

Универзитет у нишу

Електронски факултет

- пројекат -

**АНАЛИЗА СЕНТИМЕНАТА**

**ПРЕДМЕТ**: ОБРАДА ПРИРОДНИХ ЈЕЗИКА

**Ментор**: Сузана Стојковић **Студент:** Филип Трајковић 1574

Ниш, 2024.

Садржај

[1. Увод 3](#_Toc159279243)

[2. Анализа сентимената у обради природних језика 4](#_Toc159279244)

[3. Практични део 7](#_Toc159279245)

[3.1. Скупови података 7](#_Toc159279246)

[3.2. Изворни код 8](#_Toc159279247)

[3.3. Типови мреже 9](#_Toc159279248)

[3.4. Тренирање мреже 11](#_Toc159279249)

[3.4.1. Тренирање GRU мрежа 11](#_Toc159279250)

[3.4.2. Тренирање LSTM мрежа 16](#_Toc159279251)

[4. Закључак 17](#_Toc159279252)

[5. Литература 18](#_Toc159279253)

# Увод

У савременом добу где се вештачка интелигенција широм примењује у различитим областима софтверског развоја, кључно је разумети њен начин функционисања и препознати потенцијалне могућности за њену примену. Да бисмо идентификовали те могућности, прво је битно стећи увид у различите технологије, савладати бројне концепте који су развијени у последње време, а такође и разумети како применити одређене технологије у конкретним сценаријима. Кроз ова разумевања, могуће је креирати систем вештачке интелигенције способан за решавање различитих проблема.

Системи за детекцију, разумевање и контекстуализацију текстуалних записа представљају широку област која се веома брзо развија почетком треће деценије 21. века. Убрзани развој и помама за оваквим системима креће крајем 2022. године са појавом до сада невиђеног система за обраду текстова под називом „ChatGPT“. Широке корисничке масе које су се уз помоћ овог система упознале са могућности коју нуде системи за обраду текстова, додатно су убрзале даљи развој и напредак у обради природних језика стварајући нове случајеве коришћења овакве технологије.

Анализа сентимената представља један од основних случајева коришћења вештачке интелигенције у области обраде природних језика. Користе се процеси класичног машинског учења и процеси обраде природних језика како би се креирала једна симбиоза различитих области са циљем решавања специфичних проблема попут анализе сентименталних особина текстуалних записа. Анализа сентимената текстуалних записа представља област надгледаног учења у машинском учењу при чему је неопходно да се за сваки запис у скупу података наведе и припадност категорији којој припада наведени текст. Сви записи у скупу података требају припадати коначном скупу категорија. Решавање проблема анализе сентимената се своди на проблем класификације у машинском учењу.

Даљи текст овог рада је подељен на теоријски и практични део. У теоријском делу рада су обрађене технике обраде текстуалних података, док су у практичном делу рада приказани резултати ових обрада.

Теоријски део рада обухвата основну теоријску позадину иза техника обраде текстуалних података, док практични део рада обухвата примену наведених техника над специфичним скупом података са применом класификационих алгоритама машинског учења, као и уз помоћ претренираних језичких модела.

# Анализа сентимената у обради природних језика

* 1. Теоријске основе анализе сентимената

Анализа сентимената представља област у обради природних језика која се бави проучавањем означених скупова података при чему се скуп података састоји од текстуалног записа и категоричног атрибута који представља категорију припадности одређеној категорији у скупу коначних категорија.

Концепт анализе сентимената као специфичног случаја обраде текстуалних података представља процес утврђивања квалитативних својстава текстуалних записа како би се из датог скупа података извеле одређене законитости које важе између контекстуалних репрезентација посматраних записа и категоричког атрибута који представља припадност одређеном скупу.

Циљ овог процеса јесте утврђивање емоционалног контекста датих текстова и тежину коју речи природног језика имају у односу на стање сентимента који описују. Ставови тј. контекст који се преноси кроз текст се обично означавају у неколико категорија попут:

* Позитивно
* Негативно
* Неутрално
  1. Примена анализе сентимената

Процес утврђивања сентемената у тексту са собом доноси нове случајеве примене оваквог типа технологија у различитим сферама. Примена анализе сентимената има највећих бенефита у сферама које се баве обрадом и анализом мишљења корисника, оцену одређених ентитета (брендова, производа, итд.), нивоима одушевљења или разочарења посматраног корисника и сл. Овакви случајеви коришћења могу додатно допринети анализи одређених појава јер до сада није било могуће аутоматизовати анализу и процесирање текстова на овај начин.

Овакав аутоматизован начин обраде великих текстуалних скупова података са собом доноси убрзани начин обраде и анализе различитих текстова.

Коначни резултати утврђивања контекста текстова имају за циљ презентовање јасније слике о стеченим мишљењима различитих корисника над датим објектима посматрања при чему се у анализу могу укључити и додатни параметри који могу додатно категоризовати типове личности корисника.

* 1. Технике обраде природног језика

Технике обраде природног језика представљају технике којима се обрађују подаци који се у изворном облику налазе у текстуалном формату. Оваквим техникама се врши обрада и припрема података за даље кораке у развоју процеса анализе сентимената попут примене математичких метода машинског учења или тренирања модела.

Обзиром да над подацима који се изворно налазе у текстуалном формату није могуће применити споменуте даље кораке, такве податке је најпре неопходно претворити у погодан формат који се може обрађивати на хардверу који се користи у машинском учењу. Погодан формат јесте векторски/тензорски простор у којем је неопходно представити текстуални контекст посматраног скупа података.

Обрађени текстуални подаци постају вектори који се у векторском/тензорском простору представљају као низови бројева одређене дужине. Вредности које ови вектори носе са собом требају на високо-квалитативан начин описати дати текстуални податак при чему се требају одржати својства текстова попут редоследа речи у реченици, контекстуалне зависности између речи, дужина текстова, контекстуална сличности и сл.

* + 1. Препроцесирање података

Препроцесирање података се бави основним чишћењем улазних текстуалних података и подразумева неке од следећи техника:

* Чишћење текста
* Нормализација
* Токенизација
* Лематизација и стемизација
* Уклањање стоп-речи

Чишћење текста

Чишћење текста је први корак у препроцесирању текстуалних података и подразумева уклањање специјалних карактера из текста попут HTML тагова, знакова интерпункције, бројева и осталих не-текстуалних карактера. На овај начин се врши елиминација сувишних симбола из текста и крера једноставан и чистији текст.

Нормализација

Нормализација представља корак у препроцесирању текстуалних података при чему се врши конверзија текста у јединствени тј. стандардни формат за све карактере текста. Основни процес у номализацији јесте *case* конверзија при чему се текст претвара у *lower* или *upper case.*

Токенизација

Токенизација представља корак у препроцесирању текстуалних података при чему се улазна реченица дели на листу краћих текстуалних јединица које могу бити појединачне речи, реченице или параграфи. Токени представљају смислену целину која представља заједнички контекст одређене целине токенизоване реченице. У највећем броју случајева се токенизација врши поделом улазне реченице на листу речи. Оваква листа је погодна за даље кораке у препроцесирању података.

Лематизација и стемизација

# Практични део

Пројекат се састоји од два подфолдера са изворним кодом под радним називом **Code** и подацима за обраду под радним називом **Data**.

## Скупови података

У подфолдеру са подацима под радним називом **Data** се налази неколико необрађених и обрађених скупова података. Над подацима за обраду је претходно одрађена процедура екстракције како би се добила прилагођена структура података по колонама спремним за даље коришћење. Улазни необрађени подаци се налазе у фолдерима **link.csv** и **gauge.csv** .

Најбитнији скуп података из **Data** фолдера који се користи у даљем току рада овог пројекта јесте **link\_fixed.csv** . Овај скуп података садржи све неопходне фичере неопходне за тренирање различитих конфигурација неуронских мрежа.

Скупови података представљају следеће:

- **link.csv** - скуп необрађених података који садржи најбитније фичере у формату који не одговара траженом како би се успешно спровело тренирања мрежа. Да би се успешно спровело тренирање мреже неопходна је екстракција података у формат који одговара траженом

- **link\_fixed.csv -** представља обрађени скуп података са подацима у формату који одговара траженом како би се успешно извршила анализа података и тренирање мрежа

- **gauge.csv** - скуп необрађених података који садржи само бинарну информацију о томе да ли у одређеном временском запису пада киша или не. Подаци се налазе у формату који не одговара траженом како би се успешно спровело тренирања мрежа те је због тога неопходна екстракција података у формат који одговара траженом

- **gauge\_fixed.csv -** представља скуп обрађених података који се могу користити за анализу и тренирање мреже

- **link\_gauge\_fixed\_merged.csv -** представља обједињени скуп података link\_fixed.csv и gauge\_fixed.csv скупова података по заједничком параметру за временски запис (timestamp)

- **link\_fixed\_with\_A.csv -** представља скуп података изведен из link\_fixed.csv скупа података при чему је изведен додатни фичер А који представља разлику два основна фичера (baseline и rsl)

Наведени процеси обраде и екстракције података су извршени на почетку израде пројекта и изворни код ових процеса се налази у подфолдеру Code/Dataset preprocessing.

Улазни скуп података се налази у фајлу **link\_gauge\_fixed\_merged.csv** и у њему се налазе записи за временске серије са следећим колонама:

- **Timestamp**: време бележења података

- **Rsl**: Ниво примљеног сигнала

- **Baseline**: Средња вредност у том тренутку

- **Wet\_dry**: Лабела да ли има кише или нема (wet\_dry=0 - нема кише, wet\_dry=1 - имаа кише)

- **Rainrate**: Када је овај параметар једнак 0, онда је wet\_dry=0, а када је различито од нуле тада је wet\_dry = 1

## Изворни код

Изворни код садржи радне свеске за обраду података, анализу података као и тренирања различитих типова неуронских мрежа. Изворни код се налази у подфолдеру **Code,** док се пројекат даље дели по следећим подфолдерима:

- **Dataset preprocessing -** садржи изворни код уз помоћ којег је извршена екстракција и форматирање скупова података у формат који је погодан за коришћење

- **GRU -** садржи изворни код који служи за тренирање две резличите конфигурације са GRU типовима слојева неуронских мрежа

- **Helpers -** представља скуп помоћних функција које се користе у процесу обраде и анализе података, као и у процесу тренирања модела

- **LSTM -** садржи изворни код који служи за тренирање две резличите конфигурације са LSTM типовима слојева неуронских мрежа

- **MLPRegressor -** садржи изворни код и запамћене објекте који се користе за тренирање мреже уз помоћ MLPRegressor објекта

Најпре је обрађена експланаторна анализа података уз помоћ које су утврђене одређене законитости које важе у датом скупу података. Матрицом корелације је додатно испитана линеарна зависност која влада између улазних и излазних података у скупу података.

Резултати ЕDA анализе су дати у Јupyter Notebook радним свескама под радним називом EDA.ipynb.

Током анализе података утврђена је висока негативна корелација између фичера *A(baseline - rsl)* и *rsl* као и негативна корелација између *rsl* и *wet\_dry*.

Позитивна корелација је уочена код између фичера *rsl* и *baseline,* као и између фичера *A(baseline - rsl)* и *wet\_dry* .

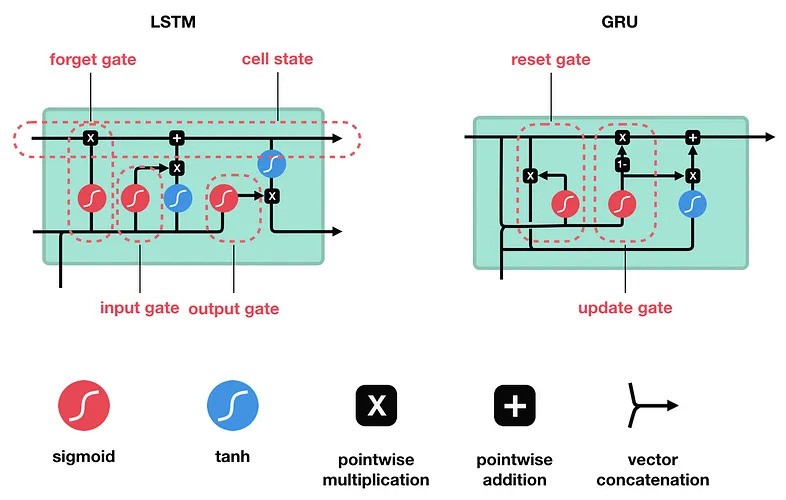
## Типови мреже

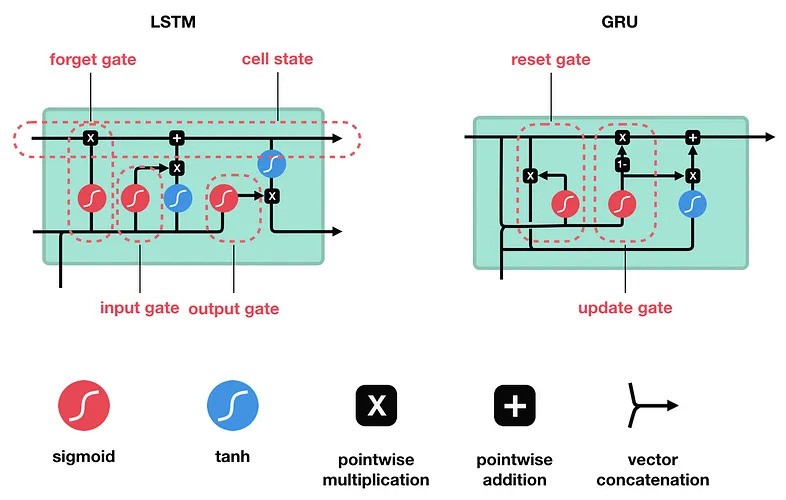
Процес тренирања је примењен на два различита типа мреже и то:

- **LSTM**

- **GRU**

LSTM (Long short-term Memory) представља тип чвора који служи за креирање LSTM слојева неуронске мреже. Креиран је како би решио проблем дугорочних зависности и заборављања информација код рекурентних неуронских мрежа. Користи се код решавања проблема који се односе на зависне секвенце међусобно условљених података као што су текстови, временске серије или геномски подаци.



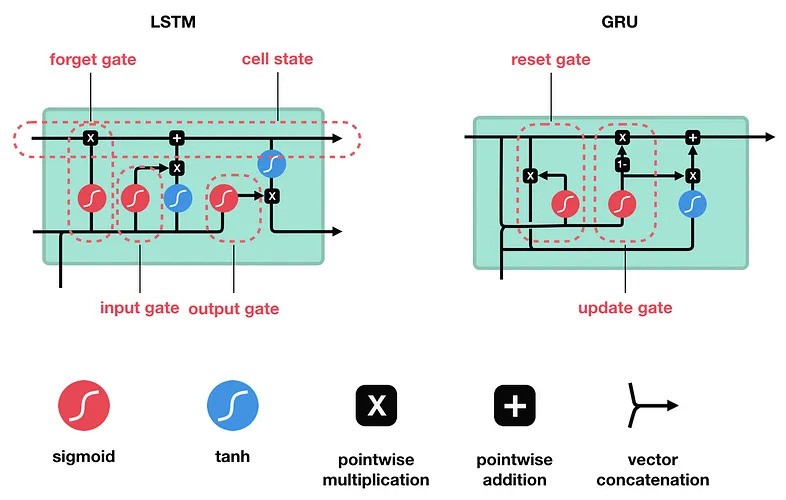


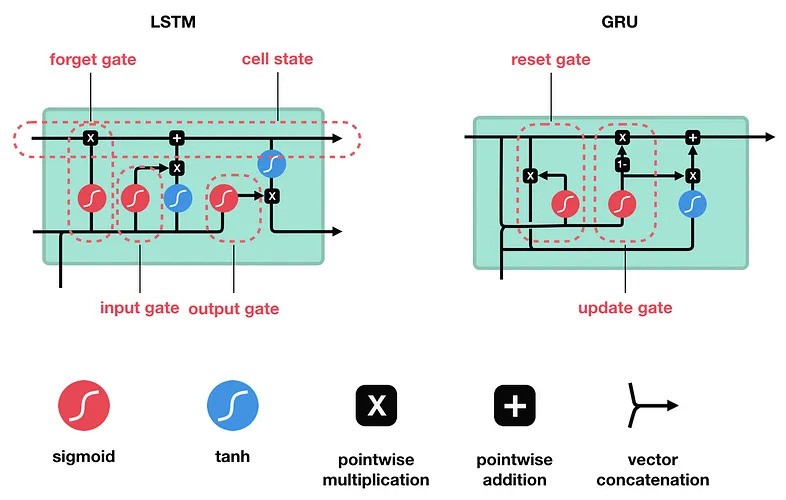
Сл. 1 – Архитектура LSTM чвора неуронске мреже

Главне компоненте овог чвора представљају:

* **Стање чвора** (eng. *Cell-State*): представља стање које се добија применом различитих нелинеарних функција и векторских операција над улазима чвора
* **Скривено стање** (eng. *Hidden-State*) – представља меморију или излаз чвора који се пропагира на улаз наредног чвора, а рачуна се на основу текућег стања чвора, претходног излазног стања и текућег улаза
* **Улазна капија** (eng. *Input-Gate*) – представља улазни логички блок LSTM чвора које се отвара у тренутку *t*. Овај логички блок служи како би ажурирао тренутно стање чвора и одређује које ће се вредности додати на текуће стање
* **Капија заборављања** (eng. *Forget Gate*) – представља капију која служи за одређивања које ће се информације задржати или одбацити на основу информација из претходнох скривеног слоја и улаза тренутног чвора

GRU (eng. *Gated Reccurent Units*) представља сличан тип чвора као и LSTM чвор и примењује се код рекурентних неуронских мрежа. Примењује се код решавања проблема дугорочних зависности између континуалних података. Користи механизам капија за контролу тока података кроз сваки појединачни чвор неуронске мреже. Представља једноставнији тип чвора од LSTM чвора јер комбинује стање чвора и скривено стање у једно заједничко стање.





Сл. 2 – Архитектура GRU чвора неуронске мреже

Главне компоненте овог чвора представљају:

* **Скривено стање** (eng. Hidden Layer) – представља слој у којем се врши рекурентно ажурирање информација у току сваког проласка информација. Ово стање се ажурира у зависности од текућег улаза чвора и претходног скривеног стања. Скривено стање представља меморију чвора који рачуна информације и прослеђује их следећем слоју неуронске мреже
* **Капија за ажурирање** (eng. Update Gate) – представља слој за ажурирање текућег стања чвора тако што непотребне информације избацује, а нове информације додаје у текуће стање. На улазу обрађује претходно скривено стање и текући улаз у чвор
* **Капија за ресетовање** (eng. Reset Gate) – представља слој GRU чвора који служи за одбацивање непотребних информација из претходног скривеног стања. На улазу се налази текући улаз у чвор и претходно скривено стање

## Тренирање мреже

За сваки од типова мреже извршено је тренирање са и без изведеног параметра А(baseline - rsl) који представља разлику нивоа примљеног сигнала и средње вредности примљеног сигнала у том тренутку. Тренирање је обављено са различитим конфигурацијама мреже.

Како ручним модификовањем конфигурација није успешно генерисана најбоља конфигурација са високим нивоом тачности алгоритма, прешао сам на аутоматско модификовање конфигурације мреже како би се утврдила најбоља конфигурација мреже која ће дати најбоље резултате предикције.

Аутоматско модификовање мреже је успостављено уз помоћ GridSearchCV методологије са унапред дефинисаним могућностима за различите хипер-параметре које мрежа може да прихвати. Уз помоћ ове методологије успешно је генерисана најбоља конфигурација са високим нивоом тачности предикције.

### Тренирање GRU мрежа

Тренирање мреже са GRU слојем је извршено у фајловима:

* project\_GRU\_with\_A(baseline-rsl) 2.0.ipynb
* project\_GRU\_without\_A(baseline-rsl) 2.0.ipynb

Оба примера тренирања се налазе у фолдеру Projekat-Detekcija\_padavina/Code/GRU .

Тренирање се састоји од неколико фаза, а то су:

* Учитавање библиотека
* Препроцесирање података
* Тренирање мреже ручним подешавањима хипер-параметара
* Тренирање мреже аутоматским подешавањем хипер-параметара

Учитавање библиотека – учитавање свих неопходних библиотека за рад са неуронским мрежама попут pandas, keras, tensorflow; као и осталих попут pandas, matplotlib, pickle, os, sys, datetime…

Препроцесирање података – представља фазу учитавања података при чему се у овој фази врши сва неопходна обрада и припрема података за касније тренирање мрежа.

Кораци у овој фази су:

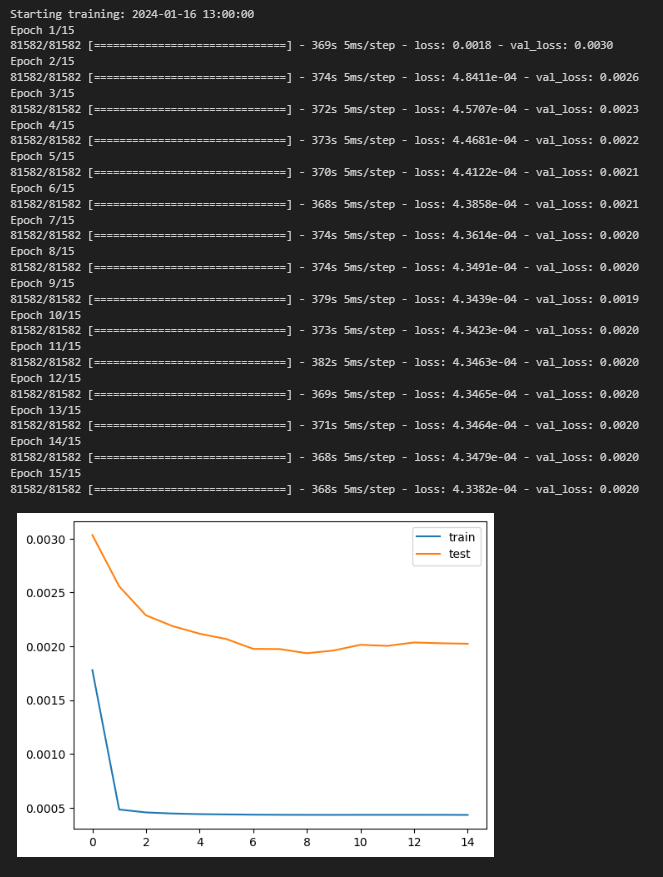
* учитавање фајла
* извацивање редова са непостојећим вредностима
* одстрањивање timestamp податка
* скалирање података MinMaxScaler методом
* подела скупа података на тренинг и тест скуп
* Креирање специјалног TimeSeriesGenerator објекта који служи за чување података у формату погодном за тренирање временских серија; представља податке у три димензије

Тренирање мреже ручним подешавањем хипер-параметара – представља покушај да се ручно пронађе конфигурација хипер-параметара која ће имати најбоље резултате код тренирања мреже. Са ручном променом хипер-параметара није било могуће утврдити тачно најбоље хипер-параметре те се због тога прешло на аутоматско генерисање различитих конфигурација.

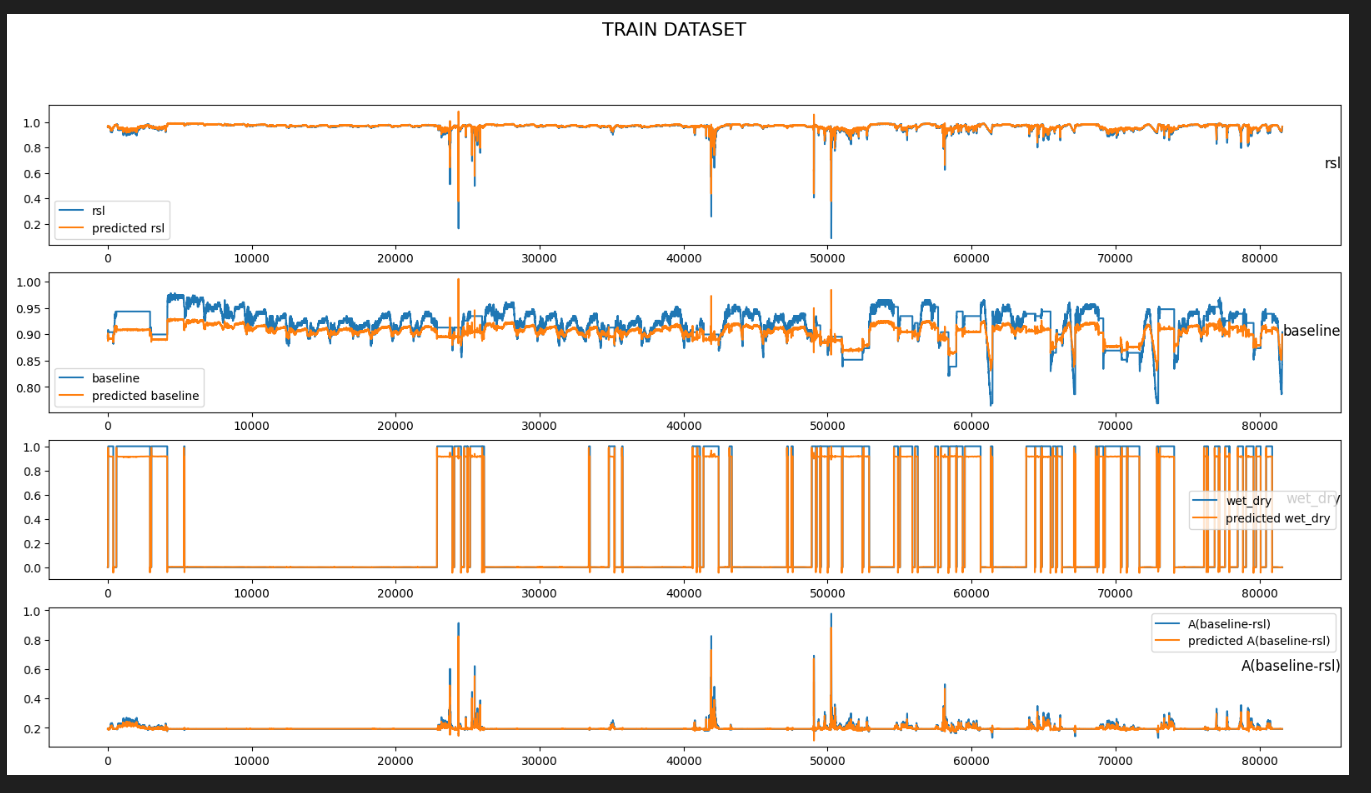
Приликом ручног подешавања хипер-параметара, на графику се приказује вредност губитака по епохама, као и график предикције свих излазних вредности и на тренинг скупу и на тест скупу података.

Иако се на графицима предикције на тренинг и тест подацима врло добро врши предикција wet\_dry параметра, остали параметри не добијају тако добре резултате, те је због тога идеја да се утврди још боља комбинација хипер-параметара.

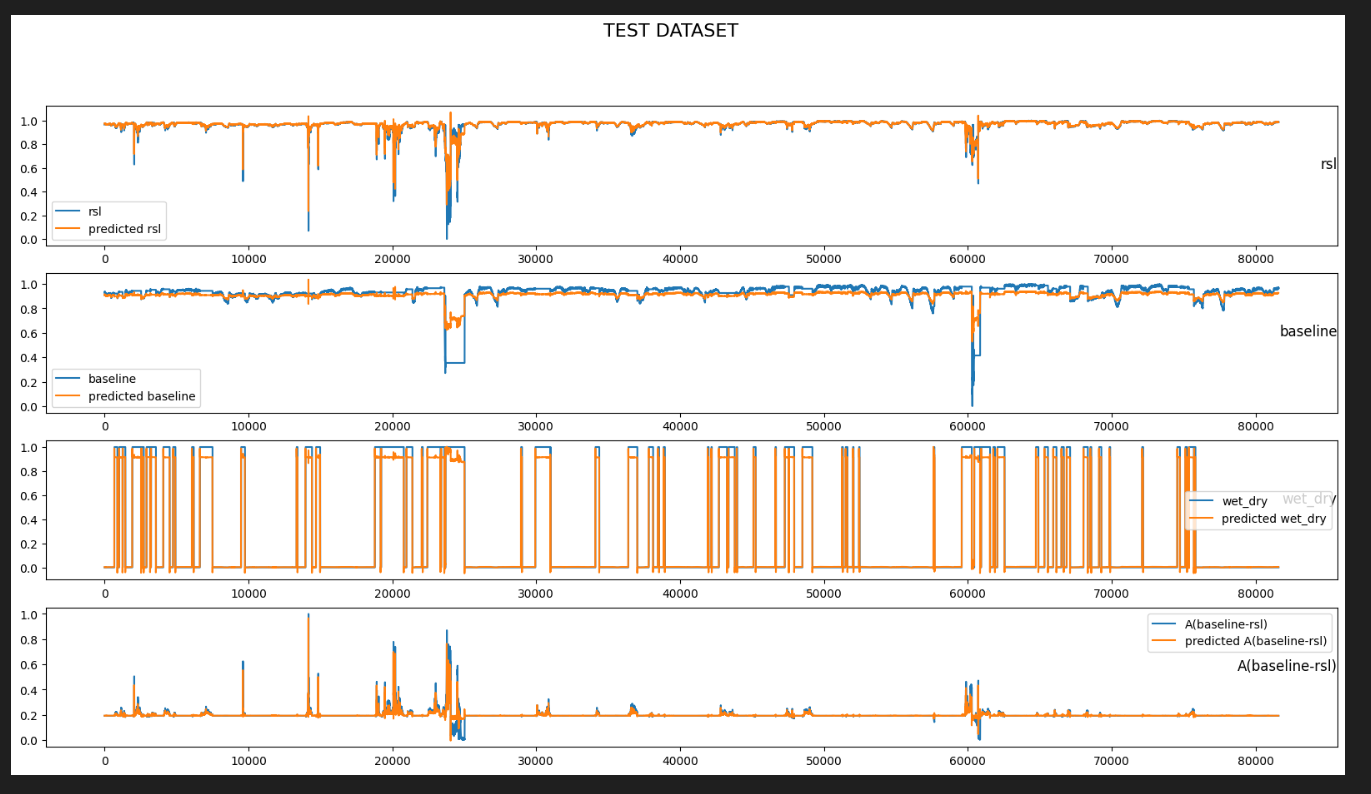
Примери резултата једне од конфигурација (Units=50, LR=0.0001, Epoch=15):



Сл. 3 – Тренирање и приказ промене тренинг и валидационих губитака по епохама



Сл. 4 – Предикција вредности на тренинг скупу



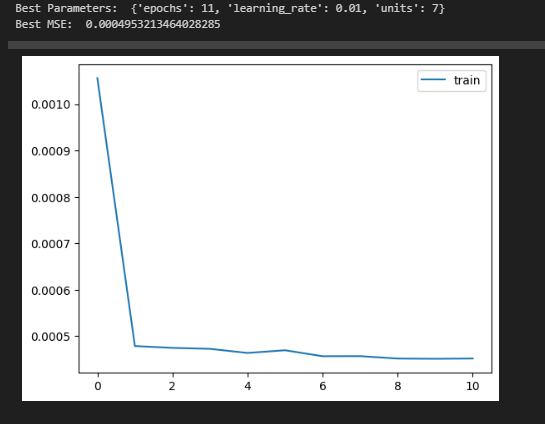
Сл. 5 – Предикција вредности на тест скупу

Тренирање мреже аутоматским подешавањем хипер-параметара – (или оптимизација хипер-параметара) представља процес тренирања мреже при чему се врши промена конфигурације параметара мреже како би се утврдила конфигурација са најмањим губицима током тренирања.

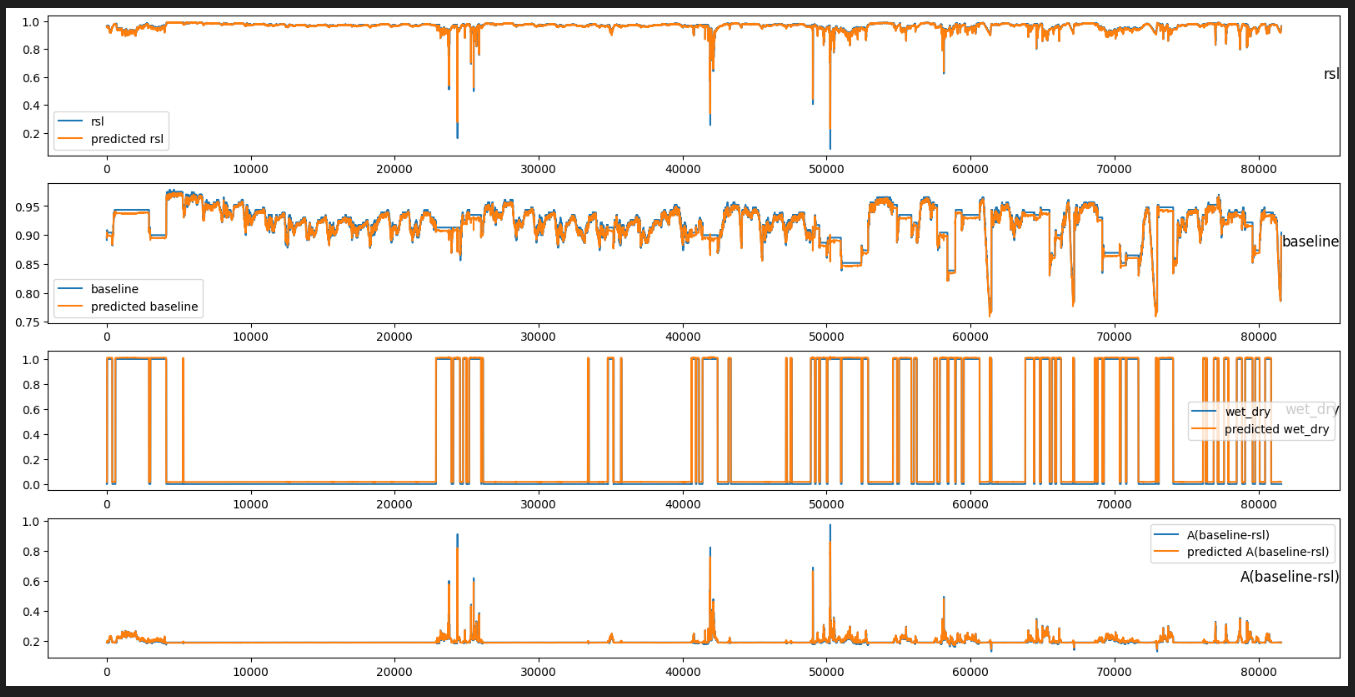
Кораци оптимизације хипер-параметара:

* креирање основне структуре мреже помоћу функције create\_gru\_model
* креирање KerasRegressor објекта
* дефинисање вредности хипер-параметара за оптимизацију помоћу param\_grid
* Креирање GridSearchCV објекта
* ручна екстракција бечева из train\_generator објекта и припрема истих за тренирање
* тренирање оптимизацијом хипер-параметара
* приказ свих резултата промене губитака, као и предикција на тренинг и тест скуповима уз помоћ модела са најбољом комбинацијом хипер-параметара

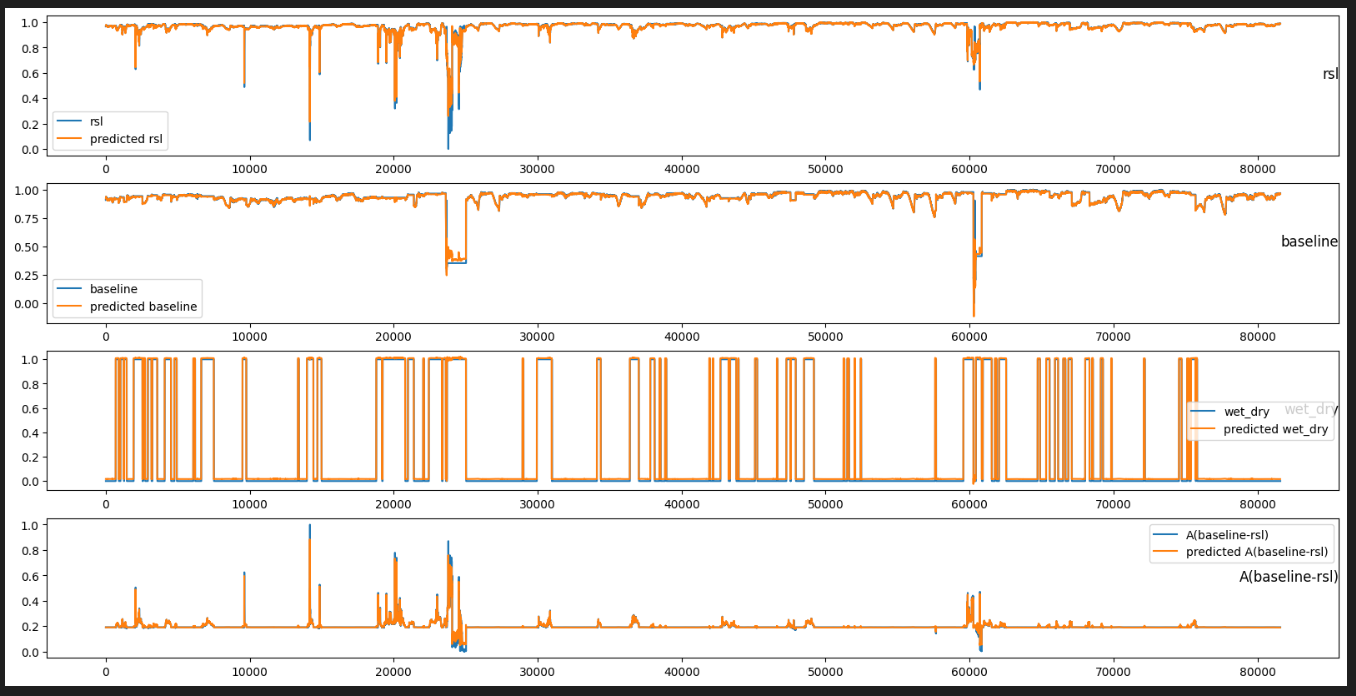
Примери резултата најбоље конфигурације:



Сл. 5 – Изабрани хипер-параметри и промена губитака по епохама



Сл. 4 – Предкција вредности на тренинг скупу



Сл. 4 – Предкција вредности на тест скупу

### Тренирање LSTM мрежа

Тренирање мреже са LSTM слојем је извршено у фајловима:

* project\_LSTM\_with\_A(baseline-rsl) 2.0.ipynb
* project\_LSTM\_without\_A(baseline-rsl) 2.0.ipynb

Оба примера тренирања се налазе у фолдеру Projekat-Detekcija\_padavina/Code/LSTM .

Тренирање се састоји од идентичних фаза као и у тачки 2.4.1. , а то су:

* Учитавање библиотека
* Препроцесирање података
* Тренирање мреже ручним подешавањима хипер-параметара
* Тренирање мреже аутоматским подешавањем хипер-параметара

Процес тренирања LSTM се своди на исти поступак као и у претходној тачки уз одређене ситне промене.

Код тренирања LSTM мреже без изведеног параметра А, одрађено је тренирање на редукованом скупу података са 2500 тренинг података и 2500 тест података. Тренирање на овако малом броју података је дало изразито добре резултате уколико се узме у обзир обим скупа података. Резултате су приказани у Jupyter радним свескама.

Друга значајна измена је тренирање аутоматским подешавањем хипер-параметара уз одстрањивање “cross-validation” методе, тј. методе преклапања скупова података. Ово одстрањивање је убрзало процес тренирања при чему су тренинг подаци очували континуалност у временској серији.

# Закључак

Приликом израде овог пројекта, било је неопходно упознати се са различитим процесима рада са скуповима података заснованим на временским серијама. Иако сам поседовао одређена предзнања из области неуронских мрежа и тренирања мрежа, значајну промену је представљало генерисање тродимензионалних података како би се подаци припремили за тренирање мрежа заснованим на временским серијама са LSTM и GRU слојевима. Изучавање LSTM и GRU слојева је неопходно у разумевању обраде серијских података и начину функционисања таквих типова неуронских мрежа.

Радом на овом пројекту сам стекао нова знања из области неуронских мрежа, обраде времеских серија и рада на креирању специфичности са таквим типовима података.

# Литература

1. „Коришћење неуронских мрежа за детекцију падавина на основу нивоа примљеног сигнала комерцијалног микроталасног линка“, Владица Ђорђевић, Оливера Пронић-Ранчић, Златица Маринковић, Марија Милијић, Вера Марковић, Uwe Siart
2. “New Method for Detection of Precipitation Based on Artificial Neural Networks”, Vladica Đorđević , Olivera Pronić-Rančić , Zlatica Marinković , Marija Milijić , Vera Marković2, Uwe Siart, Christian Chwala, Harald Kunstmann