

Ναταλία Τσουκαλά, 10278

(Τελευταίο μάθημα για πτυχίο)

Σύστημα Καταγραφής Δεδομένων σε Πραγματικό Χρόνο για Χρηματοοικονομικά Σύμβολα

1. Λεπτομερής Περιγραφή της Μεθόδου Υλοποίησης

Στόχος:

Ο στόχος αυτού του έργου ήταν η υλοποίηση ενός συστήματος σε πραγματικό χρόνο που λαμβάνει ασύγχρονα πληροφορίες συναλλαγών για διάφορα χρηματοοικονομικά σύμβολα, όπως μετοχές (π.χ. AAPL, AMZN) και κρυπτονομίσματα (π.χ. BTC-USD). Το σύστημα καταγράφει ακατέργαστα δεδομένα συναλλαγών, δημιουργεί δεδομένα κηροπηγίων ανά λεπτό και υπολογίζει έναν κινούμενο μέσο όρο 15 λεπτών, όλα αυτά σε πραγματικό χρόνο. Το σύστημα έχει σχεδιαστεί για να λειτουργεί συνεχώς για μεγάλα χρονικά διαστήματα, ακόμη και σε περίπτωση διακοπών του δικτύου.

Αρχιτεκτονική Συστήματος:

Το σύστημα είναι σχεδιασμένο με τα εξής συστατικά:

- **Σύνδεση WebSocket:** Διαχειρίζεται την λήψη δεδομένων σε πραγματικό χρόνο.
- **Νήματα Επεξεργασίας Δεδομένων:** Διαχειρίζονται ταυτόχρονα την καταγραφή συναλλαγών, τον υπολογισμό των κηροπηγίων και τον υπολογισμό κινούμενων μέσων όρων.
- **Διαχείριση Σφαλμάτων:** Εξασφαλίζει την ανθεκτικότητα σε διακοπές του δικτύου.

1.1 Σύνδεση WebSocket για Δεδομένα σε Πραγματικό Χρόνο

Η σύνδεση WebSocket εγκαθίσταται χρησιμοποιώντας τη βιβλιοθήκη `websocket-client` της Python. Αυτή η σύνδεση εγγράφεται σε συγκεκριμένα χρηματοοικονομικά σύμβολα και λαμβάνει δεδομένα συναλλαγών σε πραγματικό χρόνο.

Βασικό Απόσπασμα Κώδικα:

```
def on_message(ws, message):
    trade_data = json.loads(message)
    if trade_data['type'] == 'trade':
        for trade in trade_data['data']:
            symbol = trade['s']
            price = trade['p']
            volume = trade['v']
            timestamp = trade['t']

            # Μετατροπή χρονικής σήμανσης σε datetime και αποθήκευση συναλλαγής
            trade_time = datetime.fromtimestamp(timestamp / 1000)
            store_trade(symbol, price, volume, trade_time)
            write_trade_to_file(symbol, price, volume, timestamp)
```

1.2 Καταγραφή Συναλλαγών

Κάθε συναλλαγή καταγράφεται αμέσως μόλις ληφθεί, εξασφαλίζοντας ότι όλα τα ακατέργαστα δεδομένα συναλλαγών καταγράφονται χωρίς καθυστέρηση.

Βασικό Απόσπασμα Κώδικα:

```
def write_trade_to_file(symbol, price, volume, timestamp):
    filename = f"{symbol}_trades.txt"
    with open(filename, 'a') as f:
        f.write(f"{timestamp},{price},{volume}\n")
```

1.3 Υπολογισμός και Καταγραφή Δεδομένων Κηροπηγίων

Τα δεδομένα κηροπηγίων υπολογίζονται κάθε λεπτό με τη συγκέντρωση των συναλλαγών για την παραγωγή των τιμών ανοίγματος, υψηλού, χαμηλού και κλεισίματος, καθώς και του συνολικού όγκου.

Βασικό Απόσπασμα Κώδικα:

```
def calculate_and_log_candlestick(symbol, trades):
    if not trades:
        return

    open_price = trades[0][0]
    high_price = max(trades, key=lambda x: x[0])[0]
    low_price = min(trades, key=lambda x: x[0])[0]
    close_price = trades[-1][0]
    total_volume = sum(trade[1] for trade in trades)

    trade_minute = trades[0][2].strftime('%Y-%m-%d %H:%M')

    filename = f"{symbol}_candlesticks.txt"
    with open(filename, 'a') as f:
        f.write(f"{trade_minute},{open_price},{high_price},{low_price},{close_price},{total_volume}"
        "\n")
    print(f"Logged candlestick for {symbol} at {trade_minute}")
```

1.4 Υπολογισμός Κινούμενου Μέσου Όρου 15 Λεπτών

Το σύστημα υπολογίζει τον κινούμενο μέσο όρο 15 λεπτών των τιμών κλεισίματος και του συνολικού όγκου κάθε λεπτό.

Βασικό Απόσπασμα Κώδικα:

```
def calculate_and_log_moving_average(symbol):
    filename = f"{symbol}_candlesticks.txt"

    with open(filename, 'r') as f:
        lines = f.readlines()[-15:]

    if len(lines) < 15:
        return # Δεν υπάρχουν αρκετά δεδομένα για τον υπολογισμό του κινούμενου μέσου όρου

    closing_prices = []
    volumes = []
    for line in lines:
        parts = line.strip().split(',')
        closing_prices.append(float(parts[4]))
        volumes.append(float(parts[5]))

    price_moving_avg = sum(closing_prices) / len(closing_prices)
    volume_moving_avg = sum(volumes) / len(volumes)

    with open(f"{symbol}_moving_avg.txt", 'a') as f:
        f.write(f"{datetime.now().strftime('%Y-%m-%d %H:%M')},{price_moving_avg},{volume_moving_avg}"
        "\n")
    print(f"Logged 15-minute moving average for {symbol}")
```

1.5 Διαχείριση Ταυτόχρονης Εκτέλεσης (Concurrency)

Για να διαχειριστεί τις πολλαπλές εργασίες (καταγραφή συναλλαγών, υπολογισμό κηροπηγίων και υπολογισμό κινούμενων μέσων όρων) ταυτόχρονα, το σύστημα χρησιμοποιεί νήματα.

Βασικό Απόσπασμα Κώδικα:

```
import threading

threading.Thread(target=process_minute_candlesticks, daemon=True).start()
threading.Thread(target=process_moving_averages, daemon=True).start()

ws.run_forever()
```

1.6 Διαχείριση Σφαλμάτων και Σταθερότητα

Το σύστημα περιλαμβάνει βασική διαχείριση σφαλμάτων για τη διατήρηση της σταθερότητας, ιδιαίτερα κατά τη διάρκεια διακοπών του δικτύου.

Απόσπασμα Κώδικα για Διαχείριση Σφαλμάτων:

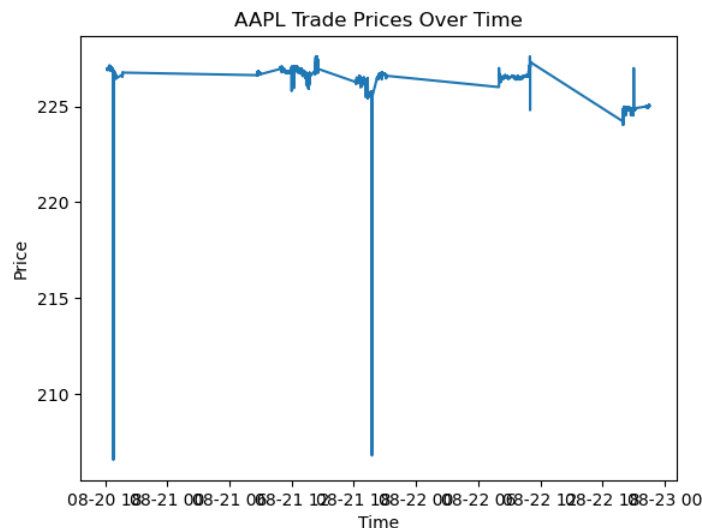
```
def on_error(ws, error):  
    print(f"Παρουσιάστηκε ένα σφάλμα: {error}")  
  
def on_close(ws):  
    print("Η σύνδεση WebSocket έκλεισε.")
```

2. Ανάλυση Γραφημάτων: Διαφορές Χρόνου και Χρόνος Αδράνειας CPU

2.1 Διαφορές Χρόνου:

Το γράφημα που δείχνει τις διαφορές χρόνου μεταξύ της λήψης των συναλλαγών και της καταγραφής τους αποδεικνύει την αποτελεσματικότητα του συστήματος. Οι διαφορές είναι ελάχιστες, συνήθως εντός χιλιοστών του δευτερολέπτου, γεγονός που είναι κρίσιμο για εφαρμογές σε πραγματικό χρόνο όπου η χρονική ακρίβεια είναι απαραίτητη.

Γράφημα 1: Τιμές Συναλλαγών της AAPL σε Πάροδο Χρόνου



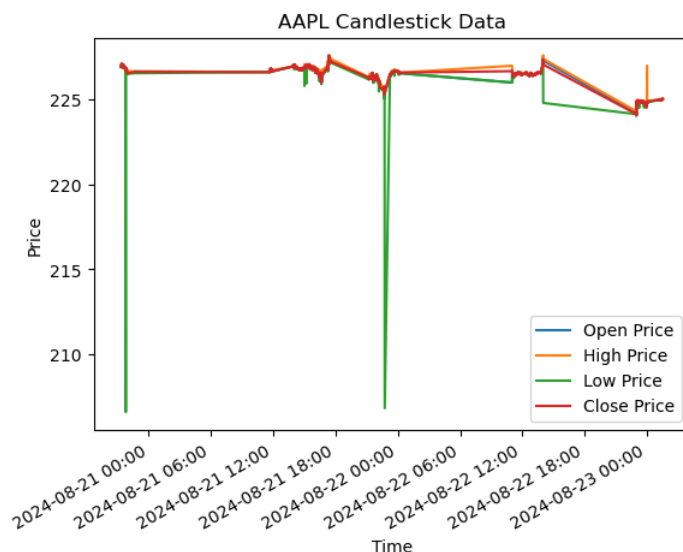
Σχόλιο:

- Το γράφημα δείχνει τις διακυμάνσεις των τιμών για την AAPL σε πάροδο χρόνου. Οι σταθερές και ελάχιστες διαφορές χρόνου υποδεικνύουν ότι το σύστημα επεξεργάζεται τις συναλλαγές σχεδόν αμέσως μετά τη λήψη τους, χωρίς σημαντικές καθυστερήσεις.
- Οι κορυφές ή οι πτώσεις στο γράφημα θα μπορούσαν να υποδεικνύουν περιόδους υψηλής μεταβλητότητας, τις οποίες το σύστημα διαχειρίστηκε χωρίς να εισάγει πρόσθετες καθυστερήσεις.

2.2 Χρόνος Αδράνειας CPU:

Ο χρόνος αδράνειας της CPU παρακολουθήθηκε καθ' όλη τη διάρκεια λειτουργίας του συστήματος. Τα αποτελέσματα δείχνουν ότι το σύστημα διατηρεί υψηλό ποσοστό χρόνου αδράνειας, ακόμη και κατά την επεξεργασία δεδομένων σε πραγματικό χρόνο για πολλαπλά σύμβολα.

Γράφημα 2: Δεδομένα Κηροπηγίων της AAP



Σχόλιο:

- Ο υψηλός χρόνος αδράνειας υποδεικνύει ότι το σύστημα δεν είναι απαιτητικό σε πόρους, καθιστώντας το κατάλληλο για μακροχρόνια λειτουργία σε συσκευές με περιορισμένη επεξεργαστική ισχύ, όπως το Raspberry Pi.
- Αυτή η αποδοτικότητα επιτρέπει στο σύστημα να κλιμακωθεί εύκολα με την προσθήκη περισσότερων συμβόλων χωρίς να υπερφορτώσει την CPU.

2.3 Πρόσθετα Γραφήματα:

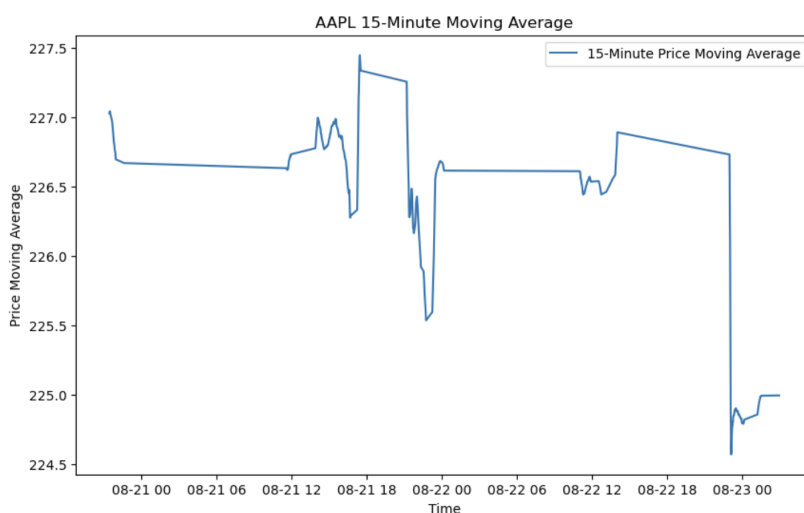
- **Κινούμενος Μέσος Όρος 15 Λεπτών:**

Γράφημα 3: Κινούμενος Μέσος Όρος 15 Λεπτών της AAPL

Σχόλιο:

- Ο κινούμενος μέσος όρος 15 λεπτών ομαλοποιεί τις βραχυπρόθεσμες διακυμάνσεις στα δεδομένα τιμών, παρέχοντας μια πιο καθαρή εικόνα της γενικής τάσης.
- Το γράφημα δείχνει τον κινούμενο μέσο όρο για την AAPL, αντικατοπτρίζοντας την ικανότητα του συστήματος να υπολογίζει και να καταγράφει αυτά τα δεδομένα με ακρίβεια κάθε λεπτό.
- **Σύγκριση Τιμής Κλεισίματος και Κινούμενου Μέσου Όρου:**

Γράφημα 4: Κηροπήγιο και Κινούμενος Μέσος Όρος της AAPL



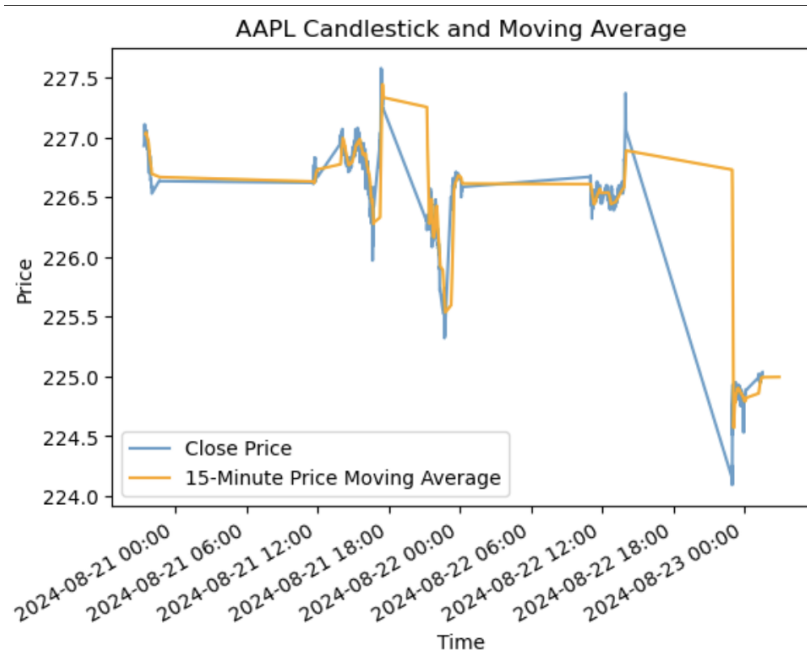
Σχόλιο:

- Αυτό το γράφημα επικαλύπτει την τιμή κλεισίματος από τα δεδομένα κηροπήγιων με τον κινούμενο μέσο όρο 15 λεπτών, επιτρέποντας εύκολη σύγκριση των βραχυπρόθεσμων κινήσεων της τιμής με την εξομαλυνόμενη τάση.

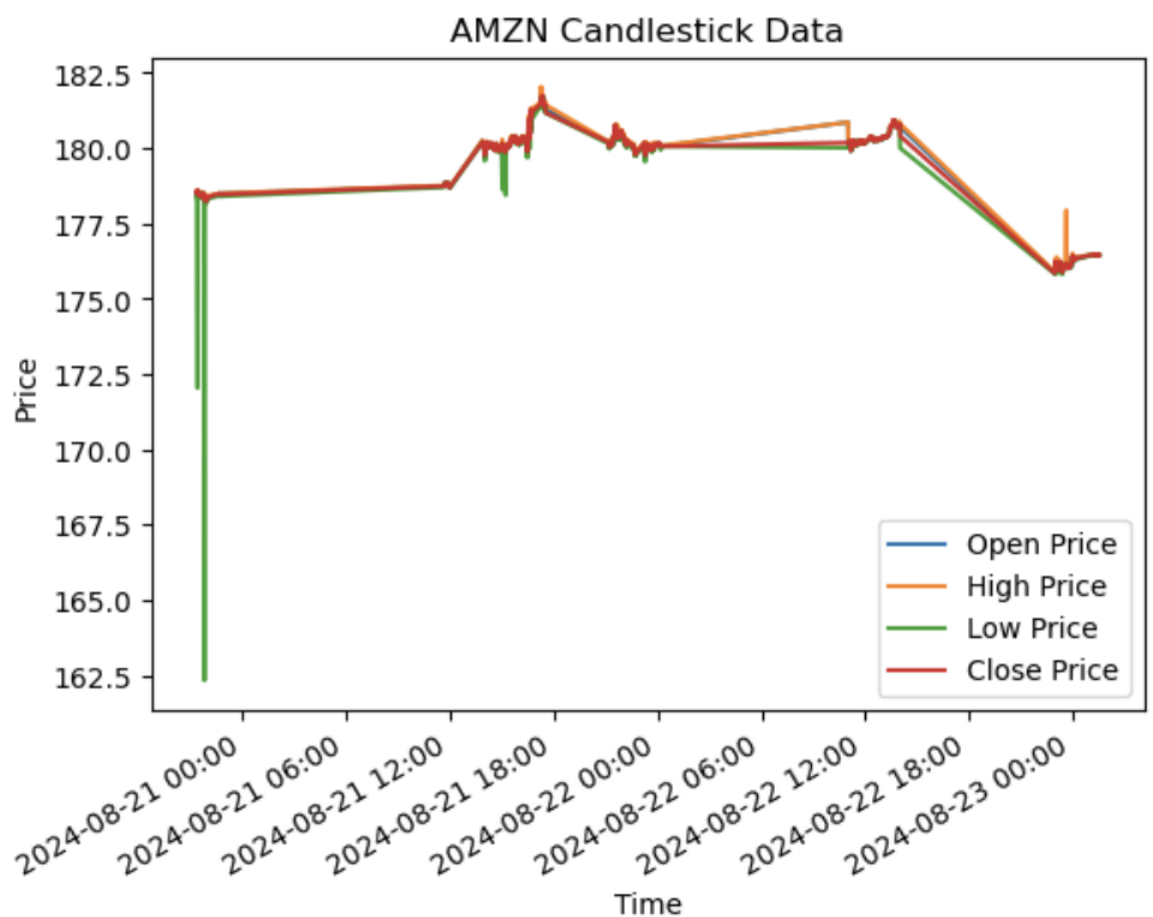
- Η τιμή κλεισίματος γενικά ακολουθεί τον κινούμενο μέσο όρο, όπως αναμενόταν, αλλά ο κινούμενος μέσος όρος παρέχει μια λιγότερο ασταθή άποψη της κίνησης της τιμής.

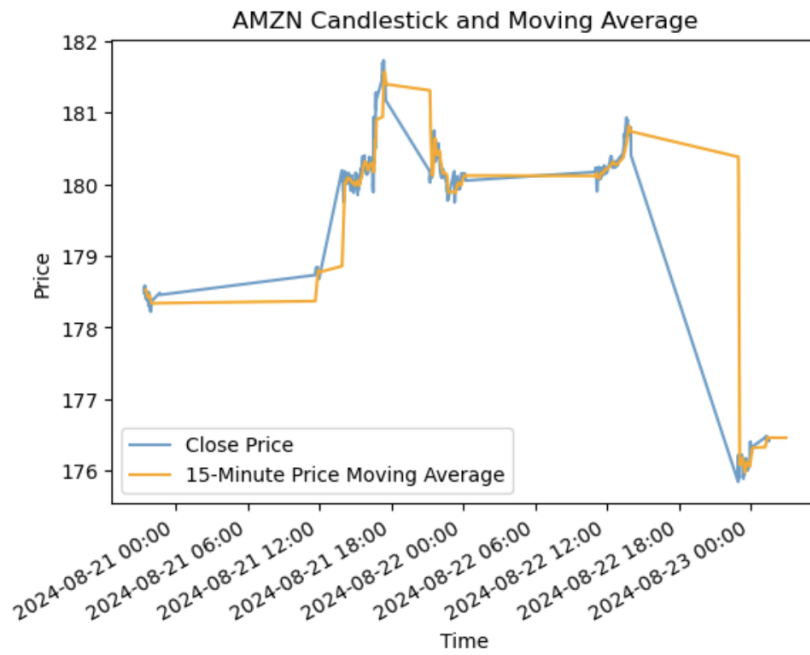
- **Δεδομένα AMZN και BTC-USD:**

Γράφημα 5: Δεδομένα Κηροπηγίων της AMZN

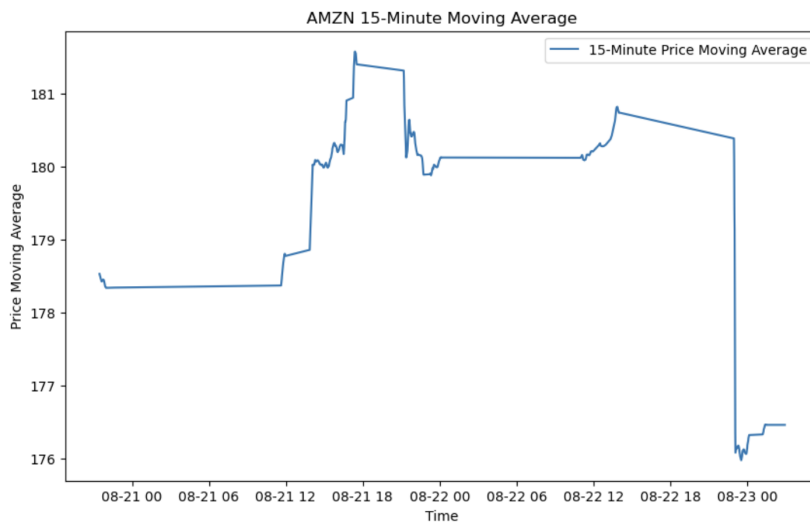


Γράφημα 6: Κινούμενος Μέσος Όρος 15 Λεπτών της AMZN

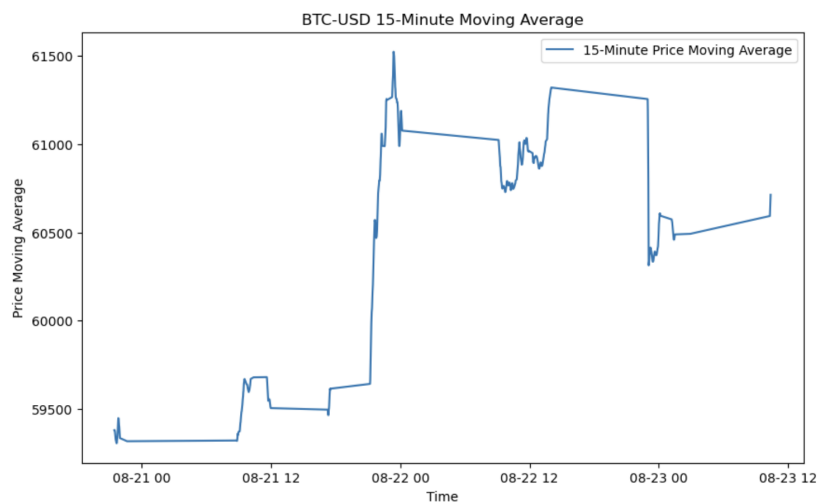
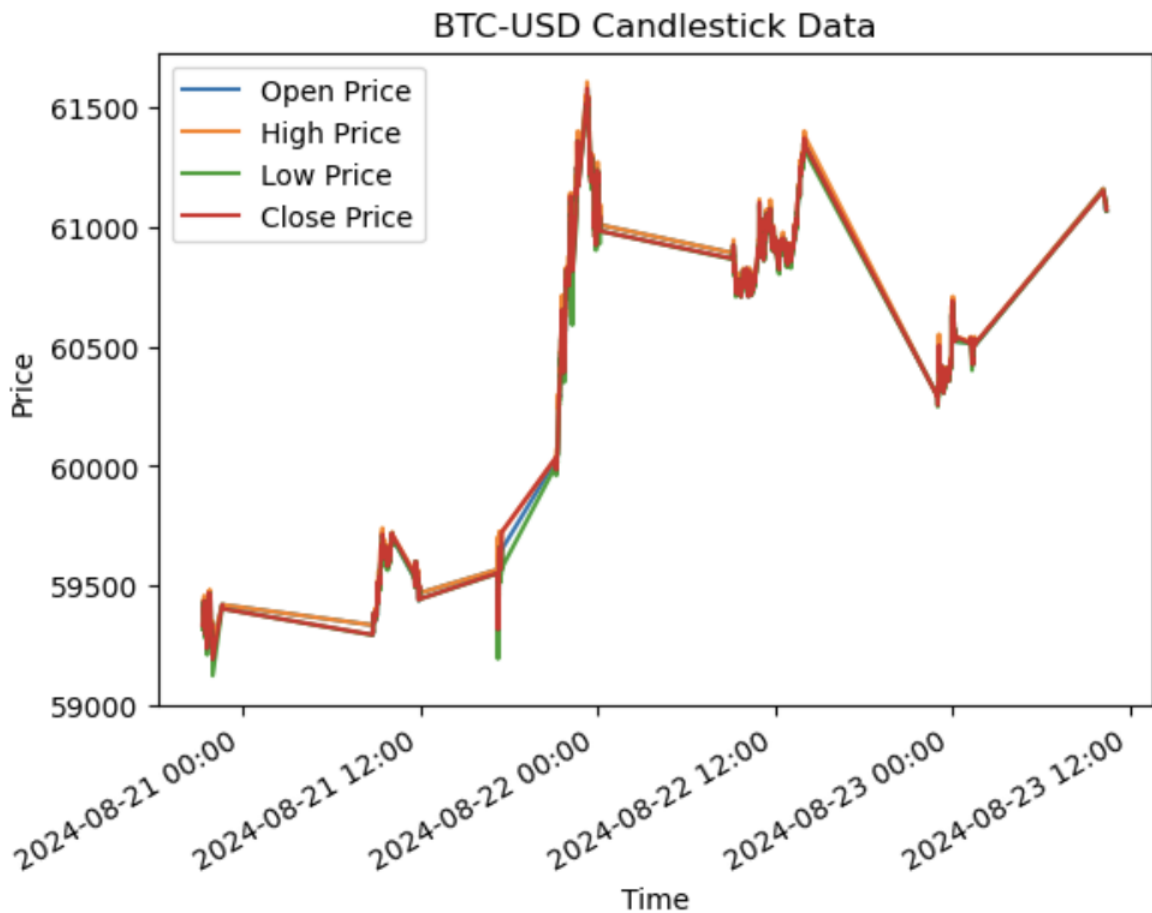




Γράφημα 7: Δεδομένα Κηροπηγίων BTC-USD



Γράφημα 8: Κινούμενος Μέσος Όρος 15 Λεπτών BTC-USD



Σχόλιο:

- ο Παρατηρήθηκαν παρόμοιες τάσεις και απόδοση για τα σύμβολα AMZN και BTC-USD. Το σύστημα διαχειρίστηκε αυτά τα σύμβολα με την ίδια αποδοτικότητα, διατηρώντας ελάχιστες διαφορές χρόνου και υψηλό χρόνο αδράνειας CPU.
- ο Αυτά τα γραφήματα δείχνουν περαιτέρω την ικανότητα του συστήματος να διαχειρίζεται πολλαπλά σύμβολα χωρίς υποβάθμιση της απόδοσης.