0.530:0",
    "df": "0\/626",
    "time": "2020-05-08T13:25:37.339Z",
    "glmode": "RECTANGULAR_2D",
    "mtext": "",
    "error": "No errors",
    "innertube.build.changelist": "310498967",
    "innertube.build.experiments.source_version": "310467747",
    "innertube.build.label": "youtube.ytfe.innertube_20200507_0_RC1",
    "innertube.build.timestamp": "1588914399",
    "innertube.build.variants.checksum": "569bb16175dd569dd785f8d6fcae8004",
    "e": "11211411,11211694,11213630,11216291,11219182,11219238,11219600,112
19646,11219917,11220078,11220349,11220398,11220543,11220650,23744176,2379588
2,23803853,23811270,23837040,23837993,23839552,23839597,23844961,23850473,23
856950,23857949,23859802,23860859,23861273,23865856,23866393,23876458,238778
89,23880389,23880619,23880720,23882034,23882502,23884386,23886825,23888831,2
3889737,23890852,23890887,23890966,23891343,23891856,23892593,23893072,23893
805,23894666,23895672,23896291,23896332,23896332,23896580,23896799,23898054,
23899473,23899886,23900839,23901518,23901875,23903199,24630898,39787779,3978
8273,39789494,39790338,39790376,39790405,39790704,45170004,9449243",
    "logged_in": "1"
}
```


5.2.2.2 Appium

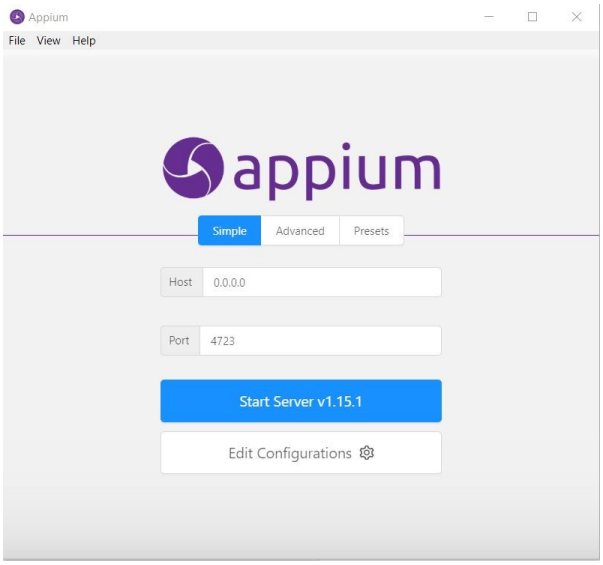
Appium je besplatan alat za automatsko testiranje nativnih, mobilnih web i hibridnih aplikacija na iOS, Android i Windows platformama [25]. Appium je web poslužitelj napisan u node.js-u koji neprekidno sluša naredbe od strane klijenta i izvršava ih na mobilnom uređaju (Slika 5.4). Činjenica da Appium

koristi klijent-server arhitekturu omogućuje testeru da se kod za testove može napisati u bilo kojem programskom jeziku.



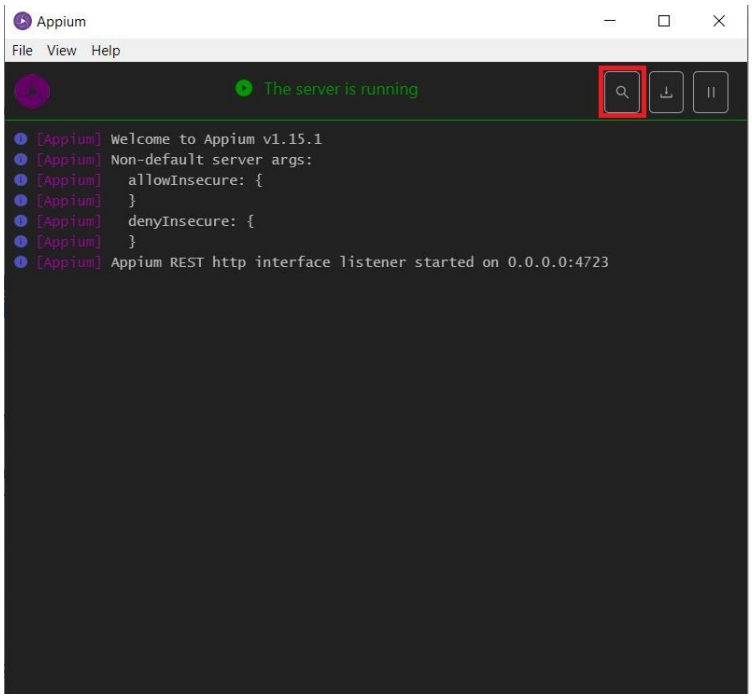
Slika 5.4 Appium arhitektura (slika preuzeta iz [26])

Izgled početnog ekrana grafičkog sučelja Apppium poslužitelja nalazi se na Slici 5.5. Klikom na gumb *start server* pokreće se poslužitelj.

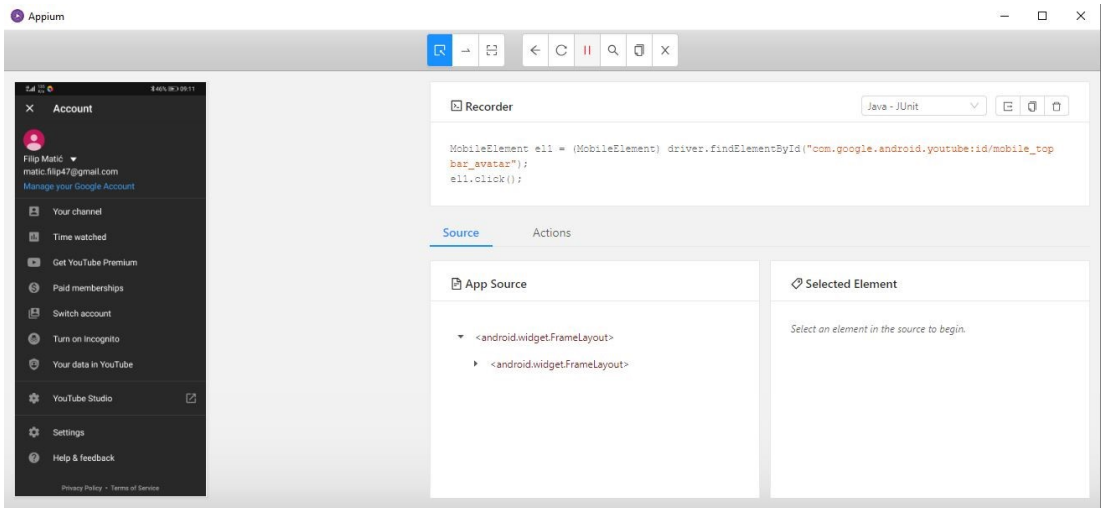


Slika 5.5 Appium: početni ekran

Nakon što je Appium poslužitelj pokrenut, na ekranu na kojem se ispisuje dnevnik (engl. *log*) poslužitelja, klikom na ikonu povećala se može pokrenuti *inspector* sučelje (Slika 5.6). *Inspector* sučelje omogućuje korisniku da sazna ID elemenata kako bi ih mogao programski dohvatiti (Slika 5.7).



Slika 5.6 Apppium: dnevnik poslužitelja



Slika 5.7 Appium: *inspector* sučelje

5.2.2.3 Skripta za prikupljanje video statistike

U sklopu ovog rada napravljena je skripta za automatizirano prikupljanje video statistike i simulaciju korisničkih interakcija. Skripta je napisana u programskom jeziku Java koristeći Appium.

Prvi korak pri izvođenju skripte je dohvaćanje liste videa iz baze video ID-eva i metapodataka (Poglavlje 4) na kojima će se prikupljati podaci. U listu se osim video ID-a pohranjuje i trajanje videa u sekundama. Prilikom dohvaćanja liste uzeti su u obzir samo videi koji zadovoljavaju sljedeće parametre (Isječak koda 5.2):

- broj pregleda je veći od pet milijuna,
- barem tisuću oznaka „sviđa mi se“,
- broj oznaka „ne sviđa mi se“ je manji od tisuću,
- dostupnost u 720p kvaliteti,
- duljina trajanja između 180 i 540 sekundi i
- datum objave u periodu između 2010. i 2019. godine.

Isječak koda 5.2 Dohvaćanje liste videa

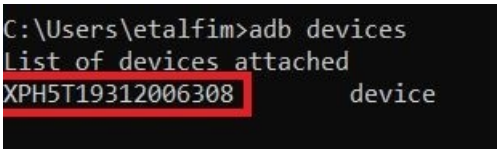
```
SELECT DISTINCT t1.videoID, t2.durationInSeconds
FROM video AS t1 NATURAL JOIN duration_in_seconds AS t2
JOIN formats AS t3 ON t1.videoID = t3.videoID AND t3.format_note='720p'
WHERE DIMENSION='2d' AND definition='hd' AND projection = 'rectangular'
AND uploadStatus = 'processed' AND privacyStatus = 'public'
AND viewCount > ? AND likeCount > ? AND dislikeCount < ?
AND durationInSeconds BETWEEN ? AND  ?
AND publishedAt BETWEEN '2010-10-01' AND '2019-12-31'
```

Nakon što je dohvaćena lista videa, sljedeći korak je uspostava veze između Appium poslužitelja i Android uređaja. Programska implementacija prikazana je u Isječku koda 5.3.

Isječak koda 5.3 Uspostavljanje veze s Android uređajem

```
public static AndroidDriver<AndroidElement> startAppiumSession() throws Exception {
        DesiredCapabilities dc = new DesiredCapabilities();
        dc.setCapability(MobileCapabilityType.DEVICE_NAME, deviceName);
        dc.setCapability(MobileCapabilityType.PLATFORM_NAME, Platform.ANDROID);
        dc.setCapability(MobileCapabilityType.NO_RESET, true);

        URL url = new URL("http://127.0.0.1:4723/wd/hub");
        return new AndroidDriver<AndroidElement>(url, dc);
  }
```

Kod povezivanja Appium poslužitelja s mobilnim uređajem potrebno je predati podatke o uređaju (*deviceName*, *platformName*). Izvršavanjem naredbe *adb devices* u konzoli se ispisuje lista uređaja spojenih USB kabelom s računalom u kojoj se nalazi parametar *deviceName* (Slika 5.8).



Slika 5.8 Primjer rezultata izvršavanja naredbe adb devices

Da bi se izbjegao gubitak informacija od trenutka otvaranja prvog videa iz liste do uključivanja statistike za štrebere (otprilike pet sekundi), najprije se otvara video koji se ne nalazi u listi čija je svrha isključivo uključivanje opcije statistika za štrebere. Otvaranje videa u YouTube aplikaciji izvedeno je putem adb konzole (Isječak koda 5.4).

Isječak koda 5.4 Otvaranje YouTube videa putem adb konzole

```
public static void openVideoWithABDCommand(String videoID) throws IOException {
   String cmd = "adb shell am start -
a android.intent.action.VIEW \"http://www.youtube.com/watch?v=" + videoID + "\"";
   ProcessBuilder processBuilder = new ProcessBuilder();
   processBuilder.command("cmd.exe", "/c", cmd);
   Process process = processBuilder.start();
}
```

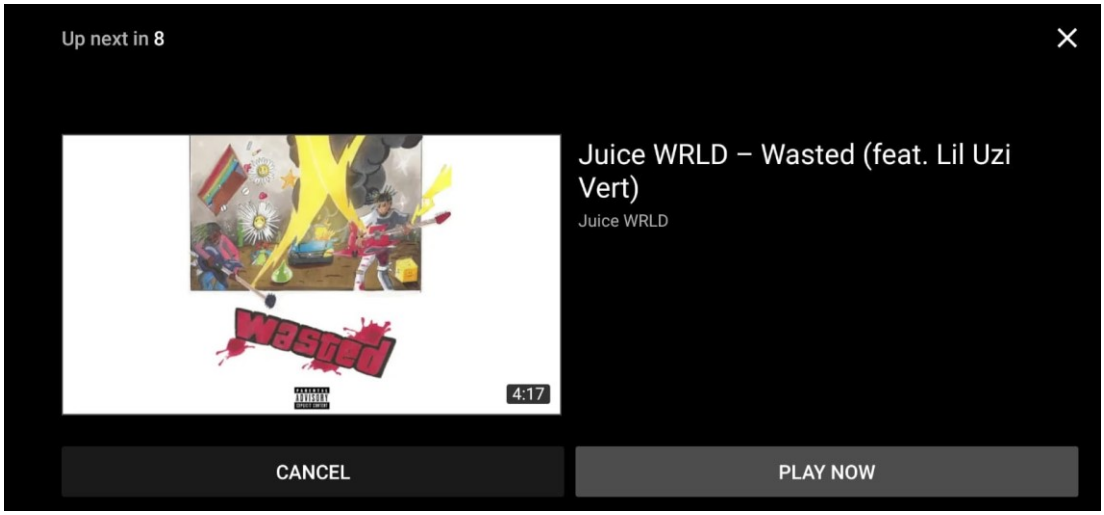
Posljednji korak skripte je iteriranje po listi videa. Za svaki video prvo se nasumičnim (engl. *random*) odabirom izabere koja će se interakcija izvesti i u kojoj sekundi. Zatim se video otvara u YouTube aplikaciji (Isječak koda 5.4) i u svakoj sekundi kopira se *debug info* (Isječak koda 5.5), te se zajedno sa vremenskom oznakom sekunde zapisuje u izlaznu tekstualnu datoteku. U odabranoj sekundi izvodi se interakcija.

Isječak koda 5.5 Kopiranje *debug info*-a

```
public static String clickCopyDebugInfo(AndroidDriver<AndroidElement> driver) {
   driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
   WebElement el=(WebElement) driver.findElementByAccessibilityId("Copy debug info");
   el.click();

   driver.getClipboard(ClipboardContentType.PLAINTEXT);
   return driver.getClipboardText();
}
```

Kada video završi s reprodukcijom, otvara se sljedeći video u listi i ponavlja se zadnji korak skripte sve dok se ne prođe kroz cijelu listu. Za završetak reprodukcije ne vrijedi uvijek $kraj\ videa = vremenska\ oznaka\ početka +$

$trajanje$ ukoliko se za vrijeme reprodukcije dogodilo zastajkivanje. Zbog toga se kao univerzalni indikator kraja videa uzima ekran na Slici 5.9 koji se prikazuje na kraju svakog videa.



Slika 5.9 Indikator kraja videa

5.2.3 Izvođenje interakcija

U ovom radu definirane su tri korisničke interakcije (Poglavlje 3). U nastavku je opisana programska izvedba definiranih interakcija.

Pauza

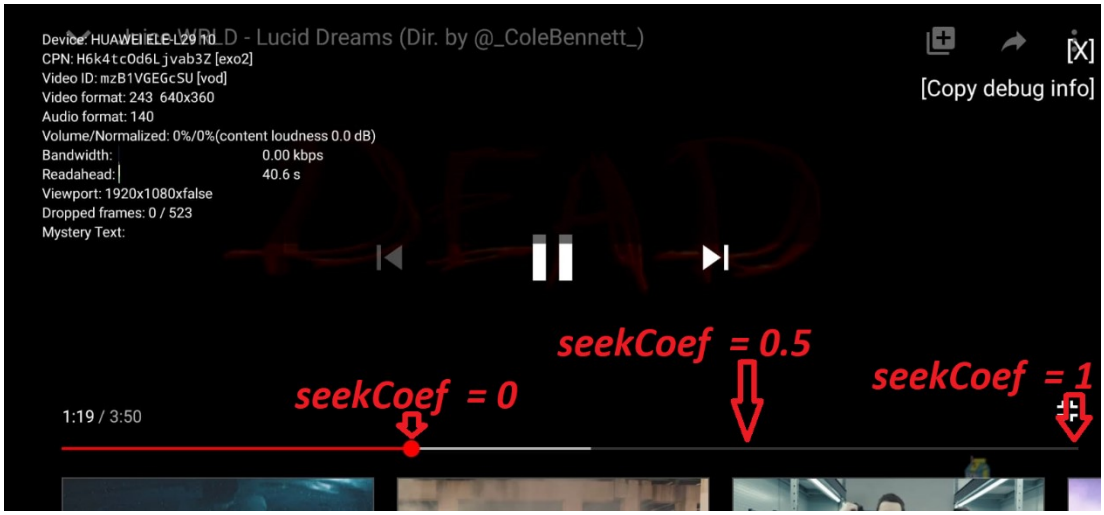
Interakcija pauze izvodi se na način da se programski prvo klikne bilo gdje na ekran da se prikaže gumb za pauzu, a zatim se klikne na gumb (Slika 3.1). Trajanje pauze u sekundama definira se kao nasumični broj između 10 i 30. Za vrijeme dok je video pauziran, *debug info* se i dalje kopira. Nakon što prođe definirano vrijeme pauze klikne se na gumb za nastavak izvođenja videa (Isječak koda 5.6).

Isječak koda 5.6 Programska izvedba interakcije pauze

```
public static void pauseVideoInteraction(AndroidDriver<AndroidElement> driver) thr
ows InterruptedException {
   copyDebugInfo(driver);

   System.out.println("PAUSE VIDEO");
   // tap on the video
   WebElement el1=(WebElement)driver.findElementByAccessibilityId("Video player");
   el1.click();
   copyDebugInfo(driver);
   // tap to pause video
   WebElement el2=(WebElement)driver.findElementByAccessibilityId("Pause video");
   el2.click();

   long startTime = System.currentTimeMillis();
   interactionTimestamp = startTime;

   // copy debug info while video is paused
   while (true) {
      long currentTime = System.currentTimeMillis();
      int currentSecond = (int) ((currentTime-startTime)/1000);

      System.out.println("Pause second: " + currentSecond);
      copyDebugInfo(driver);

      if (currentSecond >= pauseDuration) {
            break;
      }
      Thread.sleep(500);
   }
   // tap to resume video
   el2.click();
}
```

Premotavanje unaprijed

Za interakciju premotavanja unaprijed se definira koeficijent *seekCoef* koji označava za koliko će se video premotati unaprijed. Ako je vrijednost koeficijenta jednaka nula, video se neće premotati, a ako je vrijednost koeficijenta jednaka jedan, video će se premotati do kraja videa (Slika 5.10). Za potrebe ovog rada vrijednost koeficijenta je ograničena na broj između 0.4 i 0.7.



Slika 5.10 Koeficijent premotavanja unaprijed

Interakcija premotavanja unaprijed programski se izvodi na način da se prvo klikne bilo gdje na ekran da se prikaže vremenska crta (engl. *timeline*) videa, a zatim se klikne na koordinatu ekrana koja odgovara trenutku do kojega se video premotava (Isječak koda 5.7).

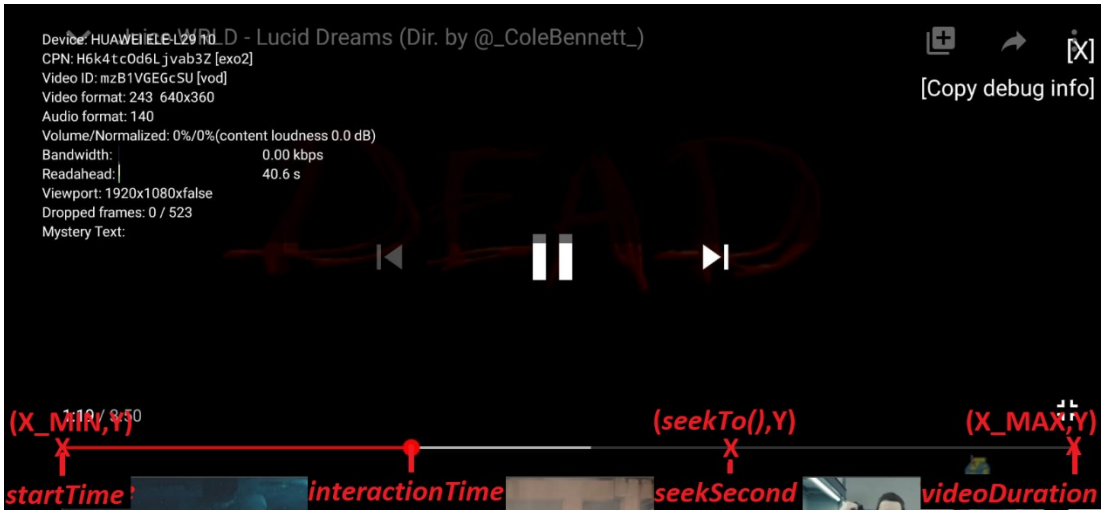
Isječak koda 5.7 Programska izvedba interakcije premotavanja unaprijed

```
public static void seekForwardInteraction(AndroidDriver<AndroidElement> driver, int
videoDuration, int interactionTime) {
   copyDebugInfo(driver);

   System.out.println("SEEK FORWARD");
   // tap on the video
   WebElement el1 = (WebElement) driver.findElementByAccessibilityId("Video player");
   el1.click();

   copyDebugInfo(driver);

   PointOption po2 = new PointOption();
   po2.withCoordinates(seekTo(videoDuration, interactionTime), Y);
   (new TouchAction<>(driver)).tap(po2).perform();
   interactionTimestamp = System.currentTimeMillis();
}
```

Koordinata ekrana računa se prema formuli na Isječku koda 5.8, a vizualna reprezentacija varijabli iz formule nalazi se na Slici 5.11.

Isječak koda 5.8 Izračun koordinate za premotavanje videa unaprijed

```
public static int seekTo(int videoDuration, int interactionTime) {
   double seekSecond = interactionTime + ((videoDuration-interactionTime)*seekCoef);
   return (int) (seekSecond/videoDuration * (X_MAX - X_MIN) + X_MIN);
}
```



Slika 5.11 Vizualna reprezentacija varijabli formule za premotavanje unaprijed

Prebacivanje videa

Interakcija prebacivanja videa izvodi se na način da se prvo klikne bilo gdje na ekran da se prikaže gumb za pokretanje sljedećeg videa, a zatim se klikne na njega (Slika 3.3). Isječak koda 5.9 prikazuje programsku izvedbu interakcije.

Isječak koda 5.9 Programska izvedba interakcije prebacivanja videa

```
public static void skipVideoInteraction(AndroidDriver<AndroidElement> driver) {
   copyDebugInfo(driver);

   System.out.println("SKIP VIDEO");
   // tap on the video
   WebElement el1 = (WebElement) driver.findElementByAccessibilityId("Video player");
   el1.click();

   copyDebugInfo(driver);

   // tap to play next video
   WebElement el2 = (WebElement) driver.findElementByAccessibilityId("Next video");
   el2.click();
   interactionTimestamp = System.currentTimeMillis();
}
```

5.3 Obrada prikupljenih podataka

Potrebno je prvo pojedinačno obraditi podatke s mrežne i s aplikacijske razine te ih nakon toga povezati za svaki video. Podaci s mrežne razine obrađeni su pomoću skripte [27]. Ulazni podatak za skriptu je datoteka formata pcap koja sadrži snimljeni mrežni promet. Izlazna datoteka koju generira skripta je formata csv (*comma separated values*) u kojoj jedan redak predstavlja jednu sekundu video sadržaja. Svaki redak sadrži vremensku oznaku (engl. *timestamp*) i 228 značajki (engl. *features*) koje su izračunate iz mrežnog prometa. Značajke su dobivene računanjem statističkih parametara iz odlaznog (engl. *uplink*) i dolaznog (engl. *downlink*) mrežnog prometa na prozorima od 1, 2, 3, 5, 10 i 20 sekundi (sve do intervala od jedne sekunde za koji se radi predviđanje). Opis statističkih parametara se nalazi u Tablici 5.2.

Tablica 5.2 Mrežne značajke [27]

Naziv	Opis
pckt_count	Broj paketa
pckt_count_gt100	Broj paketa većih od 100 bajtova
throughput	Propusnost
active_time	Postotak prozora koji se koristi za prijenos

mean_pckt_size_gt100	Srednja vrijednost veličine paketa većih od 100 bajtova
median_pckt_size_gt100	Medijan veličine paketa većih od 100 bajtova
max_pckt_size_gt100	Najveća veličina paketa većih od 100 bajtova
min_pckt_size_gt100	Najmanja veličina paketa većih od 100 bajtova
stdev_pckt_size_gt100	Standardna devijacija veličine paketa većih od 100 bajtova
mean_pckt_size	Srednja vrijednost veličine paketa
median_pckt_size	Medijan veličine paketa
max_pckt_size	Najveća veličina paketa
min_pckt_size	Najmanja veličina paketa
stdev_pckt_size	Standardna devijacija veličine paketa
mean_iat	Srednja vrijednost vremena između uzastopnih dolazaka paketa (engl. *interarrival time*)
median_iat	Medijan vremena između uzastopnih dolazaka paketa
max_iat	Najveća vrijednost vremena između uzastopnih dolazaka paketa
min_iat	Najmanja vrijednost vremena između uzastopnih dolazaka paketa
stdev_iat	Standardna devijacija vremena između uzastopnih dolazaka paketa

Za potrebe ovog rada napravljena je skripta u programskom jeziku Java za obradu podataka s aplikacijske razine. Skripta iterira po tekstualnim datotekama koje je generirala skripta u Poglavlju 5.2.2.3, te se za svaku sekundu videa određuju klase za:

- interakciju (engl. *interaction*):
    - *yes*
    - *no*
- vrstu interakcije (engl. *interactionType*):
    - *NO_INTERACTION*
    - *SKIP_VIDEO*
    - *SEEK_FORWARD*
    - *PAUSE_VIDEO*

Osim navedenih parametara se za svaku sekundu zapisuje i jedinstveni identifikator videa i vremenska oznaka sekunde. Skripta generira izlaznu datoteku formata csv.

Nakon što su obrađeni podaci s mrežne i aplikacijske razine, dvije csv datoteke se spajaju u jednu po vremenskoj oznaci (engl. *timestamp*) u jednu. Konačno, csv datoteku je potrebno prebaciti u .arff format (*Attribute-Relation File Format*) zato što program Weka, koji se u ovom radu koristi za izgradnju modela strojnog učenja, zahtjeva .arff format ulazne datoteke.

6. Izgradnja modela strojnog učenja i analiza rezultata

Ovo poglavlje se sastoji od dva dijela. U prvom dijelu su definirani pojmovi vezani uz strojno učenje i vrednovanje modela strojnog učenja. Drugi se dio sastoji od detaljnog opisa procesa izgradnje modela strojnog učenja i analize dobivenih rezuiltata.

6.1 Strojno učenje

Strojno učenje grana je umjetne inteligencije koja se bavi programiranjem računala na način da se optimizira neki kriterij uspješnosti na temelju podatkovnih primjera ili prethodnog iskustva. Situacije u kojima uporaba strojnog učenja ima svrhe su problemi koji su presloženi da bi se mogli riješiti algoritamski (npr. raspoznavanje govora), sustavi koji se dinamički mijenjaju (npr. prilagodba korisničkih sučelja) te kada raspolažemo ogromnim količinama podataka iz kojih trebamo izvući neko znanje [28].

Zbog različitih pristupa obradi podataka razlikujemo tri vrste strojnog učenja: nadzirano učenje (engl. *supervised learning*), nenadzirano učenje (engl. *unsupervised learning*) i podržano učenje (engl. *reinforcement learning*). Kod nadziranog učenja raspolažemo podacima oblika (ulaz, izlaz) iz kojih algoritmi kreiraju funkciju kojom će se neki novi podaci moći klasificirati. Ulazni podatak najčešće je vektor značajki (engl. *features*) dok je izlazni podatak jedna vrijednost (engl. *label*). S druge strane kod nenadziranog učenja raspolažemo podacima bez ciljne vrijednosti. U tom se slučaju strojno učenje koristi za pronalazak pravilnosti među podacima. Podržano učenje svodi se na učenje optimalne strategije na temelju pokušaja s odgođenom nagradom.

6.1.1 Vrednovanje modela

Matrica zabune (engl. *confusion matrix*) matrica je koja prikazuje odnos stvarnih oznaka i predikcija modela (Tablica 6.1). Vrijednosti koje se nalaze u matrici su:

- Stvarno pozitivan (engl. *true positive,* TP) - instanca ispravno klasificirana pozitivno,
- Stvarno negativan (engl. *true negative,* TN) – instanca ispravno klasificirana negativno,
- Lažno pozitivan (engl. *false positive,* FP) – instanca pogrešno klasificirana pozitivno,
- Lažno negativan (engl. *false negative,* FN) – instanca pogrešno klasificirana negativno.

Tablica 6.1 Matrica zabune

		Stvarno	
		+	-
Model	+	TP	FP
	-	FN	TN

Mjere koje se najčešće koriste za vrednovanje klasifikacijskih modela, a mogu se izračunati iz matrice zabune, su:

- Točnost (engl. *Accuracy*, Acc) je definirana kao udio točno klasificiranih primjera.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

- Preciznost (engl. *Precision,* P) je omjer točnih pozitivno klasificiranih primjera i svih pozitivno klasificiranih primjera.

$$P = \frac{TP}{TP + FP}$$

- Odziv (engl. *Recall,* R) se definira kao omjer točnih pozitivno klasificiranih primjera i svih pozitivnih primjera [29].

$$R = \frac{TP}{TP + FN}$$

6.1.2 Algoritmi strojnog učenja

U ovom potpoglavlju opisani su klasifikacijski algoritmi koji su korišteni u ovom radu. U ovom radu korištena su tri algoritma za klasifikaciju: OneR, J48 i RandomForest. Svi navedeni algoritmi implementirani su u programu Weka.

6.1.2.1 OneR

OneR, skraćeno od *One Rule*, jednostavan je klasifikacijski algoritam. Algoritam stvara jedno pravilo koje klasificira objekt na temelju jednog atributa [31]. Iako vrlo jednostavan algoritam, u praksi se pokazao samo neznatno lošijim od drugih, mnogo kompliciranijih algoritama. Glavna ideja algoritma nalazi se na Isječku koda 6.1.

Isječak koda 6.1 OneR algoritam [32]

```
for each attribute a do
    for each value v from the domain of a do
        Select the set of instances where a has value v.;
        Let c be the most frequent class in that set.;
        Add clause to rule for a: if a has value v then the class is c
    end
    Calculate the classification accuracy of this rule.
end
Use the rule with the highest classification accuracy.
```

6.1.2.2 J48

J48 algoritam treniranjem na ulaznom skupu podataka kreira stablo odlučivanja prema kojem se rade predviđanja za nove podatke. Izgradnja stabla odlučivanja odvija se na način da se jedan atribut postavi za korijenski čvor koji ima onoliko grana koliko taj atribut ima različitih vrijednosti, a zatim se na svaku granu za čvor postavlja drugi atribut koji se grana na isti način kao i prethodni čvor. Takav se postupak ponavlja sve dok se ne iskoriste svi atributi [30].

6.1.2.3 RandomForest

RandomForest algoritam na ulaznom skupu podataka kreira odabrani broj različitih stabala odlučivanja. Algoritam funkcionira na način da se iz skupa ulaznog podataka nasumično (engl. *random*) odabere podskup od svih atributa prema kojem će se izgraditi jedno stablo odlučivanja. Taj se postupak ponavlja onoliko puta koliko je zadano (najčešće oko 100). Predviđanje za novi skup podataka odvija se tako da se odabere ona vrijednost koja je dobivena na najvećem broju stabala [33].

6.1.3 Weka

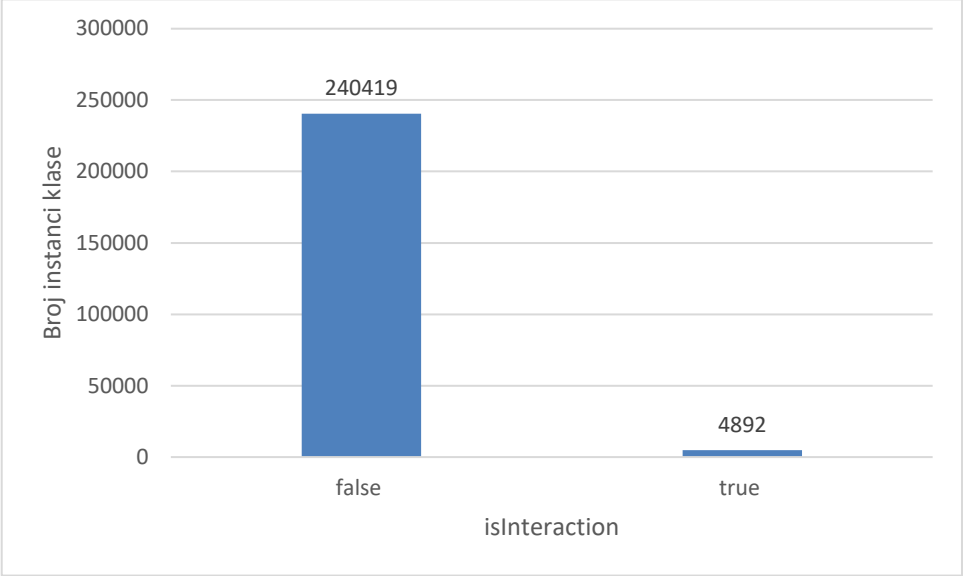
Weka (*Waikato Environment for Knowledge Analysis*) je program otvorenog koda pod GNU općom javnom licencom [30]. Napravljen je na Sveučilištu u Waikatu na Novom Zelandu i program je napisan u potpunosti u Javi što znači da je dostupan na svim platformama. Weka je zapravo kolekcija algoritama strojnog učenja koji se koriste za dubinsku analizu podataka. Također sadrži alate za obradu, klasifikaciju, regresiju i vizualizaciju podataka.

6.2 Analiza rezultata

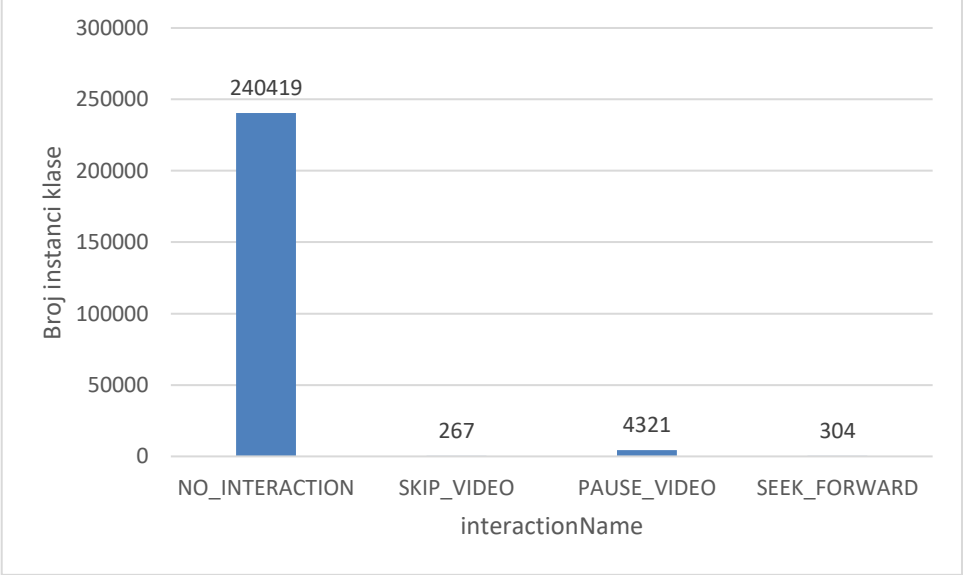
Cilj ovog rada je detekcija korisničkih interakcija na temelju analize mrežnog prometa za vrijeme strujanja video sadržaja na platformi YouTube. Za analizu podataka mrežne razine koristile su se metode strojnog učenja.

Ciljne oznake koje će se predviđati, na razini sekunde videa, u ovom radu su:

- *IsInteraction* (Slika 6.1): dvije klase („*true*“ i „*false*“) koje označavaju je li se interakcija dogodila ili nije.
- *InteractionName* (Slika 6.2): četiri klase („*NO_INTERACTION*“, „*SKIP_VIDEO*“, „*PAUSE_VIDEO*“, „*SEEK_FORWARD*“) koje označavaju koji tip interakcije se dogodio.



Slika 6.1 Distribucija uzoraka po klasama za ciljnu oznaku *isInteraction*



Slika 6.2 Distribucija uzoraka po klasama za ciljnu oznaku *interactionName*

Na slikama 6.1 i 6.2 može se primijetiti da je skup podataka za promatrane ciljne oznake nebalansiran. Modeli strojnog učenja izgrađeni na ovim podacima, koristeći algoritam RandomForest i unakrsnu validaciju za učenje i testiranje modela, imaju točnost od 98%, ali se iz matrica zabune (Tablica 6.2 i Tablica 6.3) vidi da se manjinske klase uglavnom klasificiraju kao većinske.

Tablica 6.2 Matrica zabune nebalansiranog skupa podataka za ciljnu oznaku *isInteraction*

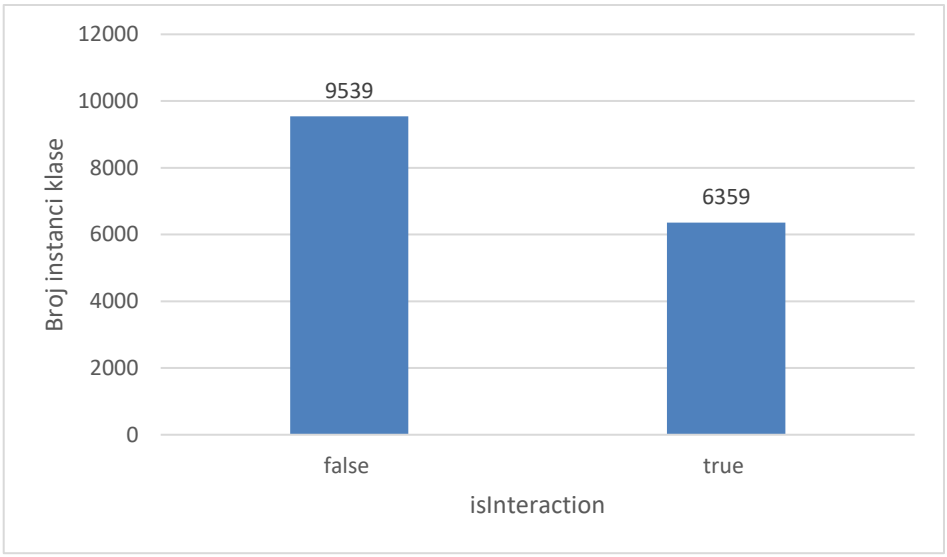
Klasificirano kao		**Prava klasa**
false	***true***	
240347	72	***false***
4590	302	***true***

Tablica 6.3 Matrica zabune nebalansiranog skupa podataka za ciljnu oznaku *interactionName*

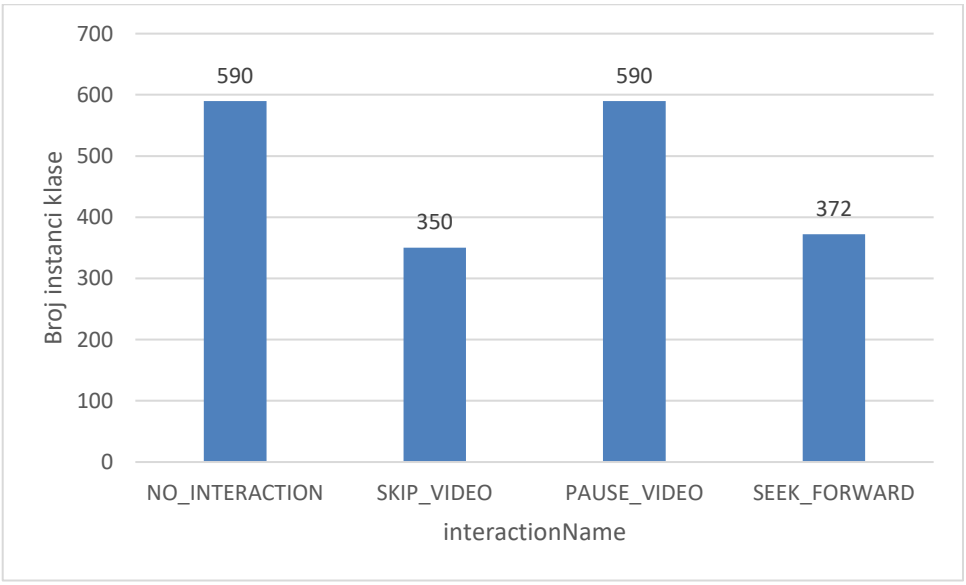
Klasificirano kao				**Prava klasa**
NO_INTERACTION	***SKIP_VIDEO***	***PAUSE_VIDEO***	***SEEK_FORWARD***	
240295	115	9	0	***NO_INTERACTION***
267	0	0	0	***SKIP_VIDEO***
3722	0	599	0	***PAUSE_VIDEO***
304	0	0	0	***SEEK_FORWARD***

Prethodno navedeni problem rješava se balansiranjem skupa podataka. U ovom se radu za balansiranje skupa podataka koriste postupci naduzorkovanja i poduzorkovanja. Naduzorkovanjem se povećava broj instanci manjinske klase na način da se originalni uzorci ponavljaju, dok se

poduzorkovanjem smanjuje broj instanci većinske klase. Balansiranje skupa izvedeno je u programu Weka pomoću filtera *SpreadSubsample* [36] i *Resample* [37]. Slika 6.3 i Slika 6.4 prikazuju raspodjelu klasa za promatrane ciljne oznake nakon što je skup balansiran.



Slika 6.3 Distribucija klasa za ciljnu oznaku *isInteraction* nakon balansiranja skupa



Slika 6.4 Distribucija klasa za ciljnu oznaku *interactionName* nakon balansiranja skupa

6.2.1 Odabir značajki

Prikupljeni skup podataka sadrži 228 različitih značajki. Nisu sve značajke jednako bitne za klasifikaciju. Neke redundantne ili nebitne značajke samo mogu otežati klasifikaciju. Uklanjanjem nebitnih značajki se smanjuje složenost modela, ubrzava se proces učenja i poboljšavaju se performanse klasifikacije. Za odabir značajki koristila se *BestFirst* [39] metoda pretraživanja i *WrapperSubsetEval* [40] kao evaluator značajki na cijelom skupu podataka. *BestFirst* metoda pretražuje prostor podskupova značajki pomoću pohlepne metode penjanja uzbrdo (engl. *greedy hillclimbing*) sa mogućnošću vraćanja (engl. *backtracking*). *WrapperSubsetEval* koristi klasifikator za evaluaciju podskupova značajki, te dodatno za procjenu točnosti koristi unakrsnu validaciju za svaki podskup. U ovom je radu odabir značajki proveden za tri algoritma: OneR, J48 i RandomForest. Za svaki algoritam odabran je drugačiji skup značajki. Odabrane značajke po algoritmima nalaze se u tablicama 6.4 i 6.5.

Tablica 6.4 Odabrane značajke za ciljnu oznaku *isInteraction*

Algoritam	**Odabrane značajke**
OneR	*dl_max_iat_w20*
J48	*dl_stdev_iat_w1, ul_stdev_pckt_size_gt100_w1, ul_max_iat_w2, ul_stdev_iat_w2, dl_min_iat_w3, dl_min_pckt_size_w5, ul_max_pckt_size_w5, ul_stdev_pckt_size_w5, ul_mean_pckt_size_w10, ul_max_pckt_size_w10, dl_median_iat_w20, ul_pckt_count_w20*
RandomForest	*ul_stdev_pckt_size_gt100_w1, ul_stdev_pckt_size_gt100_w2, dl_min_pckt_size_w5, dl_pckt_count_w10, dl_stdev_pckt_size_gt100_w10, dl_median_pckt_size_w10, dl_min_pckt_size_w10,*

	dl_min_iat_w10, ul_min_pckt_size_w10, dl_active_time_w20, dl_max_pckt_size_w20, dl_min_pckt_size_w20, dl_stdev_pckt_size_w20, dl_mean_iat_w20, dl_median_iat_w20, dl_max_iat_w20, dl_min_iat_w20, dl_stdev_iat_w20, ul_pckt_count_w20, ul_mean_pckt_size_w20, ul_max_pckt_size_w20, ul_median_iat_w20, ul_max_iat_w20, ul_min_iat_w20

Tablica 6.5 Odabrane značajke za ciljnu oznaku *interactionName*

Algoritam	**Odabrane značajke**
OneR	*ul_stdev_pckt_size_w2*
J48	*dl_median_pckt_size_gt100_w1, dl_max_pckt_size_gt100_w1, dl_min_pckt_size_gt100_w1, dl_pckt_count_w2, dl_stdev_pckt_size_w2, dl_mean_iat_w2, ul_pckt_count_w2, ul_max_pckt_size_gt100_w2, ul_stdev_pckt_size_w2, dl_throughput_w3, dl_mean_pckt_size_gt100_w3, dl_median_pckt_size_gt100_w3, dl_min_pckt_size_w3, dl_stdev_pckt_size_w3, dl_stdev_iat_w3, ul_min_pckt_size_w3, dl_min_iat_w5, ul_min_iat_w5, dl_active_time_w10, dl_stdev_pckt_size_w10, dl_stdev_pckt_size_w20, dl_median_iat_w20, dl_min_iat_w20, dl_stdev_iat_w20*

RandomForest	*dl_median_pckt_size_gt100_w1, dl_min_pckt_size_gt100_w1, dl_stdev_iat_w1, dl_pckt_count_w2, ul_min_iat_w2, dl_median_pckt_size_w3, ul_max_pckt_size_w3, ul_min_pckt_size_w3, ul_active_time_w5, ul_stdev_iat_w5, dl_stdev_pckt_size_gt100_w20, dl_max_pckt_size_w20, dl_max_iat_w20, ul_max_pckt_size_w20, ul_max_iat_w20, ul_min_iat_w20*

6.2.2 Izgradnja modela i rezultati

Za izgradnju modela strojnog učenja osim algoritma potrebno je odabrati način treniranja i testiranja modela iz podataka. Neki od najčešće korištenih metoda treniranja i testiranja modela strojnog učenja su unakrsna validacija (engl. *cross validation*) i podjela skupa podataka (engl. *dataset split*). U ovom će se radu koristiti unakrsna validacija. Unakrsna validacija dijeli skup podataka za učenje na *k* odvojenih skupova (engl. *folds*). Zatim u *k* iteracija jedan skup podataka koristi se za validaciju modela, dok se preostalih *k-1* skupova koristi za učenje. Vrijednost *k* u ovom radu iznosi 10. Za ciljnu se oznaku *isInteraction* skup podataka prikazan na Slici 6.3 koristio se kao ulazni skup i korištene su samo odabrane značajke koje se nalaze u Tablici 6.4. Za ciljnu oznaku *interactionName* kao ulazni skup podataka se koristio skup prikazan na Slici 6.4 i korištene su odabrane značajke koje se nalaze u Tablici 6.5. U nastavku se nalazi prikaz rezultata po algoritmima.

Performanse modela algoritma OneR po klasama nalaze se u tablicama 6.6 i 6.7. Točnost za ciljnu oznaku *isInteraction* iznosi 87.99%, dok je za ciljnu oznaku *interactionName* 51.16%.

Tablica 6.6 Detaljna točnost za ciljnu oznaku *isInteraction* za algoritam OneR

	TP	FP	Preciznost	Odziv	Klasa
	0.916	0.175	0.887	0.916	***false***
	0.825	0.084	0.868	0.825	***true***
Težinski prosjek	0.880	0.138	0.880	0.880	

Tablica 6.7 Detaljna točnost za ciljnu oznaku *interactionName* za algoritam OneR

	TP	FP	Preciznost	Odziv	Klasa
	0.449	0.224	0.474	0.449	***NO_INTERACTION***
	0.434	0.086	0.533	0.434	***SKIP_VIDEO***
	0.605	0.248	0.523	0.605	***PAUSE_VIDEO***
	0.535	0.115	0.531	0.535	***SEEK_FORWARD***
Težinski prosjek	0.512	0.185	0.511	0.512	

Performanse modela algoritma J48 po klasama nalaze se u tablicama 6.8 i 6.9. Točnost za ciljnu oznaku *isInteraction* iznosi 85.5%, dok je za ciljnu oznaku *interactionName* 77.18%.

Tablica 6.8 Detaljna točnost za ciljnu oznaku *isInteraction* za algoritam J48

	TP	FP	Preciznost	Odziv	Klasa
	0.868	0.164	0.888	0.868	***false***
	0.836	0.132	0.808	0.836	***true***
Težinski prosjek	0.855	0.152	0.856	0.855	

Tablica 6.9 Detaljna točnost za ciljnu oznaku *interactionName* za algoritam J48

	TP	FP	Preciznost	Odziv	Klasa
	0.708	0.115	0.735	0.708	***NO_INTERACTION***
	0.769	0.052	0.769	0.769	***SKIP_VIDEO***
	0.798	0.111	0.763	0.798	***PAUSE_VIDEO***
	0.833	0.037	0.847	0.833	***SEEK_FORWARD***
Težinski prosjek	0.772	0.087	0.772	0.772	

Performanse modela algoritma RandomForest po klasama nalaze se u tablicama 6.10 i 6.11. Točnost za ciljnu oznaku *isInteraction* iznosi 96.68%, dok je za ciljnu oznaku *interactionName* 85.49%.

Tablica 6.10 Detaljna točnost za ciljnu oznaku *isInteraction* za algoritam RandomForest

	TP	FP	Preciznost	Odziv	Klasa
	0.978	0.050	0.967	0.978	***false***
	0.950	0.022	0.966	0.950	***true***
Težinski prosjek	0.967	0.039	0.967	0.967	

Tablica 6.11 Detaljna točnost za ciljnu oznaku *interactionName* za algoritam RandomForest

	TP	FP	Preciznost	Odziv	Klasa
	0.827	0.085	0.815	0.827	***NO_INTERACTION***
	0.843	0.020	0.905	0.843	***SKIP_VIDEO***
	0.864	0.062	0.863	0.864	***PAUSE_VIDEO***
	0.895	0.035	0.863	0.895	***SEEK_FORWARD***
Težinski prosjek	0.855	0.056	0.856	0.855	

6.2.3 Usporedba rezultata

Tablica 6.12 sadrži točnosti klasifikacijskih modela za ciljnu oznaku *isInteraction* po algoritmima za dva slučaja: kada su se za učenje koristile sve značajke i samo odabrane značajke (Tablica 6.4). U oba slučaja se za treniranje i testiranje modela koristila unakrsna validacija. Sa svim značajkama najveća se točnost postigla sa algoritmom RandomForest (94.05%), dok je najmanja točnost bila za algoritam OneR (87.99%). Sa odabranim značajkama

točnost za RandomForest algoritam povećala se za 2.63 postotna boda i iznosi 96.68%. Za OneR algoritam točnost je ostala ista, dok se za J48 smanjila za 4.37 postotnih bodova. Svi korišteni algoritmi imaju vrlo visoku točnost. Zanimljivo je za primjetiti da OneR algoritam, koji je najjednostavniji i za predviđanje koristi samo jednu značajku, također ima vrlo visoku točnost od približno 88%. Konačno, može se zaključiti da se za promatranu sekundu videa, na temelju značajki mrežnog prometa, može uspješno predvidjeti je li se interakcija dogodila ili nije s vrlo visokom točnošću od 96.68%.

Tablica 6.12 Točnosti za ciljnu oznaku *isInteraction*

Algoritam	Točnost (sve značajke)	Točnost (odabrane značajke)
OneR	87.99%	87.99%
J48	89.87%	85.5%
RandomForest	94.05%	96.68%

Tablica 6.13 sadrži točnosti klasifikacijskih modela za ciljnu oznaku *interactionName* po algoritmima za dva slučaja: kada su se za učenje koristile sve značajke i samo odabrane značajke (Tablica 6.5). U oba slučaja se za treniranje i testiranje koristila unakrsna validacija. Sa svim značajkama najveću točnost ima algoritam RandomForest (84.28%), a OneR ima najmanju točnost (51.16%). Sa odabranim značajkama točnost za RandomForest algoritam povećala se za 1.21 postotni bod i iznosi 85.49%. Za OneR algoritam točnost je ostala ista, dok se za J48 smanjila za 0.69 postotnih bodova. Konačno, može se zaključiti da se za promatranu sekundu videa, na temelju mrežnih značajki, sa prihvatljivom točnošću klasifikacije od 85.49% može odrediti koji se tip interakcije dogodio.

Tablica 6.13 Točnosti za ciljnu oznaku *interactionName*

Algoritam	Točnost (sve značajke)	Točnost (odabrane značajke)
OneR	51.16%	51.16%
J48	77.87%	77.18%
RandomForest	84.28%	85.49%

Zaključak

Budući da je većina Internet prometa kriptirana to predstavlja problem davateljima mrežnih usluga pri praćenju iskustvene kvalitete, a time i pri pružanju odgovarajuće razine iskustvene kvalitete usluge krajnjim korisnicima. Uz navedeno, procjena iskustvene kvalitete dodatno je otežana prisustvom korisničkih interakcija za vrijeme prikazivanja video sadržaja zbog utjecaja interakcija na mrežni promet. U ovom radu korištena su rješenja za detekciju korisničkih interakcija koja se temelje na metodama strojnog učenja.

Za potrebe prikupljanja podataka napravljena je skripta za automatizirano prikupljanje video statistike i simulaciju korisničkih interakcija. Prikupljeni su podaci na 1100 različitih videa. Od svih videa,210 sadrži interakcije s pauzom, 303sadrži premotavanje unaprijed, 267 sadrži prebacivanje videa 320 videa ne sadrži interakcije. Za vrijeme reprodukcije video sadržaja paralelno su se prikupljali podaci s mrežne (mrežni promet) i aplikacijske (video statistika) razine.

Nakon što su podaci prikupljeni, za svaki video su se pojedinačno obradili podaci s mrežne i aplikacijske razine. Zatim su se podaci spojili u jednu datoteku koja je činila ulazni skup podataka za izgradnju modela strojnog učenja koji služi za detekciju interakcija.

Cilj ovog rada bio je na temelju poznatih mrežnih značajki detektirati korisničke interakcije na razini sekunde. U radu su se provodile dvije vrste detekcija korisničkih interakcija. Prva detekcija sastojala se u ispitivanju je li se interakcija dogodila ili nije u promatranoj sekundi. Druga detekcija sastojala se u ispitivanju koji tip interakcije se dogodio (bez interakcije, pauza, premotavanje videa unaprijed i prebacivanje videa) u promatranoj sekundi. Temeljem dobivenih rezultata može se izvući zaključak da se korisničke interakcije, za vrijeme gledanja video sadržaja, mogu detektirati u stvarnom vremenu s točnošću od 96% dok se tip interakcije može odrediti s točnošću od 85%.

Literatura

[1] Internet usage statistics: world Internet usage and population statistics. Internet World Stats. https://www.internetworldstats.com/stats.htm, 20.06.2020.

[2] Cisco Annual Internet Report, https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html, 20.06.2020.

[3] Sandvine: 2020 Mobile Internet Phenomena Report, https://www.sandvine.com/download-report-mobile-internet-phenomena-report-2020-sandvine, 20.06.2020.

[4] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming.", IEEE Communications Surveys & Tutorials, 2015, 17.1: 469-492.

[5] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP-design Principles and Standards. ", InProceedings of the second annual ACM conference on Multimedia systems 2011 Feb 23 (Vol. 2014, pp. 2-4). New York, USA: ACM.

[6] D. K. Krishnappa, D. Bhat, M. Zink, "DASHing YouTube: An Analysis of Using DASH in YouTube Video Service.", In38th Annual IEEE Conference on Local Computer Networks 2013 Oct 21 (pp. 407-415). IEEE.

[7] Press – YouTube, https://www.youtube.com/intl/en-GB/yt/about/press/, 30.03.2020.

[8] S. Möller, P. Le Callet and A. Perkis, editors, "Qualinet White Paper on Definitions of Quality of Experience. ", Technical Report Version 1.2. European Network on Quality of Experience in Multimedia Systems and Services, 2013.

[9] H. Rifai, S. Mohammed, A. Mellouk, "A Brief Synthesis of QoS-QoE Methodologies.", In: 2011 10th International Symposium on Programming and Systems. IEEE, 2011. p. 32-38.

[10] R. Schatz, T. Hoßfeld, P. Casas, "Passive Youtube Qoe Monitoring for ISPs.", In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2012. p. 358-364.

[11] R. C. Streijl, S. Winkler and D. S. Hands, "Mean Opinion Score (MOS) Revisited: Methods and Applications, Limitations and Alternatives.", Multimedia Systems (2016) 22: 213. https://doi.org/10.1007/s00530-014-0446-1

[12] Rec, I. T. U. T., "P. 10: Vocabulary for Performance and Quality of Service, Amendment 2: New Definitions for Inclusion in Recommendation ITU-T P. 10/G. 100. ", Int. Telecomm. Union, Geneva, 2008.

[13] F. Qiu and Y. Cui, "An Analysis of User Behavior in Online Video Streaming. ", In Proceedings of the international workshop on Very-large-scale multimedia corpus, mining and retrieval, pp. 49-54. 2010.

[14] Dhiraj072: "random-word-generator", https://github.com/Dhiraj072/random-word-generator, 31.03.2020.

[15] YouTube Data API (v3) - Quota Calculator, https://developers.google.com/youtube/v3/determine_quota_cost?hl=hr, 31.03.2020.

[16] youtube-dl utility, https://rg3.github.io/youtube-dl/, 31.03.2020.

[17] The FreeBSD Project, https://www.freebsd.org/, 08.05.2020.

[18] Ubuntu, https://ubuntu.com/, 08.05.2020.

[19] Net.Shark, http://www.albedotelecom.com/pages/fieldtools/src/netshark.php, 08.05.2020.

[20] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz and P. Halvorsen, "Video Streaming Using a Location-based Bandwidth-lookup Service for Bitrate Planning.", ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 8, no. 3 (2012): 1-19.

[21] J. Van Der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen and F. De Turck, "HTTP/2-based Adaptive Streaming of HEVC Video over 4G/LTE Networks.", IEEE Communications Letters 20, no. 11 (2016): 2177-2180.

[22] Integrated Multiprotocol Network Emulator/Simulator, http://imunes.net/, 08.05.2020.

[23] Analiza alata Wireshark, Nacionalni CERT, https://www.cis.hr/www.edicija/LinkedDocuments/NCERT-PUBDOC-2010-09-312.pdf

[24] Samsung Galaxy S8, GSMArena, https://www.gsmarena.com/samsung_galaxy_s8-8161.php, 28.04.2020.

[25] Appium, http://appium.io/docs/en/about-appium/intro/?lang=en, 28.04.2020.

[26] N. Verma, "Mobile Test Automation with Appium", Packt Publishing Ltd, 2017.

[27] I. Orsolic and L. Skorin-Kapov, "A Framework for in-Network QoE Monitoring of Encrypted Video Streaming," in IEEE Access, vol. 8, pp. 74691-74706, 2020, doi: 10.1109/ACCESS.2020.2988735.

[28] B. D. Bašić, J. Šnajder, "Strojno učenje: Uvod u strojno učenje", https://www.fer.unizg.hr/_download/repository/SU-2019-01-Uvod.pdf, 25.05.2020.

[29] J. Šnajder, "Strojno učenje: Vrednovanje modela", https://www.fer.unizg.hr/_download/repository/SU-2017-21-VrednovanjeModela.pdf, 25.05.2020.

[30] I. H. Witten, E. Frank and M. A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Burlington, MA: Morgan Kaufmann, 2011

[31] R.C. Holte, "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning, 11:63–91, 1993.

[32] C. G. Nevill-Manning, G. Holmes and I. H. Witten, "The Development of Holte’s 1R Classifier", In Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, 1995. Proceedings., pages 239–242. IEEE, 1995.

[33] L. Breiman, A. Cutler, "Random forests", https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm, 03.06.2020.

[34] MPEG-DASH Standard, https://mpeg.chiariglione.org/standards/mpeg-dash, 26.06.2020.

[35] Telekom Deutschland: Stream On, https://www.telekom.de/unterwegs/tarife-und-optionen/streamon, 26.06.2020.

[36] Weka: SpreadSubsample Class, https://weka.sourceforge.io/doc.dev/weka/filters/supervised/instance/SpreadSubsample.html, 26.06.2020.

[37] Weka: Resample Class, https://weka.sourceforge.io/doc.dev/weka/filters/supervised/instance/Resample.html, 26.06.2020.

[38] YouTube Video Stream Format Codes, https://gist.github.com/sidneys/7095afe4da4ae58694d128b1034e01e2, 26.06.2020.

[39] Weka: BestFirst Class, https://weka.sourceforge.io/doc.dev/weka/attributeSelection/BestFirst.html, 26.06.2020.

[40] Weka: WrapperSubsetEval, https://weka.sourceforge.io/doc.dev/weka/attributeSelection/WrapperSubsetEval.html, 26.06.2020.

Sažetak

U posljednje vrijeme na globalnoj razini bilježimo sve veći porast broja korisnika Interneta. To dovodi i do sve većeg porasta u globalnom Internet prometu. Upravo sve učestalije korištenje usluga video strujanja pridonosi tom porastu. Budući da je većina Internet prometa kriptirana to predstavlja problem davateljima mrežnih usluga pri praćenju iskustvene kvalitete usluge krajnjim korisnicima. Dodatno, korisničke interakcije koje su u vezi s prikazivanjem video sadržaja utječu na uzorke u mrežnom prometu i otežavaju procjenu iskustvene kvalitete. U sklopu ovog rada se pomoću tehnika strojnog učenja ispituje mogućnost otkrivanja korisničkih interakcija u stvarnom vremenu.

Ključne riječi: korisničke interakcije, iskustvena kvaliteta, strojno učenje, YouTube

Summary

Nowadays, on the global scale there is an ever-increasing number of Internet users. This has led to a rise in global Internet traffic, driven mostly by increased usage of video streaming services. The fact that most of the Internet traffic is encrypted poses a problem for ISPs to monitor and meet end user Quality of Experience (QoE) expectations. Furthermore, user interactions during video playback have an impact on network traffic patterns and make the assessment of QoE more difficult. This thesis analyses solutions that employ machine learning techniques to detect user interactions in real time.

Key words: user interactions, Quality of Experience, Machine learning, YouTube

Popis oznaka i kratica

QoE – iskustvena kvaliteta (engl. *Quality of Experience*)

QoS – kvaliteta usluge (engl. *Quality of Service*)

API – aplikacijsko programsko sučelje (engl. *Application Programming Interface*)

URL – ujednačeni lokator sadržaja (engl. *Uniform Resource Locator*)

HTTP – protokol za upravljanje i prijenos hiperteksta preko Interneta (engl. *HyperText Transfer Protocol*)

DASH – dinamičko prilagodljivo strujanje putem protokola HTTP (engl. *Dynamic Adaptive Streaming over HTTP*)

ISP – pružatelj internetske usluge (engl. *Internet Service Provider*)

KPI – ključni pokazatelji uspješnosti (engl. *Key Performance Indicators*)

Popis slika

Popis tablica

Popis isječaka koda