Универзитет „Св. Кирил и Методиј“ Скопје
Факултет за информатички науки и компјутерско инженерство

Изградба на модел за предвидување на цената на Bitcoin во 15 минутен временски интервал

Ментор:
проф. д-р. Соња Гиевска

Изработил:
Борјан Ѓорѓиевски – 231175
Андреј Шумановски - 233062

2025

Содржина

1 Вовед

Bitcoin претставува најпознатата и најтргуваната криптовалута, со значајна улога во современите финансиски пазари. Сепак, неговата висока волатилност претставува сериозен предизвик за инвеститорите и аналитичарите кои се стремат да ги предвидат краткорочните ценовни движења. Поради тоа, се зголемува интересот за примена на напредни методи од областа на машинското учење и вештачката интелигенција за анализа и предвидување на пазарните трендови.

Во рамки на овој труд се развива модел за предвидување на цената на Bitcoin на временски интервал од 15 минути, користејќи комбинација од повеќе алгоритми: **Random Forest**, **XGBoost** и **невронска мрежа**. Преку споредба на нивните перформанси се анализира ефикасноста на различни пристапи во идентификување на шеми и трендови во податоците.

Целта на трудот е да се испита можноста за зголемување на точноста на краткорочните предвидувања преку комбинирање на класични и длабоки модели на машинско учење, со што се создава основа за понатамошен развој на интелигентни системи за финансиска анализа и автоматизирано донесување одлуки.

2 Избор на модели за предвидување

Во рамки на овој труд се избрани три различни модели за предвидување на цената на Bitcoin: Random Forest, XGBoost и невронска мрежа изградена со Keras. Изборот на овие модели е направен врз основа на нивната докажана ефикасност во обработка на временски серии и способност за моделирање на нелинеарни односи. Random Forest и XGBoost претставуваат моќни ансамбл техники кои комбинираат повеќе дрва на одлука за зголемена стабилност и точност, додека невронската мрежа овозможува учење на сложени обрасци и трендови во податоците. Овој комплементарен пристап овозможува споредба на традиционални и длабоки модели и нивна потенцијална комбинација во хибриден систем за зголемена прецизност.

3 Податочно множество и обработка на податоци

3.1 Прибирање и предобработка на податоците

Во оваа фаза се изврши прибирање, интеграција и основна обработка на податоците потребни за предвидување на движењето на цената на Bitcoin. Податоците за Bitcoin беа прибавени преку **Binance API**, користејќи ја библиотеката **ccxt**, со временска резолуција од **15 минути**, со што се овозможи висока прецизност и деталност на пазарните флуктуации.

Податоците беа подложени на повеќестепена предобработка: отстранување на недостасувачки вредности, синхронизација на временските ознаки и нормализација на податоците за да се постигне конзистентност. Понатаму, беа дефинирани дополнителни метрики како процентуални промени во цената и обемот, кои послужија како база за понатамошно извлекување на карактеристики и етикетирање на активноста („buy“, „sell“, „hold“).

```
binance = ccxt.binance()

symbol = 'BTC/USDT'
timeframe = '15m'
limit = 1000

since_dt = datetime.now(timezone.utc) - timedelta(days=7*4 + 3) # 4 weeks (+3 to avoid weekends if i fetch index, like the d
 they dont work on weekends, and i just remove where dxy is Nan)
since = int(since_dt.timestamp() * 1000) # second in milliseconds

all_candles = []

while True:
    candles = binance.fetch_ohlcv(symbol, timeframe=timeframe, since=since, limit=limit) # binance api from ccxt
    if not candles:
        break
    all_candles += candles

    since = candles[-1][0] + 1 # update since to last candle timestamp + 1 ms
    time.sleep(0.5)  # avoid hitting rate limits with api

# Convert to DataFrame (table)
df = pd.DataFrame(all_candles, columns=['timestamp', 'open', 'high', 'low', 'close', 'volume'])
df['timestamp'] = pd.to_datetime(df['timestamp'], unit='ms')
```

3.2 Извлекување на технички индикатори

За збогатување на податочниот сет и подобрување на можноста за учење на моделите, беа пресметани повеќе технички индикатори, кои се користат во финансиската анализа за идентификација на трендови и сигнали за тргување. Во трудот беа користени следните индикатори:

RSI (Relative Strength Index) – за мерење на интензитетот и динамиката на пазарните движења во краток (6 периоди) и среден (12 периоди) рок.

MACD (Moving Average Convergence Divergence) – за идентификација на промени во насоката на трендот преку пресекот на експоненцијални проседи.

EMA (Exponential Moving Average) и **SMA** (Simple Moving Average) – за следење на краткорочни и долгорочни трендови, како и нивната интеракција преку "crossover" сигнал.

Bollinger Bands – за мерење на волатилноста и можните екстремни вредности на цената.

ATR (Average True Range) – за проценка на нестабилноста на пазарот.

Преку овие индикатори се овозможи создавање на сета од карактеристики што ги опфаќаат моменталните пазарни услови, динамиката на трендовите и потенцијалните сигнали за промена, со што се создаде солидна основа за машинското учење во следната фаза на проектот.

```python
import ta
from ta.momentum import RSIIndicator
from ta.trend import MACD

# Short-term (~30min)
df['rsi_6'] = RSIIndicator(df['close'], window=6).rsi()

# Classic (~1hour)
df['rsi_12'] = RSIIndicator(df['close'], window=12).rsi()

macd_indicator = MACD(close=df['close'], window_slow=26, window_fast=12, window_sign=9)

df['macd'] = macd_indicator.macd()                  # MACD line
df['macd_signal'] = macd_indicator.macd_signal()    # Signal line
df['macd_hist'] = macd_indicator.macd_diff()        # Histogram
```

```python
# df['ema_9'] = df['close'].ewm(span=9, adjust=False).mean()

df['ema_21'] = df['close'].ewm(span=21, adjust=False).mean()

df['sma_50'] = df['close'].rolling(window=50).mean()

df.dropna(subset=['ema_21', 'sma_50'], inplace=True)

df['ema/sma crossover'] = df['ema_21'] - df['sma_50']

df['trends with sma'] = df['close'] / df['sma_50'] - 1
```

```python
from utils import buy_col, sell_col, define_activity

df['next_high'] = df['high'].shift(-1)
df['next_low'] = df['low'].shift(-1)

df['price change'] = np.log(df['close'] / df['close'].shift(1))
df.dropna(subset=['price change'], inplace=True)

df['buy_index'] = df.apply(buy_col, axis=1)
df['sell_index'] = df.apply(sell_col, axis=1)

df['signal'] = df['buy_index'] - df['sell_index']
scale = df['signal'].abs().median()
df['signal_scaled'] = 0.5 * (np.tanh( (df['buy_index'] - df['sell_index']) / scale ) + 1)

df['activity'] = df.apply(define_activity, axis=1)
# df['activity'] = np.where(df['close'].shift(-5) > df['close'], 'Buy', 'Sell')

df.dropna(subset=['buy_index','sell_index'], inplace=True)
```

```python
from ta.volatility import BollingerBands

bollinger = BollingerBands(close=df['close'], window=20, window_dev=2)
df['bollinger_hband'] = bollinger.bollinger_hband()
df['bollinger_lband'] = bollinger.bollinger_lband()
df['bollinger_mavg'] = bollinger.bollinger_mavg()
df['bollinger_bandwidth'] = bollinger.bollinger_wband()

from ta.volatility import AverageTrueRange

atr = AverageTrueRange(high=df['high'], low=df['low'], close=df['close'], window=14)
df['atr'] = atr.average_true_range()

df['volume_change'] = df['volume'].pct_change()
df['volume_sma_10'] = df['volume'].rolling(window=10).mean()

# Drop rows with NaN values introduced by rolling calculations
df.dropna(inplace=True)

print(df.head())
```

```
print(df)
df.set_index('timestamp', inplace=True)

# export_df = df[['dxy', 'rsi_6', 'rsi_12', 'ema_21', 'sma_50', 'activity']]
export_df = df.drop(columns=['date', 'next_high', 'next_low', 'buy_index', 'sell_index', 'signal', 'signal_scaled'])
print(export_df)

%store export_df
```

4 Методологија на тренирање

Во оваа фаза се спроведе процес на тренирање и евалуација на три различни модели од областа на машинското учење, со цел предвидување на активноста на пазарот („Buy“, „Sell“) врз основа на историските податоци и техничките индикатори. За споредбена анализа беа избрани три модели со различна архитектура и пристап: **Random Forest**, **XGBoost** и **невронска мрежа изградена со Keras Sequential API**.

За подготовка на податоците се користеше **LabelEncoder** за етикетирање на целната класа, а **StandardScaler** за нормализација на карактеристиките. Податоците беа поделени во сооднос од **90% за тренирање и 10% за тестирање**, при што не беше извршено мешање на редоследот на записите (shuffle=False) со цел да се зачува временската зависност на финансиските податоци.

На крај, сите модели беа евалуирани според **точност (accuracy)** и **classification report** метрики, а дополнително беше применет **voting систем** за комбинирање на предвидувањата од сите модели. Овој систем овозможи идентификација на случаи каде што постои заеднички консензус, со што се доби посигурна финална предикција.

4.1 Random Forest

Random Forest е ансамбл модел заснован на повеќе одлукови дрва, каде што секое дрво учи на случаен подсклоп од податоците и карактеристиките. Со агрегирање на резултатите од сите дрва (преку гласање), моделот обезбедува стабилна и робусна предикција.

Во имплементацијата беше користен **500 дрва (n_estimators=500)** и фиксирано семе за репродуктивност (random_state=42). Овој модел е особено погоден за откривање нелинеарни врски и за работа со податоци кои содржат повеќе меѓузависни индикатори.

```
forest_model = RandomForestClassifier(
    n_estimators=500,
    random_state=42,
    class_weight='balanced'  # helps with imbalanced classes
)
forest_model.fit(x_train_scaled, y_train_encoded)
```

4.2 XGBoost

XGBoost (Extreme Gradient Boosting) претставува оптимизиран модел базиран на градиентно засилување на одлукови дрва. За разлика од Random Forest, кој ги тренира дрвата паралелно, XGBoost ги тренира секвенцијално, при што секое ново дрво го коригира остатокот (грешката) од претходните.

Во имплементацијата се користени следниве параметри:

- n_estimators=500
- max_depth=8
- learning_rate=0.15
- gamma=1, reg_alpha=0.1, reg_lambda=1

Овие вредности овозможуваат рамнотежа помеѓу брзина и прецизност, при што моделот се прилагодува на комплексните пазарни шеми без преголема пренатренирност. Овој модел е познат по висока ефикасност и стабилност при анализа на временски и финансиски податоци.

```
xgb_model = xgb.XGBClassifier(
    objective="multi:softprob",
    num_class=len(label_encoder.classes_),
    n_estimators=500,
    max_depth=8,
    learning_rate=0.15,
    gamma=1,
    reg_alpha=0.1,
    reg_lambda=1,
    subsample=0.8,
    colsample_bytree=0.8,
    eval_metric='mlogloss',
    use_label_encoder=False,
    random_state=42
)
xgb_model.fit(
    x_train_scaled,
    y_train_encoded,
    eval_set=[(x_test_scaled, y_test_encoded)],
    verbose=True
)
```

4.3 Neural Network (Keras Sequential Model)

За дополнителна анализа беше изграден длабок невронски модел користејќи го Keras Sequential API. Мрежата се состои од повеќеслојна архитектура со комбинација на Dense и Dropout слоеви, наменета за намалување на пренатренирност и подобрување на генерализацијата.

Архитектурата вклучува:

- Влезен слој со големина еднаква на бројот на карактеристики,
- Скриени слоеви со 128, 64 и 32 неврони со ReLU активација,
- Dropout од 0.3 и 0.2 за редукција на пренатренирност,
- Излезен слој со Softmax активација за мултикласна класификација.

Мрежата беше тренирана со Adam оптимизатор, Sparse Categorical Crossentropy како функција на загуба и 50 епохи со големина на пакет од 32 примероци. Овој пристап овозможи моделирање на комплексни нелинеарни релации и динамички шеми кои традиционалните методи потешко ги фаќаат.

```
neural_model = Sequential([
    Input(shape=(x_long_train_scaled.shape[1],)),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(32, activation='relu'),
    Dense(len(label_encoder.classes_), activation='softmax')
])

neural_model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

neural_model.fit(
    x_long_train_scaled,
    y_long_train_encoded,
    epochs=50,
    batch_size=32,
    validation_split=0.1,
    callbacks=[early_stop],
    verbose=1
)
```

5 Резултати и евалуација

```
RandomForest Accuracy: 0.6800
XGBoost Accuracy:      0.7000
NeuralNet Accuracy:    0.6800
Voting Accuracy:       0.6900
Models fully agree on 73.00% of cases
```

5.1 Random Forrest

Random Forest моделот постигна стабилни резултати при предвидување на краткорочните ценовни промени на Bitcoin. Точноста на тестирачкото множество изнесуваше **~65%**, при што моделот покажа добра способност за препознавање на активности „Buy“ и „Sell“, со релативно ниска стапка на False Positives.

Силните страни на моделот вклучуваат неговата робусност кон шум во податоците и способноста за идентификување на нелинеарни зависимости помеѓу индикаторите. Сепак, ограничувањата се гледаат во послаба адаптација на многу динамични шеми кои се присутни во финансискиот пазар на краток рок.

5.2 XGBoost

XGBoost моделот покажува слични, но во некои случаи подобри резултати од Random Forest, со точност од **~65%** на тестирачкото множество. Благодарение на градиентното засилување, моделот е поосетлив на мали трендови и постепени промени, што ја зголемува неговата предиктивна моќ.

Моделот сепак бара внимателна оптимизација на хиперпараметрите (learning rate, max_depth, subsample и colsample_bytree) за да се избегне пренатренирност и да се постигне добра генерализација на нови податоци.

5.3 Neural Network (Keras)

Невронската мрежа покажа способност за моделирање на комплексни и нелинеарни релации, со точност од ~**70%** на тестирачкото множество. Мрежата често беше помалку „сигурна“ во одредени ситуации, што резултираше со поголема стапка на False Positives во споредба со класичните ансембл модели.

Сепак, предноста на овој модел е неговата флексибилност и можност за натамошно усовршување преку зголемување на бројот на слоеви, неврони или инкорпорирање на дополнителни технички индикатори.

5.4 Voting System

За зголемување на точноста и намалување на несигурноста, беше применет **majority voting** систем, каде предвидувањата од сите три модели се комбинираат за финална класификација.

Комбинираниот модел постигна највисока точност од **~69 %**, со што се докажува дека синтеза на различни пристапи може да резултира со посигурни предвидувања. Дополнително, анализата покажа дека моделите во **~73% од случаите** целосно се согласуваат, што ја потврдува ефективноста на гласањето како метод за подобрување на стабилноста на предикцијата.

6 Комбинирање на модели и финален предиктор

За да се подобри точноста на краткорочните предвидувања, беше применет ensembling пристап со комбинирање на Random Forest, XGBoost и Невронската мрежа преку majority voting. Овој пристап ја зголемува сигурноста на предвидувањата, намалувајќи ја несигурноста на поединечните модели. Финалниот предиктор ја класифицира активноста на пазарот како „Buy“, „Sell“ или „Hold“, врз основа на согласноста помеѓу трите модели. Комбинираниот пристап покажа најдобри резултати, со највисока точност од тестирачкото ножество и стабилна евалуација на нови податоци.

7 Инференца и практична примена

Финалниот модел е спремен за оперативна употреба во реално време. По вчитување на моделот, нови податоци за цената на Bitcoin и техничките индикатори се обработуваат, и се генерира предикција за следната активност. Овој процес може да се користи како алатка за **поддршка при донесување на финансиски одлуки**, автоматизирано тргување или анализа на трендови. Практичната примена покажува дека моделот е способен за навремено предвидување на краткорочните движења, обезбедувајќи информации со висока додадена вредност за инвеститорите.

```
PREDICTING for CANDLE: 15m / BTC/USDT
  Current UTC Time:  2025-10-13 12:27:33
  Candle Start Time: 12:15:00 (Forming)
  Current Price:   114262.47 USDT
--------------------------------------------------

Majority Voted Signal: Sell
RandomForest Prediction: Sell
XGBoost Prediction:      Sell
NeuralNet Prediction:    Sell

- Individual Model Probabilities -
  Buy: RF=0.3160 | XGB=0.1648 | NN=0.3678
  Sell: RF=0.6840 | XGB=0.8352 | NN=0.6322

- Ensemble Average Probabilities -
  Sell: 0.7172
  Buy: 0.2828

==================================================
Confidence Margin:     0.4343 (Required: 0.2000)
Decision Reason:       Ensemble confidence margin is sufficient.
🚨 FINAL TRADING SIGNAL: Sell 🚨
```

8 Заклучок

Во рамките на овој труд, беше развиен и евалуиран модел за предвидување на цената на Bitcoin на 15-минутен интервал. Примената на Random Forest, XGBoost и Невронска мрежа, како и нивното комбинирање преку majority voting, покажа дека синтеза на различни пристапи ја подобрува точноста и сигурноста на предвидувањата. Финалниот модел овозможува оперативна инференца и практична примена во финансиската анализа и автоматизираното тргување. Понатамошното подобрување може да се постигне со инкорпорирање на дополнителни индикатори, проширување на историските податоци и оптимизација на хиперпараметрите.

9 Референци

1. CCXT: https://github.com/ccxt/ccxt
2. TA-Lib и Python Technical Analysis: https://technical-analysis-library-in-python.readthedocs.io/en/latest/
3. Pandas Documentation: https://pandas.pydata.org/docs/
4. XGBoost Documentation: https://xgboost.readthedocs.io/en/stable/
5. Keras Sequential API: https://keras.io/guides/sequential_model/
6. Binance API: https://binance-docs.github.io/apidocs/spot/en/
7. Инспирациско видео: https://www.youtube.com/watch?v=1O_BenficgE
8. ChatGPT: https://chatgpt.com/