

# Gerçek Zamanlı Büyük Veri Analitiği ile Anomali Tespiti: Spark ve Kafka Entegrasyonu

Mahmut Satıcı(211307070),  
Deniz Ilgın Koca(211307017)  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
<https://github.com/Denizilgin/BigDataAnalysis>

**Özet**–Bu proje, Kafka ve Spark teknolojilerini kullanarak gerçek zamanlı veri işleme ve anomali tespiti gerçekleştirmeyi amaçlamaktadır. Seçilen veri seti üzerinde eksik verilerin temizlenmesi, normalizasyon, kategorik verilerin dönüştürülmesi gibi ön işlemler yapılmış ve veri dağılımı görselleştirilmiştir. Random Forest, Decision Tree ve Linear Regression gibi makine öğrenmesi modelleri kullanılarak başarı metrikleri değerlendirilmiştir. Kafka ile veri akışı sağlanmış, Spark Streaming ile gerçek zamanlı anomali tespiti yapılarak sonuçlar ilgili Kafka topic'lerine iletilmiştir. Proje çıktıları detaylı bir şekilde analiz edilip raporlanarak GitHub deposunda paylaşılmıştır.

**Anahtar Kelimeler** – Kafka, Spark Streaming, Anomali Tespiti, Makine Öğrenmesi, Gerçek Zamanlı Veri İşleme, Veri Görselleştirme

## I. GİRİŞ

### • Proje Amacı:

Bu proje, gerçek zamanlı veri işleme ve anomali tespiti yapmak için Kafka ve Spark Streaming teknolojilerini kullanmayı hedeflemektedir.

### • Kapsam:

Veri seti üzerinde ön işleme, görselleştirme ve makine öğrenmesi modellerinin geliştirilmesi. Kafka ile gerçek zamanlı veri akışı sağlanarak Spark ile anomali tespiti gerçekleştirilmesi.

### • Uygulanan Yöntemler:

#### 1. Veri Ön İşleme:

Eksik verilerin temizlenmesi, verilerin normalize edilmesi ve kategorik değişkenlerin dönüştürülmesi.

#### 2. Makine Öğrenmesi:

Random Forest, Decision Tree ve Linear Regression gibi algoritmaların kullanımı.

#### 3. Gerçek Zamanlı İşleme:

Kafka ve Spark entegrasyonu ile anlık veri analizi ve tespit edilen anomalilerin Kafka topic'lerine iletilmesi.

## II. PROJE ADIMLARI

### 1. Veri Seti Seçimi

Bu projede, "Automobile.csv" veri seti seçilmiştir. Bu veri seti, otomobillerin teknik özellikleri (motor gücü, ağırlık, hızlanma vb.) ile bazı kategorik bilgilerini (köken, model yılı gibi) içermektedir. Ayrıca, veri seti, belirli bir hedef değişken üzerinden anomali tespiti yapılabilecek yeterli çeşitliliğe sahiptir. Anomali tespiti için hem sürekli hem de kategorik değişkenlere sahip olması ve eksik veri içeriyor olması, bu veri setini projeye uygun hale getirmiştir. Aşağıda seçim nedenleri listelenmektedir:

**Çeşitlilik:** Veri seti, sayısal (horsepower, weight, acceleration) ve kategorik (origin, model\_year) özelliklerin bir kombinasyonunu içerir.

**Eksik Veri İçeriği:** Veri setinde bazı sütunlarda eksik değerler bulunması, veri ön işleme adımlarını uygulamak için fırsat sağlamıştır.

**Anlamlı Anomaliler:** Motor gücü, ağırlık gibi özelliklerde olabilecek aykırı değerler, otomobillerin performans analizine dair çıkarımlar yapmayı sağlar.

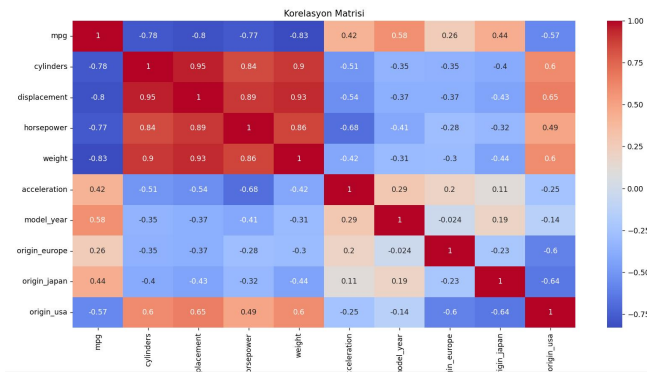
```
Proje > Automobile.csv > data
1 name,mpg,cylinders,displacement,horsepower,weight,acceleration,model_year,origin
2 chevrolet chevelle malibu,18,8,307,130,3504,12,70,usa
3 buick skylark 320,15,8,350,165,3693,11.5,70,usa
4 plymouth satellite,18,8,318,150,3436,11,70,usa
5 amc rebel sst,16,8,304,150,3433,12,70,usa
6 ford torino,17,8,302,140,3449,10.5,70,usa
7 ford galaxie 500,15,8,429,198,4341,10,70,usa
8 chevrolet impala,14,8,454,220,4354,9,70,usa
9 plymouth fury iii,14,8,440,215,4312,8.5,70,usa
10 pontiac catalina,14,8,455,225,4425,10,70,usa
11 amc ambassador dpl,15,8,390,190,3850,8.5,70,usa
12 dodge challenger se,15,8,383,170,3563,10,70,usa
13 plymouth 'cuda 340,14,8,340,160,3609,8,70,usa
14 chevrolet monte carlo,15,8,400,150,3761,9.5,70,usa
15 buick estate wagon (sw),14,8,455,225,3086,10,70,usa
16 toyota corona mark ii,24,4,113,95,2372,15,70,japan
17 plymouth duster,22,6,198,95,2833,15.5,70,usa
18 amc hornet,18,6,199,97,2774,15.5,70,usa
19 ford maverick,21,6,300,85,2587,16,70,usa
20 datsun pl510,27,4,97,88,2130,14.5,70,japan
21 volkswagen 1131 deluxe sedan,26,4,97,46,1835,20.5,70,europe
22 peugeot 504,25,4,110,87,2672,17.5,70,europe
23 audi 100 ls,24,4,107,90,2430,14.5,70,europe
24 saab 99e,25,4,104,95,2375,17.5,70,europe
25 bmw 2002,26,4,121,113,2234,12.5,70,europe
26 amc gremlin,21,6,258,90,2640,15,70,usa
```

(Veri setine ait bir görüntü)

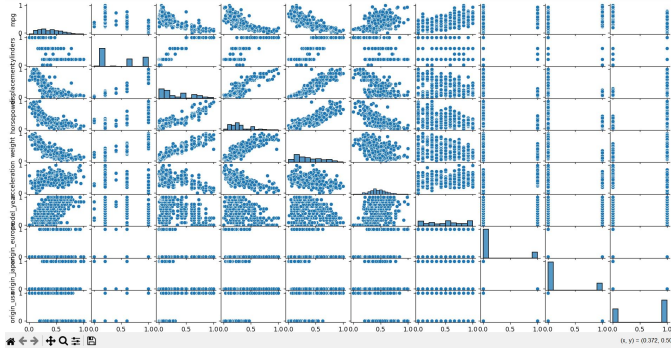
## 2. Kodda Kullandığımız Kütüphaneler ve Uygulanan İşlemler

### Kütüphaneler:

- pandas: Veri işleme ve tablo manipülasyonu için kullanıldı.
- numpy: Eksik verilerin doldurulması gibi matematiksel işlemler için kullanıldı.
- seaborn ve matplotlib: Veri görselleştirme (korelasyon matrisleri, dağılım grafikleri) için kullanıldı.
- sklearn.preprocessing.MinMaxScaler: Verilerin normalize edilmesi için kullanıldı.



(Korelasyon Matrisine ait bir resim)



(Oluşturulan grafiklere ait bir resim)

### Uygulanan Adımlar:

- Eksik Verilerin Doldurulması: horsepower sütununda eksik değerler bulunduğu tespit edildi ve bu sütun eksik değerlerin ortalaması ile dolduruldu.
- Kategorik Değişkenlerin Kodlanması: origin sütunu, one-hot encoding yöntemiyle sayısal değerlere dönüştürüldü. Örneğin, origin\_usa, origin\_europe ve origin\_japan gibi sütunlar oluşturuldu.

- Normalizasyon:

Tüm sayısal sütunlar Min-Max Scaling yöntemiyle [0, 1] aralığına getirildi. Bu, farklı ölçeklerdeki verilerin makine öğrenmesi algoritmaları tarafından daha iyi işlenmesini sağlar.

- Görselleştirme:

Korelasyon matrisi, veri setindeki özellikler arasındaki ilişkileri ortaya koymak için görselleştirildi.

Dağılım grafikleri ile verinin genel dağılımı incelendi.

- İşlenmiş Verinin Kaydedilmesi:

Ön işlemden geçirilen veri, Processed\_Automobile.csv adıyla yeni bir dosya olarak kaydedildi.

## 3. Yapay Zeka Modeli Geliştirme

Bu adımda, işlenmiş veriler kullanılarak makine öğrenmesi modelleri geliştirilmiş ve performansları karşılaştırılmıştır. Hedef, otomobillerin silindir sayısını tahmin edebilecek en iyi modeli belirlemektir. Silindir sayısı (cylinders), hedef değişken olarak belirlenmiş ve diğer tüm özellikler tahmin için kullanılmıştır.

```
1 # Veriyi yükle
2 data = pd.read_csv('Processed_Automobile.csv')
3
4 # Özellikler (X) ve hedef değişken (y)
5 X = data.drop('cylinders', axis=1) # target_column burada hedef değişken olmalı
6 y = data['cylinders'] # hedef değişken
7
8 # Veriyi eğitim ve test alanına ayır
9 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
10
11 # Modelleri tanımla
12 rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
13 dt_model = DecisionTreeRegressor(random_state=42)
14 lr_model = LinearRegression()
15
16 # Modelleri eğit
17 rf_model.fit(X_train, y_train)
18 dt_model.fit(X_train, y_train)
19 lr_model.fit(X_train, y_train)
20
21 # Tahminler
22 rf_pred = rf_model.predict(X_test)
23 dt_pred = dt_model.predict(X_test)
24 lr_pred = lr_model.predict(X_test)
25
26 # Başarı metriklerini hesapla
27 def print_metrics(model_name, y_true, y_pred):
28     mse = mean_squared_error(y_true, y_pred)
29     r2 = r2_score(y_true, y_pred)
30     print(f'{model_name} - MSE: {mse:.4f}')
31     print(f'{model_name} - R^2 Score: {r2:.4f}')
32
33 # Sonuçları yazdır
34 print_metrics('Random Forest', y_test, rf_pred)
35 print_metrics('Decision Tree', y_test, dt_pred)
36 print_metrics('Linear Regression', y_test, lr_pred)
```

(Modelleri oluşturan ve skorları üreten kod parçacığı)

- Veri Bölme:

İşlenmiş veri, modelin öğrenmesi ve test edilmesi için iki bölüme ayrılmıştır:

-Eğitim Verisi (Training Data): Modelin öğrenmesi için kullanılan veri (%80).

-Test Verisi (Test Data): Modelin performansını değerlendirmek için ayrılan veri (%20).

Bu ayırım, modelin gerçek hayatta hiç görmediği veriler üzerinde nasıl bir performans göstereceğini anlamak için yapılmıştır.

Eksik Değer Sayısı:

```
name      0
mpg        0
cylinders  0
displacement  0
horsepower 6
weight     0
acceleration 0
model_year 0
origin     0
dtype: int64
```

İşlenmiş Veri Kümesi:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
```

Data columns (total 10 columns):

#	Column	Non-Null Count	Dtype
0	mpg	398 non-null	float64
1	cylinders	398 non-null	float64
2	displacement	398 non-null	float64
3	horsepower	398 non-null	float64
4	weight	398 non-null	float64
5	acceleration	398 non-null	float64
6	model_year	398 non-null	float64
7	origin_europe	398 non-null	bool
8	origin_japan	398 non-null	bool
9	origin_usa	398 non-null	bool

(main.py dosyasının çıktısı)

- Kullanılan Modeller:

Üç farklı model kullanılarak tahminleme yapılmıştır:

- Random Forest Regressor:

Karar ağaçlarının bir topluluğunu kullanarak tahmin yapar. Ağaç sayısı (n\_estimators) 100 olarak belirlenmiştir. Daha karmaşık ilişkileri öğrenme kapasitesine sahiptir ve aşırı öğrenme riskini azaltır.

- Decision Tree Regressor:

Veriyi bir karar ağacı mantığıyla bölerek tahmin yapar. Hızlı bir şekilde çalışır, ancak aşırı öğrenmeye (overfitting) yatkındır.

- Linear Regression:

Özellikler ve hedef değişken arasındaki ilişkiyi doğrusal bir modelle tahmin eder. Daha basit ilişkilerde etkili, ancak doğrusal olmayan karmaşık ilişkilerde yetersizdir.

- Performans Ölçütleri:

Model performansını değerlendirmek için aşağıdaki metrikler kullanılmıştır:

Mean Squared Error (MSE):

Tahminlerin hedef değişkenden ne kadar sapma gösterdiğini ölçer. Daha düşük bir MSE, daha iyi bir model anlamına gelir.

R<sup>2</sup> (R-Kare) Skoru:

Modelin veriye ne kadar iyi uyduğunu gösterir. 1.0'a ne kadar yakınsa, model o kadar iyi demektir.

```
C:\Users\st_fa\Desktop\big data 2\Proje>p
Random Forest - MSE: 0.0027
Random Forest - R^2 Score: 0.9785
Decision Tree - MSE: 0.0020
Decision Tree - R^2 Score: 0.9839
Linear Regression - MSE: 0.0130
Linear Regression - R^2 Score: 0.8953

C:\Users\st_fa\Desktop\big data 2\Proje>
```

(Skor sonuçları)

- Sonuçlar:

- ✓ Random Forest:

MSE: 0.0027

R<sup>2</sup>: 0.9785

Çok düşük hata oranı ve yüksek doğrulukla en iyi sonucu vermiştir.

- ✓ Decision Tree:

MSE: 0.0020

R<sup>2</sup>: 0.9839

Random Forest'a benzer şekilde başarılı sonuçlar vermiştir.

- ✓ Linear Regression:

MSE: 0.0130

R<sup>2</sup>: 0.8953

Daha yüksek bir hata oranı ve daha düşük doğrulukla diğer modellere kıyasla daha zayıf bir performans göstermiştir.

- Yorum:

Karar ağaçlarına dayalı modeller (Random Forest ve Decision Tree), karmaşık ilişkileri daha iyi öğrenmiş ve Linear Regression modeline göre çok daha iyi sonuçlar vermiştir. Random Forest modeli, karar ağaçlarının topluluk yöntemiyle çalışması sayesinde aşırı öğrenme riskini azaltarak daha genellenebilir sonuçlar sağlamıştır. Bu nedenle, veri seti üzerindeki en uygun model olarak değerlendirilmiştir.

#### 4. Zookeeper, Kafka ve Spark Yükleme ve Kullanma Aşamaları

##### I. Zookeeper ve Kafka Yükleme

Zookeeper ve Kafka, gerçek zamanlı veri akışı işlemleri için birlikte çalışır. Kafka, mesajların üretildiği ve tüketildiği bir sistemdir; Zookeeper ise Kafka'nın koordinasyonu için gereklidir.

- ❖ Zookeeper ve Kafka'nın İndirilmesi:

Apache Kafka'nın resmi web sitesinden uygun Kafka sürümü indirilir.

Zookeeper indirilirken Kafka ile uyumlu sürümünün indirilmesi gerekir.



## ❖ Zookeeper'ın Çalıştırılması:

Zookeeper'ın başlatılması için bin/zookeeper-server-start.sh dosyası kullanılır.

```
Komut İstemi - zkserver
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\st_fa>zkserver

C:\Users\st_fa>call "C:\Java\jdk-1.8\bin\java "-Dzookeeper.log.dir
=C:\apache-zookeeper-3.8.4\bin\..\logs" "-Dzookeeper.log.file=zooke
eper-st_fa-server-MAHMUT.log" "-XX:+HeapDumpOnOutOfMemoryError" "-X
X:OnOutOfMemoryError=cmd /c taskkill /pid %p /t /f" -cp "C:\apache
-zookeeper-3.8.4\bin\..\build\classes;C:\apache-zookeeper-3.8.4\bin
\..\build\lib\*;C:\apache-zookeeper-3.8.4\bin\..\lib\*;C:\apache-zooke
per-3.8.4\bin\..\lib\*;C:\apache-zookeeper-3.8.4\bin\..\conf" org.a
pache.zookeeper.server.quorum.QuorumPeerMain "C:\apache-zookeeper-3
.8.4\bin\..\conf\zoo.cfg"
2024-12-21 12:28:53,765 [myid:] - INFO [main:o.a.z.s.q.QuorumPeerC
onfig@177] - Reading configuration from: C:\apache-zookeeper-3.8.4\
bin\..\conf\zoo.cfg
2024-12-21 12:28:53,800 [myid:] - WARN [main:o.a.z.s.u.VerifyingFi
leFactory@65] - C:\apache-zookeeper-3.8.4\data is relative. Prepend .
\ to indicate that you're sure!
2024-12-21 12:28:53,805 [myid:] - INFO [main:o.a.z.s.q.QuorumPeerC
onfig@440] - clientPortAddress is 0.0.0.0:2181
```

(Zookeeper'ın başlatılması)

## ❖ Kafka Broker'ın Başlatılması:

Kafka, bir broker üzerinden çalışır ve veri akışı buradan yönetilir.

Kafka'yı başlatmak için bin/kafka-server-start.sh komutu kullanılır ve Kafka yapılandırma dosyasına bağlı olarak çalıştırılır.

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\kafka_2.13-3.7.2\bin\windows>kafka-server-start.bat ..\..\config
\server.properties
[2024-12-21 12:29:24,465] INFO Registered kafka:type=kafka.Log4jCon
troller MBean (kafka.utils.Log4jControllerRegistration$)
[2024-12-21 12:29:24,799] INFO Setting -D jdk.tls.rejectClientIniti
atedRenegotiation=true to disable client-initiated TLS renegotiatio
n (org.apache.zookeeper.common.X509Util)
[2024-12-21 12:29:24,924] INFO starting (kafka.server.KafkaServer)
[2024-12-21 12:29:24,925] INFO Connecting to zookeeper on localhost
:2181 (kafka.server.KafkaServer)
[2024-12-21 12:29:24,945] INFO [ZooKeeperClient Kafka server] Initi
alizing a new session to localhost:2181. (kafka.zookeeper.ZooKeeper
Client)
[2024-12-21 12:29:24,955] INFO Client environment:zookeeper.version
=3.8.4-9316c2a7a97e1666d8f4593f34dd6fc36ecc436c, built on 2024-02-1
2 22:16 UTC (org.apache.zookeeper.ZooKeeper)
[2024-12-21 12:29:24,955] INFO Client environment:host.name=mahmut
(org.apache.zookeeper.ZooKeeper)
[2024-12-21 12:29:24,956] INFO Client environment:java.version=1.8.
```

(Kafkanın başlatılması)

## II. Kafka'nın Yapılandırılması

Kafka, mesajların akışı için topic adlı veri kanallarını kullanır. Veri üretmek için Producer, tüketmek için ise Consumer kullanılır.

### ❖ Topic Oluşturma:

Kafka üzerinde mesajların aktarılacağı bir topic oluşturulur. Bu topic, verinin depolandığı bir kanal gibi çalışır.

Topic oluştururken partition ve replication gibi parametreler ayarlanır. Bu parametreler, veri paralelizmini ve güvenliğini etkiler.

### ❖ Kafka Producer ve Consumer Kullanımı:

Producer, topic'e veri gönderir. Veriler, gerçek zamanlı olarak veya belirli aralıklarla Kafka'ya aktarılır.

Consumer, bu verileri ilgili topic'ten okur ve işleme alır.

Kafka'nın tasarımı sayesinde aynı topic'i birden fazla consumer grubu dinleyebilir.

### ❖ Anomali Tespiti:

Buradaki anomali tespitinde beygir gücü 200'den fazla veya litre hacmi 10'dan küçük olan arabaların tespiti yapılmakta ve işaretlenmektedir.

```
C:\kafka_2.13-3.7.2\bin\windows>kafka-console-consumer --bootstrap
server localhost:9092 --topic araba_i --from-beginning
{"name": "buick skylark 320", "mpg": 15.0, "cylinders": 8, "displac
ement": 350.0, "horsepower": 165.0, "weight": 3693, "acceleration":
11.5, "model_year": 70, "origin": "usa"}
{"name": "amc rebel sst", "mpg": 16.0, "cylinders": 8, "displacemen
t": 384.0, "horsepower": 150.0, "weight": 3433, "acceleration": 12.
0, "model_year": 70, "origin": "usa"}
{"name": "plymouth cuda 340", "mpg": 14.0, "cylinders": 8, "displa
cement": 340.0, "horsepower": 160.0, "weight": 3609, "acceleration"
: 8.0, "model_year": 70, "origin": "usa"}
{"name": "ford maverick", "mpg": 21.0, "cylinders": 6, "displacemen
t": 200.0, "horsepower": 85.0, "weight": 2587, "acceleration": 16.0
, "model_year": 70, "origin": "usa"}
{"name": "datsun pl510", "mpg": 27.0, "cylinders": 4, "displacement"
: 97.0, "horsepower": 88.0, "weight": 2130, "acceleration": 14.5,
"model_year": 70, "origin": "japan"}
{"name": "peugeot 504", "mpg": 25.0, "cylinders": 4, "displacement"
: 110.0, "horsepower": 87.0, "weight": 2672, "acceleration": 17.5,
"model_year": 70, "origin": "europe"}
{"name": "toyota corona", "mpg": 25.0, "cylinders": 4, "displacemen
```

(Şekil 1. Producer topic)

```
origin:"usa", "anomalies": "High horsepower"}
{"name": "toyota corona", "mpg": 25.0, "cylinders": 4, "displacemen
t": 110.0, "horsepower": 87.0, "weight": 2672, "acceleration": 17.5,
"model_year": 70, "origin": "europe", "anomalies": "High horsepower"}
{"name": "peugeot 504", "mpg": 25.0, "cylinders": 4, "displacement"
: 110.0, "horsepower": 87.0, "weight": 2672, "acceleration": 17.5,
"model_year": 70, "origin": "europe", "anomalies": "High horsepower"}
{"name": "datsun pl510", "mpg": 27.0, "cylinders": 4, "displacement"
: 97.0, "horsepower": 88.0, "weight": 2130, "acceleration": 14.5,
"model_year": 70, "origin": "japan", "anomalies": "High horsepower"}
{"name": "ford maverick", "mpg": 21.0, "cylinders": 6, "displacemen
t": 200.0, "horsepower": 85.0, "weight": 2587, "acceleration": 16.0
, "model_year": 70, "origin": "usa", "anomalies": "High horsepower"}
{"name": "plymouth cuda 340", "mpg": 14.0, "cylinders": 8, "displa
cement": 340.0, "horsepower": 160.0, "weight": 3609, "acceleration"
: 8.0, "model_year": 70, "origin": "usa", "anomalies": "High horsepower"}
{"name": "amc rebel sst", "mpg": 16.0, "cylinders": 8, "displacemen
t": 384.0, "horsepower": 150.0, "weight": 3433, "acceleration": 12.
0, "model_year": 70, "origin": "usa", "anomalies": "High horsepower"}
{"name": "buick skylark 320", "mpg": 15.0, "cylinders": 8, "displac
ement": 350.0, "horsepower": 165.0, "weight": 3693, "acceleration":
11.5, "model_year": 70, "origin": "usa", "anomalies": "High horsepower"}
```

(Şekil 1.2 Consumer Topic)

### III. Spark Yükleme

Apache Spark, büyük veri işleme platformudur ve Kafka ile entegre çalışarak gerçek zamanlı veri akışı sağlar.

#### ❖ Spark'ın İndirilmesi ve Kurulumu:

Apache Spark'ın resmi web sitesinden Spark sürümü indirilir. Spark, bağımsız bir sistemde çalışabileceği gibi, Hadoop gibi bir altyapı ile entegre edilerek de kullanılabilir.

#### ❖ Spark Ortamının Hazırlanması:

Spark, bir yerel modda (local mode) çalıştırılabilir veya daha büyük veri işleme için bir küme (cluster) modunda kullanılabilir. Spark, veri işlemek için RDD, DataFrame veya Dataset gibi veri yapıları sunar.

#### ❖ Spark ve Kafka Entegrasyonu:

Spark Streaming, Kafka'dan gelen verileri gerçek zamanlı olarak işler. Kafka ile Spark entegrasyonu, Spark Streaming modülünün Kafka bağlayıcılarını kullanarak gerçekleştirilir. Bu entegrasyon ile Kafka'dan gelen veriler işlenip sonuçlar diğer sistemlere aktarılabilir.

### IV. Spark ve Kafka ile Gerçek Zamanlı İşlem

Spark ve Kafka entegrasyonu, verinin üretilip tüketilmesi için kullanılır. Bu süreçte:

Kafka'dan gelen veri, Spark Streaming üzerinden alınır. Spark, bu veriyi işler ve sonuçlar tekrar Kafka'ya veya bir başka hedef sisteme gönderilir. Gerçek zamanlı veri analitiği ve anomali tespiti gibi işlemler bu süreçte gerçekleştirilir.

### KAYNAKLAR

- [1] M. King, B. Zhu, and S. Tang, "Optimal path planning," *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.
- [2] H. Simpson, *Dumb Robots*, 3<sup>rd</sup> ed., Springfield: UOS Press, 2004, pp.6-9.
- [3] M. King and B. Zhu, "Gaming strategies," in *Path Planning to the West*, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.
- [4] B. Simpson, et al, "Title of paper goes here if known," unpublished.
- [5] J.-G. Lu, "Title of paper with only the first word capitalized," *J. Name Stand. Abbrev.*, in press.
- [6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Translated J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [*Digest 9<sup>th</sup> Annual Conf. Magnetism Japan*, p. 301, 1982].
- [7] M. Young, *The Technical Writer's Handbook*, Mill Valley, CA: University Science, 1989.
- [8] J. K. Author, "Title of paper," in *Unabbreviated Name of Conf.*, City of Conf., Abbrev. State (if given), year, pp xx-xxx.
- [9] R. J. Vidmar. (1994). *On the use of atmospheric plasmas as electromagnetic reflectors* [Online]. Available FTP: atmnext.usc.edu



BvagProje.pdf

Directory: pub/etext/1994 File: atmosplasma.txt