

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2209

**STVARNO-VREMENSKA PROCJENA ISKUSTVENE
KVALITETE USLUGE YOUTUBE TEMELJENA NA
STROJNOM UČENJU I NEOVISNA O POSTOJANJU
KORISNIČKIH INTERAKCIJA**

Blaž Planinić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2209

**STVARNO-VREMENSKA PROCJENA ISKUSTVENE
KVALITETE USLUGE YOUTUBE TEMELJENA NA
STROJNOM UČENJU I NEOVISNA O POSTOJANJU
KORISNIČKIH INTERAKCIJA**

Blaž Planinić

Zagreb, lipanj 2020.

DIPLOMSKI ZADATAK br. 2209

Pristupnik: **Blaž Planinić (0036493548)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: izv. prof. dr. sc. Lea Skorin-Kapov

Zadatak: **Stvarno-vremenska procjena iskustvene kvalitete usluge YouTube temeljena na strojnom učenju i neovisna o postojanju korisničkih interakcija**

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. Prikupljene podatke potrebno je analizirati i ispitati mogućnost definiranja mrežnih značajki koje ne ovise o postojanju korisničkih interakcija, te izgradnje modela za procjenu iskustvene kvalitete koji adresiraju i slučajeve bez i sa korisničkim interakcijama. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Rok za predaju rada: 30. lipnja 2020.

Prije svega, zahvaljujem se mentorici izv. prof. dr. sc. Lei Skorin-Kapov, asistentima Ireni i Ivanu na pruženoj prilici te pomoći i savjetima prilikom izrade diplomskog rada. Hvala mojim prijateljima bez kojih bi studiranje bilo puno teže i nezanimljivije. Veliko hvala mojoj obitelji, a posebno roditeljima, sestri i bratu, koja me je pratila kroz cijelo vrijeme školovanja i koja je bila tu kada je bilo najpotrebnije. Vi imate najveće zasluge zbog onoga što sam danas postao, a to je izgrađena osoba koja zna što želi ostvariti u životu. I na kraju, najveće hvala mojoj djevojci Anamariji kojoj i posvećujem ovaj rad. Nijedan moj uspjeh, pa tako ni ovaj, ne bi bio moguć bez tebe!

Sadržaj

Uvod	1
1. Platforma YouTube i tehnologije na kojima se temelji	3
1.1. Prilagodljivo strujanje putem protokola <i>HTTP</i>	5
1.2. Video na zahtjev	7
2. Iskustvena kvaliteta i kvaliteta usluge	8
2.1. Kvaliteta usluge.....	8
2.2. Iskustvena kvaliteta.....	9
2.2.1. Subjektivno vrednovanje iskustvene kvalitete	11
2.2.2. Objektivno vrednovanje iskustvene kvalitete	11
3. Stvaranje baze i prikupljanje podataka o YouTube videozapisima	14
3.1. Baza podataka o videozapisima	14
3.1.1. Programski alat WAMP	17
3.1.2. YouTube-dl.....	18
3.2. Prikupljanje podataka o YouTube videozapisima.....	19
3.2.1. YouTube Data API v3.....	19
3.2.2. Skripta za prikupljanje podataka o YouTube videozapisima ..	21
4. Metodologija prikupljanja i obrade podataka	23
4.1. Laboratorijsko okruženje	23
4.2. Prikupljanje podataka s aplikacijske razine	25
4.2.1. Radni okvir Appium	34
4.3. Prikupljanje podataka s mrežne razine	36
4.3.1. Alat tcpdump	36
4.3.2. IMUNES	37
4.4. Obrada prikupljenih podataka	38

4.4.1. Obrada prikupljenih podataka s aplikacijske razine	38
4.4.2. Obrada prikupljenih podataka s mrežne razine	41
5. Izgradnja modela strojnog učenja i analiza rezultata	43
5.1. Strojno učenje	43
5.1.1. Tehnika i algoritam učenja modela	45
5.1.2. Vrednovanje uspješnosti naučenog modela	46
5.1.3. WEKA.....	48
5.2. Izgradnja modela pomoću značajki odabranih na skupu podataka bez interakcija.....	49
5.3. Izgradnja modela pomoću značajki odabranih na mješovitom skupu podataka	53
5.4. Analiza rezultata	57
Zaključak.....	70
Literatura.....	72
Sažetak	77
Summary	78
Dodaci.....	79
Popis slika	79
Popis tablica	82
Popis oznaka i kratica	83

Uvod

Proteklih godina sve je veći porast količine Internet prometa, a postoji nekoliko razloga zašto. Danas je otprilike 4.5 milijardi korisnika Interneta, a procjene govore da bi broj korisnika mogao porasti i do 5.3 milijarde do 2023. godine [1]. Osim porasta korisnika, veliku ulogu, a možda i najveću, u porastu količine Internet prometa ima masovno korištenje usluga strujanja videa. Brojke govore da je od 2015. godine do 2020. godine Internet video promet porastao za 31%, a trenutni udio iznosi 79% u odnosu na cijeli Internet promet [2].

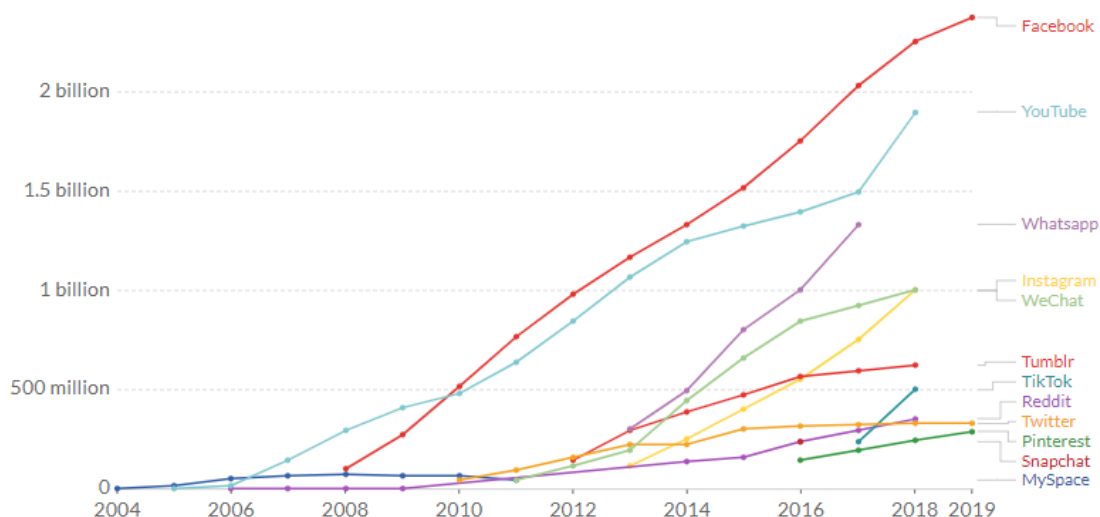
U ovakim uvjetima, kada se većina Internet prometa odnosi na strujanje videa, davatelji mrežnih usluga suočeni su s izazovom efikasnog upravljanja ograničenim mrežnim resursima, a da se pri tome krajnjim korisnicima koji gledaju videozapise osigura i pruži zadovoljavajuća razina iskustvene kvalitete. S obzirom da je većina mrežnog prometa kriptirana, čak i samo praćenje performansi usluga strujanja videa u mreži predstavlja problem. Potencijalno rješenje za praćenje iskustvene kvalitete se pronalazi u primjeni metoda strojnog učenja kod procjene iskustvene kvalitete krajnjih korisnika na temelju analize mrežnog prometa.

Međutim, korisničko ponašanje prilikom gledanja videozapisa u obliku interakcija utječe na uzorke u mrežnom prometu, a samim time i na procjenu iskustvene kvalitete. Stoga, cilj ovog rada je uspostaviti okruženje za automatizirano prikupljanje podataka vezanih uz prijenos YouTube video sadržaja na zahtjev na Android mobilnom uređaju. Unutar okruženja potrebno je implementirati simulirane korisničke interakcije koje će se programski aktivirati prilikom reproduciranja videozapisa. Za vrijeme izvođenja navedenog procesa potrebno je istovremeno prikupljati podatke s aplikacijske i mrežne razine. Prikupljene podatke potrebno je analizirati i ispitati mogućnost definiranja mrežnih značajki koje ne ovise o postojanju korisničkih interakcija kako bi se izgradio model strojnog učenja za procjenu iskustvene kvalitete koji adresira slučajeve bez obzira na prisutnost korisničkih interakcija.

Ovaj rad sastoji se od šest poglavlja. U prvom poglavlju opisuje se platforma YouTube i tehnologije koje platforma koristi za reproduciranje videozapisa. U drugom poglavlju objašnjeni su pojmovi kvalitete usluge i iskustvene kvalitete. Treće poglavlje opisuje proces prikupljanja podataka o videozapisima s YouTube-a te koji se sve alati i tehnologije koriste u svrhu prikupljanja. U četvrtom poglavlju objašnjena je metodologija prikupljanja i obrade podataka. Opisuje je laboratorijsko okruženje u kojem su se mjerenja radila, automatizirani proces prikupljanja podataka s aplikacijske razine i mrežne razine te obrada prikupljenih podataka. U petom poglavlju opisano je strojno učenje i izgradnja modela strojnog učenja. Također se analiziraju rezultati dobiveni izgradnjom modela te se pokušavaju otkriti značajke koje su neovisne o korisničkim interakcijama. Na samom kraju rada nalazi se popis literature, sažeci na hrvatskom i engleskom jeziku te popis slika, tablica i oznaka i kratica.

1. Platforma YouTube i tehnologije na kojima se temelji

Platforma YouTube je jedna od najvećih mrežnih usluga na današnjem tržištu koja svojim korisnicima pruža usluge postavljanja i pregledavanja videozapisa. Kao takva, trenutno u svijetu broji preko 300 milijuna dnevno aktivnih korisnika, a zanimljiva je činjenica ta da se svaki dan pogleda otprilike 1 milijarda sati videozapisa [3]. Također, YouTube se smatra drugom najvećom društvenom mrežom (slika 1.1) jer brojke govore da čak 79% korisnika Interneta posjeduje YouTube profil [3]. Usluga je besplatna te se videozapisi mogu pregledavati bez registracije korisnika (osim sadržaja koji nije primjeren za osobe mlađe od 18 godina) dok je za postavljanje videozapisa nužna registracija.



Slika 1.1 - Broj korisnika društvenih mreža [4]

Platforma je osnovana u veljači 2005. godine od strane bivših zaposlenika PayPal-a, tvrtke koja omogućuje obavljanje uplata i novčanih prijenosa u potpunosti preko Interneta. YouTube je otvoren za javnost tek nekoliko mjeseci nakon osnivanja, a već ubrzo privlači velike ulagače. Tako u razdoblju od studenog 2005. godine do travnja 2006. godine investitorska tvrtka Sequoia Capital ulaže 11,5 milijuna dolara. Nedugo zatim tvrtka Google

najavljuje preuzimanje i otkup svih dionica u vrijednosti 1,65 milijardi dolara. Ugovor je zaključen u studenom 2006. godine te Google postaje i ostaje vlasnik sve do danas [5].

Glavni slogan YouTube-a je taj da svi trebaju imati mogućnost slobode govora te da je svijet bolje mjesto u kojem slušamo, dijelimo i gradimo zajednicu kroz svoje priče. Njihove vrijednosti se baziraju na četiri esencijalne slobode koje ih definiraju [6]:

- 1.) Sloboda izražavanja – svi ljudi bi trebali imati mogućnost slobode govora, dijeljenja mišljenja i poticanja otvorenog dijaloga,
- 2.) Sloboda informacija – svi ljudi bi trebali imati otvoren pristup informacijama; video je snažan izvor edukacije, gradnje razumijevanja i dokumentiranja svjetskih događaja,
- 3.) Sloboda prilika – svi ljudi bi trebali imati priliku biti otkriveni, izgraditi posao i uspjeti u svojim uvjetima te da ljudi sami odluče što je popularno,
- 4.) Sloboda pripadanja – svi ljudi bi trebali imati mogućnost pronalaska zajednice u kojoj će zajedno dijeliti interese i strasti.

YouTube u današnje vrijeme mnogim ljudima predstavlja priliku za zaradu. Zaraditi se može na više načina, no prvi je uvjet prikupiti određen broj pretplatnika (engl. *subscribers*) na svome kanalu i ostvarivati velik broj pregleda (engl. *views*) na pojedinom videozapisu. Razlog je taj što se većina zarade bazira na reklamama (engl. *ads*). Naime, svaki pregled reklame donosi određeni iznos zarade. Tako za 20.000 pregleda videozapisa na dnevnoj razini se zaradi otprilike od 180 do 320 kuna [7]. Upravljanje reklamama provodi se pomoću programa Google AdSense [8]. To je program koji omogućuje izdavačima da ostvare prihod od svojeg *online* sadržaja. AdSense funkcionira tako da pridružuje oglase *web*-lokacijama na temelju sadržaja i posjetitelja tih *web*-lokacija (u ovom slučaju videozapisima). Oglase izrađuju i plaćaju oglašivači koji žele promovirati svoje proizvode. Budući da ti oglašivači plaćaju različite cijene za različite oglase, iznos zarade može se razlikovati [8].

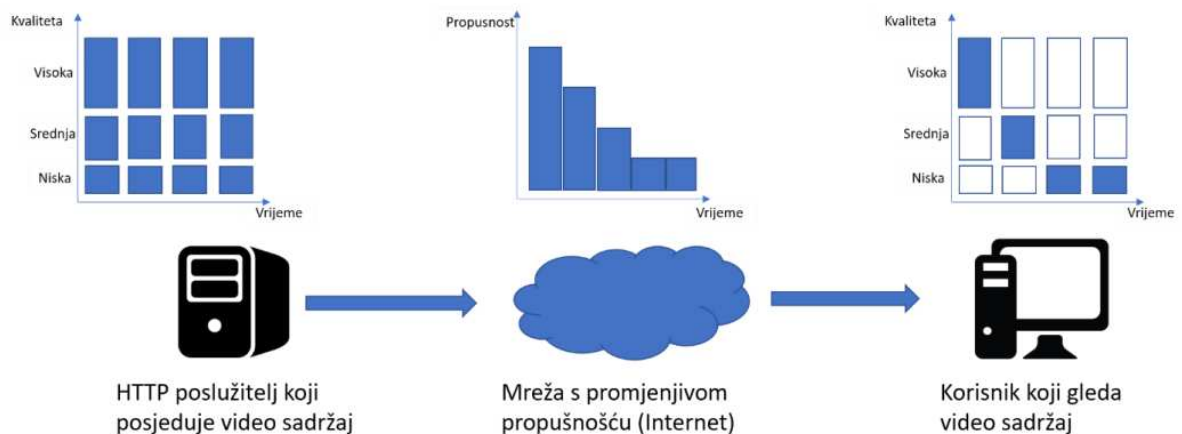
Reklame se mogu pojavljivati čak i nekoliko puta prilikom gledanja videozapisa. To korisniku koji gleda videozapis može uvelike postati naporno. Rješenje koje YouTube nudi je *YouTube Premium Account*. Kao što i samo ime kaže, to je povlašteni račun koji korisniku omogućuje nesmetano gledanje videozapisa ne prikazujući mu reklame (engl. *ad-free*). Osim toga, moguće je i privremeno preuzimanje video sadržaja kako bi ga korisnik mogao gledati bez pristupa Internetu te neprekidno izvođenje videozapisa u pozadini (engl. *background play*). *Premium Account* u sebi uključuje i *YouTube Music*, Google-ov glazbeni servis za strujanje [9]. Za potrebe ovog rada koristit će se *YouTube Premium Account*.

1.1. Prilagodljivo strujanje putem protokola *HTTP*

Platforma YouTube koristi tehniku dinamički prilagodljivog strujanja putem protokola *HTTP* (engl. *Dynamic Adaptive Streaming over HTTP, DASH*), također poznatog kao *MPEG-DASH* jer se tehnika temelji na *MPEG* standardu [10]. Ova tehnika se zasniva na tehnologiji prilagodljivog strujanja putem protokola *HTTP* (engl. *HTTP Adaptive Streaming, HAS*). To je jedna od najčešće korištenih tehnologija kod usluga koje se bave prikazivanjem videozapisa. *HAS* je kombinacija učitavanja i istovremenog prikazivanja video sadržaja. Razina kvalitete videozapisa prilagođava se trenutnim uvjetima u mreži te time omogućuje neprekidno reproduciranje videozapisa. Osim navedene tehnologije, postoje i mnoge druge tehnologije prilagodljivog strujanja putem protokola *HTTP* kao što su Apple-ov *HLS*, Microsoft-ov *Smooth Streaming* i mnoge druge. YouTube koristi dvije vrste transportnog protokola, a to su *TCP* (engl. *Transmission Control Protocol*) i *QUIC* (engl. *Quick UDP Internet Connections*) [11]. *QUIC* je transportni protokol razvijen od strane Google-a čiji je glavni cilj smanjiti kašnjenje u usporedbi s *TCP* protokolom [12]. Temelji se na *UDP* (engl. *User Datagram Protocol*) transportnom protoklu koji nema mogućnost provjere primitka poruke.

Podaci videozapisa se prenose do korisnika putem *HTTP* veze gdje se pohranjuju u aplikacijski spremnik (engl. *buffer*). Kada je dovoljna količina

podataka preuzeta, odnosno pohranjena u spremnik, video će krenuti s reprodukcijom. Sve dok ima podataka u spremniku videozapis će se prikazivati, a u suprotnom video će zastati (engl. *stalling*) dok se spremnik ponovno ne napuni dovoljnom količinom podataka potrebnom da bi se videozapis nastavio reproducirati. Ovo se događa kada su uvjeti u mreži loši, te kada je potrebno više vremena za preuzimanje podataka. Razina kvalitete, odnosno rezolucija i brzina kodiranja (engl. *bitrate*) videozapisa, se automatski prilagođava uvjetima u mreži. Videozapis uvijek teži biti reproduciran u što većoj kvaliteti koju trenutno izmjereni uvjeti mogu podržati [10]. Slika 1.2 prikazuje kako se kvaliteta video zapisa prilagođava dinamičkim uvjetima u mreži. Videozapis se na poslužitelju pohranjuje u raznim verzijama koje se razlikuju po kvaliteti, odnosno mogu biti različite rezolucije ili brzine kodiranja. Svaka verzija videozapisa podijeljena je u segmente koji obično traju otprilike dvije do deset sekundi, te se učitava segment po segment. Kada klijent procijeni da se propusnost mreže smanjila, dohvatit će se sljedeći segment u nižoj kvaliteti.



Slika 1.2 - Prilagodljivo strujanje putem protokola *HTTP*

1.2. Video na zahtjev

YouTube je jedna od mnogih usluga koja nudi mogućnost gledanja Videana-zahtjev (engl. *Video-on-demand*, *VOD*). *VOD* sustavi pružaju korisniku mogućnost da pretražuju, odabiru i gledaju višemedijske sadržaje u vrijeme kada oni to žele na osnovi kataloga programa koji nudi pružatelj višemedijskih usluga. Sustavi mogu biti oblika za strujanje kako bi se video mogao odmah gledati na sustavu ili za preuzimanje za naknadno gledanje videa. YouTube nudi oba oblika, no preuzimanje je dodatna usluga koja se naplaćuje (*Premium Account* opisan u poglavlju 2). U novije vrijeme, sve više su popularne *VOD* usluge koje se bave strujanjem filmova i serija, poput Netflix-a i HBO GO-a. Ljudi iz naslonjača svojih domova mogu u visokoj kvaliteti gledati svoje omiljene serije i filmove kada oni to požele.

Pojavom *VOD* usluga internetski promet je značajno porastao. Statistika kaže da danas čak tri četvrtine internetskog prometa otpada na prijenos video sadržaja. Također, tvrtka Cisco je napravila predviđanje u kojem kažu da će taj promet čak narasti i do 82% do 2022. godine, od čega će se većina odnositi na *VOD* usluge [13].

2. Iskustvena kvaliteta i kvaliteta usluge

U ovom poglavlju opisani su i definirani pojmovi koji se koriste u svrhu analize video sadržaja i njihove kvalitete: iskustvena kvaliteta i kvaliteta usluge. Osim njihovih definicija, opisano je na koji se način vrednuje iskustvena kvaliteta te koji su to faktori koji imaju utjecaja na nju.

2.1. Kvaliteta usluge

Kvaliteta usluge (engl. *Quality of Service, QoS*) je koncept koji opisuje ukupan učinak određene usluge kao što su usluge telekomunikacije ili računalne mreže. U prošlosti, kada je analogna telefonija bila zastupljena, *QoS* je označavao vjerojatnost dostupnosti puta u telekomunikacijskoj mreži. No danas mu je definicija potpuno drugačija. Međunarodna unija za telekomunikacije *ITU* je definirala *QoS* u preporuci *ITU-T-a E.800* kao „Ukupan efekt performansi određene usluge koja se određuje zadovoljstvom krajnjeg korisnika“ [14]. Ova definicija govori da je zapravo korisnik krajnji „sudac“ dobre ili loše kvalitete usluge. Različite aplikacije zahtijevaju različite kvalitete usluga. Za neke aplikacije je kritično minimalno kašnjenje i vrijeme odgovora (kao što je stvarno-vremenska komunikacija (engl. *real-time communication*)), dok je za neke aplikacije bitnija kvaliteta slike (kao što su videozapisi na zahtjev usluge YouTube).

QoS obuhvaća parametre kao što su broj pogrešnih paketa, razina gubitka paketa, kašnjenje u prijenosu paketa, kolebanje kašnjenja i propusnost mreže. Takvi parametri direktno utječu na kvalitetu usluge, pa onda konačno i na iskustvenu kvalitetu korisnika (opisano u sljedećem potpoglavlju). Tablica 2.1 prikazuje i sažima pet kategorija parametara kvalitete usluge [15].

Tablica 2.1 - Pet kategorija QoS parametara [15]

Kategorija	Primjer parametra
Orijentirani performansama	<i>E2E</i> (engl. <i>end-to-end</i>) kašnjenja i <i>bit rate</i>
Orijentirani formatu	Rezolucija video zapisa, broj okvira u sekundi, format pohrane
Sinkrono-orijentirani	Asimetrija između početaka audio i video sekvenci
Troškovno-orijentirani	Naknade konekcije i prijenosa podataka
Korisnički-orijentirani	Kvaliteta slika i zvuka

2.2. Iskustvena kvaliteta

Iskustvena kvaliteta (engl. *Quality of Experience, QoE*) je koncept koji se bavi mjerenjem razine korisničkog zadovoljstva korištenjem određene usluge ili proizvoda [16]. Sve većim razvojem višemedijskih usluga, naglasak s tehnologije pomaknuo se na korisnike i njihove zahtjeve. Kako je danas velik broj različitih usluga, a svakim danom izlaze nove, iskustvena kvaliteta je otvoreno područje istraživanja. Nažalost, zbog subjektivnosti iskustvene kvalitete nemoguće je definirati gotov model za mjerenje iskustvene kvalitete koji bi vrijedio za sve telekomunikacijske usluge, ali je zato moguće definirati pojedine modele za procjenu iskustvene kvalitete za različite tipove usluga. Definicija iskustvene kvalitete glasi [16]:

„Iskustvena kvaliteta je stupanj zadovoljstva ili iritiranosti od strane korisnika aplikacije ili usluge. To proizlazi iz ostvarenja njegovih ili njezinih očekivanja s uvažavanjem korisnosti i/ili uživanja aplikacije ili usluge s obzirom na korisničku osobnost i trenutno stanje“.

Puno je faktora koji utječu na samo zadovoljstvo korisnika, pa samim time i na iskustvenu kvalitetu. Svaki korisnik na svoj način doživljava kvalitetu korištene usluge ili aplikacije (slika 2.1). Zadovoljstvo korisnika ne mora ovisiti samo o aplikaciji ili usluzi, nego može ovisiti i o samom raspoloženju korisnika. Iz tog razloga, istraživanja koja se bave mjerenjem i predviđanjem iskustvene kvalitete u mrežnom okruženju često pokušavaju zanemariti veliki broj subjektivnih faktora koji utječu na iskustvenu kvalitetu krajnjeg korisnika. Dakle, takva istraživanja, prema definiciji od strane *ITU-T*-a, bi se trebala fokusirati na utjecaj tehničkih parametara na iskustvenu kvalitetu [17]. Tehnički parametri su objektivni parametri koji utječu na zadovoljstvo krajnjeg korisnika, ne uzimajući u obzir njegovu osobnost, emocije, raspoloženje i mnoge druge faktore.



Slika 2.1 – Primjer faktora koji utječu na zadovoljstvo korisnika umrežene usluge

Postoje dva načina vrednovanja iskustvene kvalitete, a to su subjektivni i objektivni [18]. Svaki od navedenih načina zahtjeva različit pristup mjerenju i vrednovanju. Objektivna mjerenja ne ovise o mišljenju korisnika (npr. vrijeme potrebno za rješavanje zadatka mjereno u sekundama), dok subjektivna mjerenja u velikoj mjeri ovise o mišljenju korisnika (npr. zadovoljstvo korisnika usluge) [19]. U nastavku rada detaljnije su opisani navedeni pojmovi.

2.2.1. Subjektivno vrednovanje iskustvene kvalitete

Subjektivno vrednovanje iskustvene kvalitete provodi se na više načina, a najčešći oblici takvog ispitivanja su anketiranje i ispitivanje. Takva mjerenja se rade u testnim okruženjima koja pokušavaju simulirati stvarno okruženje. Postoji velik nedostatak kod ovakve vrste ispitivanja jer testna okruženja najčešće ne mogu dočarati stvarnu situaciju. Jedan od primjera je taj da se korisnik pri testiranju ne ponaša i ne osjeća isto kao kad bi tu istu stvar radio od doma gdje ga nitko ne promatra. Kod takvih se mjerenja najčešće koristi mjera srednje ocjene kvalitete (engl. *Mean Opinion Score*, *MOS*). *MOS* vrijednosti se izražavaju pomoću *ACR* (engl. *Absolute Category Rating*) ljestvice (tablica 2.2). No mora se naglasiti kako ni ova mjera nije najbolja za ocjenu kvalitete usluge jer različiti su kriteriji ocjenjivanja pojedinih korisnika [20].

Tablica 2.2 - *MOS* vrijednosti

Ocjena	Opis
5	Odlično
4	Dobro
3	Prihvatljivo
2	Loše
1	Vrlo loše

2.2.2. Objektivno vrednovanje iskustvene kvalitete

Kao što je već rečeno, kompleksnost subjektivnog vrednovanja iskustvene kvalitete ukazuje na to da se s aspekta korisnika teško mogu donijeti kvalitetni zaključci. To je razlog da se često iskustvena kvaliteta usluge nastoji vrednovati pomoću modela i objektivnih metrika. Kod definiranja pouzdanog modela vrednovanja iskustvene kvalitete potrebni su mjerljivi

parametri koji utječu na iskustvenu kvalitetu korisnika, te su vezani uz različite dijelove usluge i različiti za svaku uslugu.

ITU-T je u preporuci *P.1203* predstavio prvi standardizirani model za procjenu iskustvene kvalitete usluga postupnog preuzimanja i prilagodljivog strujanja audiovizualnog sadržaja putem pouzdanog prijenosa [21]. *P.1203* je skup dokumenata koji opisuju modele algoritama za praćenje kvalitete video sjednica koji koriste TCP transportni protokol ili slične transportne protokole. Skup sadrži tri dokumenta od kojih svaki opisuje funkcionalnosti različitih modela:

- 1.) *P.1203.1*. – modul za procjenu video kvalitete (detaljan opis u [22]),
- 2.) *P.1203.2* – modul za procjenu audio kvalitete (detaljan opis u [23]),
- 3.) *P.1203.3* – modul za audiovizualnu integraciju i izračun završne procjene kvalitete (detaljan opis u [24]).

Kod *VOD* usluga postoje mnogi parametri koji utječu na iskustvenu kvalitetu. Ti parametri zovu se ključni pokazatelji uspješnosti (engl. *Key Performance Indicators, KPI*). Neki od njih su početno kašnjenje (engl. *initial delay*), zastajkivanje (engl. *stalling*) i kvaliteta video zapisa. U nastavku su ukratko opisani navedeni *KPI*-evi.

Početno kašnjenje je vrijeme proteklo od trenutka kada korisnik pokrene reprodukciju do trenutka kada ta reprodukcija zaista krene. Radi se o neizbježnoj pojavi kod svakog strujanja videozapisa na zahtjev, a njegovo trajanje je *KPI* koji utječe na iskustvenu kvalitetu. Sve dok se dovoljno podataka ne prikupi u spremniku videozapis se neće pokrenuti.

Zastajkivanje je slično početnom kašnjenju, no ono se ne manifestira na početku videozapisa, već nakon što videozapis krene s reprodukcijom. Događa se kada se spremnik isprazni te nema dovoljno podataka za daljnju reprodukciju videozapisa. Vrijeme potrebno da se spremnik ponovno dovoljno napuni i da videozapis nastavi s reprodukcijom naziva se vremenom zastajkivanja (engl. *stalling duration*).

Kvaliteta videozapisa je isto jedan od faktora koji najviše utječu na iskustvenu kvalitetu. Segmenti videozapisa mogu se razlikovati po kvaliteti,

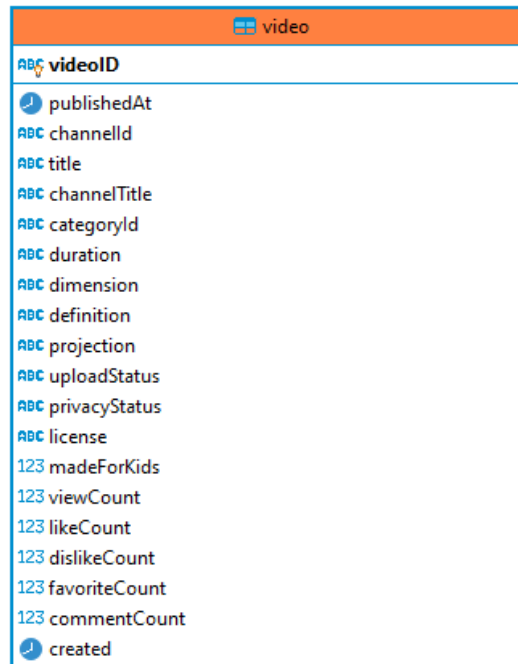
primjerice u smislu rezolucije ili brzine kodiranja. *KPI*-evi koji se odnose na kvalitetu videozapisa u velikoj mjeri utječu na razinu zadovoljstva korisnika, ali primjerice, korisnik će biti zadovoljniji ako videozapis bez zastajkivanja može pogledati u nižoj rezoluciji nego sa zastajkivanjem u višoj rezoluciji.

3. Stvaranje baze i prikupljanje podataka o YouTube videozapisima

U ovom poglavlju opisan je način prikupljanja i pohrane podataka koji su potrebni kako bi se napravila mjerenja opisana u poglavlju 4. U prvom potpoglavlju opisana je baza podataka u koju se spremaju podaci o YouTube videozapisima, a u drugom potpoglavlju opisana je skripta pomoću koje se podaci prikupljaju.

3.1. Baza podataka o videozapisima

Prije nego li se počnu raditi mjerenja, potrebno je stvoriti bazu podataka identifikatora (u nastavku ID) o videozapisima s YouTube-a. Mjerenja koja su opisana u poglavlju 4. koriste bazu ID-eva kako bi se YouTube videozapisi mogli automatizirano otvarati i reproducirati. Baza podataka napravljena je pomoću programskog alata WAMP koji je ukratko objašnjen u odjeljku 3.1.1. Osim ID-eva, potrebno je i prikupiti ostale informacije koje se mogu saznati o videozapisima prilikom njihovog dohvata. Tako će se osim ID-eva prikupljati i kada je videozapis objavljen, ID kanala, naslov videozapisa, naslov kanala, ID kategorije, trajanje, dimenzija (npr. 2D ili 3D), definicija (standardna ili visoka definicija), oblik, status učitavanja, status privatnosti, vlasnik licence, je li videozapis primjeren za djecu, broj pregleda, broj sviđanja (engl. *like*), broj nesviđanja (engl. *dislike*), broj ljudi koji je videozapis označio kao omiljeni (engl. *favourite*), broj komentara i kada je kreiran videozapis. Slika 3.1 prikazuje *ER* dijagram (dijagram entiteta i veza, engl. *entity relationship diagram*) tablice *video* u koju se navedeni podaci spremaju.



Slika 3.1 - ER dijagram tablice video

YouTube pohranjuje trajanje videozapisa u formatu ISO 8601 čiji je oblik $P[n]Y[n]M[n]DT[n]H[n]M[n]S$, odnosno $P<date>T<time>$. Slovo P (engl. *period*) označava period trajanja u godinama (engl. *years*, Y), mjesecima (engl. *months*, M) i danima (engl. *days*, D), dok T (engl. *time*) označava vrijeme trajanja u satima (engl. *hours*, H), minutama (engl. *minutes*, M) i sekundama (engl. *seconds*, S). Oznaka $[n]$ se zamjenjuje vrijednostima ispred svake od oznaka reprezentacija. Također, nije potrebno ispunjavanje vrijednosti za svaku od oznaka, već je dovoljno ispuniti samo jednu. Tako je i $PT[n]S$ valjan oblik. Slika 3.2 predstavlja isječak iz tablice *video* koji prikazuje trajanje pojedinog videozapisa u formatu ISO 8601.

	ABC videoID	ABC duration
1	--U8PbrTdo	PT3M9S
2	--0cnjeDXQA	PT20M15S
3	--1W3Kq_pfk	PT3M36S
4	--BHuKeveg4	PT4M41S
5	--c9QY66Z6Q	PT11M24S
6	--dKptn22OM	PT12M27S
7	--dZb0dGxUE	PT2H37M39S
8	--H74_vwBs	PT46S
9	--hbv95pZV0	PT11M46S
10	--KDqrP5x4e	PT4M28S

Slika 3.2 - Isječak iz tablice *video*: format ISO 8601

Ovakav format trajanja videozapisa nije prigodan za potrebe računanja i manipulacije koje su potrebne u svrhu mjerenja. Iz tog razloga potrebno je napraviti pretvorbu trajanja iz formata ISO 8601 u trajanje u sekundama. U tu svrhu napisana je procedura koja za svaki redak tablice *video* računa trajanje u sekundama te ga zapisuje u novu tablicu *duration_in_seconds*. Slika 3.3 prikazuje proceduru koja pretvara trajanje videozapisa u sekunde.

```

create temporary table tempTableSecs
as
SELECT videoID,
CASE
WHEN duration LIKE '%DT%H%M%S' THEN TIME_TO_SEC(STR_TO_DATE(duration, '%dDT%H%M%S'))
WHEN duration LIKE '%DT%H%S' THEN TIME_TO_SEC(STR_TO_DATE(duration, '%dDT%H%S'))
WHEN duration LIKE '%DT%H%M' THEN TIME_TO_SEC(STR_TO_DATE(duration, '%dDT%H%M'))
WHEN duration LIKE '%DT%H' THEN TIME_TO_SEC(STR_TO_DATE(duration, '%dDT%H'))
WHEN duration LIKE '%DT%M' THEN TIME_TO_SEC(STR_TO_DATE(duration, '%dDT%M'))
WHEN duration LIKE '%DT%S' THEN TIME_TO_SEC(STR_TO_DATE(duration, '%dDT%S'))
WHEN duration LIKE '%DT' THEN TIME_TO_SEC(STR_TO_DATE(duration, '%dDT'))
WHEN duration LIKE 'PT%H%M%S' THEN TIME_TO_SEC(STR_TO_DATE(duration, 'PT%H%M%S'))
WHEN duration LIKE 'PT%H' THEN TIME_TO_SEC(STR_TO_DATE(duration, 'PT%H'))
WHEN duration LIKE 'PT%H%M' THEN TIME_TO_SEC(STR_TO_DATE(duration, 'PT%H%M'))
WHEN duration LIKE 'PT%H%S' THEN TIME_TO_SEC(STR_TO_DATE(duration, 'PT%H%S'))
WHEN duration LIKE 'PT%M' THEN TIME_TO_SEC(STR_TO_DATE(duration, 'PT%M'))
WHEN duration LIKE 'PT%M%S' THEN TIME_TO_SEC(STR_TO_DATE(duration, 'PT%M%S'))
WHEN duration LIKE 'PT%S' THEN TIME_TO_SEC(STR_TO_DATE(duration, 'PT%S'))
END as durationInSecs
FROM video
where videoID not in (select videoID from duration_in_seconds);

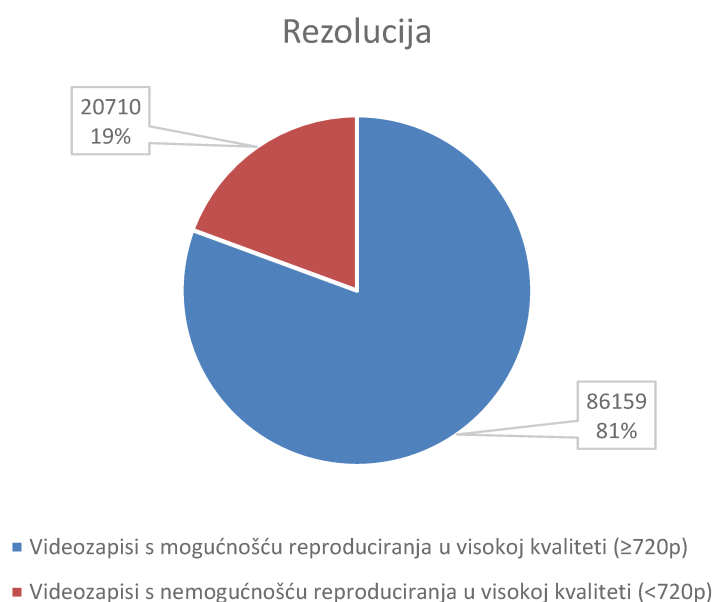
INSERT INTO duration_in_seconds
select videoID, durationInSecs
from tempTableSecs;

drop table tempTableSecs;

```

Slika 3.3 - Procedura za pretvorbu trajanja videozapisa iz ISO 8601 formata u sekunde

Nakon što se prikupe informacije o videozapisima, potrebno je otkriti koji videozapisi imaju mogućnost reproduciranja u visokoj kvaliteti ($\geq 720p$). Naime, da bi modeli strojnog učenja koji će se izgraditi iz podataka dobivenih nakon automatiziranog prikupljanja (potpoglavlje 4.2.) bili pouzdani, svaki videozapis koji će se reproducirati mora imati mogućnost gledanja u visokoj kvaliteti. U tu svrhu koristit će se program Youtube-dl koji je ukratko opisan u odjeljku 3.1.2. Pomoću ovog programa za svaki videozapis će se dohvatiti koje su dostupne rezolucije te koja je prosječna brzina kodiranja za svaku rezoluciju. Ukupan broj skupljenih informacija o videozapisima je 106.869, a većina njih ima mogućnost reproduciranja u visokoj kvaliteti (slika 3.4).



Slika 3.4 - Graf koji prikazuje odnos mogućnosti reproduciranja videozapisa u visokoj i niskoj kvaliteti

3.1.1. Programski alat WAMP

WAMP je programski alat koji služi za pokretanje i simulaciju poslužitelja na lokalnom računalu [25]. WAMP je akronim za Windows, Apache, MySQL i PHP. W (Windows) označava da je program kompatibilan s operacijskim sustavom Windows, dok ostatak skraćenice WAMP-a, AMP,

(Apache, MySQL i PHP) označava da su navedeni alati sadržani u WAMPu. U nastavku su ukratko objašnjeni svaki od pojmova:

- 1.) Apache je implementacija *HTTP web* poslužitelja. On je besplatan projekt otvorenog koda koji služi za pokretanje i postavljanje (engl. *to host*) jednostavnih *web* stranica na Internet.
- 2.) MySQL je također sustav otvorenog koda koji služi za upravljanje relacijskim bazama podataka.
- 3.) PHP (rekurzivni akronim od engleske riječi *Hypertext Preprocessor*) je skriptni jezik specijaliziran za dinamičko *web* programiranje. On može biti direktno ugrađen u HTML dokumente.

Dakle, WAMP je ništa drugo nego paket u koji su ugrađeni Apache, MySQL i PHP te njegovom instalacijom se instaliraju sve potrebne stvari koje su potrebne za razvoj web aplikacije. Za potrebe ovog rada koristi će se samo MySQL poslužitelj na kojem je napravljena baza podataka.

3.1.2. YouTube-dl

YouTube-dl je program naredbenog retka čija je glavna svrha preuzimanje videozapisa s YouTube-a i mnogih drugi stranica kao što su Twitch, Twitter, Facebook i mnogi drugi [26]. Ovo je besplatan program otvorenog koda napisan u programskom jeziku Python. YouTube-dl nudi jako puno mogućnosti. Tako se osim preuzimanja samih videozapisa, mogu preuzimati i liste videozapisa, zvučne verzije videozapisa (engl. *audio-only from video*), informacije vezane za videozapis koje u sebi uključuju i informacije vezane za dostupne formate videozapisa te mnoge druge mogućnosti. Slika 3.5 prikazuje primjer dohvata dostupnih formata za pojedini videozapis.


```
C:\Users\Blaz Planinic>youtube-dl --list-formats https://www.youtube.com/watch?v=MQaD3zVofho
[youtube] MQaD3zVofho: Downloading webpage
[info] Available formats for MQaD3zVofho:
format code  extension  resolution note
249          webm      audio only tiny 59k , opus @ 50k (48000Hz), 4.01MiB
250          webm      audio only tiny 78k , opus @ 70k (48000Hz), 5.09MiB
140          m4a       audio only tiny 130k , m4a_dash container, mp4a.40.2@128k (44100Hz), 10.17MiB
251          webm      audio only tiny 147k , opus @160k (48000Hz), 9.57MiB
278          webm      256x144      144p 97k , webm container, vp9, 25fps, video only, 6.98MiB
160          mp4       256x144      144p 110k , avc1.4d400c, 25fps, video only, 5.93MiB
242          webm      426x240      240p 226k , vp9, 25fps, video only, 12.56MiB
133          mp4       426x240      240p 276k , avc1.4d4015, 25fps, video only, 10.56MiB
243          webm      640x360      360p 413k , vp9, 25fps, video only, 22.69MiB
134          mp4       640x360      360p 537k , avc1.4d401e, 25fps, video only, 20.41MiB
244          webm      854x480      480p 754k , vp9, 25fps, video only, 36.51MiB
135          mp4       854x480      480p 826k , avc1.4d401e, 25fps, video only, 31.38MiB
247          webm      1280x720     720p 1506k , vp9, 25fps, video only, 68.35MiB
136          mp4       1280x720     720p 1551k , avc1.4d401f, 25fps, video only, 53.51MiB
248          webm      1920x1080    1080p 2677k , vp9, 25fps, video only, 174.51MiB
137          mp4       1920x1080    1080p 4571k , avc1.640028, 25fps, video only, 210.35MiB
18          mp4       640x360      360p 616k , avc1.42001E, mp4a.40.2@ 96k (44100Hz), 48.42MiB
22          mp4       1280x720     720p 810k , avc1.64001F, mp4a.40.2@192k (44100Hz) (best)
```

Slika 3.5 - Dohvat dostupnih formata za pojedini videozapis pomoću YouTube-dl-a

3.2. Prikupljanje podataka o YouTube videozapisima

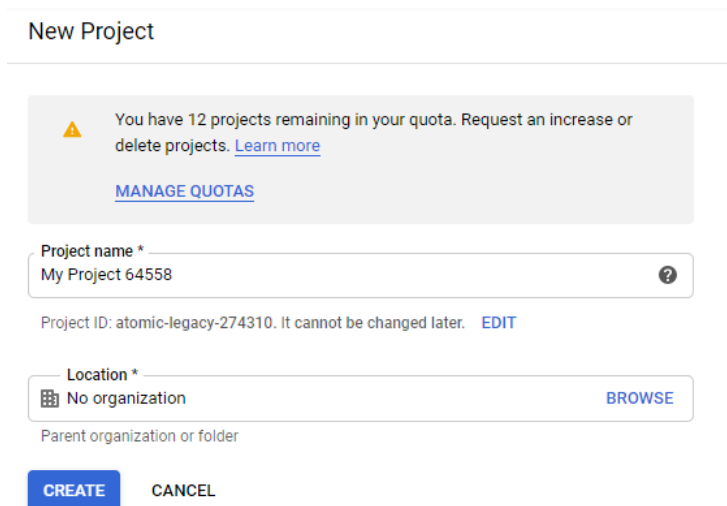
U svrhu prikupljanja podataka o YouTube videozapisima napravljena je skripta u programskom jeziku Java koja putem poziva na odgovarajući YouTube API dohvaća potrebne podatke. Potpoglavlje je podijeljeno u dva dijela. Prvi dio opisuje YouTube API koji služi za dohvat podataka, a drugi dio opisuje samu implementaciju skripte koja koristi navedeni API kako bi se prikupili podaci.

3.2.1. YouTube Data API v3

YouTube Data API (v3) je programsko sučelje koje omogućuje da se podacima s YouTube-a pristupa i manipulira ne samo preko YouTube aplikacije, nego i putem vlastitih aplikacija ili web stranica [27]. Gotovo sve funkcionalnosti ugrađene su unutar API-ja te sve funkcije koje se mogu izvršavati na YouTube-u mogu se izvršiti i putem API-ja. Tako je osim dohvata podataka o videozapisima moguće i unositi, ažurirati i brisati podatke za koje je prethodno potrebno imati dozvolu.

Kako bi se moglo pristupiti YouTube API-ju, podrazumijeva se da korisnik ima izrađen Google račun. Upravljanje računom za potrebe razvoja aplikacija odvija se na web aplikaciji <https://console.developers.google.com/>.

Aplikacija koja će koristiti API mora imati autorizacijska prava (engl. *authorization credentials*) kako bi mogla pristupiti API-ju. Prvi korak je napraviti projekt na svome računu (slika 3.6). Maksimalan broj projekata na pojedinom računu je 12.



New Project

You have 12 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
My Project 64558

Project ID: atomic-legacy-274310. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

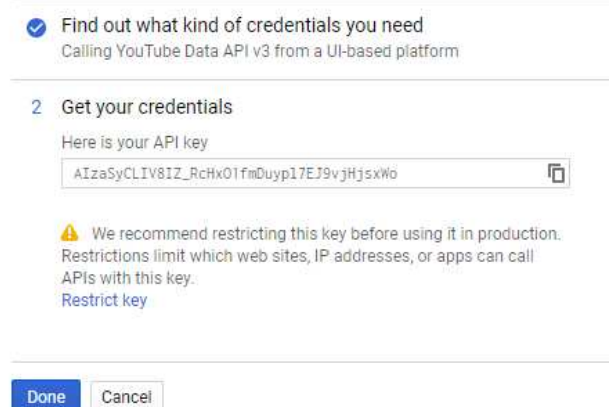
Parent organization or folder

[CREATE](#) [CANCEL](#)

Slika 3.6 - Kreiranje projekta na vlastitom Google računu

Nakon što je projekt kreiran, na njemu je potrebno omogućiti YouTube Data API te kreirati autorizacijska prava za pristup API-ju. U tu svrhu izrađuje se API ključ koji će služiti za autorizaciju prilikom upita na API (slika 3.7).

Add credentials to your project



✓ Find out what kind of credentials you need
Calling YouTube Data API v3 from a UI-based platform

2 Get your credentials

Here is your API key

AIzaSyCLIV8Iz_RcHx01fmDuyP17EJ9v_jHjSxWo

⚠ We recommend restricting this key before using it in production. Restrictions limit which web sites, IP addresses, or apps can call APIs with this key.
[Restrict key](#)

[Done](#) [Cancel](#)

Slika 3.7 - API ključ

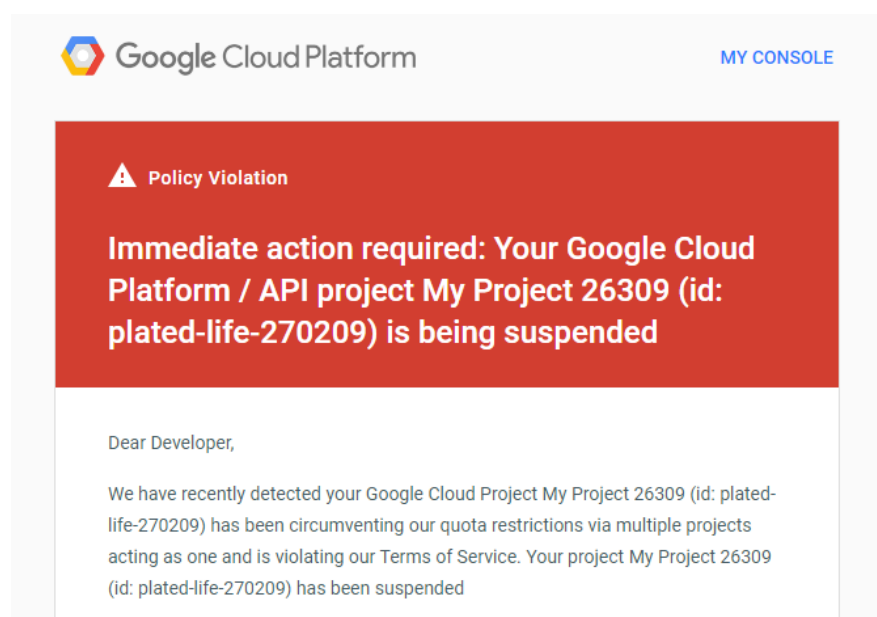
Projekti koji imaju omogućen YouTube Data API imaju ograničenje u broju poziva i zahtjeva za podacima. Dnevno ograničenje je deset tisuća jedinica (engl. *units*) [27]. Jedinice su YouTube-ov interni način računanja koliko je poziv zahtjevan te ne postoji jedinstveni javno dostupan kalkulator za njihovo računanje. Tako za poziv koji zahtjeva samo jedan resurs (npr. ID videozapisa) potrošit će se 1 jedinica, dok za poziv koji traži puno više resursa (npr. ID, naziv, trajanje, broj komentara itd...) potrošit će se i više od jedne jedinice.

3.2.2. Skripta za prikupljanje podataka o YouTube videozapisima

Za potrebe ovoga rada potrebno je prikupiti velik broj informacija o videozapisima kako bi se mjerenja mogla raditi na dovoljnom broju videozapisa koji zadovoljavaju postavljene uvjete (npr. videozapisi s trajanjem do 5 minuta i minimalno 50.000 pregleda). Kao što je opisano u odjeljku 3.2.1., YouTube Data API ima svoja ograničenja te se samim time javlja problem prikupljanja većeg broja podataka. S jednim projektom i ključem moguće je dnevno prikupiti maksimalno tristotinjak informacija o videozapisima. Razlog je taj što se za dohvat svih potrebnih informacija o jednom videozapisu potroši otprilike tridesetak jedinica. U svrhu ubrzanja prikupljanja podataka, napravljeno je više Google računa te je na svakom od računa napravljen maksimalni broj projekata.

Skripta funkcionira tako da za svaki projekt i njegov pripadni ključ napravi maksimalni dopušteni broj poziva na YouTube API. Skripta je podijeljena u dva dijela koji se odnose na pozive prema API-ju. Prvi dio skripte uključuje odabir ključne riječi koja će se tražiti na YouTube-u pomoću besplatne i javno dostupne knjižnice (engl. *library*) za generiranje slučajne riječi te poziv prema API-ju kojim se dohvaća ID videozapisa. Drugi dio je zadužen za prikupljanje detaljnih informacija o svakom videozapisu dobivenom u rezultatu. Dakle, za svaki ID videozapisa iz rezultata radi se drugi poziv prema API-ju te se dohvaćaju detaljne informacije. Podaci koji se prikupljaju opisani su u potpoglavlju 3.1.

Skripta završava kada se za svaki ključ napravi maksimalni broj poziva. Moguće ju je izvršiti jednom dnevno za iste projekte i njihove pripadne API ključeve. Naime, da bi stvari bile još kompliciranije, YouTube ne dopušta ovakav način rada skripte. Razlog je taj što više projekata i API ključeva napravljenih na Google računu koristi ista aplikacija (skripta). Jedna aplikacija smije koristiti samo jedan projekt i njegov pripadni API ključ. YouTube Data API prepoznaje ovakve slučajeve te suspendira projekt napravljen na Google računu (slika 3.8).

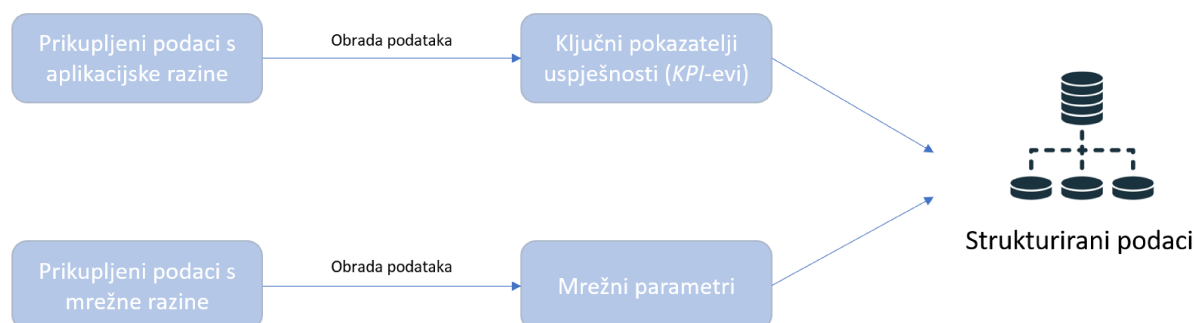


Slika 3.8 - Obavijest o suspendiranom API projektu

Najprije dolazi upozorenje u kojem stoji objašnjenje o kršenju pravila te nakon tri dana projekt bude suspendiran. Za to vrijeme moguće je prikupljati podatke. Na ovaj način prikupljene su informacije o 106.869 videozapisa.

4. Metodologija prikupljanja i obrade podataka

U ovom poglavlju je opisan cijeli proces prikupljanja podataka s aplikacijske i mrežne razine u laboratoriju te njihova obrada za kasnije razvijanje modela za predviđanje iskustvene kvalitete pomoću metoda strojnog učenja (slika 4.1). Podaci s aplikacijske razine prikupljaju se iz YouTube aplikacije Android mobilnog uređaja koja nudi opciju kopiranja podataka o videozapisu (engl. *Copy debug info*). Obradom prikupljenih podataka s aplikacijske razine računaju se *KPI*-evi koji predstavljaju podatke koji utječu na razinu korisničkog zadovoljstva (*QoE*). Istovremeno s aplikacijskim parametrima se prikupljaju i mrežni parametri na zasebnom računalu koji zajedno predstavljaju skup podataka za razvijanje modela strojnog učenja. Prilikom prikupljanja podataka, uvjeti u mreži su kontrolirani pomoću skripte u IMUNES-u (odjeljak 4.3.2.) koja svake sekunde mijenja karakteristike mreže.

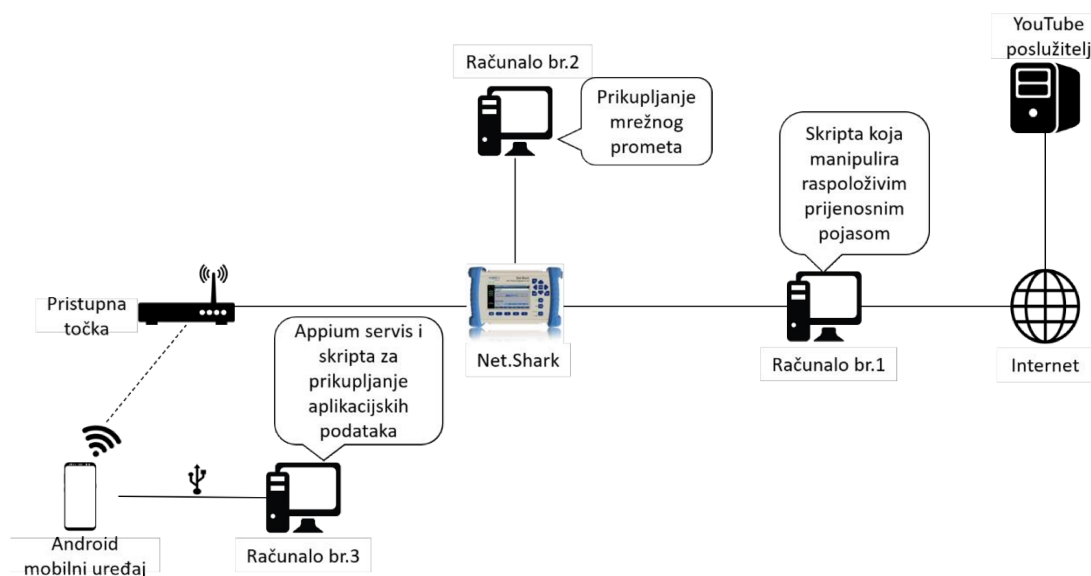


Slika 4.1 - Prikupljanje i obrada podataka

4.1. Laboratorijsko okruženje

Laboratorijsko okruženje postavljeno je na način da osigura emuliranje željenih uvjeta u mreži i prikupljanje podataka s aplikacijske i mrežne razine. Slika 4.2 prikazuje način na koji je okruženje postavljeno.

Prilikom gledanja YouTube videozapisa na Android mobilnom uređaju, mrežni paketi se šalju s YouTube poslužitelja preko Interneta do računala br.1. Računalo br.1 ima ugrađene dvije mrežne kartice od kojih je jedna direktno povezana na Internet, a druga je povezana s uređajem Net.Shark [28]. Navedeno računalo pokreće operacijski sustav FreeBSD na kojem je instaliran alat IMUNES pomoću kojeg se manipulira mrežnim uvjetima. Kao što je rečeno, računalo br.1 dalje preusmjerava promet na uređaj Net.Shark čija je svrha kopiranje podataka mrežnog prometa na računalo br.2. Računalo br.2 pokreće operacijski sustav Linux te ono pomoću alata tcpdump (odjeljak 4.3.1.) snima i pohranjuje mrežni promet u obliku .pcap datoteke. Net.Shark, osim što kopira mrežni promet, također ga i propušta na pristupnu točku na koju se Android mobilni uređaj povezuje putem *WiFi* mreže. Računalo br.3 pokreće operacijski sustav Windows 10 čija je svrha upravljanje mobilnim uređajem. Na računalu br.3 je pokrenuta skripta za automatizirano prikupljanje podataka (potpoglavlje 4.2.) koja pomoću Appium (opisan u odjeljku 4.2.1.) poslužitelja upravlja mobilnim uređajem. Kako bi računalo br.3 moglo komunicirati s Android mobilnim uređajem, ono je s njim povezano putem *USB* kabla.



Slika 4.2 - Shema laboratorijskog okruženja

4.2. Prikupljanje podataka s aplikacijske razine

U svrhu prikupljanja podataka s aplikacijske razine napisana je skripta u programskom jeziku Java koja pomoću Appium poslužitelja komunicira s mobilnim uređajem. Osim samog prikupljanja podataka implementirana je simulacija korisničkih interakcija prilikom reproduciranja videozapisa. Skripta simulira četiri načina gledanja videozapisa:

- 1.) Gledanje videozapisa bez interakcija,
- 2.) Gledanje videozapisa s interakcijom pauze (engl. *pause*),
- 3.) Gledanje videozapisa s interakcijom premotavanja unaprijed (engl. *seek forward*),
- 4.) Gledanje videozapisa s interakcijom prekida reprodukcije i prebacivanja na sljedeći videozapis (engl. *abandonment*).

Na ovaj način ostvarena je automatizacija procesa prikupljanja podataka s aplikacijske razine i istovremenog simuliranja korisničkih interakcija. Podaci koji se prikupljaju nalaze se u opciji Statistika za štrebere (engl. *Stats for nerds*) aplikacije YouTube. Opcija u sebi sadrži detaljne informacije o reprodukciji videozapisa u trenutnoj sekundi: ID videozapisa, rezolucija, audio i video *codec*-i, brzina konekcije i mnoge druge informacije. U nastavku teksta opisan je rad skripte u koracima:

1. Inicijalizacija procesa

Na početku skripte otvara se nova Appium sjednica. Appium poslužitelju potrebno je poslati podatke o uređaju s kojim će komunicirati (slika 4.3).

```
public static AndroidDriver<AndroidElement> startAppiumSession() throws MalformedURLException {
    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);
}
```

Slika 4.3 - Isječak koda - otvaranje Appium sjednice

Nakon što je sjednica uspješno otvorena, potrebno je pokrenuti YouTube mobilnu aplikaciju te pripremiti okruženje za prikupljanje podataka i izvođenje interakcija. Najprije se otvara pripremni (engl. *dummy*) video (3.a.) na kojem će se napraviti rotacija zaslona u horizontalni položaj (engl. *landscape*) te će se nakon toga otvoriti opcija Statistika za štrebere (slika 4.4). Ovo je potrebno napraviti zbog toga što okretanje zaslona i otvaranje Statistike za štrebere traje otprilike 5 sekundi te se tako podaci na početku videozapisa ne bi mogli prikupiti.

```
public static void showStatsForNerds(AndroidDriver<AndroidElement> driver) {
    // tap on the video
    WebElement e11 = (WebElement) driver.findElementByAccessibilityId("Video player");
    e11.click();

    // tap on more options menu
    WebElement e15 = (WebElement) driver.findElementByAccessibilityId("More options");

    // tap to show stats for nerds
    try {
        WebElement e16 = (WebElement) driver.findElementByXPath(
            "/hierarchy/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/"
            + "android.widget.LinearLayout/android.widget.ListView/android.widget.RelativeLayout[5]/android.widget.TextView");
        e16.click();
    } catch (Exception e) {
        WebElement e16 = (WebElement) driver.findElementByXPath(
            "/hierarchy/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/"
            + "android.widget.LinearLayout/android.widget.ListView/android.widget.RelativeLayout[3]/android.widget.TextView");
        e16.click();
    }
}
```

Slika 4.4 - Isječak koda - otvaranje Statistike za štrebere

2. Dohvat ID-eva videozapisa iz baze podataka

Nakon što je sve podešeno za početak prikupljanja podataka i izvođenje interakcija, potrebno je dohvatiti ID-eve videozapisa iz baze podataka (potpoglavlje 3.1.) na kojima će se navedeni proces izvršavati. Na temelju zadanih parametara filtera u SQL upitu dohvaćaju se ID-evi i trajanje videozapisa koje je potrebno radi izračuna vremena zbivanja i trajanja interakcija (slika 4.5). Mjerenja će se raditi na videozapisima dostupnim u visokoj kvaliteti ($\geq 720p$). U ovom radu filteri su postavljeni na sljedeće vrijednosti:

- Minimalni broj pregleda videozapisa: 5.000.000
- Minimalni broj oznaka „sviđa mi se“ (*like*) na videozapisu: 10.000
- Maksimalni broj oznaka „ne sviđa mi se“ (*dislike*) na videozapisu: 10.000
- Minimalna duljina videozapisa: 3 minute
- Maksimalna duljina videozapisa: 9 minuta

```

public static ArrayList<YouTubeVideo> getVideoList() throws SQLException, ClassNotFoundException {
    String myDriver = "com.mysql.cj.jdbc.Driver";
    String myUrl = "jdbc:mysql://localhost/youtube?serverTimezone=UTC";
    Class.forName(myDriver);
    Connection conn = DriverManager.getConnection(myUrl, "root", "root");

    ArrayList<YouTubeVideo> videoList = new ArrayList<YouTubeVideo>();
    String query = "SELECT DISTINCT t1.videoID, t2.durationInSeconds \r\n"
        + "FROM video AS t1 \r\n"
        + "NATURAL JOIN duration_in_seconds AS t2 \r\n"
        + "JOIN formats AS t3 ON t1.videoID = t3.videoID AND t3.format_note = '720p'\r\n"
        + "WHERE DIMENSION = '2d' AND definition = 'hd' AND projection = 'rectangular' AND uploadStatus = 'processed' AND privacyStatus = 'public' \r\n"
        + "AND viewCount > ? AND likeCount > ? AND dislikeCount < ? \r\n"
        + "AND durationInSeconds BETWEEN ? AND ?\r\n"
        + "AND publishedAt BETWEEN '2010-10-01' AND '2019-12-31'\r\n" + "LIMIT ?";

    PreparedStatement ps = conn.prepareStatement(query);
    ps.setInt(1, viewCount);
    ps.setInt(2, likeCount);
    ps.setInt(3, dislikeCount);
    ps.setInt(4, durationMin);
    ps.setInt(5, durationMax);
    ps.setInt(6, limit);

    ResultSet rs = ps.executeQuery();
    while (rs.next()) {
        YouTubeVideo video = new YouTubeVideo();
        video.setVideoID(rs.getString("videoID"));
        video.setDurationInSeconds(rs.getInt("durationInSeconds"));
        videoList.add(video);
    }
    rs.close();
    ps.close();
    return videoList;
}

```

Slika 4.5 - Isječak koda - dohvat videozapisa iz baze podataka

3. Iteracija po videozapisima i izvođenje procesa prikupljanja podataka i simuliranja korisničkih interakcija

Nakon dohvata ID-eva videozapisa, započinje proces prikupljanja podataka i simuliranja korisničkih interakcija za svaki YouTube videozapis. Prije nego li se otvori videozapis, određuje se koja će se interakcija izvoditi (3.c.) te u kojem trenutku videozapisa će se odabrana interakcija izvršiti. Kada je interakcija i vrijeme interakcije određeno, videozapis se otvara (3.a.) i kreće se u proces prikupljanja podataka, odnosno kopiranja Statistike za štrebere za svaku sekundu reprodukcije videozapisa (3.b.). Kopirani podaci spremaju se u tekstualnu datoteku čiji naziv odgovara ID-u videozapisa. U trenutku kada odabrano vrijeme interakcije bude jednako trenutnom vremenu reprodukcije videozapisa, interakcija se izvodi. Izvođenja interakcija opisana

su u koracima 3.d., 3.e. i 3.f. U slučaju kada je za interakciju odabrana opcija ne izvođenja interakcije, videozapis nastavlja s reprodukcijom. Skripta prepoznaje kada je videozapis završen te započinje s reproduciranjem sljedećeg videozapisa. Kada se ciklus izvrši za svaki ID videozapisa iz liste ID-eva, skripta završava.

a. Otvaranje YouTube videozapisa

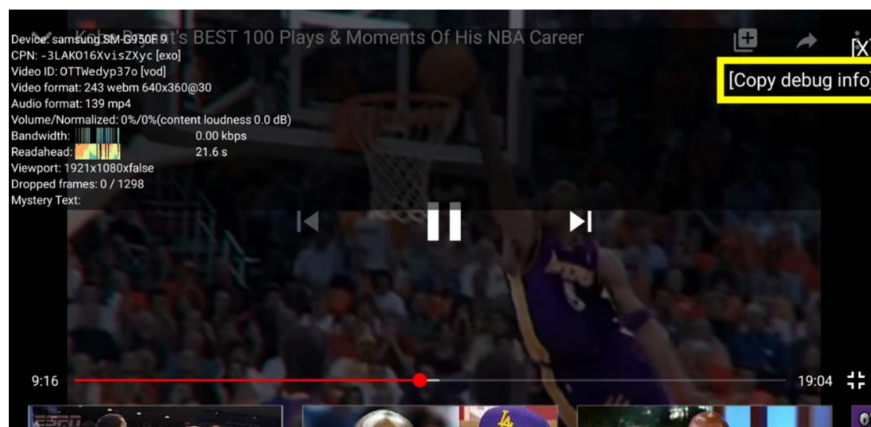
Otvaranje videozapisa u skripti vrši se putem *adb* konzolne naredbe (slika 4.6). Ovom naredbom se mobilna aplikacija YouTube otvara te se u njoj odmah učitava željeni videozapis (ID videozapisa koji se preda).

```
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);
    processBuilder.start();
}
```

Slika 4.6 - Isječak koda - otvaranje YouTube videozapisa

b. Kopiranje Statistike za štrebere

Za svaku sekundu videozapisa prikupljaju se podaci iz opcije Statistike za štrebere. Ako je opcija uključena, omogućen je gumb (slika 4.7 - označeno žutim okvirom) čijim se pritiskom kopiraju podaci u međuspremnik (engl. *clipboard*) u *JSON* formatu.



Slika 4.7 - Gumb za kopiranje Statistike za štrebere

JSON se pohranjuje u izlaznu datoteku s prikladajućom vremenskom oznakom. Slika 4.8 prikazuje programsku implementaciju pritiska gumba za kopiranje Statistike za štrebere.

```
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();
}
```

Slika 4.8 - Isječak koda - kopiranje Statistike za štrebere

c. Odabir interakcije

Za svaki videozapis na slučajan način odabire se koja će se interakcija izvoditi (slika 4.9). Vjerojatnost za odabir svake od interakcija je jednaka i iznosi 25%. Ako je odabrana interakcija pauziranja videozapisa, potrebno je odrediti koliko će dugo videozapis biti pauziran (slučajan broj iz intervala [10, 30], u sekundama). Ako je odabrana interakcija premotavanja videozapisa unaprijed, potrebno je odrediti koeficijent koji definira koliko će se videozapis vremenski premotati (slučajan broj iz intervala [0.4,0.7]). Svrha koeficijenta je detaljnije opisana u koraku 3.e.

```
public static Interaction chooseInteraction() {
    Interaction interaction = Interaction.values()[new Random().nextInt(Interaction.values().length)];

    switch (interaction) {
        case PAUSE_VIDEO:
            pauseDuration = pauseVideoDuration();
            break;
        case SEEK_FORWARD:
            seekCoef = getRandomNumberInRange(MIN_SEEK, MAX_SEEK) / 100.;
            break;
        default:
            break;
    }

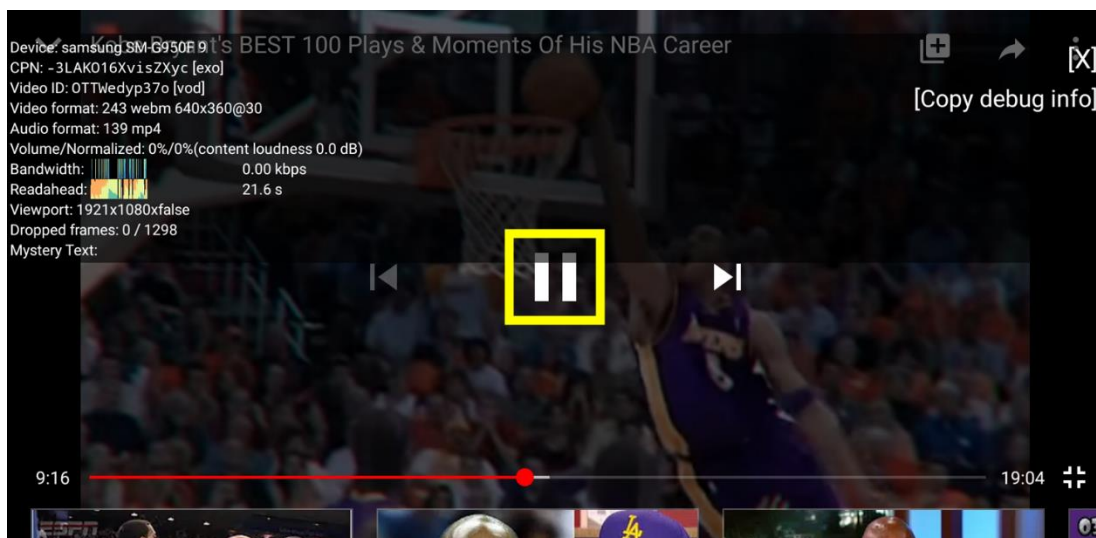
    return interaction;
}
```

Slika 4.9 - Isječak koda - odabir interakcije

d. Izvođenje interakcije pauziranja videozapisa

Kako bi se videozapis pauzirao prvo je potrebno pritisnuti po ekranu. Pritiskom ekrana pojavljuje se gumb za pauzu čijim se naknadnim

pritisakom videozapis pauzira (slika 4.10 - označeno žutim okvirom).



Slika 4.10 - Gumb za pauziranje videozapisa

Videozapis će biti pauziran onoliko dugo koliko je određeno prilikom odabira interakcije (3.c.). Za vrijeme trajanja ove interakcije prikuplja se Statistika za štrebere svake sekunde. Slika 4.11 prikazuje implementaciju opisanog procesa izvršavanja pauze.

```

public static void pauseVideoInteraction(AndroidDriver<AndroidElement> driver) throws 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);
    }

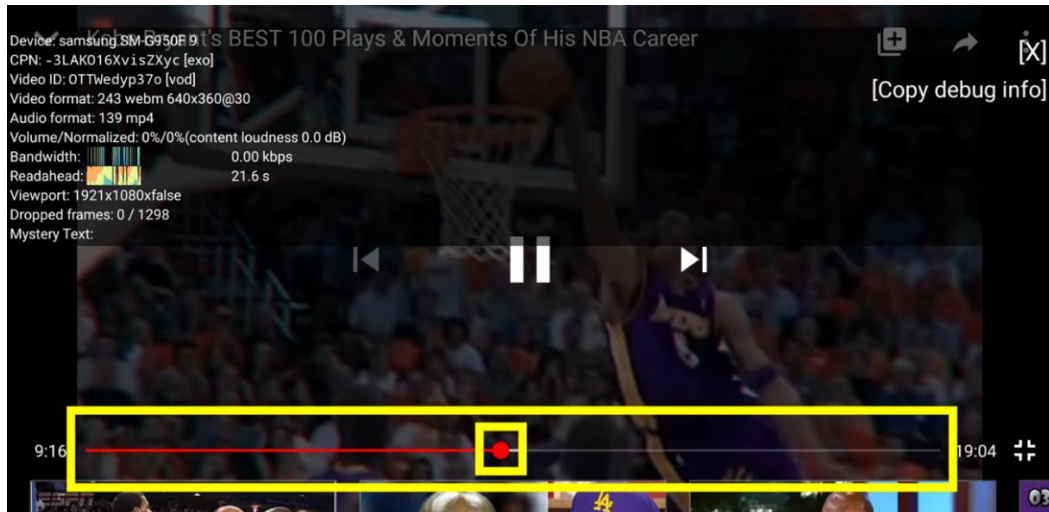
    // tap to resume video
    el2.click();
}

```

Slika 4.11 - Isječak koda - interakcija pauziranja videozapisa

e. Izvođenje interakcije premotavanja videozapisa unaprijed

Kako bi se videozapis premotao unaprijed potrebno je prvo pritisnuti ekran. Pritiskom ekrana pojavljuje se reproduksijska lenta i na njoj točka koje zajedno predstavljaju trenutak u kojem se videozapis nalazi (slika 4.12 – označeno žutim okvirom).



Slika 4.12 – Reprodukcijska lentu videozapisa

Naknadnim pritiskom po lenti reprodukcije videozapis se premotava u željeno vrijeme. Premotavanje će se vršiti samo unaprijed tako da je potrebno pritisnuti lentu desno od trenutne pozicije točke kako bi se videozapis premotao. Koeficijent koji je definiran prilikom odabira interakcije određuje odmak premotavanja. Množenjem koeficijenta s nereproduciranim trajanjem videozapisa te zbrajanjem s vremenom zbivanja interakcije dobiva se trenutak u koji će se video premotati. Slika 4.13 prikazuje isječak koda implementacije opisane interakcije.

```

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

    System.out.println("SEEK FORWARD");
    // tap on the video
    WebElement e11 = (WebElement) driver.findElementByAccessibilityId("Video player");
    e11.click();

    copyDebugInfo(driver);

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

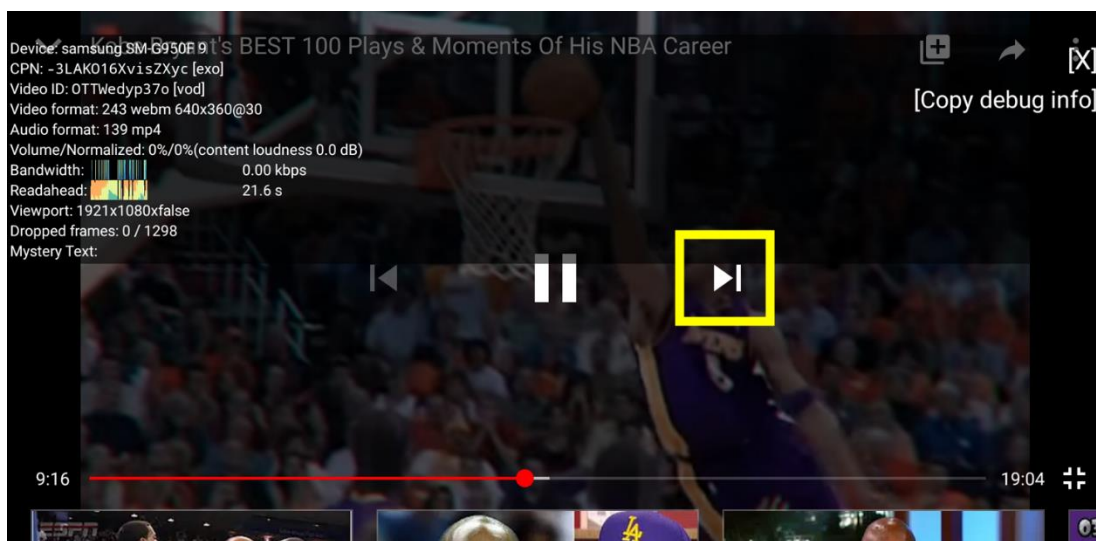
public static int seekTo(int videoDuration, int interactionTime) {
    double seekSecond = interactionTime + ((videoDuration - interactionTime) * seekCoef);
    return (int) (seekSecond / videoDuration * (X_MAX - X_MIN) + X_MIN);
}

```

Slika 4.13 - Isječak koda - premotavanje videozapisa unaprijed

f. Izvođenje interakcije prekida reproduciranja i prebacivanja na idući videozapis

Da bi se videozapis prebacio na idući, potrebno je pritisnuti po ekranu kako bi se pojavila opcija prebacivanja na sljedeći videozapis (slika 4.14 – označeno žutim okvirom).



Slika 4.14 - Gumb za prebacivanje videozapisa na sljedeći videozapis

Pritiskom na gumb prebacivanja, učitava se sljedeći videozapis koji se predlaže od strane YouTube-a. Kako za taj videozapis ne postoje informacije koje su potrebne u svrhu prikupljanja podataka, programski se učitava (3.a.) sljedeći videozapis čiji ID je sljedeći na redu iz liste ID-eva. Slika 4.15 prikazuje implementaciju prebacivanja na sljedeći videozapis.

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

    System.out.println("SKIP VIDEO");
    // tap on the video
    WebElement e11 = (WebElement) driver.findElementByAccessibilityId("Video player");
    e11.click();

    copyDebugInfo(driver);

    // tap to play next video
    WebElement e12 = (WebElement) driver.findElementByAccessibilityId("Next video");
    e12.click();
    interactionTimestamp = System.currentTimeMillis();
}
```

Slika 4.15 - Isječak koda - prebacivanje na sljedeći videozapis

4.2.1. Radni okvir Appium

Appium je alat otvorenog koda za automatizaciju nativnih, mobilnih *web* i hibridnih aplikacija za iOS mobilne uređaje, Android mobilne uređaje te Windows *desktop* platformu. Alat je „*cross-platform*“ što znači da se isti API koristi za pisanje testova za različite platforme. Ovime se omogućuje da se testni slučajevi napisani za testiranje aplikacije na jednoj platformi iskoriste za testiranje aplikacije napisane na drugoj platformi [29].

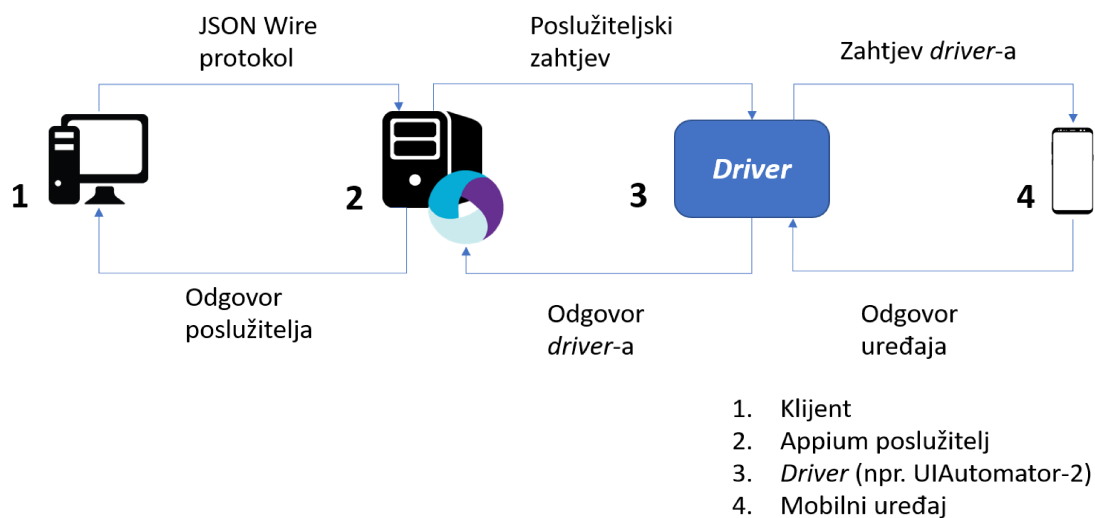
Appium je osmišljen kako bi zadovoljio potrebe mobilne automatizacije bazirajući se na četiri načela [29]:

- 1.) Ne bi se smjelo rekompajlirati (engl. *recompile*) aplikaciju ili uređivati kako bi ju se automatiziralo,
- 2.) Pisanje i izvođenje testova ne bi smjelo biti ograničeno na određeni jezik ili radni okvir (engl. *framework*),
- 3.) Kada je u pitanju automatizacija API-ja, radni okvir za mobilnu automatizaciju se ne bi trebao ponovno izrađivati,
- 4.) Radni okvir bi trebao biti otvorenog koda.

Appium je u svojoj srži zasnovan na arhitekturi klijent-poslužitelj (engl. *client-server architecture*). On se ponaša kao poslužitelj koji sadrži svoj API. Implementacija poslužitelja napisana je u programskom jeziku Node.js. Appium prima konekcije od klijenata, sluša naredbe, izvršava te naredbe na

mobilnom uređaju te odgovara s *HTTP* odgovorom koji predstavlja rezultat izvođenja naredbe. Kako je ovaj alat zasnovan na arhitekturi klijent-poslužitelj, on sam po sebi nudi bezbroj mogućnosti. Testovi se mogu pisati u bilo kojem jeziku koji podržava opciju *HTTP* klijenta u sebi. Appium nudi već gotovu implementaciju *WebDriver* klijenta u sklopu knjižnice (engl. *library*) za šest različitih programskih jezika: Java, Ruby, Python, PHP, JavaScript i C#. Prema preporuci iz dokumentacije, stabilnije je i bolje koristiti Appium knjižnicu, nego samostalno implementirati *WebDriver* klijenta [29].

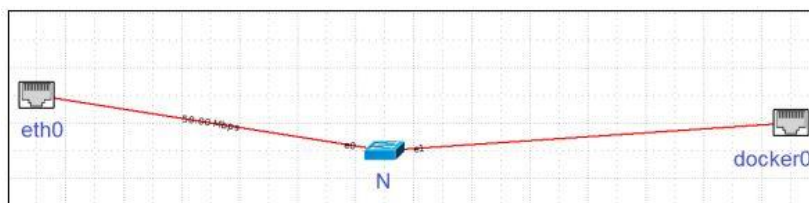
Slika 4.16 prikazuje proces izvršavanja zahtjeva pomoću Appium poslužitelja i gotove implementacije *WebDriver*-a. Klijent šalje upit na Appium poslužitelj putem *JSONWire* protokla. Ovaj protokol koristi *HTTP* zahtjeve koji u sebi sadrže *JSON* objekte. Appium poslužitelj pomoću određenog *drivera* komunicira s mobilnim uređajem te izvršava naredbe koje je zaprimio od klijenta. Odgovor se s mobilnog uređaja preko *driver-a* šalje na poslužitelj koji na kraju vraća odgovor klijentu.



Slika 4.16 - Appium arhitektura [30]

4.3. Prikupljanje podataka s mrežne razine

Prikupljanje mrežnog prometa odvija se za vrijeme izvršavanja skripte za prikupljanje podataka s aplikacijske razine. Prije svega, potrebno je postaviti mrežnu topologiju u alatu IMUNES na računalo br.1 (slika 4.17). Pomoću topologije određuje se smjer prosljeđivanja mrežnog prometa. Promet se s računala br.1. prosljeđuje do uređaja Net.Shark koji promet šalje dalje do pristupne točke. Osim što prosljeđuje promet, Net.Shark ga i kopira te šalje na računalo br.2, na kojem se izvršava prikupljanje mrežnog prometa.



Slika 4.17 - Topologija u IMUNES-u

Prilikom prikupljanja podataka uvjeti u mreži su kontrolirani pomoću skripte napisane u programskom jeziku Python koja manipulira raspoloživim prijenosnim pojasom [31][32]. Skripte simuliraju 3G i 4G mobilnu mrežu na način da dinamički ograničavaju propusnost prijenosnog pojasa prethodno definiranog u alatu IMUNES. Osim 3G i 4G mrežnih uvjeta, simulirana je jedna od danas najpopularnijih opcija dostupnih kod telekom operatera, *stream-on* opcija [33]. Opcija nudi besplatan pristup sadržaja uz pristanak na ograničenje prijenosnog pojasa prilikom pristupa istom na 1.5 Mbit/s.

Mrežni promet se prikuplja pomoću alata tcpdump na računalo br.2. Alat prikuplja i sprema podatke u izlaznu datoteku oblika *.pcap*. Kasnijom obradom prikupljenih podataka dobiva se skup podataka s mrežne razine koji će zajedno s obrađenim podacima s aplikacijske razine predstavljati skup podataka za učenje modela strojnog učenja.

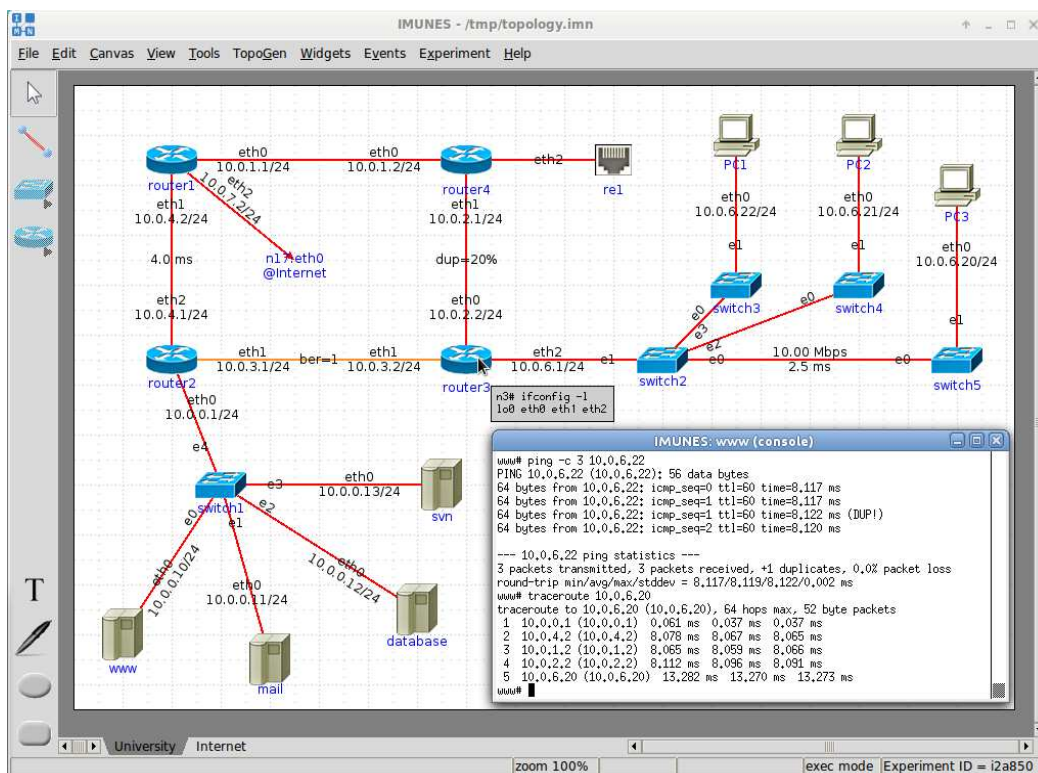
4.3.1. Alat tcpdump

Tcpdump je besplatan „*command-line*“ alat čija je glavna svrha pregled i prikupljanje internetskih paketa koji prolaze mrežom na koju je računalo

priključeno. Alat je besplatan i dolazi pod licencom naziva *The 3-Clause BSD License* [34]. Dostupan je za gotovo sve operacijske sustave koji su bazirani na Linux-u/Unix-u [35]. Alat omogućuje korisniku ispis TCP/IP i drugih paketa koji su primljeni ili poslani u njegovoj mreži u izlazni dokument formata *.pcap*. Ovakva vrsta dokumenta može se pregledati putem *tcpdump* naredbe ili *Wireshark*-a, programskog alata za analizu mrežnog protokola [36].

4.3.2. IMUNES

IMUNES (engl. *Integrated Multiprotocol Network Emulator/Simulator*) je alat za emuliranje i simuliranje IP mrežnih topologija [37]. Ovaj besplatan alat otvorenog koda nudi bogato interaktivno grafičko sučelje za izgradnju mrežnih topologija (slika 4.18). Trenutno je dostupan za Linux i FreeBSD operacijske sustave [37]. Jedna od mogućnosti koje nudi ovaj alat su postavke ograničenja na propusnost mreže (engl. *bandwidth*). Osim fiksnog postavljanja propusnosti, moguće je i napisati skripte koje dinamički mijenjaju propusnost mreže.



Slika 4.18 - Prikaz grafičkog sučelja IMUNES-a [37]

4.4. Obrada prikupljenih podataka

Prikupljene podatke potrebno je obraditi na način da se njihovom obradom dobije skup podataka spreman za učenje modela strojnog učenja. Odjeljak 4.4.1. opisuje obradu podataka dobivenih s aplikacijske razine, a odjeljak 4.4.2. opisuje obradu podataka dobivenih s mrežne razine. Obradeni podaci se naknadno spajaju pomoću vremenskih oznaka u jedan skup podataka.

4.4.1. Obrada prikupljenih podataka s aplikacijske razine

Podaci prikupljeni s aplikacijske razine spremaju se u tekstualnu datoteku od kojih svaka predstavlja jedan reproducirani videozapis. Kako je opisano u koraku 3.b potpoglavlja 4.2., u datoteku se za svaku sekundu videozapisa s pripadajućom vremenskom oznakom sprema Statistika za štrebere u obliku *JSON* formata (slika 4.19).

```
1587636413589
{"cplatform":"mobile","cbr":"com.google.android.youtube","c":"android","cplayer":"ANDROID_EXOPLAYER",
"cmode1":"SM-G950F","cos":"Android","csdk":"28","cbrver":"14.49.51","cver":"14.49.51","cosver":"9",
cbrand":"samsung","videoid":"0AwBokwCGuY","cpn":"0f7T7ylfoqUGqih9","fmt":"248 wehm 1920x1080@30","afmt":"140 mp4",
"bh":2733,"conn":3,"volume":0,"loudness":"-5.472","bat":"0.330:1","df":"0\4","time":"2020-04-23T10:06:57.080Z",
"glmode":"RECTANGULAR_2D","mtext":"","error":"No errors","innertube.build.changelist":"307653843",
"innertube.build.experiments.source_version":"307990178","innertube.build.label":"youtube.ytfe.innertube_20200421_6_RCO",
"innertube.build.timestamp":"1587496363","innertube.build.variants.checksum":"89732145b3768973ea7931dd4f4f3050","a":"11211411,
11211694,11213630,11216291,11218995,11219182,11219238,11219600,11219647,11220353,23735347,23744176,23803853,23813225,23817290,
23837040,23837993,23839552,23839597,23844961,23850473,23856510,23856510,23856950,23857949,23859129,23859802,23860859,23866661,
23869401,23873928,23875532,23876129,23876178,23876458,23876644,23876669,23877025,23878653,23880389,23880620,23882034,23882502,
23885294,23885572,23886825,23887548,23888075,23889151,23892927,23895290,23895446,23895671,23896333,23896333,23898053,39786617,
39787779,39788273,39788645,39789494,39790256,39790282,45170005,9449243","logged_in":"1"}
```

Slika 4.19 - Primjer zapisa jedne sekunde videozapisa

U svrhu obrade navedenih podataka napisana je skripta u programskom jeziku Java. Skripta obrađuje svaku tekstualnu datoteku na način da generira ključne pokazatelje uspješnosti (*KPI*-eve) na temelju *JSON*-a za svaku vremensku oznaku. U nastavku su navedeni generirani *KPI*-evi:

- **Zastajkivanje** – je li u toj sekundi bio zastoj videozapisa

Moguće vrijednosti:

- *TRUE*
- *FALSE*

- **Prosječna brzina kodiranja trenutne sekunde videozapisa**

Moguće vrijednosti:

- *HIGH* (≥ 1500 Kbps)
- *LOW* (< 1500 Kbps)

- **Rezolucija trenutne sekunde videozapisa**

Moguće vrijednosti:

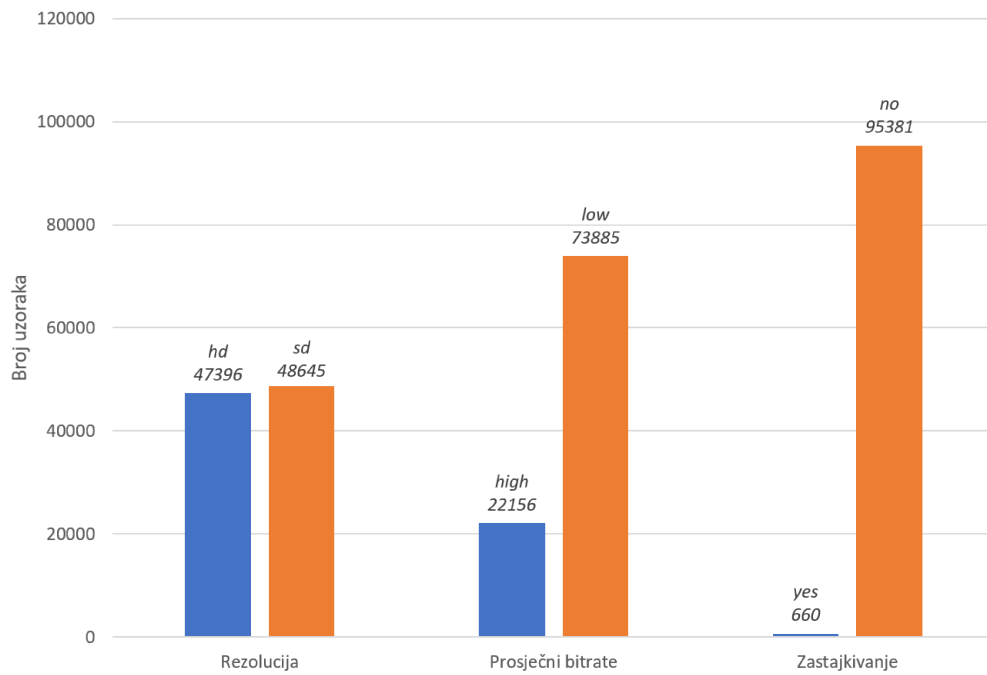
- *HIGH* ($\geq 720p$)
- *LOW* ($< 720p$)

Osim *KPI*-eva, za svaku sekundu videozapisa ispisuje se je li u tom trenutku napravljena interakcija. Obradeni podaci se zajedno s pripadnim vremenskim oznakama i ID-em videozapisa spremaju u izlaznu datoteku oblika .csv (slika 4.20).

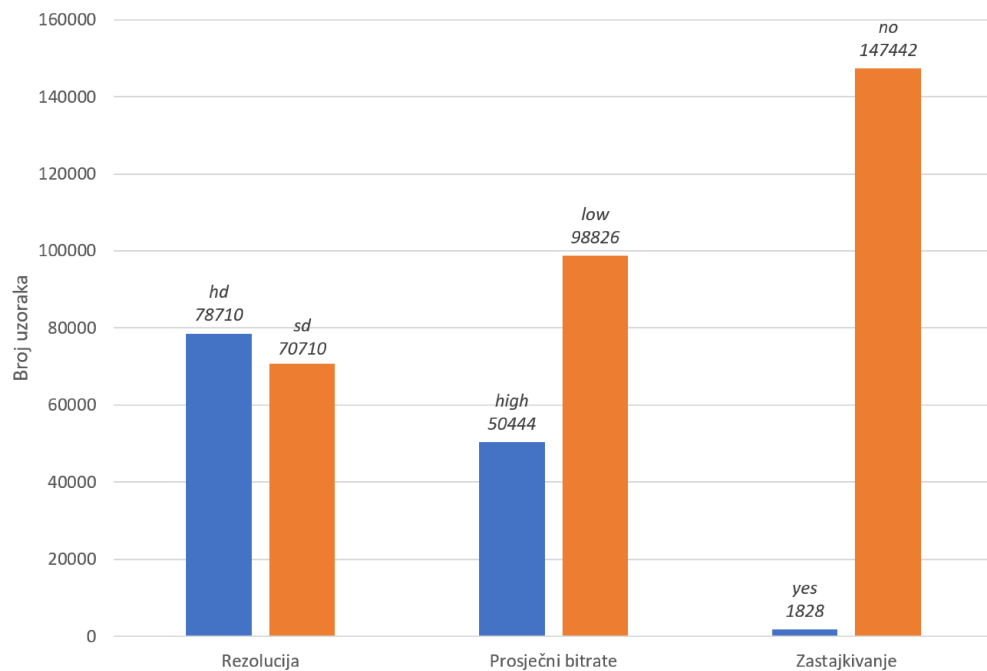
videoID	timestamp	stalling	avgBitrate	resolution	isInteraction	interactionName
fs0SyW980YI	1588750783	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750784	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750787	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750788	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750789	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750790	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750791	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750792	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750793	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750794	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750795	FALSE	LOW	LOW	FALSE	null
fs0SyW980YI	1588750796	FALSE	LOW	LOW	FALSE	null

Slika 4.20 - Obradeni podaci s aplikacijske razine

U nastavku je prikazana statistika obrađenih podataka za svaku sekundu videozapisa. Slika 4.21 prikazuje broj uzoraka u pojedinoj klasi za obrađene podatke videozapisa u kojima se nije pojavila interakcija, dok slika 4.22 isto prikazuje za obrađene podatke videozapisa u kojima se pojavila interakcija pauze, premotavanja unaprijed ili prekida videozapisa i prebacivanja na sljedeći.



Slika 4.21 – Broj uzoraka videozapisa u pojedinoj klasi za obrađene podatke videozapisa bez interakcija



Slika 4.22 - Broj uzoraka videozapisa u pojedinoj klasi za obrađene podatke videozapisa s interakcijama

4.4.2. Obrada prikupljenih podataka s mrežne razine

Podaci prikupljeni s mrežne razine pohranjeni su u datoteku oblika *.pcap*. Takvu datoteku najprije je potrebno obraditi na način da se iz nje izvuku podaci potrebni za izračun ulaznih parametara za učenje modela strojnog učenja.

Za obradu *.pcap* datoteke koristi se alat Tshark [38]. Tshark je mrežni analizator protokola koji omogućuje snimanje paketa podataka u mreži ili čitanje paketa iz prethodno sačuvane *.pcap* datoteke. Izvršavanjem konzolne naredbe

```
tshark -r XY.pcap -T fields -e frame.time_epoch -e ip.src -e ip.dst -e udp.srcport -e udp.dstport -e tcp.srcport -e tcp.dstport -e frame.cap_len -E header=, -E separator=, -E quote=d -E occurrence=f > traffic.csv
```

dobivaju se željeni podaci mrežnog prometa koji u sebi uključuju vremensku oznaku, *IP* adresu, portove protokola *TCP* i *UDP* te veličinu paketa koji se prenosi u toj sekundi. Podaci se spremaju u datoteku oblika *.csv* koju je dalje potrebno obraditi (slika 4.23).

frame.time_epoch	ip.src	ip.dst	udp.srcport	udp.dstport	tcp.srcport	tcp.dstport	frame.cap_len
1588750433	74.125.99.168	10.19.128.130			443	44074	1462
1588750433	10.19.128.130	74.125.99.168			44074	443	66
1588750433	10.19.128.130	31.13.84.4			57372	443	66
1588750433	10.19.128.130	31.13.84.4			57372	443	66
1588750433	74.125.99.168	10.19.128.130			443	44074	1462
1588750433	10.19.128.130	74.125.99.168			44074	443	66
1588750433	10.19.128.130	31.13.84.4			57372	443	66
1588750433	74.125.99.168	10.19.128.130			443	44074	1462
1588750433	10.19.128.130	74.125.99.168			44074	443	66
1588750433	74.125.99.168	10.19.128.130			443	44074	1462
1588750433	10.19.128.130	74.125.99.168			44074	443	66
1588750433	74.125.99.168	10.19.128.130			443	44074	1462
1588750433	10.19.128.130	74.125.99.168			44074	443	78

Slika 4.23 - Obradeni podaci iz *.pcap* datoteke

Dobivene podatke naknadno je potrebno obraditi na način da se matematičkim izračunima dobiju željeni ulazni parametri za učenje modela strojnog učenja. U svrhu obrade korištena je skripta napisana u programskom jeziku Python [39]. Skripta na temelju izvorišnih i odredišnih *IP* adresa te veličina pojedinih paketa u prethodnih 1, 2, 3, 5, 10 i 20 sekundi izračunava statistiku za odlazni (engl. *uplink*, *ul*) i dolazni (engl. *downlink*, *dl*) promet.

Tablica 4.1 sadrži sažeti popis svih značajki za svaki od prozora, te za odlazni i dolazni promet. Konačni format naziva značajke dobiva se dodavanjem oznake za odlazni ili dolazni promet i oznake veličine prozora (npr. *ul_pckt_count_w5*).

Tablica 4.1 - Popis mrežnih značajki koje će se koristiti za klasifikaciju *KPI*-eva [39]

Značajka	Opis značajke
<i>pckt_count</i>	Broj paketa.
<i>pckt_count_gt100</i>	Broj paketa čija je veličina veća od 100B.
<i>throughput</i>	Prosječna propusnost.
<i>active_time</i>	Postotak prozora koji se koriste za prijenos.
<i>mean_pckt_size_gt100</i>	Srednja vrijednost veličine paketa u prozoru čija je veličina veća od 100B.
<i>median_pckt_size_gt100</i>	Medijan veličine paketa u prozoru čija je veličina veća od 100B.
<i>max_pckt_size_gt100</i>	Maksimalna vrijednost veličine paketa u prozoru čija je veličina veća od 100B.
<i>min_pckt_size_gt100</i>	Minimalna vrijednost veličine paketa u prozoru čija je veličina veća od 100B.
<i>stdev_pckt_size_gt100</i>	Standardna devijacija veličine paketa u prozoru čija je veličina veća od 100B.
<i>mean_pckt_size</i>	Srednja vrijednost veličine paketa u prozoru.
<i>median_pckt_size</i>	Medijan veličine paketa u prozoru.
<i>max_pckt_size</i>	Maksimalna vrijednost veličine paketa u prozoru.
<i>min_pckt_size</i>	Minimalna vrijednost veličine paketa u prozoru.
<i>stdev_pckt_size</i>	Standardna devijacija veličine paketa u prozoru.
<i>mean_iat</i>	Srednja vrijednost vremenskog intervala u prozoru.
<i>median_iat</i>	Medijan vremenskog intervala u prozoru.
<i>max_iat</i>	Maksimalna vrijednost vremenskog intervala u prozoru.
<i>min_iat</i>	Minimalna vrijednost vremenskog intervala u prozoru.
<i>stdev_iat</i>	Standardna devijacija vremenskog intervala u prozoru.

5. Izgradnja modela strojnog učenja i analiza rezultata

U ovom poglavlju opisuje se proces izgradnje modela za predviđanje *KPI*-eva te njihova analiza. Proces obuhvaća pripremu ulaznih podataka, odabir značajki, izgradnju i testiranje modela. Nakon što se model izgradi, radi se ponovna procjena (engl. *re-evaluate*) modela na podacima videozapisa u kojima se pojavljuju interakcije. Odabir značajki vrši se odabirom iz skupa svih 228 značajki koje daju najbolju predikciju za model. U svrhu odabira značajki koristi se metoda pretrage *BestFirst* [40], a kao procijenitelj značajki *wrapper* metoda [41].

Prije svega, potrebno je dokazati da postoji utjecaj korisničkih interakcija na predviđanje *KPI*-eva (rezolucija, prosječna brzina kodiranja i zastajkivanje) što je opisano u potpoglavlju 5.2., a nakon toga je potrebno odrediti postoje li značajke koje ne ovise o utjecaju korisničkih interakcija te za njih izgraditi model strojnog učenja što je opisano u potpoglavlju 5.3. U zadnjem potpoglavlju detaljno su analizirani rezultati dobiveni u potpoglavljima 5.2. i 5.3. te se daje konačni zaključak o postojanju značajki koje nisu ovisne o korisničkim interakcijama. Potpoglavlje 5.1. opisuje strojno učenje, tehniku i algoritam učenja modela koji će se koristiti u ovome radu, vrednovanje modela strojnog učenja te alat WEKA-u.

5.1. Strojno učenje

Strojno učenje (engl. *machine learning, ML*) je grana umjetne inteligencije i dio računalne znanosti koji se bavi oblikovanjem algoritama koji optimiziraju svoju učinkovitost na temelju empirijskih podataka. Modeli strojnog učenja definirani su pomoću parametara, a riječ učenje označava izvršavanje računalnog programa koji optimizira parametre modela koristeći podatkovne primjere ili iskustvo iz prošlosti. Postoje tri vrste modela [42]:

- 1.) Prediktivni – predviđanje budućnosti,
- 2.) Deskriptivni – znanje o podacima,

3.) Prediktivni i deskriptivni.

U većini slučajeva grade se modeli koji pokrivaju definiciju i prediktivnog i deskriptivnog modela. Najprije se želi dobiti nekakvo znanje o podacima, a nakon toga se želi stvoriti model koji na temelju istih podataka predviđa svojstva novih, još neviđenih podataka.

Strojno učenje koristi teoriju statistike u izgradnji matematičkih modela čija je glavna svrha donositi zaključke iz podataka. Kao takva, računalna znanost ima dvostruku ulogu [42]:

- 1.) U treniranju podataka potrebni su učinkoviti algoritmi za rješavanje problema optimizacije, kao i za pohranu i procesiranje velike količine stečenih podataka,
- 2.) Jednom kada je model naučen, njegovo reprezentativno i algoritamsko rješenje u zaključivanju mora biti učinkovito.

U današnjem svijetu strojno učenje ima široku primjenu: od teških problema koji zahtijevaju raspoznavanje uzoraka i obradu prirodnog govora preko dinamičkih sustava kao što je robotika pa sve do obrade ogromnih količina podataka i izvlačenja znanja iz njih. Baš zbog svoje specifičnosti i primjena, strojno učenje se dijeli na tri vrste [42]:

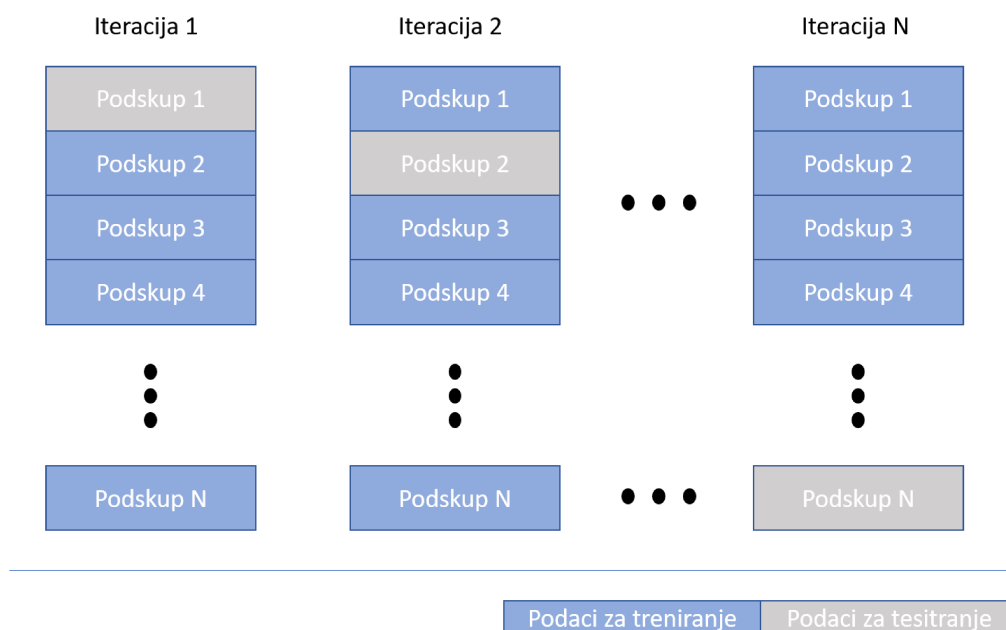
- 1.) Nadzirano učenje (engl. *supervised learning*) – Cilj je pronaći $y=f(x)$ za parove ulaznih podataka (x,y) . Ako je y kontinuirana varijabla govori se o regresiji, a ako je diskretna govori se o klasifikaciji,
- 2.) Nenadzirano učenje (engl. *unsupervised learning*) – Cilj je pronaći pravilnosti u podacima koji nemaju zadane ciljne vrijednosti. Dijeli se na grupiranje (engl. *clustering*), procjenu gustoće (engl. *density estimation*) i smanjenje dimenzionalnosti (engl. *dimensionality reduction*),
- 3.) Podržano/ojačano učenje (engl. *reinforcement learning*) - Učenje optimalne strategije na temelju pokušaja s odgođenom nagradom.

5.1.1. Tehnika i algoritam učenja modela

U ovom odjeljku definirani su algoritam učenja za treniranje modela i tehnika učenja prilikom treniranja modela. Za učenje modela u ovome radu odabran je Random Forest algoritam, dok će se za treniranje modela koristiti tehnika k-struke unakrsne validacije. U nastavku su opisani tehnika i algoritam učenja.

K-struka unakrsna validacija (engl. *k-fold cross-validation*) je tehnika učenja koja dijeli skup podataka na k jednakih podskupova. Nakon što je skup podataka podijeljen, tehnika ponavlja sljedeći proces k puta za svaki podskup (slika 5.1):

- odabrani podskup koristi za testiranje,
- ostalih k-1 podskupova koristi za učenje.



Slika 5.1 - K-struka unakrsna validacija

Random Forest (RF) je klasifikacijski algoritam nadziranog učenja [43]. Kao što i samo ime kaže, cilj algoritma je izgraditi veliki broj nasumičnih stabla odluke koji se ponašaju kao cijelina. Svako stablo odluke u RF-u predviđa klasu te klasa s najvećim brojem glasova postaje predikcija modela. Postoji

direktna poveznica između broja stabala koje RF koristi i uspješnosti rezultata: što je broj stabala veći to je rezultat uspješniji. U nastavku je naveden pseudokod algoritma [43]:

```
Precondition: A training set  $S := (x_1, y_1), \dots, (x_n, y_n)$ , features  $F$ , and number of trees in forest  $B$ .  
function RandomForest( $S, F$ )  
   $H \leftarrow \emptyset$   
  for  $i \in 1, \dots, B$  do  
     $S(i) \leftarrow$  A bootstrap sample from  $S$   
     $h_i \leftarrow$  RandomizedTreeLearn( $S(i), F$ )  
     $H \leftarrow H \cup \{h_i\}$   
  end for  
  return  $H$   
end function  
function RandomizedTreeLearn( $S, F$ )  
  At each node:  
     $f \leftarrow$  very small subset of  $F$   
    Split on best feature in  $f$   
  return The learned tree  
end function
```

5.1.2. Vrednovanje uspješnosti naučenog modela

Naučene modele strojnog učenja potrebno je ocijeniti i vrednovati na način da se odredi razina uspješnosti. Za vrednovanje modela će se u ovom radu koristiti mjere dobivene izračunom iz matrice zabune.

Matrica zabune (engl. *confusion matrix*) je matrica čiji su redci i stupci vrijednosti klasa koje se predviđaju. Redak predstavlja klasu koju je model predvidio, a stupac koja je stvarna vrijednost predviđene klase. Matrica se sastoji od četiri kategorije:

- *True positive (TP)* vrijednosti,
- *False positive (FP)* vrijednosti,
- *True negative (TN)* vrijednosti,
- *False negative (FN)* vrijednosti.

Matrica zabune može biti višedimenzionalna, a to je slučaj kada vrijednosti klasa koje se predviđaju mogu poprimiti više od dvije vrijednosti. Slika 5.2 prikazuje dvodimenzionalnu matricu zabune.

		Stvarne vrijednosti	
		1	0
Predviđene vrijednosti	1	TP	FP
	0	FN	TN

Slika 5.2 - Matrica zabune

Iz matrice zabune mogu se izračunati mjere koje pokazuju uspješnost klasifikacijskog modela. U ovom radu će se koristiti sljedeće mjere:

- 1.) **Točnost** (engl. *accuracy*) – postotak točno klasificiranih primjera u odnosu na sve primjere. Formula za točnost je:

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

- 2.) **Preciznost** (engl. *precision*) - postotak pozitivno klasificiranih primjera u odnosu na ukupan broj pozitivno klasificiranih primjera. Formula za preciznost je:

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

- 3.) **Odziv** (engl. *recall*) – postotak pozitivno klasificiranih primjera u odnosu na ukupan broj stvarno pozitivnih primjera. Formula za odziv je:

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

Osim navedenih mjera postoje i druge mjere kao što su specifičnost, *fall-out*, mjera *F1* i mjera *F-beta*, no one se neće koristiti u ovome radu u svrhu vrednovanja uspješnosti modela.

5.1.3. WEKA

WEKA (engl. *Waikato Environment for Knowledge Analysis*) je program otvorenog koda napisan u programskom jeziku Java koji sadrži skup algoritama strojnog učenja i alata za obradu podataka [44]. Razvijena je na Sveučilištu Waikato na Novom Zelandu s namjerom da na brz i pouzdan način osigura isprobavanje postojećih metoda i algoritama na novim skupovima podataka. WEKA omogućuje podršku za cijeli proces eksperimentalnog rudarenja podataka (engl. *data mining*) [45]:

- Pripremanje ulaznih podataka,
- Učenje modela,
- Vizualizacija ulaznih podataka i rezultata učenja.

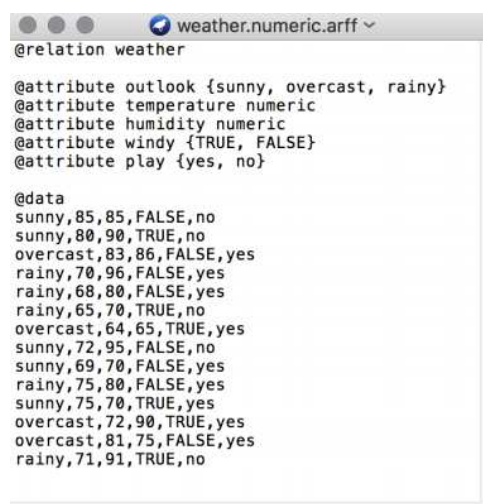
WEKA se može koristiti putem gotovog grafičkog sučelja, terminala ili putem Java *API*-ja. Pokreće se na gotovo svakoj platformi i testirana je na Linux, Windows i Macintosh operacijskim sustavima [45]. Postoji nekoliko načina korištenja ovog programa:

1. Jedan način korištenja WEKA-e je taj da se metode učenja primjenjuju na skupovima podataka i da se njihov rezultat analizira u svrhu stvaranja znanja o podacima,
2. Drugi način je taj da se naučeni modeli koriste u svrhu predviđanja novih, prethodno neviđenih podataka,
3. Te treći način je taj da se primjeni nekoliko različitih metoda učenja te uspoređuje njihova učinkovitost i svojstva kako bi se odabrala jedna za predviđanje.

Ulazni skupovi podataka su najčešće prezentirani u tablicama i bazama podataka. Međutim, WEKA za spremanje podataka koristi ARFF format. Ovaj

format zapisa podataka se sastoji od 3 dijela (slika 5.3). Naslov skupa podataka,

- 1.) Nazivi atributa s vrijednostima koje mogu poprimiti,
- 2.) Podaci koji su odvojeni zarezom.



```
weather.numeric.arff
@relation weather
@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

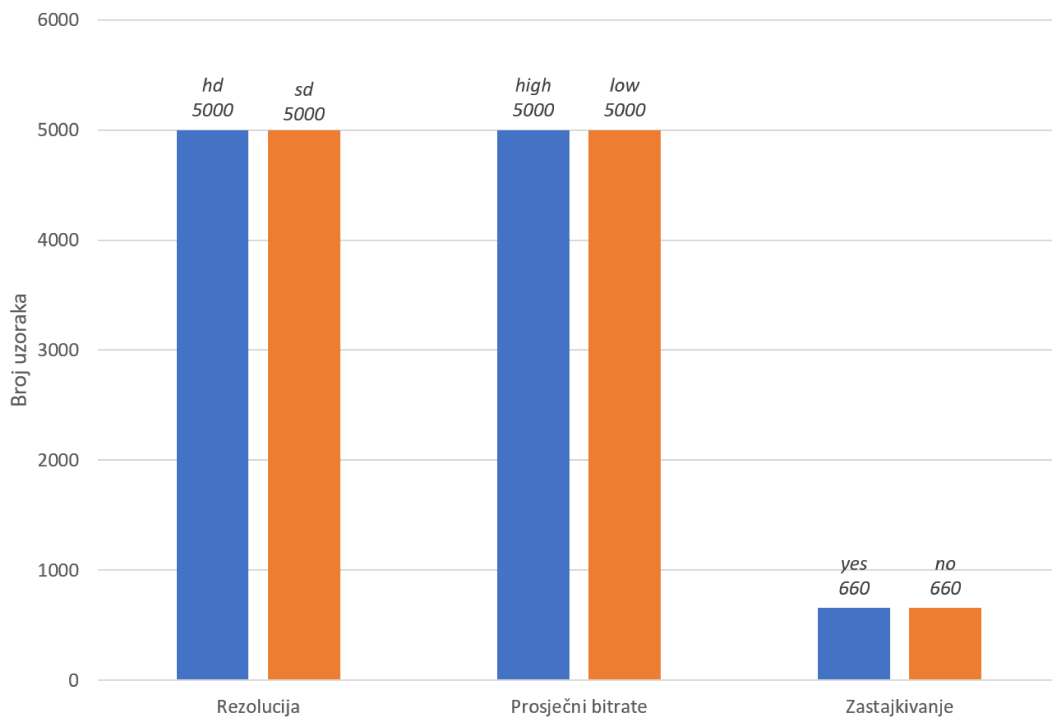
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

Slika 5.3 – Datoteka ARFF formata [45]

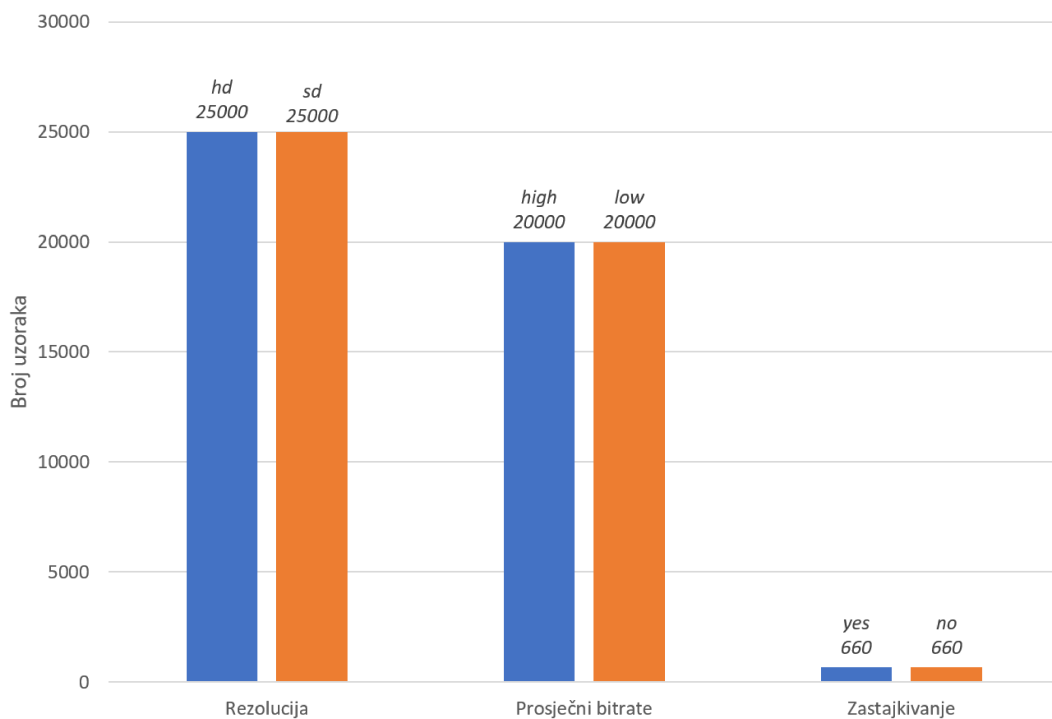
Većina programa za čuvanje podataka omogućuje izvoz (engl. *export*) podataka u datoteku formata *.csv*. WEKA iz navedenog formata omogućuje transformaciju u *.arff* format datoteke čime je ulazni skup podataka spreman za učitavanje u program.

5.2. Izgradnja modela pomoću značajki odabranih na skupu podataka bez interakcija

Prije svega, potrebno je dokazati da korisničke interakcije imaju utjecaj na predviđanje *KPI*-eva. U tu svrhu odabir značajki, treniranje i testiranje modela vrši se na podacima videozapisa bez interakcija. Nakon toga, model će se ponovno procijeniti na podacima videozapisa s interakcijama kako bi se vidio utjecaj korisničkih interakcija na predikciju modela. Skup podataka u ovisnosti o vrijednostima klasa koji će se koristiti za odabir značajki prikazan je na slici 5.4, a skup podataka za treniranje na slici 5.5.



Slika 5.4 - Skup podataka videozapisa bez interakcija za odabir značajki



Slika 5.5 - Skup podataka videozapisa bez interakcija za treniranje modela

Tablica 5.1 prikazuje odabrane značajke za svaki *KPI* koji se predviđa te preciznost, točnost i odziv za model izgrađen na podacima videozapisa bez interakcija.

Tablica 5.1 - Tablica prikaza performansi modela izgrađenog na podacima videozapisa bez interakcija pomoću značajki odabranih na skupu podataka bez interakcija

<i>KPI</i>	Odabrane značajke	Točnost	Preciznost	Odziv
Rezolucija <i>sd / hd</i>	<i>ul_pkt_count_w5,</i> <i>dl_throughput_w20,</i> <i>dl_max_pkt_size_w20,</i> <i>dl_median_iat_w20,</i> <i>dl_max_iat_w20,</i> <i>ul_median_pkt_size_w20,</i> <i>ul_min_pkt_size_w20,</i> <i>ul_max_iat_w20,</i> <i>ul_min_iat_w20</i>	0.95508	<i>sd: 0.943,</i> <i>hd: 0.969</i>	<i>sd: 0.970,</i> <i>hd: 0.940</i>
Prosječna brzina kodiranja <i>low / high</i>	<i>dl_max_pkt_size_w10,</i> <i>dl_min_pkt_size_w10,</i> <i>dl_active_time_w20,</i> <i>dl_mean_pkt_size_w20,</i> <i>dl_max_pkt_size_w20,</i> <i>dl_median_iat_w20,</i> <i>ul_median_iat_w20,</i> <i>ul_max_pkt_size_w20,</i> <i>ul_median_iat_w20,</i> <i>ul_max_iat_w20,</i> <i>ul_min_iat_w20</i>	0.9732	<i>low: 0.982,</i> <i>high: 0.965</i>	<i>low: 0.964,</i> <i>high: 0.982</i>
Zastajkivanje <i>yes / no</i>	<i>dl_pkt_count_w1,</i> <i>ul_median_pkt_size_w1,</i> <i>dl_mean_pkt_size_w3,</i> <i>dl_min_pkt_size_w3,</i> <i>dl_median_iat_w3,</i> <i>ul_throughput_w3,</i> <i>ul_mean_pkt_size_w5,</i> <i>ul_max_pkt_size_w5,</i> <i>dl_pkt_count_w10,</i> <i>dl_median_iat_w10,</i> <i>dl_min_iat_w10,</i> <i>ul_max_pkt_size_w10,</i> <i>ul_max_iat_w10,</i>	0.963855	<i>no: 0.968,</i> <i>yes: 0.957</i>	<i>no: 0.972,</i> <i>yes: 0.952</i>

	<ul style="list-style-type: none"> <i>ul_stdev_iat_w10,</i> <i>dl_throughput_w20,</i> <i>ul_pckt_count_w20,</i> <i>ul_stdev_pckt_size_w20</i> 			
--	---	--	--	--

Model za svaki *KPI* koji se predviđa daje visoku točnost, preciznost i odziv. Kako bi se dokazalo da interakcije utječu na predviđanje *KPI*-eva, na izgrađenom modelu potrebno je provesti ponovnu procjenu modela na podacima videozapisa s interakcijama. Tablica 5.2 prikazuje točnost, preciznost i odziv koji su dobiveni ponovnom procjenom modela na podacima videozapisa za sve tri vrste interakcija.

Tablica 5.2 - Tablica uspješnosti ponovne procjene modela na podacima videozapisa s interakcijama izgrađenog na podacima videozapisa bez interakcija pomoću značajki odabranih na skupu podataka bez interakcija

Interakcija	<i>KPI</i>	Točnost	Preciznost	Odziv
Pauza	Rezolucija <i>sd / hd</i>	0.784541	<i>sd: 0.733,</i> <i>hd: 0.828</i>	<i>sd: 0.781,</i> <i>hd: 0.787</i>
	Prosječna brzina kodiranja <i>low / high</i>	0.912289	<i>low: 0.957,</i> <i>high: 0.843</i>	<i>low: 0.903,</i> <i>high: 0.929</i>
	Zastajkivanje <i>yes / no</i>	0.950706	<i>no: 1.0,</i> <i>yes: -</i>	<i>no: 0.951</i> <i>yes: -</i>
Premotavanje unaprijed	Rezolucija <i>sd / hd</i>	0.835227	<i>sd: 0.851,</i> <i>hd: 0.812</i>	<i>sd: 0.866,</i> <i>hd: 0.793</i>
	Prosječna brzina kodiranja <i>low / high</i>	0.93122	<i>low: 0.996,</i> <i>high: 0.794</i>	<i>low: 0.911,</i> <i>high: 0.990</i>
	Zastajkivanje <i>yes / no</i>	0.70471	<i>no: 0.640,</i> <i>yes: 0.884</i>	<i>no: 0.938,</i> <i>yes: 0.471</i>

Prekid reprodukcije i prebacivanje na sljedeći videozapis	Rezolucija <i>sd / hd</i>	0.873889	<i>sd: 0.866,</i> <i>hd: 0.884</i>	<i>sd: 0.899,</i> <i>hd: 0.846</i>
	Prosječna brzina kodiranja <i>low / high</i>	0.930964	<i>low: 0.973,</i> <i>high: 0.854</i>	<i>low: 0.925,</i> <i>high: 0.944</i>
	Zastajkivanje <i>yes / no</i>	0.857784	<i>no: 0.892,</i> <i>yes: 0.829</i>	<i>no: 0.814,</i> <i>yes: 0.901</i>

Iz dobivenih rezultata može se vidjeti da interakcije pauze i premotavanja unaprijed više utječu na predviđanje *KPI*-eva od interakcije prekida reprodukcije i prebacivanja na sljedeći videozapis. Za interakciju pauze je dobivena loša predikcija za rezoluciju, ali relativno dobra za prosječnu brzinu kodiranja i zastajkivanje. Za interakciju premotavanja videozapisa unaprijed može se vidjeti da su za rezoluciju i zastajkivanje dobivene loše predikcije, dok je za prosječnu brzinu kodiranja dobivena dobra predikcija. Dakle, korisničke interakcije pauziranja i premotavanja videozapisa unaprijed imaju najveći, uglavnom negativni utjecaj na predviđanje rezolucije i zastajkivanja.

5.3. Izgradnja modela pomoću značajki odabranih na mješovitom skupu podataka

U prethodnom potpoglavlju pokazano je da korisničke interakcije pauziranja i premotavanja videozapisa imaju najveći utjecaj na predviđanje rezolucije i zastajkivanja. Kako bi se pronašao model koji će dobro predviđati slučajeve bez obzira jesu li interakcije prisutne ili ne, potrebno je otkriti značajke koje su neovisne o korisničkim interakcijama. Nakon što se odaberu značajke, model je potrebno izgraditi na skupu podataka videozapisa bez interakcija te ga nakon toga ponovno procijeniti na skupu podataka videozapisa s interakcijama. Ako model pokaže visoku točnost, preciznost i odziv na skupu podataka videozapisa s interakcijama, može se pretpostaviti da su odabrane značajke neovisne o korisničkim interakcijama.

Postupak odabira značajki provodit će se na mješovitom skupu podataka koji će uključivati jednak broj podataka videozapisa s i bez interakcija. Prije svega, nužno je skalirati ulazni skup podataka za odabir značajki kako bi se dobio jednak broj podataka o sekundama videozapisa u kojima se pojavljuju i ne pojavljuju korisničke interakcije. Podaci o sekundama videozapisa s interakcijama uzeti su na sljedeći način:

- Slučaj pauze – sve sekunde u kojima je prisutna pauza,
- Slučaj premotavanja unaprijed – trenutak premotavanja i dvadeset sekundi poslije,
- Slučaj prekida videozapisa i prebacivanja na sljedeći – trenutak prekida i dvadeset sekundi poslije.

Broj podataka o sekundama videozapisa s interakcijama (prethodno opisanih) je 16.207, dok je broj podataka o sekundama videozapisa bez interakcija 96.041. Ulazni skup podataka za odabir značajki potrebno je skalirati na način da se dobije jednak broj podataka o sekundama videozapisa s i bez interakcija te da je jednak broj vrijednosti klasa koje se predviđaju. Broj podataka koji će se koristiti za odabir značajki u svrhu izgradnje modela za predviđanje rezolucije i prosječne brzine kodiranja je 10.000, a za zastajkivanje 1.600.

Tablica 5.3 prikazuje preciznost, točnost i odziv za model izgrađen na podacima bez interakcija pomoću značajki odabranih na mješovitom skupu podataka.

Tablica 5.3 - Tablica prikaza performansi modela izgrađenog na podacima videozapisa bez interakcija pomoću značajki odabranih na mješovitom skupu podataka

<i>KPI</i>	Odabrane značajke	Točnost	Preciznost	Odziv
Rezolucija <i>sd / hd</i>	<i>dl_median_pkt_size_w20,</i> <i>dl_max_pkt_size_w20,</i> <i>dl_median_iat_w20, dl_max_iat_w20,</i> <i>ul_median_pkt_size_w20,</i> <i>ul_max_pkt_size_w20,</i> <i>ul_median_iat_w20, ul_max_iat_w20,</i> <i>ul_min_iat_w20</i>	0.96414	<i>sd: 0.956,</i> <i>hd: 0.973</i>	<i>sd: 0.974,</i> <i>hd: 0.954</i>
Prosječna brzina kodiranja <i>low / high</i>	<i>dl_throughput_w20,</i> <i>dl_median_pkt_size_w20,</i> <i>dl_max_pkt_size_w20,</i> <i>dl_mean_iat_w20, dl_median_iat_w20,</i> <i>dl_max_iat_w20,</i> <i>ul_pkt_count_gt100_w20,</i> <i>ul_median_pkt_size_w20,</i> <i>ul_max_pkt_size_w20,</i> <i>ul_stdev_pkt_size_w20,</i> <i>ul_max_iat_w20, ul_min_iat_w20</i>	0.980025	<i>low: 0.986,</i> <i>high: 0.974</i>	<i>low: 0.974,</i> <i>high: 0.986</i>
Zastajkivanje {ne:n, da:y}	<i>dl_max_pkt_size_gt100_w1,</i> <i>ul_pkt_count_w1,</i> <i>ul_min_pkt_size_w1,</i> <i>ul_min_pkt_size_w2,</i> <i>dl_pkt_count_w3,</i> <i>dl_stdev_iat_w3,</i> <i>ul_pkt_count_w3,</i> <i>ul_active_time_w3,</i> <i>ul_mean_pkt_size_w5,</i> <i>ul_max_pkt_size_w5,</i> <i>ul_min_pkt_size_w5,</i> <i>dl_pkt_count_w10,</i> <i>ul_throughput_w10, ul_median_iat_w20</i>	0.93012	<i>no: 0.943,</i> <i>yes: 0.911</i>	<i>no: 0.941,</i> <i>yes: 0.914</i>

Sva tri modela, jedan za predviđanje rezolucije, drugi za predviđanje prosječne brzine kodiranja te treći za predviđanje zastajkivanja, su pokazala veliku uspješnost. Dobiveni su rezultati visoke preciznosti, točnosti i odziva. Kako bi se pokazalo da model dobro predviđa podatke bez obzira je su li interakcije prisutne, potrebno ga je ponovno procijeniti na skupu podataka

videozapisa s interakcijama. Tablica 5.4 prikazuje uspješnost ponovne procjene modela za sve tri vrste interakcija.

Tablica 5.4 - Tablica uspješnosti ponovne procjene modela na podacima videozapisa s interakcijama izgrađenog na podacima videozapisa bez interakcija pomoću značajki odabranih na mješovitom skupu podataka

Interakcija	KPI	Točnost	Preciznost	Odziv
Pauza	Rezolucija <i>sd / hd</i>	0.800046	<i>sd: 0.740,</i> <i>hd: 0.855</i>	<i>sd: 0.822,</i> <i>hd: 0.783</i>
	Prosječna brzina kodiranja <i>low / high</i>	0.904652	<i>low: 0.951,</i> <i>high: 0.834</i>	<i>low: 0.898,</i> <i>high: 0.917</i>
	Zastajkivanje <i>yes / no</i>	0.943765	<i>no: 1.0,</i> <i>yes: -</i>	<i>no: 0.944.,</i> <i>yes: -</i>
Premotavanje unaprijed	Rezolucija <i>sd / hd</i>	0.851347	<i>sd: 0.848,</i> <i>hd: 0.857</i>	<i>sd: 0.906,</i> <i>hd: 0.777</i>
	Prosječna brzina kodiranja <i>low / high</i>	0.939223	<i>low: 0.995,</i> <i>high: 0.816</i>	<i>low: 0.923,</i> <i>high: 0.987</i>
	Zastajkivanje <i>yes / no</i>	0.756341	<i>no: 0.694,</i> <i>yes: 0.879</i>	<i>no: 0.918,</i> <i>yes: 0.594</i>
Prekid reprodukcije i prebacivanje na sljedeći videozapis	Rezolucija <i>sd / hd</i>	0.906034	<i>sd: 0.886,</i> <i>hd: 0.932</i>	<i>sd: 0.942,</i> <i>hd: 0.867</i>
	Prosječna brzina kodiranja <i>low / high</i>	0.940204	<i>low: 0.982,</i> <i>high: 0.866</i>	<i>low: 0.929,</i> <i>high: 0.965</i>
	Zastajkivanje <i>yes / no</i>	0.872754	<i>no: 0.919,</i> <i>yes: 0.836</i>	<i>no: 0.817,</i> <i>yes: 0.928</i>

Kao što je već rečeno u potpoglavlju 5.2, interakcija prekida reprodukcije i prebacivanja na sljedeći videozapis ima mali, gotovo zanemariv utjecaj na predviđanje rezolucije, prosječne brzine kodiranja i zastajkivanja što je pokazano i u ovom testiranju. Također, u ovom testiranju potvrđeno je da je prosječna brzina kodiranja ovisna i o druge dvije interakcije te da interakcija pauze slabije utječe na predviđanje zastajkivanja. Što se tiče predviđanja rezolucije za interakcije pauziranja i premotavanja unaprijed, dobiveni su poboljšani rezultati u odnosu na rezultate dobivene u potpoglavlju 5.2. Za interakciju premotavanja unaprijed i predviđanja zastajkivanja također su dobiveni bolji rezultati u odnosu na rezultate dobivene u potpoglavlju 5.2, no još uvijek niže točnosti, preciznosti i odziva. U sljedećem potpoglavlju nalazi se detaljna analiza rezultata o ovisnostima mrežnih značajki i interakcija.

5.4. Analiza rezultata

U potpoglavlju 5.2. pokazano je da interakcije imaju utjecaj na predviđanje *KPI*-eva. Za odabir značajki, treniranje i testiranje modela korišteni su podaci videozapisa u kojima nisu bile prisutne interakcije, a model se naknadno ponovno procjenjivao na podacima videozapisa s interakcijama. Tablica 5.5 prikazuje značajke odabrane na skupu podataka bez interakcija s kojima se trenirao model.

Tablica 5.5 - Značajke odabrane na skupu podataka videozapisa bez interakcija

<i>KPI</i>	Odabrane značajke
Rezolucija	<i>ul_pkt_count_w5, dl_throughput_w20, dl_max_pkt_size_w20, dl_median_iat_w20, dl_max_iat_w20, ul_median_pkt_size_w20, ul_min_pkt_size_w20, ul_max_iat_w20, ul_min_iat_w20</i>
Prosječna brzina kodiranja	<i>dl_max_pkt_size_w10, dl_min_pkt_size_w10, dl_active_time_w20, dl_mean_pkt_size_w20, dl_max_pkt_size_w20, dl_median_iat_w20, ul_median_iat_w20, ul_max_pkt_size_w20, ul_median_iat_w20, ul_max_iat_w20, ul_min_iat_w20</i>

Zastajkivanje	<i>dl_pkt_count_w1, ul_median_pkt_size_w1, dl_mean_pkt_size_w3, dl_min_pkt_size_w3, dl_median_iat_w3, ul_throughput_w3, ul_mean_pkt_size_w5, ul_max_pkt_size_w5, dl_pkt_count_w10, dl_median_iat_w10, dl_min_iat_w10, ul_max_pkt_size_w10, ul_max_iat_w10, ul_stdev_iat_w10, dl_throughput_w20, ul_pkt_count_w20, ul_stdev_pkt_size_w20</i>
---------------	--

Ponovnom procjenom modela su dobivene lošije predikcije u odnosu na dobivene rezultate uspješnosti modela izgrađenog na podacima videozapisa bez interakcija. U potpoglavlju 5.3 napravljen je isti postupak kao i u potpoglavlju 5.2, ali za odabir značajki koristio se mješoviti skup podataka koji je u sebi uključivao podatke videozapisa s i bez interakcija. Za rezoluciju je odabrano 6 istih značajki, za prosječnu brzinu kodiranja 5 istih značajki, a za zastajkivanje su odabrane samo 3 iste značajke kao i u potpoglavlju 5.2. Tablica 5.6 prikazuje značajke odabrane na mješovitom skupu podataka s kojima se trenirao novi model na podacima videozapisa bez interakcija te presjek značajki s onima koje su dobivene u potpoglavlju 5.2.

Tablica 5.6 - Značajke odabrane na mješovitom skupu podataka i presjek značajki s onima koje su odabrane na skupu podataka videozapisa bez interakcija

<i>KPI</i>	Odabrane značajke	Presjek značajki
Rezolucija	<i>dl_median_pkt_size_w20, dl_max_pkt_size_w20, dl_median_iat_w20, dl_max_iat_w20, ul_median_pkt_size_w20, ul_max_pkt_size_w20, ul_median_iat_w20, ul_max_iat_w20, ul_min_iat_w20</i>	<i>dl_max_pkt_size_w20, dl_median_iat_w20, dl_max_iat_w20, ul_median_pkt_size_w20, ul_max_iat_w20, ul_min_iat_w20</i>
Prosječna brzina kodiranja	<i>dl_throughput_w20, dl_median_pkt_size_w20, dl_max_pkt_size_w20, dl_mean_iat_w20, dl_median_iat_w20, dl_max_iat_w20, ul_pkt_count_gt100_w20, ul_median_pkt_size_w20, ul_max_pkt_size_w20, ul_stdev_pkt_size_w20, ul_max_iat_w20, ul_min_iat_w20</i>	<i>dl_max_pkt_size_w20, dl_median_iat_w20, ul_max_pkt_size_w20, ul_max_iat_w20, ul_min_iat_w20</i>
Zastajkivanje	<i>dl_max_pkt_size_gt100_w1, ul_pkt_count_w1, ul_min_pkt_size_w1, ul_min_pkt_size_w2, dl_pkt_count_w3, dl_stdev_iat_w3, ul_pkt_count_w3,</i>	<i>ul_mean_pkt_size_w5, ul_max_pkt_size_w5, dl_pkt_count_w10</i>

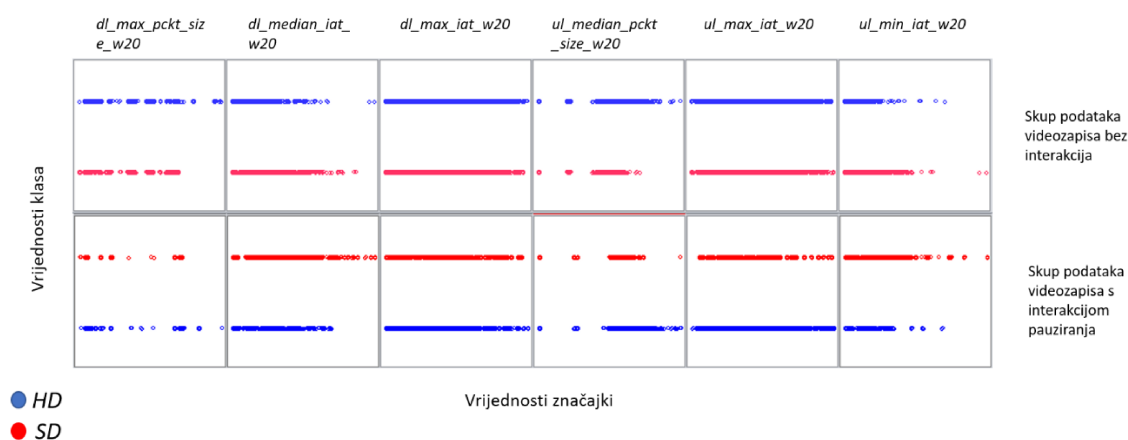
	<ul style="list-style-type: none"> <i>ul_active_time_w3, ul_mean_pkt_size_w5,</i> <i>ul_max_pkt_size_w5, ul_min_pkt_size_w5,</i> <i>dl_pkt_count_w10, ul_throughput_w10,</i> <i>ul_median_iat_w20</i> 	
--	---	--

Ponovnom procjenom novog modela su dobivene lošije predikcije u odnosu na dobivene rezultate uspješnosti novog modela izgrađenog na podacima videozapisa bez interakcija. Međutim, dobivena su mala poboljšanja u predikciji *KPI*-eva u odnosu na rezultate ponovne procjene dobivene u potpoglavlju 5.2. Za predviđanje rezolucije i prosječne brzine kodiranja odabrane su značajke s veličinom prozora od 20 sekundi, dok su za predviđanje zastajkivanja odabrane značajke s različitim veličinama prozora (od 1 do 10 sekundi). Time se može naslutiti da se bolje predikcije za rezoluciju i prosječnu brzinu kodiranja dobivaju sa značajkama s većom veličinom prozora nego s onima s manjom veličinom prozora. U nastavku se nalazi analiza i usporedba značajki koje su dobivene presjekom značajki u oba potpoglavlja za svaku od interakcija. Za prikaz vrijednosti klasa korišteno je 5.000 nasumično odabranih podataka iz skupa podataka bez interakcija i 5.000 podataka iz skupa podataka s interakcijom koja se razmatra.

1. Interakcija pauziranja videozapisa

Usporedbom rezultata ponovne procjene dobivenih u potpoglavlju 5.3 u odnosu na one dobivene u potpoglavlju 5.2 može se vidjeti malo poboljšanje rezultata. Za predviđanje prosječne brzine kodiranja i zastajkivanja rezultati su skoro pa identični, a za predviđanje rezolucije je dobiveno poboljšanje točnosti za 2 postotna boda. Rezultati ponovne procjene su zadovoljavajući, no ipak je za rezoluciju dobivena lošija točnost u usporedbi s točnošću prosječne brzine kodiranja i zastajkivanja. Stoga se može pretpostaviti da interakcija pauziranja videozapisa ima negativniji utjecaj na predviđanje rezolucije, nego na predviđanje prosječne brzine kodiranja i zastajkivanja. U nastavku se nalazi analiza značajki za svaki *KPI* koji se predviđa.

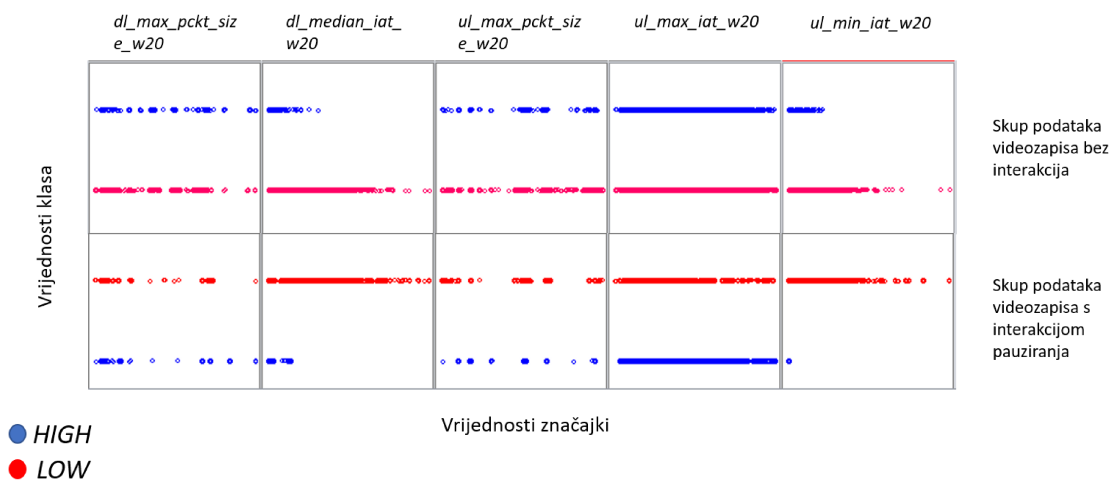
Slika 5.6 prikazuje vrijednosti klase rezolucije za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija pauziranja videozapisa. Usporedbom vrijednosti klase u grafovima može se vidjeti da se mrežne značajke ponašaju slično bez obzira radi li se o podacima videozapisa s ili bez interakcije pauziranja. Razlike u podacima postoje, no može se reći da one nisu velike. Jedina veća odstupanja u vrijednostima mogu se vidjeti u prvom grafu za značajku *dl_max_pkt_size_w20*, dok u ostalim grafovima takvih odstupanja u vrijednostima nema. Razlog ovakvih rezultata može biti u tome zato što se za vrijeme interakcije pauze trajanja do 30 sekundi i dalje preuzima video sadržaj. Stoga pretpostavka je da bi se značajke, u kojima ne postoje veća odstupanja, mogle proglasiti onima koje su gotovo neovisne o interakciji pauziranja videozapisa maksimalnog trajanja 30 sekundi u predviđanju rezolucije.



Slika 5.6 – Grafovi usporedbe skupa podataka bez i s interakcijom pauze za prikaz vrijednosti klase rezolucije za odabrane značajke mrežnog prometa

Slika 5.7 prikazuje vrijednosti klase prosječne brzine kodiranja za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija pauziranja videozapisa. Usporedbom vrijednosti klase u grafovima

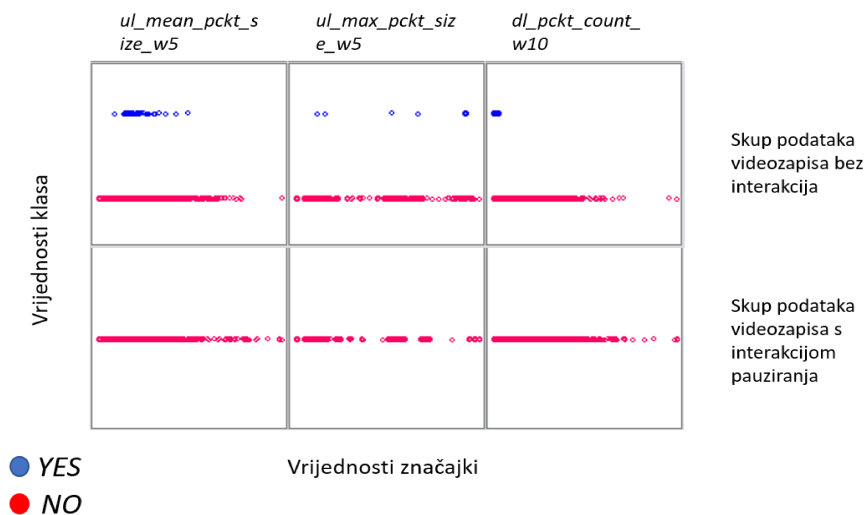
može se vidjeti da se za značajke *ul_max_iat_w20* i *dl_median_iat_w20* dobivaju približno slične vrijednosti u oba skupa podataka, dok se za ostale tri značajke vide veća odstupanja u vrijednostima. Stoga je pretpostavka da se ove značajke, osim navedene dvije, ne mogu proglasiti onima koje su neovisne o interakciji pauziranja videozapisa maksimalnog trajanja 30 sekundi u predviđanju prosječne brzine kodiranja.



Slika 5.7 - Grafovi usporedbe skupa podataka bez i s interakcijom pauze za prikaz vrijednosti klase prosječne brzine kodiranja za odabrane značajke mrežnog prometa

Slika 5.8 prikazuje vrijednosti klase zastajkivanja za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija pauziranja videozapisa. Kada je u pitanju interakcija pauze, ne postoji slučaj zastajkivanja tako da su sve vrijednosti klase negativne. Promatranjem negativnih vrijednosti klase može se vidjeti da se podaci ponašaju vrlo slično te da razlike u podacima gotovo nema. Kao što je već rečeno, razlog toga može biti što se tijekom pauze videozapisa kratkog trajanja i dalje preuzima video sadržaj. Stoga je pretpostavka da bi se ove značajke mogle proglasiti onima koje su

gotovo neovisne o interakciji pauziranja videozapisa maksimalnog trajanja 30 sekundi u predviđanju zastajkivanja.



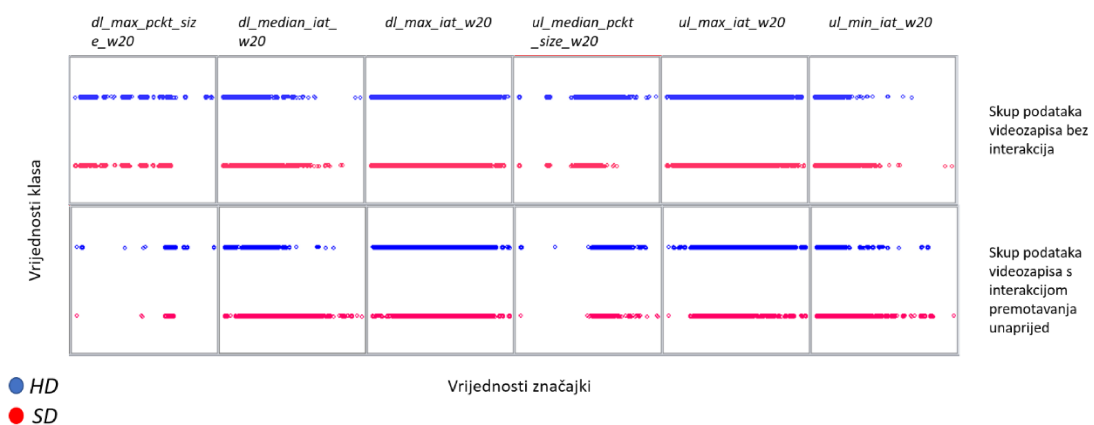
Slika 5.8 - Grafovi usporedbe skupa podataka bez i s interakcijom pauze za prikaz vrijednosti klase zastajkivanja za odabrane značajke mrežnog prometa

2. Interakcija premotavanja videozapisa unaprijed

Usporedom rezultata ponovne procjene dobivenih u potpoglavlju 5.3 u odnosu na one dobivene u potpoglavlju 5.2 može se vidjeti malo poboljšanje rezultata. Za predviđanje prosječne brzine kodiranja rezultati su približno jednaki, a za predviđanje rezolucije i zastajkivanja su dobiveni poboljšani rezultati. Za rezoluciju je dobiveno poboljšanje točnosti za 2 postotna boda, dok je za zastajkivanje dobiveno poboljšanje točnosti za 5 postotnih bodova. Rezultati uspješnosti ponovne procjene su za rezoluciju i prosječnu brzinu kodiranja zadovoljavajući te se zbog toga može pretpostaviti da interakcija premotavanja videozapisa unaprijed ima minimalni utjecaj na predviđanje navedena dva *KPI*-a. Međutim, za zastajkivanje su još uvijek dobiveni loši rezultati u ponovnoj procjeni novog modela te se zbog toga može reći da interakcija premotavanja videozapisa unaprijed ima veći utjecaj na predviđanje

navedenog *KPI*-a. U nastavku se nalazi analiza značajki za svaki *KPI* koji se predviđa.

Slika 5.9 prikazuje vrijednosti klase rezolucije za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija premotavanja videozapisa unaprijed. Usporedbom grafova istih značajki može se vidjeti da se vrijednosti klase za sve značajke slično ponašaju, osim za značajku *dl_max_pkt_size_w20*. Razlog može biti u tome što kada se dogodi premotavanje videozapisa unaprijed, a prethodno je bio reproduciran videozapis s visokom razlučivosti za koju su slani paketi veće veličine, početak će se slati paketi manje veličine kako ne bi došlo do zastajkivanja pri čemu će videozapis biti reproduciran u nižoj razini razlučivosti. Međutim, kako je maksimalna zabilježena veličina paketa u prethodnih 20 sekundi velika, značajka će za nisku razinu razlučivosti zabilježavati visoke vrijednosti. Za ostale značajke, u kojima nema značajnijih odstupanja u vrijednostima, može se reći da su gotovo neovisne o interakciji premotavanja videozapisa unaprijed u predviđanju rezolucije.



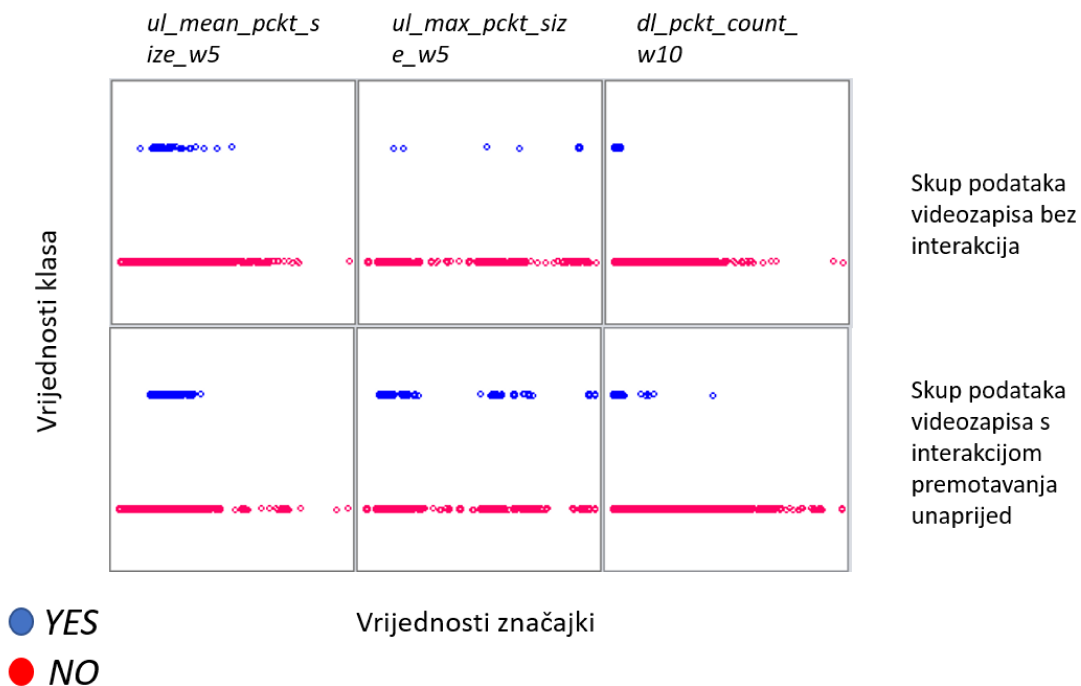
Slika 5.9 - Grafovi usporedbe skupa podataka bez i s interakcijom premotavanja videozapisa unaprijed za prikaz vrijednosti klase rezolucije za odabrane značajke mrežnog prometa

Slika 5.10 prikazuje vrijednosti klase prosječne brzine kodiranja za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija premotavanja videozapisa unaprijed. Usporedbom vrijednosti klase u grafovima može se vidjeti da za značajke *dl_max_pkt_size_w20* i *ul_max_pkt_size_w20* postoje razlike u vrijednostima, dok za ostale značajke većih razlika nema. Jedina razlika koju navedene dvije značajke imaju je ta da se jedna odnosi na odlazni, a druga na dolazni promet. Kod značajke koja se odnosi na odlazni promet može se vidjeti sličnost u ponašanju podataka, dok za drugu ne. Potencijalni uzrok ovog odstupanja u vrijednostima za navedenu značajku dolaznog prometa objašnjen je za prethodni *KPI*. Ovo se može tvrditi zbog toga što se rezolucija i prosječna brzina kodiranja ponašaju slično (viša rezolucija pretpostavlja veću brzinu kodiranja, i obrnuto). Za značajke, koje nemaju većih odstupanja u vrijednostima, može se pretpostaviti da su gotovo neovisne o interakciji premotavanja videozapisa unaprijed u predviđanju prosječne brzine kodiranja.



Slika 5.10 - Grafovi usporedbe skupa podataka bez i s interakcijom premotavanja videozapisa unaprijed za prikaz vrijednosti klase prosječne brzine kodiranja za odabrane značajke mrežnog prometa

Slika 5.11 prikazuje vrijednosti klase zastajkivanja za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija premotavanja videozapisa unaprijed. Usporedbom grafova istih značajki može se vidjeti da za značajke *ul_mean_pkt_size_w5* i *dl_pkt_count_w10* postoje manja odstupanja u vrijednostima klase nego za značajku *ul_max_pkt_size_w5*. Potencijalni uzrok razlike u vrijednostima klase zastajkivanja za značajku *ul_max_pkt_size_w5* može se kriti u samim podacima koji su se koristili za prikaz jer je puno više slučajeva kada ne dolazi do zastajkivanja. Za značajke *ul_mean_pkt_size_w5* i *dl_pkt_count_w10* može se pretpostaviti da su manje ovisne o interakciji premotavanja videozapisa unaprijed u predviđanju zastajkivanja.

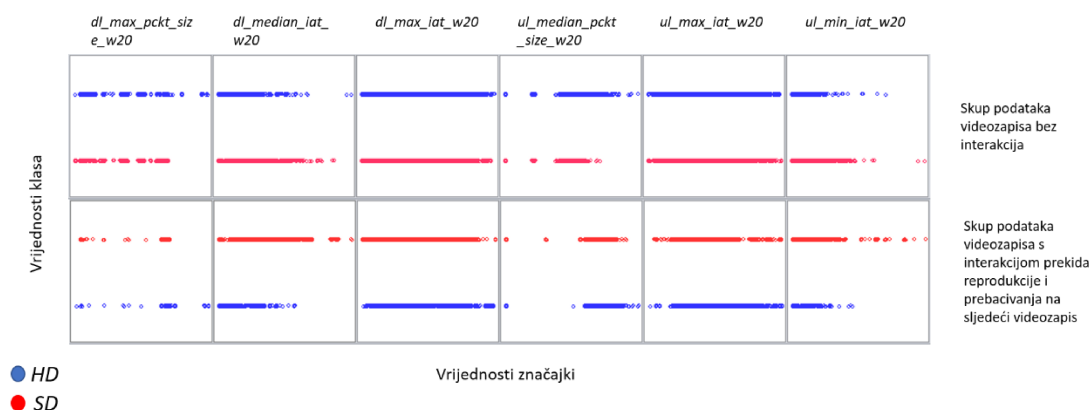


Slika 5.11 - Grafovi usporedbe skupa podataka bez i s interakcijom premotavanja videozapisa unaprijed za prikaz vrijednosti klase zastajkivanja za odabrane značajke mrežnog prometa

3. Interakcija prekida reprodukcije i prebacivanja na sljedeći videozapis

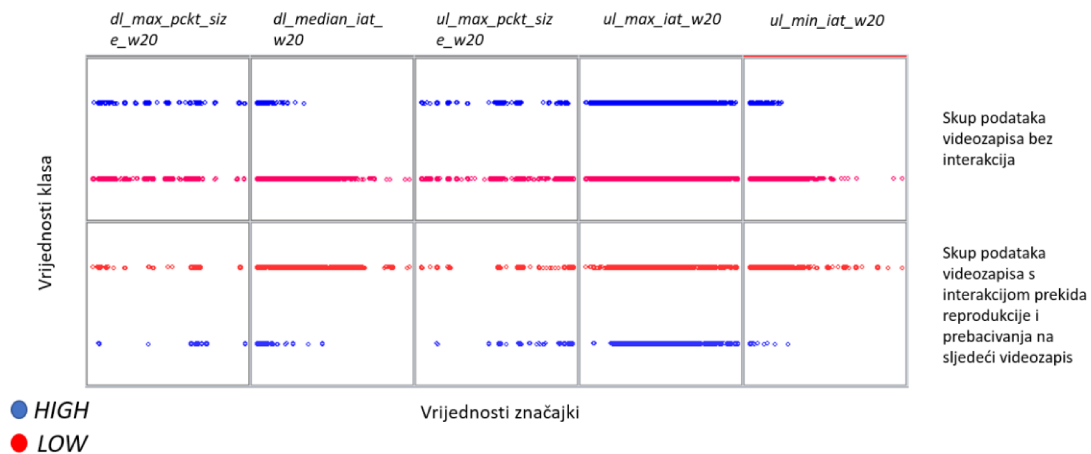
Usporedom rezultata ponovne procjene dobivenih u potpoglavlju 5.3 u odnosu na one dobivene u potpoglavlju 5.2 može se vidjeti poboljšanje rezultata za svaki *KPI* koji se predviđao. U predviđanju rezolucije, prosječne brzine kodiranja i zastajkivanja dobivena su poboljšanja točnosti redom za 3, 1 i 2 postotna boda. Rezultati uspješnosti ponovne procjene su vrlo visoki i zadovoljavajući te se može reći da interakcija prekida videozapisa i prebacivanja na sljedeći videozapis ima minimalni učinak na predviđanje navedenih *KPI*-eva. U nastavku se nalazi analiza značajki za svaki *KPI* koji se predviđa.

Slika 5.12 prikazuje vrijednosti klase rezolucije za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija prekida reprodukcije i prebacivanja na sljedeći videozapis. Usporedbom vrijednosti klase istih značajki u grafovima može se vidjeti da za značajku *dl_max_pkt_size_w20* postoje veća odstupanja u vrijednostima, dok su za ostale značajke odstupanja manja. Potencijalni uzrok razlike u vrijednostima za značajku *dl_max_pkt_size_w20* objašnjen je za interakciju premotavanja videozapisa unaprijed. Razlog može biti isti jer je prekid reprodukcije i prebacivanje na sljedeći videozapis sličan interakciji premotavanja videozapisa u dio u kojem u aplikacijskom spremniku nema podataka o toj sekundi videozapisa. Kada se videozapis prekine i prebaci na sljedeći, spremnik se mora ispočetka puniti, a na isti način se spremnik mora puniti kada dođe do premotavanja videozapisa unaprijed u dio u kojem u spremniku nema podataka o toj sekundi. Za ostale značajke, u kojima su odstupanja u vrijednostima minimalna, može se pretpostaviti da su gotovo neovise o korisničkoj interakciji prekida reprodukcije i prebacivanja na sljedeći videozapis.



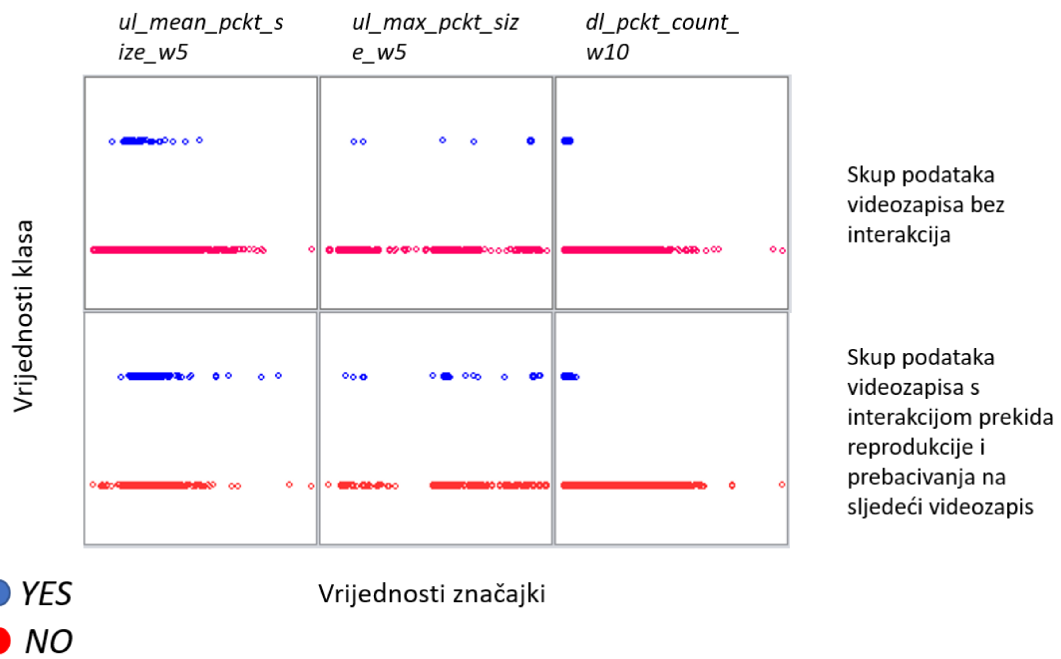
Slika 5.12 - Grafovi usporedbe skupa podataka bez i s interakcijom prekida reprodukcije i prebacivanja na sljedeći videozapis za prikaz vrijednosti klase rezolucije za odabrane značajke mrežnog prometa

Slika 5.13 prikazuje vrijednosti klase prosječne brzine kodiranja za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija prekida reprodukcije i prebacivanja na sljedeći videozapis. Usporedbom grafova istih značajki može se vidjeti da za značajke *dl_max_pkt_size_w20* i *ul_max_pkt_size_w20* postoje razlike u vrijednostima klase, dok za ostale značajke nema značajnijih odstupanja. Kao što je već rečeno u paragrafu iznad za predviđanje rezolucije, interakcija prekida videozapisa i prebacivanja na sljedeći je slična interakciji premotavanja videozapisa u dio u kojem nema podataka o toj sekundi videozapisa. Potencijalni uzrok promjene u podacima objašnjen je u dijelu teksta koji se odnosi na premotavanje videozapisa unaprijed i predviđanje rezolucije i prosječne brzine kodiranja. Za ostale značajke mrežnog prometa, može se pretpostaviti da su gotovo neovise o korisničkoj interakciji prekida reprodukcije i prebacivanja na sljedeći videozapis.



Slika 5.13 - Grafovi usporedbe skupa podataka bez i s interakcijom prekida reprodukcije i prebacivanja na sljedeći videozapis za prikaz vrijednosti klase prosječne brzine kodiranja za odabrane značajke mrežnog prometa

Slika 5.14 prikazuje vrijednosti klase zastajkivanja za značajke koje su dobivene presjekom značajki odabranih u potpoglavljima 5.2 i 5.3 za skup podataka bez interakcija i skup podataka u kojima je bila interakcija prekida reprodukcije i prebacivanja na sljedeći videozapis. Kao i za prethodne dvije klase, i ovdje se može vidjeti isto ponašanje u vrijednostima klase kao što se vrijednosti ponašaju za podatke videozapisa s interakcijom premotavanja videozapisa unaprijed. Za značajku *ul_max_pkt_size_w5* se vidi veće odstupanje u vrijednostima klase, dok se za *ul_mean_pkt_size_w5* i *dl_pkt_count_w10* odstupanja manje vide. Stoga se za te dvije značajke može pretpostaviti da su gotovo neovise o korisničkoj interakciji prekida reprodukcije i prebacivanja na sljedeći videozapis.



Slika 5.14 - Grafovi usporedbe skupa podataka bez i s interakcijom prekida reprodukcije i prebacivanja na sljedeći videozapis za prikaz vrijednosti klase zastajkivanja za odabrane značajke mrežnog prometa

Zaključak

Posljednjih godina količina Internet prometa je u sve većem porastu, a najveće zasluge za to imaju masovna korištenja usluga strujanja videa. Velik dio tog prometa povezan je s uslugama temeljenim na prilagodljivom strujanju putem protokola *HTTP*. Trenutno jedna od vodećih platformi za strujanje videa na Internetu je YouTube.

Imajući na uvidu navedene tvrdnje, davatelji mrežnih usluga suočeni su s izazovom efikasnog upravljanja ograničenim mrežnim resursima, a da se pri tome krajnjim korisnicima koji gledaju videozapise osigura i pruži zadovoljavajuća razina iskustvene kvalitete. S obzirom da je većina mrežnog prometa kriptirana, čak i samo praćenje performansi usluga strujanja videa u mreži predstavlja problem. U tom kontekstu, rješenje za praćenje iskustvene kvalitete temeljeno na metodama strojnog učenja pokazalo se kao obećavajuće. Međutim, korisničko ponašanje u obliku korisničkih interakcija ima utjecaj na uzorke u mrežnom prometu pa samim time i na procjenu iskustvene kvalitete. Cilj ovoga rada bio je prikupiti veliki broj podataka s mrežne i aplikacijske razine usluge YouTube na Android mobilnom uređaju prilikom strujanja videa sa simuliranim korisničkim interakcijama kako bi se otkrile značajke mrežnog prometa koje su neovisne o korisničkim interakcijama. U svrhu prikupljanja podataka s aplikacijske razine napravljena je skripta koja, osim prikupljanja aplikacijskih parametara, simulira korisničke interakcije prilikom reprodukcije videozapisa.

Na prikupljenim podacima bilo je potrebno odrediti koje interakcije imaju utjecaj na predviđanje *KPI*-eva, a nakon toga cilj je bio otkriti značajke koje su manje ili potpuno neovisne o korisničkim interakcijama. Iz provedene analize, pretpostavlja se da interakcija prekida reprodukcije i prebacivanja na sljedeći ima manji utjecaj na predviđanje *KPI*-eva, dok interakcije pauziranja i premotavanja unaprijed imaju veći utjecaj. Poslije toga provedeno je istraživanje u kojem su pronađene značajke mrežnog prometa za koje se može pretpostaviti da su manje ili potpuno neovisne o postojanju korisničkih interakcija. Na njima je izgrađen model strojnog učenja na kojem su dobiveni

poboljšani rezultati u predviđanju *KPI*-eva čime je procjena iskustvene kvalitete unaprijeđena.

Literatura

- [1] Cisco Annual Internet Report (2018-2023) White Paper, Poveznica: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf>
- [2] VNI Complete Forecast Highlights 2020., Poveznica: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2020_Forecast_Highlights.pdf
- [3] Maryam Mohsin, „10 Youtube Stats Every Marketer Should Know in 2020“, Poveznica: <https://www.oberlo.com/blog/youtube-statistics>, Zadnji posjet: 03.04.2020.
- [4] Esteban Ortiz-Ospina, „The Rise of Social Media“, Poveznica: <https://ourworldindata.org/rise-of-social-media>, Zadnji posjet: 03.04.2020.
- [5] YouTube, Wikipedia, Poveznica: <https://hr.wikipedia.org/wiki/YouTube>, Zadnji posjet: 03.04.2020.
- [6] YouTube About, Poveznica: <https://www.youtube.com/about/>, Zadnji posjet: 03.04.2020.
- [7] „How Much do YouTubers Make? – A YouTuber’s Pocket Guide [Calculator]“, Poveznica: <https://influencermarketinghub.com/how-much-do-youtubers-make/>, Zadnji posjet: 06.04.2020.
- [8] „Kako funkcionira AdSense?“, Poveznica: <https://support.google.com/adsense/answer/6242051?hl=hr>, Zadnji posjet: 06.04.2020.
- [9] YouTube Premium Account, Poveznica: <https://www.youtube.com/premium>, Zadnji posjet: 06.04.2020.
- [10] Seufert, Michael, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hoßfeld, and Phuoc Tran-Gia. "A Survey on Quality of Experience of HTTP Adaptive Streaming." IEEE Communications Surveys & Tutorials 17, no. 1 (2014): 469-492.

[11] Monica Allevan, "YouTube Driving More QUIC-based Traffic on Mobile", Poveznica: <https://www.fiercewireless.com/wireless/youtube-driving-more-quic-based-traffic-mobile-vasona>, Zadnji posjet: 24.06.2020.

[12] Langley, Adam, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang et al. "The QUIC Transport Protocol: Design and Internet-Scale Deployment." In Proceedings of the Conference of the ACM Special Interest Group on Data Communication, pp. 183-196. 2017.

[13] Thomas Barnett, Jr., Shruti Jain, Usha Andra, Taru Khurana; Cisco Visual Networking Index (VNI) Complete Forecast Update, 2017–2022 (prosinac 2018)

[14] Quality of telecommunication services: concepts, models, objectives and dependability planning – Terms and definitions related to the quality of telecommunication services, Recommendation ITU-T E.800 (09/2008)

[15] Pattinson, Xinping Guo Colin. "Quality of Service Requirements for Multimedia Communications.", Paper presented at Time and the Web, Staffordshire University, 19th June 1997.

[16] Brunnström, Kjell, Sergio Ariel Beker, Katrien De Moor, Ann Dooms, Sebastian Egger, Marie-Neige Garcia, Tobias Hossfeld et al. "Qualinet White Paper on Definitions of Quality of Experience." (2013).

[17] Schatz, Raimund, Tobias Hoßfeld, Lucjan Janowski, and Sebastian Egger. "From Packets to People: Quality of Experience as a New Measurement Challenge." In Data traffic monitoring and analysis, pp. 219-263. Springer, Berlin, Heidelberg, 2013.

[18] Multimedia Quality of Service and performance – Generic and user-related aspects, Reference guide to quality of experience assessment methodologies, Recommendation ITU-T G.1011 (07/2016)

[19] Human Factors (HF); Quality of Experience (QoE) requirements for real-time communication services; ETSI TR 102 643 V1.0.2 (2010-01)

[20] International Telecommunication Union - Telecommunications Sector (ITU-T) Recommendation P.800.1, 2006. Mean Opinion Score (MOS) Terminology

[21] ITU-T - P.1203: “Recommendation ITU-T P.1203 – Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport”, 10.2017.

[22] ITU-T - P.1203.1: “Recommendation ITU-T P.1203.1 – Parametric Bitstream-based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport – Video Quality Estimation Module“, 01.2019.

[23] ITU-T - P.1203.2: “Recommendation ITU-T P.1203.2 – Parametric Bitstream-based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport – Audio Quality Estimation Module”, 10.2017.

[24] ITU-T - P.1203.3: “Recommendation ITU-T P.1203.3 – Parametric Bitstream-based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport – Quality Integration Module”, 01.2019.

[25] Programski alat WAMP, Poveznica: <https://www.wampserver.com/en/>, Zadnji posjet: 24.06.2020.

[26] YouTube-dl, Poveznica: <https://ytdl-org.github.io/youtube-dl/index.html>, Zadnji posjet: 27.04.2020.

[27] YouTube Data API V3, Poveznica: <https://developers.google.com/youtube/v3/docs>, Zadnji posjet: 15.04.2020.

[28] Net.Shark, Poveznica: <http://www.albedotelecom.com/pages/fieldtools/src/netshark.php>, Zadnji posjet: 27.05.2020.

[29] Appium about, Poveznica: <http://appium.io/docs/en/about-appium/intro/>, Zadnji posjet: 05.05.2020.

[30] Paul, „A Deconstruction of the Appium Architecture“, Poveznica: <https://www.edureka.co/blog/appium-architecture/>, Zadnji posjet: 05.05.2020.

[31] Riiser, Haakon, Tore Endestad, Paul Vigmostad, Carsten Griwodz, and Pål 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.

[32] Van Der Hooft, Jeroen, Stefano Petrangeli, Tim Wauters, Rafael Huysegems, Patrice Rondao Alface, Tom Bostoen, and Filip De Turck. "HTTP/2-based Adaptive Streaming of HEVC Video over 4G/LTE Networks." *IEEE Communications Letters* 20, no. 11 (2016): 2177-2180.

[33] StreamOn, Poveznica: <https://www.telekom.de/unterwegs/tarife-und-optionen/streamon>, Zadnji posjet: 14.05.2020.

[34] Tcpcdump, Poveznica: <https://www.tcpcdump.org/license.html>, Zadnji posjet: 07.05.2020.

[35] Tcpcdump, Wikipedia, Poveznica: <https://en.wikipedia.org/wiki/Tcpcdump>, Zadnji posjet: 07.05.2020.

[36] Wireshark, Poveznica: <https://www.wireshark.org/>, Zadnji posjet: 07.05.2020.

[37] Imunes, Poveznica: <http://imunes.net/>, Zadnji posjet: 07.05.2020.

[38] Tshark - Dump and analyze network traffic, Poveznica: <https://www.wireshark.org/docs/man-pages/tshark.html>, Zadnji posjet: 21.05.2020.

[39] Orsolice, Irena, and Lea Skorin-Kapov. "A Framework for in-Network QoE Monitoring of Encrypted Video Streaming." *IEEE Access* 8 (2020): 74691-74706.

[40] BestFirst attribute selection, WEKA, Poveznica: <https://weka.sourceforge.io/doc.dev/weka/attributeSelection/BestFirst.html>, Zadnji posjet: 07.06.2020.

[41] Ron Kohavi, George H. John (1997). Wrappers for feature subset selection. *Artificial Intelligence*. 97(1-2):273-324.

[42] Alpaydin, Ethem. *Introduction to machine learning*. MIT press, 2020.

[43] Random Forests, Poveznica:
<http://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/ensembles/RandomForests.pdf>,
Zadnji posjet: 01.06.2020.

[44] WEKA, Poveznica: <https://www.cs.waikato.ac.nz/ml/weka/>, Zadnji posjet:
27.05.2020.

[45] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA
Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools
and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

Sažetak

Gledanje videa na zahtjev je jedna od najpopularnijih aktivnosti na Internetu danas. Iz tog razloga, davateljima mrežnih usluga u cilju je pružiti što bolju uslugu krajnjim korisnicima, a da pritom efikasno upravljaju ograničenim mrežnim resursima. Podaci koji utječu na iskustvenu kvalitetu korisnika su u mreži enkriptirani te davatelji mrežnih usluga nemaju uvid u njih. Rješenje za praćenje iskustvene kvalitete pronalazi se u metodama strojnog učenja. Međutim, korisničko ponašanje prilikom gledanja videozapisa u obliku interakcija utječe na uzorke u mrežnom prometu pa samim time i na procjenu iskustvene kvalitete. Stoga, u ovom radu je ispitana mogućnost definiranja mrežnih značajki koje ne ovise u postojanju korisničkih interakcija prilikom gledanja videozapisa na YouTube-u, te izgradnje modela za procjenu iskustvene kvalitete koji adresiraju slučajeve s i bez korisničkih interakcijama.

Ključne riječi: YouTube, iskustvena kvaliteta, video na zahtjev, strojno učenje

Summary

Nowadays, video on demand (*VOD*) streaming is one of the most popular activities on the Internet. Therefore, a challenge faced by network operators is to fulfill customer demands and expectations while efficiently managing their limited network resources. Data affecting the Quality of Experience (*QoE*) are encrypted, so network operators do not have direct insight into them. Solutions which include machine learning methods in predicting *QoE* based on the analysis of encrypted network traffic are promising. However, customer interactions while watching *VoD* affect network traffic, and consequently *QoE* estimation. Hence, this thesis explores possibilities of defining network features which do not depend on customer interactions while watching YouTube videos, and training models for *QoE* estimation that include data which may and may not depend on customer interactions.

Key words: YouTube, Quality of Experience, video on demand, machine learning

Dodaci

Popis slika

Slika 1.1 - Broj korisnika društvenih mreža [4]	3
Slika 1.2 - Prilagodljivo strujanje putem protokola <i>HTTP</i>	6
Slika 2.1 – Primjer faktora koji utječu na zadovoljstvo korisnika umrežene usluge	10
Slika 3.1 - <i>ER</i> dijagram tablice video	15
Slika 3.2 - Isječak iz tablice <i>video</i> : format ISO 8601	16
Slika 3.3 - Procedura za pretvorbu trajanja videozapisa iz ISO 8601 formata u sekunde	16
Slika 3.4 - Graf koji prikazuje odnos mogućnosti reproduciranja videozapisa u visokoj i niskoj kvaliteti	17
Slika 3.5 - Dohvat dostupnih formata za pojedini videozapis pomoću YouTube-dl-a	19
Slika 3.6 - Kreiranje projekta na vlastitom Google računu	20
Slika 3.7 - API ključ	20
Slika 3.8 - Obavijest o suspendiranom API projektu	22
Slika 4.1 - Prikupljanje i obrada podataka	23
Slika 4.2 - Shema laboratorijskog okruženja	24
Slika 4.3 - Isječak koda - otvaranje Appium sjednice	25
Slika 4.4 - Isječak koda - otvaranje Statistike za štrebere	26
Slika 4.5 - Isječak koda - dohvat videozapisa iz baze podataka	27
Slika 4.6 - Isječak koda - otvaranje YouTube videozapisa	28
Slika 4.7 - Gumb za kopiranje Statistike za štrebere	28
Slika 4.8 - Isječak koda - kopiranje Statistike za štrebere	29
Slika 4.9 - Isječak koda - odabir interakcije	29
Slika 4.10 - Gumb za pauziranje videozapisa	30
Slika 4.11 - Isječak koda - interakcija pauziranja videozapisa	31
Slika 4.12 – Reprodukcijska lenta videozapisa	32
Slika 4.13 - Isječak koda - premotavanje videozapisa unaprijed	32

Slika 4.14 - Gumb za prebacivanje videozapisa na sljedeći videozapis...	33
Slika 4.15 - Isječak koda - prebacivanje na sljedeći videozapis	34
Slika 4.16 - Appium arhitektura [30]	35
Slika 4.17 - Topologija u IMUNES-u	36
Slika 4.18 - Prikaz grafičkog sučelja IMUNES-a [37]	37
Slika 4.19 - Primjer zapisa jedne sekunde videozapisa	38
Slika 4.20 - Obradeni podaci s aplikacijske razine	39
Slika 4.21 – Broj uzoraka videozapisa u pojedinoj klasi za obrađene podatke videozapisa bez interakcija	40
Slika 4.22 - Broj uzoraka videozapisa u pojedinoj klasi za obrađene podatke videozapisa s interakcijama	40
Slika 4.23 - Obradeni podaci iz .pcap datoteke.....	41
Slika 5.1 - K-struka unakrsna validacija	45
Slika 5.2 - Matrica zabune	47
Slika 5.3 – Datoteka ARFF formata [45]	49
Slika 5.4 - Skup podataka videozapisa bez interakcija za odabir značajki	50
Slika 5.5 - Skup podataka videozapisa bez interakcija za treniranje modela	50
Slika 5.6 – Grafovi usporedbe skupa podataka bez i s interakcijom pauze za prikaz vrijednosti klase rezolucije za odabrane značajke mrežnog prometa	60
Slika 5.7 - Grafovi usporedbe skupa podataka bez i s interakcijom pauze za prikaz vrijednosti klase prosječne brzine kodiranja za odabrane značajke mrežnog prometa	61
Slika 5.8 - Grafovi usporedbe skupa podataka bez i s interakcijom pauze za prikaz vrijednosti klase zastajkivanja za odabrane značajke mrežnog prometa	62
Slika 5.9 - Grafovi usporedbe skupa podataka bez i s interakcijom premotavanja videozapisa unaprijed za prikaz vrijednosti klase rezolucije za odabrane značajke mrežnog prometa.....	63

Slika 5.10 - Grafovi usporedbe skupa podataka bez i s interakcijom premotavanja videozapisa unaprijed za prikaz vrijednosti klase prosječne brzine kodiranja za odabrane značajke mrežnog prometa 64

Slika 5.11 - Grafovi usporedbe skupa podataka bez i s interakcijom premotavanja videozapisa unaprijed za prikaz vrijednosti klase zastajkivanja za odabrane značajke mrežnog prometa 65

Slika 5.12 - Grafovi usporedbe skupa podataka bez i s interakcijom prekida reprodukcije i prebacivanja na sljedeći videozapis za prikaz vrijednosti klase rezolucije za odabrane značajke mrežnog prometa 67

Slika 5.13 - Grafovi usporedbe skupa podataka bez i s interakcijom prekida reprodukcije i prebacivanja na sljedeći videozapis za prikaz vrijednosti klase prosječne brzine kodiranja za odabrane značajke mrežnog prometa 68

Slika 5.14 - Grafovi usporedbe skupa podataka bez i s interakcijom prekida reprodukcije i prebacivanja na sljedeći videozapis za prikaz vrijednosti klase zastajkivanja za odabrane značajke mrežnog prometa 69

Popis tablica

Tablica 2.1 - Pet kategorija QoS parametara [15].....	9
Tablica 2.2 - <i>MOS</i> vrijednosti.....	11
Tablica 4.1 - Popis mrežnih značajki koje će se koristiti za klasifikaciju <i>KPI</i> -eva [39]	42
Tablica 5.1 - Tablica prikaza performansi modela izgrađenog na podacima videozapisa bez interakcija pomoću značajki odabranih na skupu podataka bez interakcija	51
Tablica 5.2 - Tablica uspješnosti ponovne procjene modela na podacima videozapisa s interakcijama izgrađenog na podacima videozapisa bez interakcija pomoću značajki odabranih na skupu podataka bez interakcija .	52
Tablica 5.3 - Tablica prikaza performansi modela izgrađenog na podacima videozapisa bez interakcija pomoću značajki odabranih na mješovitom skupu podataka	55
Tablica 5.4 - Tablica uspješnosti ponovne procjene modela na podacima videozapisa s interakcijama izgrađenog na podacima videozapisa bez interakcija pomoću značajki odabranih na mješovitom skupu podataka	56
Tablica 5.5 - Značajke odabrane na skupu podataka videozapisa bez interakcija.....	57
Tablica 5.6 - Značajke odabrane na mješovitom skupu podataka i presjek značajki s onima koje su odabrane na skupu podataka videozapisa bez interakcija.....	58

Popis oznaka i kratica

<i>QoE</i>	<i>Quality of experience</i>
<i>QoS</i>	<i>Quality of service</i>
<i>ICT</i>	<i>Information and communications technology)</i>
<i>ITU-T</i>	<i>International Telecommunication Union-Telecommunication</i>
<i>KPI</i>	<i>Key Performance Indicators</i>
<i>MOS</i>	<i>Mean Opinion Score</i>
<i>HTTP</i>	<i>HyperText Transfer Protocol</i>
<i>ID</i>	<i>identifikator</i>
<i>ER</i>	<i>Entity Relationship</i>
<i>API</i>	<i>Application programming interface</i>
<i>WiFi</i>	<i>Wireless Fidelity</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>JSON</i>	<i>JavaScript Object Notation</i>
<i>adb</i>	<i>Android Debug Bridge</i>