

# GÜVENİLİRLİK SKORU ALGORİTMASI RAPORU

## ÖZET

Bu rapor, Veerify sisteminde bir haberin güvenilirliğini hesaplamak için kullanılan **Credibility Score Algoritması**'nı matematiksel ve pratik perspektiften açıklamaktadır.

### Ana Formül:

$$\text{Score} = (\sum \text{Kaynak Skoru} \times \text{Similarity}) / \text{Benzersiz Sayı} \times \text{Çeşitlilik Faktörü}$$

Bu algoritma, benzer haberlerin kaynağının güvenilirliğini ve haberlerin benzerlik derecesini dikkate alarak, 0.0 ile 1.0 arasında bir güvenilirlik puanı (Score) hesaplamaktadır.

## 1. GİRİŞ

### 1.1 Problem Tanımı

Dezenformasyon çağında, bir haberin doğruluğunu belirlemek çok katmanlı bir problem:

- Tekli kaynak problemi:** Bir kaynağın kendisine güvenmenin riski
- Benzerlik problemi:** Aynı sahte haber birden fazla sitede yayılabilir
- Kaynak güvenilirliği:** Tüm kaynaklar eşit derecede güvenilir değildir
- Zamansal esneklik:** Haberler zamanla değişebilir, pero esansiyel mesaj aynı kalır

### 1.2 Çözüm Yaklaşımı

**Güvenilirlik Çeşitlendirmesi İlkesi:** Bir haber ne kadar çeşitli ve güvenilir kaynaklar tarafından raporlanırsa, o kadar güvenilir olma ihtimali yüksektir.

#### Örnek:

- BBC, DW, NTV ve Sözcü aynı haberi raporlarsa → Yüksek güvenilirlik
- Sadece 1-2 kaynak raporlarsa → Düşük güvenilirlik
- Hiçbir güvenilir kaynak raporlamazsa → Şüpheli

---

## 2. ALGORİTMA TASARIMI

### 2.1 Temel Prensipler

#### Prensip 1: Çeşitlilik > Tekrarlama

Mantık: Aynı yayın grubundaki siteler birbirini kopyalarken, farklı yayın gruplarında aynı haber varsa daha güvenilir.

Örnek:

✓ BBC (Independent) + NTV (Doğan) + Sözcü (Independent)  
= 3 farklı çıkmazda aynı haber = GÜVENİLİR

✗ Sözcü + Sözcü Haber + Sözcü TV  
= Aynı grup, tekrarlar = DÜŞÜK GÜVEN

#### Prensip 2: Güvenilirlik Ağırlıklandırması

Mantık: Tüm kaynaklar eşit değildir.

Örnek:

✓ BBC (0.95 credibility) raporlarsa = ağır  
⚠ Bilinmeyen blog (0.40 credibility) raporlarsa = hafif

#### Prensip 3: Benzerlik Eşiği

Mantık: %75+ benzerlik = aynı haber  
%50-75% benzerlik = benzer/ilişkili  
<%50% benzerlik = farklı haber

Cosine similarity metriği ile ölçülür.

### 2.2 Algoritmanın Yapı Taşları

#### Bileşen 1: Cosine Similarity

İki metin arasındaki benzerliği ölçer (0.0-1.0).

#### Bileşen 2: Kaynak Güvenilirlik Skoru

Her kaynağın inherent güvenilirliği (0.0-1.0).

### Bileşen 3: Çeşitlilik Faktörü

Kaç benzersiz kaynaktan haber raporlanması.

## 3. MATEMATİKSEL FORMÜLLER

### 3.1 Ana Formül (Detaylı)

$$S = \frac{\sum_{i=1}^n (\text{credibility}_i \times \text{similarity}_i)}{n} \times D(n)$$

Nerede:

- **S** = Credibility Score (Final puanı)
- **n** = Benzersiz kaynak sayısı
- **credibility<sub>i</sub>** = i'inci kaynağın güvenilirlik skoru (0.0-1.0)
- **similarity<sub>i</sub>** = i'inci kaynağın haberiyle orijinal haberin benzerliği (0.0-1.0)
- **D(n)** = Çeşitlilik faktörü (diversity factor)

### 3.2 Cosine Similarity (Benzerlik Hesaplaması)

$$\text{similarity} = \frac{A \cdot B}{\|A\| \times \|B\|}$$

Nerede:

- **A** = Orijinal haberin embedding vektörü (512-dim)
- **B** = Benzer haber adayının embedding vektörü (512-dim)
- **A · B** = Nokta çarpımı (dot product)
- **||A||** = Vektör A'nın normu (magnitude)

Örnek Hesaplama:

A = [0.12, -0.45, 0.89, 0.34, ...] (512 boyut)

B = [0.15, -0.40, 0.85, 0.36, ...]

Dot Product (A · B) = 0.12×0.15 + (-0.45)×(-0.40) + 0.89×0.85 + ...  
= 0.018 + 0.18 + 0.757 + ...

= ~450 (512 sayısal işlem sonrası)

$$||A|| = \sqrt{(0.12^2 + (-0.45)^2 + 0.89^2 + \dots)} = \sqrt{1.0} = 1.0$$

$$||B|| = \sqrt{(0.15^2 + (-0.40)^2 + 0.85^2 + \dots)} = \sqrt{1.0} = 1.0$$

similarity = 450 / (1.0 × 1.0) = 0.88 (88% benzer)

#### Eksiksiz Açıklama:

- **similarity = 1.0** → Tamamen aynı metin
- **similarity = 0.75** → Aynı haberin, farklı yazınla anlatımı
- **similarity = 0.50** → Kısmen ilişkili
- **similarity = 0.0** → Tamamen farklı

### 3.3 Çeşitlilik Faktörü

$$D(n) = \min(n / \text{threshold}, 1.0)$$

#### Nerede:

- **n** = Benzersiz kaynak sayısı
- **threshold** = Doyum noktası (genellikle 10)

#### Örnek Hesaplamalar:

n	Hesaplama	D(n)	Açıklama
	a		
1	1/10	0.10	Çok az kaynak
3	3/10	0.30	Yetersiz çeşitlilik
5	5/10	0.50	Orta çeşitlilik
10	10/10	1.00	Yeterli çeşitlilik
15	15/10=1.0	1.00	Tam puan (cap at 1.0)

#### Mantık:

- Tek kaynakla: En fazla 0.10 × score (risk çok yüksek)
- 10 kaynakla: 1.0 × score (tam puan, çeşitlilik sağlandı)
- 15+ kaynakla: 1.0 × score (daha fazlası yardımcı değil)

## 4. ADIM-ADIM ÖRNEK HESAPLAMA

### 4.1 Senaryo: "Yapay Zeka İçin Yeni Düzenleme" Haberi

#### Giriş Verisi:

Orijinal URL: <https://techsite.com/ai-regulation-news>

Haber Başlığı: "Avrupa'da Yapay Zeka Düzenlemesi Kabul Edildi"

Haber İçeriği: ~500 kelime

### 4.2 Step 1: Embedding Oluşturma

Input: "Avrupa'da Yapay Zeka Düzenlemesi Kabul Edildi..."

spaCy + Sentence Transformers:

Model: paraphrase-multilingual-MiniLM-L12-v2

Output: Embedding Vector (512-dim)

[0.123, -0.456, 0.789, 0.234, ..., -0.567]

### 4.3 Step 2: Benzer Haberler Bulma

MongoDB Atlas Search özelliği kullanılır

#### Sonuçlar:

Sıra	Kaynak	Başlık	Similarity	Credibility
1	BBC Türkçe	"EU Passes AI Act"	0.91	0.95
2	Deutsche Welle	"Künstliche Intelligenz Verordnung"	0.88	0.93
3	NTV	"Avrupa AI'yı Düzenledi"	0.85	0.80
4	Euronews	"Europe's AI Regulation"	0.83	0.88
5	Sözcü	"Yapay Zekanın Düzenlenmesi"	0.82	0.75
6	Cumhuriyet	"AB, Yapay Zeka Yasası..."	0.79	0.78

7	Reuters	"EU AI Rulebook Finalized"	0.77	0.96
---	---------	----------------------------	------	------

**Filtreleme:** Similarity > 0.75 olan 7 haber seçildi.

#### 4.4 Step 3: Benzerlik × Güvenilirlik Çarpımı

Haber 1:  $0.91 \times 0.95 = 0.8645$

Haber 2:  $0.88 \times 0.93 = 0.8184$

Haber 3:  $0.85 \times 0.80 = 0.6800$

Haber 4:  $0.83 \times 0.88 = 0.7304$

Haber 5:  $0.82 \times 0.75 = 0.6150$

Haber 6:  $0.79 \times 0.78 = 0.6162$

Haber 7:  $0.77 \times 0.96 = 0.7392$

Toplam ( $\Sigma$ ) = 5.6637

#### 4.5 Step 4: Ortalama Hesaplama

Ağırlıklı Ortalama =  $\Sigma / n$

=  $5.6637 / 7$

= 0.8091

#### 4.6 Step 5: Çeşitlilik Faktörü Uygulama

Benzersiz kaynak sayısı (n): 7

Threshold: 10

$D(7) = \min(7/10, 1.0) = 0.70$

#### 4.7 Step 6: Final Skoru Hesaplama

Final Score = Ağırlıklı Ortalama × D(n)

=  $0.8091 \times 0.70$

= 0.5664

≈ 0.57 (57%)

---

## 5. DURUM SINIFLANDIRMASI

Final skor hesaplandıktan sonra, haberin durumu belirlenir:

## 5.1 Sınıflandırma Tablosu

Score Aralığı	Kaynak Sayısı	Durum	Açıklama	Icon
≥ 0.80	≥ 10	● VERIFIED	Çok güvenilir	✓
0.70-0.79	≥ 7	● LIKELY TRUE	Muhtemelen doğru	✓
0.50-0.69	≥ 5	● UNCERTAIN	Kesin Değil	⚠
0.30-0.49	≥ 3	● DISPUTED	Anlaşmazlık var	✗
< 0.30	< 3	● UNVERIFIED	Doğrulanamadı	✗

## 5.2 Örneğimize Uygulanması

Score = 0.57

Kaynak Sayısı = 7

Tablo'da: 0.50-0.69 aralığı

→ Durum: ● UNCERTAIN

→ Açıklama: "Emin değiliz, daha fazla doğrulama gerekir"

## 6. İLERİ KONULAR

### 6.1 Kaynak Güvenilirlik Skoru Nasıl Belirlenir?

#### Yöntem 1: Manuel Kurulum (MVP)

BBC Türkçe = 0.95 (yüksek standart)

Deutsche Welle = 0.93 (uluslararası)

NTV = 0.80 (ana akım)

Bilinmeyen Blog = 0.40 (düşük standart)

#### Yöntem 2: Dinamik Hesaplama (v2)

```
credibility_score = (  
    accuracy_history × 0.40 +  
    editorial_standards × 0.30 +  
    transparency_level × 0.20 +  
    correction_speed × 0.10
```

)

## 6.2 Similarity Threshold Neden 0.75?

### Deneysel Bulgular:

< 0.50: Farklı haberler (False positive riski düşük)  
0.50-0.75: Kısmen ilişkili haberler  
> 0.75: Aynı haberin farklı varyasyonları

Threshold 0.75 = İyi dengeleme

## 6.3 Çeşitlilik Faktörü İçin Neden 10?

### İstatistiksel Analiz:

n=1: 1 kaynakla %10 puan alabiliyor (riskli)  
n=3: 3 kaynakla %30 puan (yetersiz)  
n=5: 5 kaynakla %50 puan (orta)  
n=10: 10 kaynakla %100 puan (yeterli)

Sosyal bilimler araştırmalarında:  
"3-5 bağımsız kaynakta aynı bulgu" = yeterli  
"10+ kaynak" = statistically significant

## 6.4 Anomali Tespiti

### Örnek 1: Çok Yüksek Skor Ama Düşük Kaynak

Score = 1.0, n=1  
→ İşaret: Tüm benzer haberler aynı kaynaktan?  
→ İşlem: Manuel review

### Örnek 2: Çok Düşük Skor Ama Yüksek Kaynak

Score = 0.2, n=15  
→ İşaret: Benzer haber yok mu?  
→ İşlem: Haber çok yeni/özel mi?

---



## 7. EDGE CASES

### 7.1 Haber Henüz Tarandığında

Durum: Haber 30 dakika önce yayınlandı

Benzer Haberleri: 0 tane

n = 0

Çözüm 1 (Bugünkü): Score = 0.0 (UNVERIFIED)

Çözüm 2 (v2): "Breaking News - Beklemede" durumu

### 7.2 Kontroversiyel Konu

Durum: Politikacı A vs B haberi

Haber Sayısı: 50+ (çelişkili)

Benzer + Güvenilir: 15 kaynak

Benzer + Güvenilmez: 35 kaynak

Sorun: Similarity aynı ama credibility farklı

Çözüm: Minimum credibility threshold (e.g., 0.50)

### 7.3 Viral Sahte Haber

Durum: False haber 100 sitede var (ama hepsi birbirini kopyaladı)

Benzer Haberler: 100

Benzersiz Kaynaklar: 1 (ilk yayınlayan)

Score =  $0.8091 \times 0.10 = 0.0809$  (8%) = UNVERIFIED ✓

Çeşitlilik faktörü yardım etti!

---

## 8. OPTİMİZASYONLAR VE İYİLEŞTİRMELER

### 8.1 Sezonsal Ayarlama

Son 7 gün: full weight

8-30 gün:  $0.8 \times \text{weight}$

1-3 ay:  $0.5 \times \text{weight}$

3+ ay:  $0.2 \times \text{weight}$

Mantık: Eski haberler değeri düşüyor

## 8.2 Zaman Damgası Boosting

Aynı saatte yayımlanan kaynaklar:

Normal: 1.0x

Aynı saatte yayınlanan: 1.2x (daha güvenilir)

Mantık: Koordineli raporlama = daha güvenilir

## 8.3 Coğrafi Çeşitlilik

$D_{\text{coğrafi}}(n) = \min(\text{unique\_countries} / 5, 1.0)$

Mantık: Farklı ülkelerden kaynaklı → daha güvenilir

Örnek: TR, DE, GB, US = 4 ülke → 0.8x boost

## 8.4 İyileştirilmiş Formula (v2 Önerisi)

Score =  $(\sum \text{credibility}_i \times \text{similarity}_i / n)$   
×  $D_{\text{çeşitlilik}}(n)$   
×  $D_{\text{zaman}}(\text{age})$   
×  $D_{\text{coğrafi}}(\text{countries})$

Nerede:

$D_{\text{çeşitlilik}} = \min(n/10, 1.0)$

$D_{\text{zaman}} = 1 - (\text{age\_days}/365)$  if age < 365

$D_{\text{coğrafi}} = \min(\text{unique\_countries}/5, 1.0)$

---

# 9. PERFORMANS ANALIZI

## 9.1 Hesaplama Karmaşıklığı

Zaman Kompleksitesi:

$O(m \times d)$

Nerede:

m = Benzer haber sayısı (50)

d = Embedding boyutu (512)

Tipik Süre:

- Embedding: 2s

- Similarity search: 1s (PostgreSQL pgvector)
- Score calculation: 100ms
- TOPLAM: ~3.1 saniye

Optimizasyon: Batch processing ile parallelizable

## 9.2 Doğruluk Metrikleri

### Precision (Yanlış Pozitif):

"Sistem verified dediği haberler gerçekten doğru mu?"

Target: >85%

### Recall (Yanlış Negatif):

"Sistem unverified dediği haberler gerçekten yanlış mı?"

Target: >75%

### F1-Score:

$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$

Target: >0.80

---

## 10. TEST SENARYOLARI

### 10.1 Test Case 1: Güvenilir Haber

Input: BBC'nin "Hükümet Açıklaması" haberi

Beklenen Sonuç:  $\geq 0.70$  (VERIFIED veya LIKELY TRUE)

Sebebi: Yüksek güvenilirlik kaynak + birden fazla kaynaktan

### 10.2 Test Case 2: Şüpheli Haber

Input: Anonim blog "Ünlü öldü (yalan)" haberi

Beklenen Sonuç:  $< 0.30$  (UNVERIFIED)

Sebebi: Düşük güvenilirlik + hiçbir resmi kaynaktan yok

### 10.3 Test Case 3: Kontroversiyel Haber

Input: "Politikacı X Yaptı/Yapmadı" haberi  
Beklenen Sonuç: 0.40-0.60 (UNCERTAIN)  
Sebeup: Benzer haberler var ama eliřkili kaynaklar

#### 10.4 Test Case 4: Breaking News

Input: Yeni olayı ilk raporlayan haber  
Beklenen Sonuç: 0.0 (UNVERIFIED - henüz)  
Sebeup: Benzer haber yok, ama zamanla artacak

---

## 11. KARŐILAŐTIRMA: DİĐER YÖNTEMLERLE

### 11.1 Basit Sayım (Naive Count)

Score\_naive =  $n / 100$  (0-1 arasına normalize)

Örnek:

10 kaynakta → 0.10 (10%)

50 kaynakta → 0.50 (50%)

- ✗ Sorun: Kaynak kalitesi görmezden geliyor
- ✗ Sorun: Sahte haberler 100 sitede olabilir

### 11.2 Sadece Credibility Ortalaması

Score =  $\sum \text{credibility}_i / n$

Örnek:

$(0.95 + 0.93 + 0.80) / 3 = 0.89$

- ✗ Sorun: Benzerlik görmezden geliyor
- ✗ Sorun: Farklı haberler de yüksek score alabilir

### 11.3 Önerilen Yöntem (Hybrid)

Score =  $(\sum \text{credibility}_i \times \text{similarity}_i / n) \times D(n)$

- ✓ Avantaj: Kalite + Benzerlik + eřitlilik
- ✓ Avantaj: Robust ve sağlam
- ✓ Avantaj: False positives minimize

---

## 12. UYGULAMA KODU (Python)

```
def calculate_credibility_score(similar_articles):
    """
    Güvenilirlik skorunu hesapla

    Args:
        similar_articles: [
            {
                'source_credibility': 0.95,
                'similarity': 0.91,
                'source_id': 1
            },
            ...
        ]

    Returns:
        float: 0.0-1.0 arası credibility score
    """

    if not similar_articles:
        return 0.0

    # Step 1: Benzersiz kaynakları say
    unique_sources = len(set(
        article['source_id']
        for article in similar_articles
    ))

    # Step 2: Similarity × Credibility çarpımını hesapla
    weighted_sum = sum(
        article['source_credibility'] * article['similarity']
        for article in similar_articles
    )

    # Step 3: Ağırlıklı ortalamayı hesapla
    weighted_average = weighted_sum / len(similar_articles)

    # Step 4: Çeşitlilik faktörünü hesapla
    diversity_factor = min(unique_sources / 10.0, 1.0)

    # Step 5: Final skoru hesapla
```

```
final_score = weighted_average * diversity_factor
```

```
return round(final_score, 2)
```

```
def determine_status(score, source_count):  
    """  
    Score'a göre doğrulama durumunu belirle  
    """  
    if score >= 0.80 and source_count >= 10:  
        return "verified", "🟢 Doğrulanmış"  
    elif score >= 0.70 and source_count >= 7:  
        return "likely_true", "🟢 Muhtemelen Doğru"  
    elif score >= 0.50 and source_count >= 5:  
        return "uncertain", "🟡 Belirsiz"  
    elif score >= 0.30 and source_count >= 3:  
        return "disputed", "🔴 Anlaşmazlık"  
    else:  
        return "unverified", "🔴 Doğrulanamadı"
```

```
# ÖRNEK KULLANIM
```

```
similar_articles = [  
    {'source_credibility': 0.95, 'similarity': 0.91, 'source_id': 1},  
    {'source_credibility': 0.93, 'similarity': 0.88, 'source_id': 2},  
    {'source_credibility': 0.80, 'similarity': 0.85, 'source_id': 3},  
    {'source_credibility': 0.88, 'similarity': 0.83, 'source_id': 4},  
    {'source_credibility': 0.75, 'similarity': 0.82, 'source_id': 5},  
    {'source_credibility': 0.78, 'similarity': 0.79, 'source_id': 6},  
    {'source_credibility': 0.96, 'similarity': 0.77, 'source_id': 7},  
]
```

```
score = calculate_credibility_score(similar_articles)
```

```
status, label = determine_status(score, len(set(a['source_id'] for a in similar_articles)))
```

```
print(f"Score: {score}")
```

```
print(f"Status: {status} ({label})")
```

```
# Output: Score: 0.57
```

```
#      Status: uncertain (🟡 Belirsiz)
```

---

## 13. SONUÇ VE ÖNERİLER

## 13.1 Algoritmanın Güçlü Yönleri

### ✓ Çeşitlilik Duyarlılığı

- Tek kaynağa güven almıyor

### ✓ Kalite-Benzerlik Dengesi

- Hem kaynağın kalitesi hem benzerliği önemli

### ✓ Skalabilite

- 1 kaynaktan 100 kaynağa ölçeklenir

### ✓ Basitlik

- Anlaşılması ve uygulanması kolay

## 13.2 Sınırlılıklar

### ⚠ Sabit Thresholdlar

- Konuya göre değişebilir (politika vs spor)

### ⚠ Semantic Gaps

- Farklı dilde yazılan haberler yanlış skorlanabilir

### ⚠ Zamansal İssüer

- Breaking news'te doğru çalışmayabilir

### ⚠ Kaynak Güvenilirliğinin Statikliği

- Zamanla değişebilir (demoted/promoted sources)

## 13.3 Gelecek İyileştirmeler (v2)

1. Dinamik kaynak credibility (gerçek-zamanlı güncelleme)
2. Coğrafi çeşitlilik faktörü
3. Semantik hata oranı
4. Makine öğrenmesi ile threshold optimizasyonu
5. A/B testing ile validation

---

## 14. REFERANSLAR

1. **Cosine Similarity:** Manning, Raghavan, Schütze (2008). "Introduction to Information Retrieval"
  2. **News Credibility:** Castillo et al. (2011). "Credibility Assessment of Textual Claims"
  3. **Fake News Detection:** Zhou et al. (2020). "A Survey on Fake News Detection"
  4. **Sentence Embeddings:** Reimers & Gupta (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks"
-