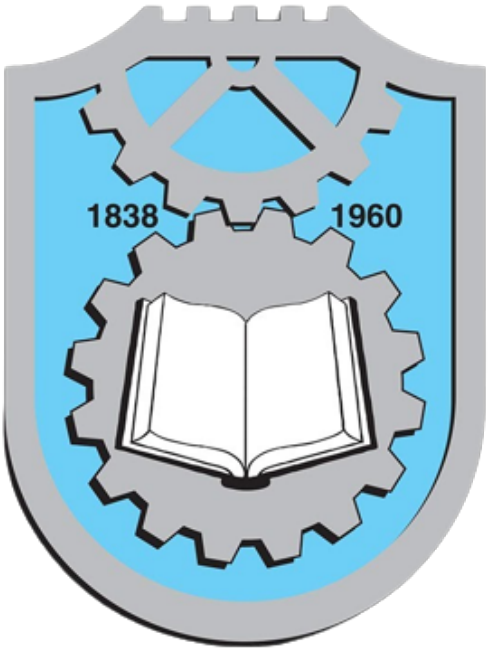
Универзитет у Крагујевцу
Факултет инжењерских наука



Неуронске мреже

Документација

Четврти домаћи задатак
Предвиђање цена акција на берзи

Професор:
Проф. др Весна Ранковић

Студент:
Каришић Ђорђе 393/2023

__.__.2023.

Садржај

1 Увод

Предвиђање цена акција на тржишту акција је изазован задатак, како због сложености података тако и због сложености самих зависности између њих. Подаци, често представљени графичким форматом, значајно зависе временски једни од других.

Цена у тренутку t може зависити од цене у тренутку $t-1$, али и $t-2$ и тако даље до неког крајњег $t-\alpha$ тренутка, али може бити и независна, при чему се претпоставља јак екстерни утицај на тржиште.

Како је потребно моделирати сложене временске зависности, предлаже се употреба дубоког учега, конкретно модела неуронских мрежа – мрежа са дуготрајном краткорочном меморијом.

Овај тип мрежа, уз помоћ своје специфичне и једноставне структуре може моделирати веоме дугорочне зависности у подацима и тиме адекватно апроксимирати функцију рачунања излаза на основу улаза чије су вредности временске серије. Једноставније речено, на основу цене претходних $N-1$ дана, овај тип модела може прецизно и брзо одредити цену у N дану.

Такав модел на улазу треба имати адекватне податке, уређене, скалиране и "чисте" од било каквих аномалија и нежељених вредности – потребно их је адекватно процесирати.

Како би се пружио увид о сложености, структури и везама међу подацима из скупа података потребно је, такође, визуелизовати скуп података, специфично за сврху коју ће обављати. Како се у овом случају ради о подацима који представљају цену акције у датом временском тренутку на тржишту акција, потребно их је испитати у таквом окружењу.

Скуп података употребљен у оквиру овог рада је CAC40 Dataset (preprocessed), скуп временски зависних података сакупљених са Француског тржишта акцијама за 40 најрелевантнијих компанија – од којих је за овај рад одабрана Peugeot.

2 Анализа скупа података

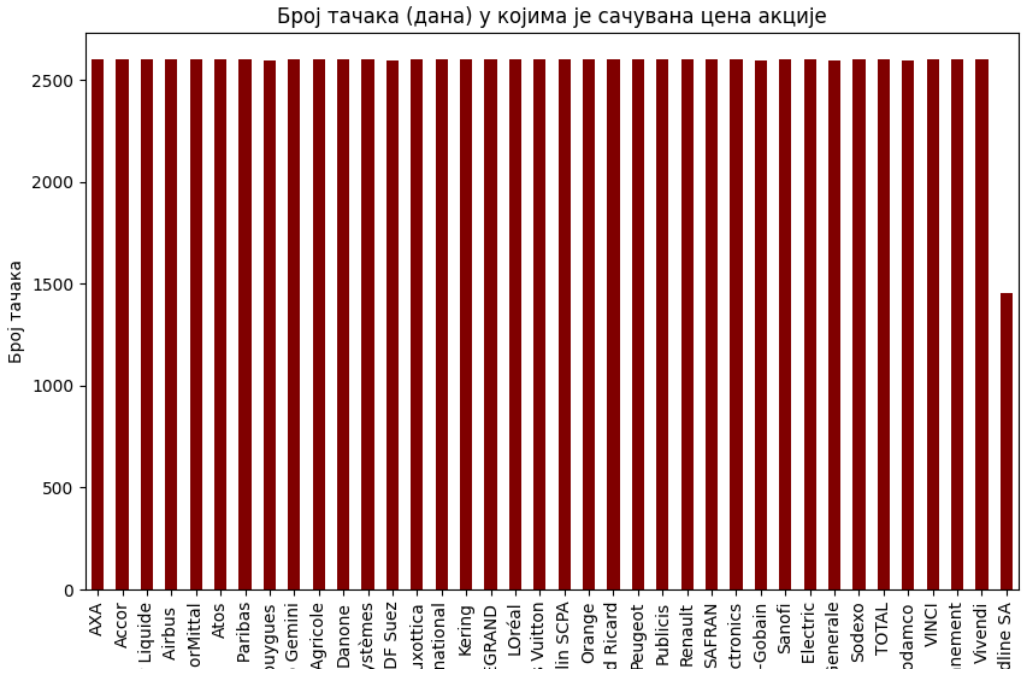
Скуп података састоји се из инстанци које карактеришу наредне карактеристике:

- **Date** – Дан за који је вршен прорачун
- **Daily Low** – Најнижа вредност акције у току дана
- **Daily High** – Највећа вредност акције у току дана
- **Closing Price** – Цена акције на крају дана
- **Open** – Цена акције на почетку дана
- **Volume** – Број продатих акција за дати дан – Обим

Циљ овог рада је прецизно предвидети карактеристику **Closing Price**. Ова карактеристика ће играти улогу излаза мреже.
Како би се видело да ли је могуће на основу осталих карактеристика предвидети излаз, потребно је посматрати колико те карактеристике међусобно зависе, као и колико излаз зависи од њих.
Слика 1 приказује корелациону матрицу ових карактеристика.

Слика 1: Корелациона матрица

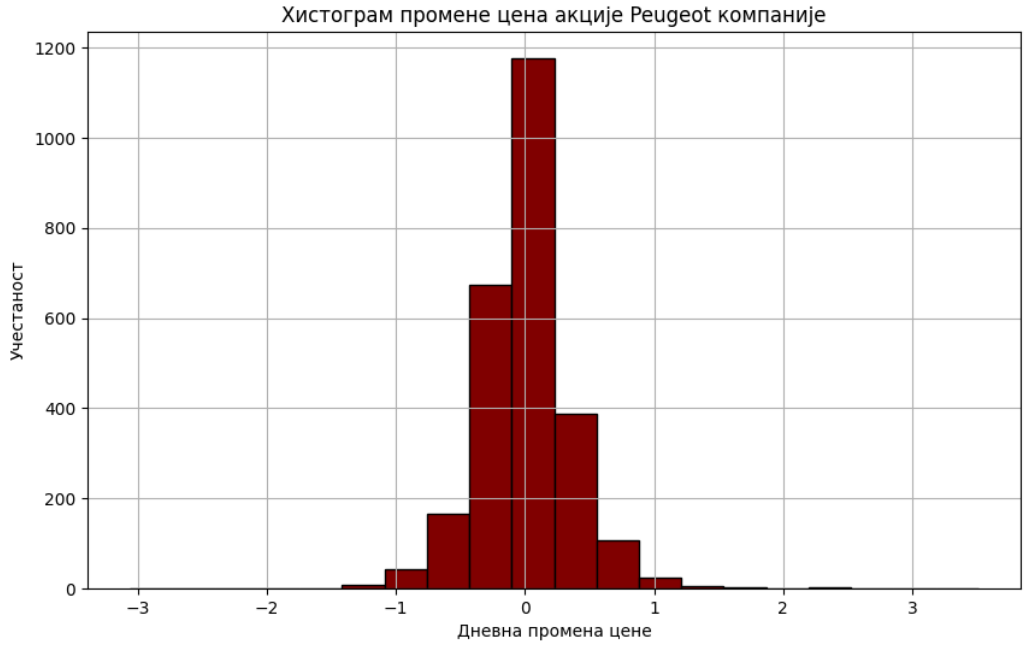
Може се приметити, као што је очекивано, да су готово све карактеристике међусобно потпуно пропорционалне. Када једне расте, и остале расту и обрнуто. Ово их чини непотребним, како не доносе нове информације са собом.
У оквиру рада, изабрана је само карактеристика **Closing Price**, односно вредност те карактеристике у временским тренуцима $t - l_{seq}$ до $t - 1$, где је l_{seq} неки коначан број већи од 1, како би се одредила вредност **Closing Price** у тренутку t.
Како скуп података садржи најуспешнијих 40 француских компанија, потребно је одабрати једну за даљи развој пројекта.
Слика 2 представља број инстанци сваке компаније у оквиру скупа података.



Слика 2: Број инстанци скупа података по компанији

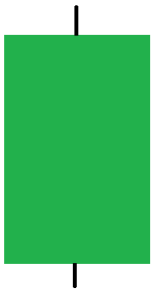
На основу 2 примећује се да све компаније, осим једне, имају идентичан број тачака или инстанци у оквиру овог скупа података, што је јако повољно јер омогућава слободан избор компаније за даљи рад.
Свака инстанца представља вредности карактеристика за један дан једне компаније на тржишту акцијама
За сврху израде овог пројекта одабрана је компанија **Peugeot**, и њена историја на француском тржишту акција приказана у 2601 инстанци (информације дате за 2601 дан).

Како се овај рад бави предвиђањем вредности у датом тренутку на основу вредности из претходних тренутака, потребно је посматрати како се крећу промене вредности **Closing Price** карактеристике из дана у дан за изабрану компанију, са акцентом на разлику вредности карактеристике између две суседне инстанце.
Слика 3 представља визуелни приказ промене вредности **Closing Price** карактеристике међу суседним инстанцама.



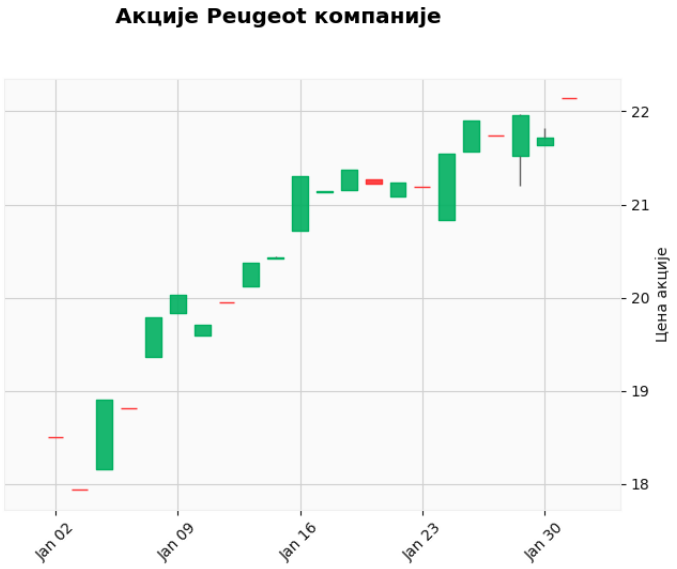
Слика 3: Промена вредности **Closing Price** карактеристике

Са слике 3 може се приметити да су разлике у вредностима **Closing Price** минималне и готово да прате нормалну расподелу, што олакшава моделу дугорочне траткотрајне меморије да ефикасније врши предвиђање.
Како се ради о тржишту акција, поучно би било приказати прикладном визуелизацијом потом сагледати скуп података.
Такав приказ подразумева представљање података помоћу **Candlestick** графика. Представљање података помоћу **Candlestick** графика је метода у финансијама која се користи за визуелизацију цена на тржишту у одређеном временском периоду. Ова врста графикона састоји се из "свећа", где свака свећа представља цену атрибута (у овом случају **Closing Price**) у одређеном временском оквиру (у овом случају дан), док крајеви линија представљају највишу и најмању цену у току дана. Боја свеће указује на то да ли је цена порасла (зелена) или опала (црвена) у односу на вредност из претходног временског тренутка. Сликом 4 дат је пример свеће.

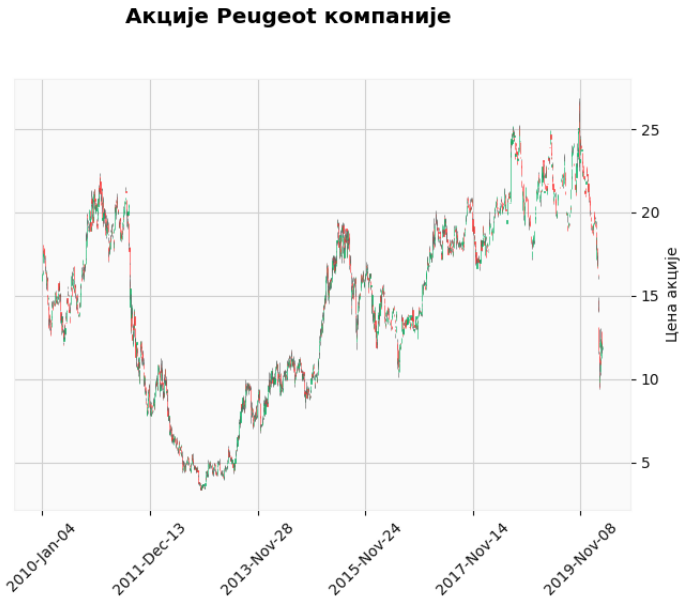


Слика 4: Пример свеће у **Candlestick** графику

Сликама 5 и 6 дат је графички приказ података помоћу **Candlestick** графика.



Слика 5: Вредност акције **Peugeot** компаније у једном месецу

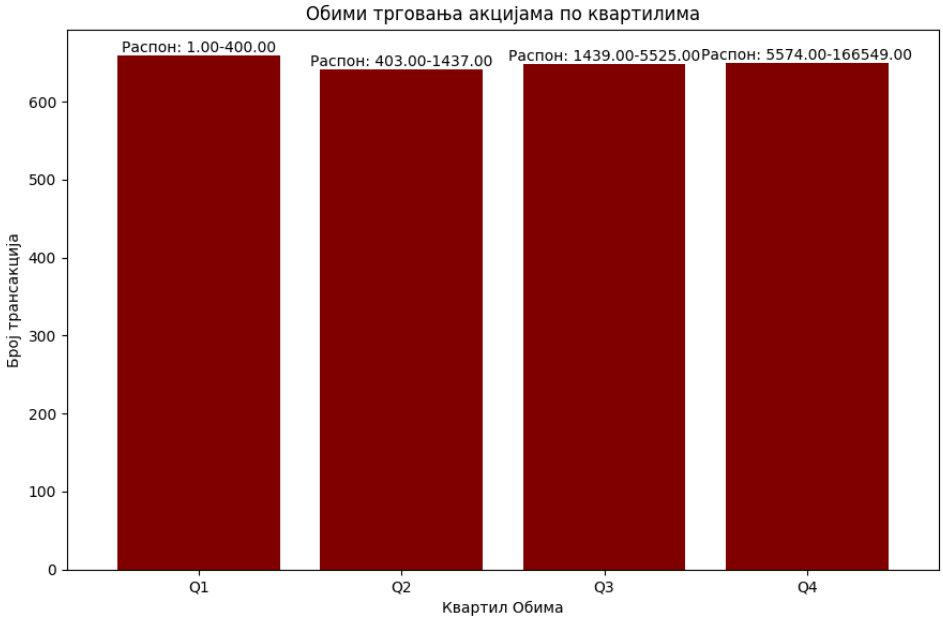


Слика 6: Вредност акције **Peugeot** компаније 2010.–2020.

Са слика 5 и 6 може се приметити да, иако су промене у вредностима између суседних инстанци минималне (слика 3), долази до великих флуктуација у вредности **Closing Price** карактеристике.
Једноставнији приказ вредности **Closing Price** карактеристике у функцији времена дат је сликом 7.

Слика 7: Једноставнији приказ промене вредности акција

Претходно је поменут број продатих акција за дати дан као важан фактор у моделирању временске зависности у овом проблему, и иако је та карактеристика уклоњена, посматрањем њених својства могу се добити важне информације.
Сликом 8 дат је обим трговања акцијом **Peugeot** компаније подељен на четири кватрала идентичне величине.



Слика 8: Једноставнији приказ промене вредности акција

Обим трговања акцијом одабране компаније се може поделити у четири квартила идентичне величине бирањем распона датих на слици 8. На основу резултата те поделе може се приметити да се трговање акцијама одабране компаније обављало у најмањој могућој, као и у значајно великој количини у неким временским тренуцима, што може бити веома важан фактор. Међутим, са корелационе матрице (слика 1) примећује се да је обим заправо обрнуто пропорционалан већини осталих карактеристика, што значи да га је могуће занемарити за сада.

3 Преглед софтверског решења

Преглед софтверског решења подељен је у више секција:

- Визуелиција
- Претпроцесирање
- Креирање, тренирање и тестирање модела мреже

3.1 Визуелизација

Зависности подпрограма за визуелизацију, функција за исцртавање графичког приказа модела, функција за исцртавање графика за приказ броја инстанци по компанији и функција за исцртавање матрице корелације дати су исечком 1, док су исечком 2 дате остале функције за исцртавање.

```
1  from matplotlib import pyplot as plt
2  import pandas as pd
3  import mplfinance as mpf
4  import seaborn as sns
5  from preprocess import load_data
6  import datetime as dt
7  from tensorflow.keras.models import load_model
8  import visualkeras
9  from PIL import ImageFont
10
11 def plot_ltsm_model(path):
12     model = load_model(path)
13     font = ImageFont.truetype("arial.ttf", 12)
14     visualkeras.layered_view(model, legend=True, to_file='gen/model_architecture.png', font=font)
15 def plot_komp_cnt(name_counts):
16     name_counts.plot(kind='bar', figsize=(10, 6), color='maroon')
17     plt.title('Број тачака (дана) у којима је сачувана цена акције')
18     plt.xlabel('Име компаније')
19     plt.ylabel('Број тачака')
20     plt.savefig("gen/brdana_komp.png")
21     plt.close()
22 def plot_corr(df):
23     cm = df.corr()
24     plt.figure(figsize=(10, 8))
25     sns.heatmap(cm, annot=True, cmap='rocket', fmt=".2f", linewidths=0.5)
26     plt.title('Матрица корелације')
27     plt.savefig("gen/cm.png")
28     plt.close()
```

Listing 1: Зависности и функције за исцртавање

```
def plot_closing_price(df):
    plt.figure(figsize=(12, 6))
    plt.plot(df['Date'], df['Closing_Price'], label='Цена затварања акција', color='maroon')
    plt.title('Peugeot - цена затварања акција у функцији времена')
    plt.xlabel('Датум')
    plt.ylabel('Цена затварања акција')
    plt.legend()
    plt.savefig("gen/cena_zatv_akc.png")
def plot_daily_price_changes(df):
    plt.figure(figsize=(10, 6))
    df['Daily_Price_Change'] = df['Closing_Price'].diff()
    df['Daily_Price_Change'].hist(bins=20, edgecolor='black', color='maroon')
    plt.title('Хистограм промене цена акције Peugeot компаније')
    plt.xlabel('Дневна промена цене')
    plt.ylabel('Учестаност')
    plt.savefig("gen/promena_cene.png")
def plot_candlestick_daytoday(df):
    sd, ed = pd.Timestamp(2019,1,1), pd.Timestamp(2019,1,31)
    df_month = df[(df['Date'] >= sd) & (df['Date'] <= ed)]
    ohlc = df_month[['Date', 'Open', 'Daily_High', 'Daily_Low', 'Closing_Price']]
    .set_index('Date')
    ohlc.columns = ['Open', 'High', 'Low', 'Close']
    mpf.plot(ohlc, type='candle', style='yahoo',
    title='Акције Peugeot компаније', ylabel='Цена акције',
    show_nontrading=False,savefig="gen/marketplot_jan.png")
def plot_volume(df):
    df['Volume_Quartiles'] = pd.qcut(df['Volume'], q=[0, 0.25, 0.5, 0.75, 1.0],\
    labels=['Q1', 'Q2', 'Q3', 'Q4'])
    volume_quartile_counts = df['Volume_Quartiles'].value_counts()
    quartile_ranges = df.groupby('Volume_Quartiles')['Volume'].agg(['min', 'max'])
    plt.figure(figsize=(9, 6))
    volume_quartile_counts.sort_index().plot(kind='bar',\
    color='maroon', width=0.8)
    bars = plt.bar(volume_quartile_counts.index, volume_quartile_counts.sort_index(),\
    color='maroon', width=0.8)
    for bar, (q, (min_val, max_val)) in zip(bars, quartile_ranges.iterrows()):
        plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
                f'Распон: {min_val:.2f}-{max_val:.2f}', ha='center', va='bottom')
    plt.title('Обими трговања акцијама по квартилима')
    plt.xlabel('Квартил Обима')
    plt.ylabel('Број трансакција')
    plt.tight_layout()
    plt.savefig("gen/volume_quartiles_plot.png")
```

Listing 2: Остале функције за исцртавање

3.2 Претпроцесирање

Исечком 3 дата је функција за учитавање и "чишћење" скупа података.

```
def load_data(start,end):
    df = pd.read_csv("preprocessed_CAC40.csv")
    df = df.interpolate(method='linear')
    df['Date'] = pd.to_datetime(df['Date'])
    df.drop(df.filter(regex="Unname"),axis=1, inplace=True)
    df = df[df["Name"] == "Peugeot"]
    df = df.loc[(df['Date'] > start) & (df['Date'] < end), :]
    df['Volume'] = pd.to_numeric(df['Volume'].str.replace(',', ''), errors='raise')
    df['Volume'] = df['Volume'].ffill()
    df = df.sort_values('Date')
    return df
```

Listing 3: Функција за учитавање и "чишћење" скупа података

Ова функција отклања проблеме код којих су неке карактеристике инстанци без вредности, на више различитих начина.
Исечком 4 дата је функција за креирање секвенци.

```
def create_sequences(arr, seq_len):
    X_train, y_train = [], []
    for x in range(seq_len, len(arr)):
        X_train.append(arr[x - seq_len : x, -1])
        y_train.append(arr[x, -1])
    X_train, y_train = np.array(X_train), np.array(y_train)
    return X_train, y_train
```

Listing 4: Функција за креирање секвенци

На основу параметра временског корака, генеришу се низови улазних података. За сваку инстанцу везујемо претходних N инстанци како би се моделирала временска зависност.
Поред ових функција, врши се и скалирање. Скалирање се врши у главном програму , како је постојала потреба за тиме због зависности појединих објеката. Исечком 5 дата је функција скалирања (која се налази у главном програму).

```
scaler = MinMaxScaler(feature_range=(0, 1))
def preprocess_data(data, scaler, cols):
    temp = data.copy(deep=True)
    temp[cols] = scaler.fit_transform(temp[cols])
    return temp[cols].values
```

Listing 5: Функција за скалирање

3.3 Креирање, тренирање и тестирање модела мреже

Зависности су дате исечком 6.

```
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout, BatchNormalization
from preprocess import load_data, create_sequences
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau, TensorBoard
from sklearn.preprocessing import MinMaxScaler
import datetime as dt
```

Listing 6: Зависности главног програма

Процес креирања модела мреже обавља функција дата исечком 7.

```
def build_lstm_model(input_shape=(60, 1)): #vremenski korak = 60
    model = Sequential()
    model.add(LSTM(50, return_sequences=True, input_shape=input_shape))
    model.add(BatchNormalization())
    model.add(Dropout(0.2))
    model.add(LSTM(units=50, return_sequences=True))
    model.add(BatchNormalization())
    model.add(Dropout(0.2))
    model.add(LSTM(units=50))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model
```

Listing 7: Функција за креирање мреже

Функције за тренирање мреже и припрему тестног скупа су дата исечком 8.

```
def train_model(model, X_train, y_train, epochs=100, batch_size=32, validation_split=0.15):
    checkpointer = ModelCheckpoint(filepath='best_weights.h5',
        monitor='val_loss',
        verbose=1,
        save_best_only=True,
        mode='min')
    early_stopping = EarlyStopping(monitor='val_loss',
        patience=15,
        restore_best_weights=True)
    reduce_lr = ReduceLROnPlateau(monitor='val_loss',
        factor=0.2,
        patience=3,
        min_lr=1e-5)
    tensorboard = TensorBoard(log_dir='logs',
        histogram_freq=1,
        write_graph=True,
        write_images=True)
    model.fit(X_train,
        y_train,
        epochs=epochs,
        batch_size=batch_size,
        validation_split=validation_split,
        callbacks=[checkpointer, early_stopping, reduce_lr, tensorboard])

def prepare_test_data(train_data, test_data, scaler, seq_length=60):
    actual_prices = test_data['Closing_Price'].values
    total_dataset = pd.concat((train_data['Closing_Price'], test_data['Closing_Price']), axis=0)
    model_inputs = total_dataset[len(total_dataset) - len(test_data) - seq_length:].values
    model_inputs = model_inputs.reshape(-1, 1)
    model_inputs = scaler.transform(model_inputs)
    X_test, _ = create_sequences(model_inputs, seq_length)
    X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
    return actual_prices, X_test
```

Listing 8: Функције за тренинг и припрему тестног скупа података

Функција за приказ резултата мреже и главни програм приказани су исечком 9

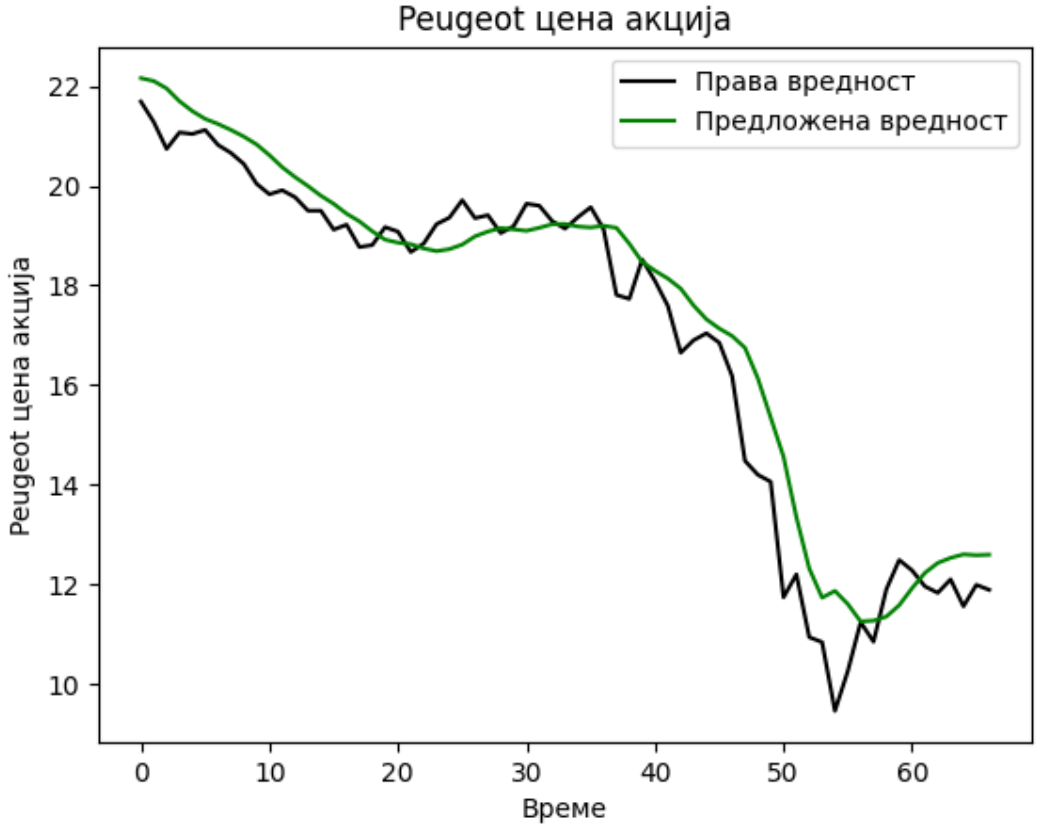
```
def plot_results(actual_prices, predicted_prices, title):
    print(np.shape(actual_prices))
    plt.plot(actual_prices, color='black', label="Права вредност")
    plt.plot(predicted_prices, color='green', label="Предложена вредност")
    plt.title(title)
    plt.xlabel("Време")
    plt.ylabel("Peugeot цена акција")
    plt.legend()
    plt.savefig("gen/rezultat.png")
    plt.close()

START_DATE = dt.datetime(2015, 1, 1)
END_DATE = dt.datetime(2019, 12, 30)
START_DATE_TEST = END_DATE

scaler = MinMaxScaler(feature_range=(0, 1))
cols = ['Closing_Price']

train_data = load_data(START_DATE, END_DATE)
temp_train = preprocess_data(train_data, scaler, cols)
X_train, y_train = create_sequences(temp_train, 60)

model = build_lstm_model(input_shape=(60, 1))
train_model(model, X_train, y_train)

test_data = load_data(START_DATE_TEST, dt.datetime(2020, 12, 30))
actual_prices, X_test = prepare_test_data(train_data, test_data, scaler)

predicted_prices = model.predict(X_test)
predicted_prices = scaler.inverse_transform(predicted_prices)

plot_results(actual_prices, predicted_prices, "Peugeot цена акција")
```

Listing 9: Функција за приказ резултата мреже и главни програм

Функција за приказ резултата мреже генерише график који приказује предвиђања мреже за 67 дана. Тренирање мреже вршило се над подацима од 1.1.2015. до 30.12.2019. (31.12.2019. не садржи икакве податке), док су за тестни скуп узети подаци од 1.1.2020. до 4.3.2020. На графику се исцртавају предвиђања као и тачне вредности, ради лакшег поређења.

4 Резултати

Улаз мреже је димензија (60,1), како је претходно постављен временски корак од вредности 60, што значи да ће мрежа за цену акције C_t у обзир узети цене акција $C_{t-1}...C_{t-60}$.
Тренирање мреже вршено је над 100 епоха са димензијом подскупа 32 (**batch size**), где је постављено пар правила које условљавају мрежу да прекида са обучавањем уколико не дође до побољшања, као и динамичка промена стопе учења и чување најефикаснијих тежина модела мреже.
Модел над тренинг скупом након 100 епоха остварује грешку (**MSE**) у вредности од 0.0018, док над валидационим скупом (узето 15% од оригиналног тренинг скупа) остварује грешку од 0.00199.
Како је речено у претходној секцији, мрежа се тестира над скупом података цена акција од 1.1.2020. до 4.3.2020.
Резултати које је модел остварио дати су сликом 9.



Слика 9: Резултат – поређење предвиђања мреже и правих вредности

Са слике 9 може се приметити да је мрежа остварила добар резултат, и што се тиче појединих цена акција и праћења линије тренда цене.