T.C.

DÜZCE ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

TWITTER PROFİL ANALİZİ

HAMZA TAŞ – 151001020

ABDULLAH CANGUL – 151001042

MUHAMMED ERDEM SİYAM – 151001024

LİSANS BİTİRME TEZİ

DANIŞMAN

DR.ÖĞR. ÜYESİ ABDULLAH TALHA KABAKUŞ

DÜZCE, 2019

BEYAN

Bu tez çalışmasının kendi çalışmamız olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımızın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimizi, bu tez çalışmasıyla elde edilen bütün bilgi ve yorumlara kaynak gösterdiğimizi ve bu kaynakları da kaynaklar listesine aldığımızı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımızın olmadığını beyan ederiz.

27 Mayıs 2019

Hamza TAŞ

Abdullah CANGUL

Muhammed Erdem SİYAM

TEŞEKKÜR

Lisans öğrenimimizde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocamız Dr. Öğr. Üyesi Abdullah Talha KABAKUŞ'a en içten dileklerimizle teşekkür ederiz.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili ailemize ve çalışma arkadaşlarımıza sonsuz teşekkürlerimizi sunarız.

27 Mayıs 2019

Hamza TAŞ

Abdullah CANGUL

Muhammed Erdem SİYAM

İçindekiler

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	**Açıklamalar**
VT	Virus Total
TA	Twitter API
SVM	Support Vector Machine
TT	Trending Topic
DM	Direct Message
ABD	Amerika Birleşik Devleri
API	Application Programming Interface

ÖZET

TWITTER PROFİL ANALİZİ

Hamza TAŞ, Abdullah CANGUL, Muhammed Erdem SİYAM

Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Düzce Üniversitesi, Düzce, TÜRKİYE

E-Posta Adresleri : hamza.tas@yahoo.com , erdemsiyam@gmail.com , abdullahcangul@gmail.com

Düzce Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Lisans Bitirme Tezi

Danışman: Dr. Öğr. Üyesi Abdullah Talha KABAKUŞ

Günümüzde çok fazla sayıda sosyal ağ sitesi bulunmaktadır. Sosyal ağ sitelerine örnek vermek gerekirse Facebook, Twitter, İnstagram, Reddit vb. sitelerdir. Sosyal ağlarda paylaşılan içerikler toplumsal duyarlılığında en üst noktaya çıkarmıştır. Toplumun etkileyen büyük olaylar sosyal ağlar aracılığıyla dakikalar içinde milyarlarca kişinin haberdar olmasına veya toplumsal bir hareketin ortaya çıkmasına neden olabilir. Bu nedenle sosyal ağlar toplumu yönlendirir hale gelmiştir. Bu durum sosyal ağları ve burada paylaşılan verileri çok önemli kılmaktadır. Bu veriler bireyler ve toplum hakkında büyük izler taşımaktadır. Verilerin önemi aynı zamanda bu veriler üzerinde yapılacak analizleri de önemli kılmaktadır. Durum böyleyken bu platformlar kötü niyetli içerik ve link yayması için popüler hale gelmiştir ve bu içerik üreticilerin tespiti önem kazanmıştır.

Anahtar Kelimeler: Twitter Profil Analizi, Twitter Spam Tespiti, Random Forest Algoritması, Sınıflandırma, Makine Öğrenmesi, Twitter API, Weka

ABSTRACT

TWITTER PROFİLE ANALYSIS

Hamza TAŞ, Abdullah CANGUL, Muhammed Erdem SİYAM

Department of Computer Engineering, Faculty of Engineering, Düzce University, Düzce, TURKEY

E-:Mail Addresses: hamza.tas@yahoo.com , erdemsiyam@gmail.com , abdullahcangul@gmail.com

Düzce University

Faculty of Engineering

Department of Computer Engineering

Undergraduate Graduation Thesis

Supervisor: Asst. Prof. Abdullah Talha KABAKUŞ

Today there are many social networking sites such as Facebook, Twitter, Instagram, Reddit and so on. The content shared on the social networks has increased to the highest level in social awareness. Large events that affect the community can cause millions of people to be aware or social movement in minutes through social networks. For this reason, social networks have become a factor to direct the society. This situation makes the social networks and the data shared here very important. These data take place great marks about individuals and society. The importance of the data also makes the analysis on these data important. As such, these platforms have become popular for malicious content and link propagation, and the detection of these content producers has gained importance.

Keywords: Twitter Profile Analysis, Twitter Spam Detection, Random Forest Algorithm, Classification, Machine Learning, Twitter API, Weka

1. GİRİŞ

1.1. Çalışmanın Amacı

Sosyal ağlarda paylaşılan içerikler toplumsal duyarlılığı en üst noktaya çıkarmıştır. Toplumun etkileyen büyük olaylar sosyal ağlar aracılığıyla dakikalar içinde milyarlarca kişinin haberdar olmasına veya toplumsal bir hareketin ortaya çıkmasına neden olabilir. Bu nedenle sosyal ağlar toplumu yönlendirir hale gelmiştir. Bu durum sosyal ağları ve burada paylaşılan verileri çok önemli kılmaktadır. Bu veriler bireyler ve toplum hakkında büyük izler taşımaktadır. Verilerin önemi aynı zamanda bu veriler üzerinde yapılacak analizleri de önemli kılmaktadır. Durum böyleyken bu platformlar kötü niyetli içerik ve link yayması için popüler hale gelmiştir ve bu içerik üreticilerin tespiti önem kazanmıştır. Bu çalışmanın temel amacı da spam olarak adlandıran bu içerik üreticilerinin tespitini gerçekleştirmektir.

1.2. Sosyal Ağlar Nedir?

Sosyal ağlar; iletişim kurabilen nesnelerin duygu düşünce ve hislerini sözlü veya yazılı bir şekilde belli bir sayıda kişi veya herkesin ulaşabileceği şekilde paylaşılabilen sanal ortamdır. Aslına bakıldığı zaman Sosyal ağları anlamak için insanoğlunun var oluşuna kadar gitmemiz gerekir. Tarih boyunca insanlar birbirleri ile iletişimde bulunmuşlardır. İnsanların bir birleri ile olan etkileşimleri ile zaman içerisinde insanlar birleşmiş ve bir bütün güç oluşturmuştur ve büyük kavimler ortaya çıkmıştır. İnsanlar zaman geçtikçe farklı iletişim araçları ile birbirleri ile iletişimde bulunmuşlardır ve internetin bulunmasıyla insanlar arasındaki iletişim daha maliyetsiz hale gelmiştir. Bunun sonucunda uzak iki nokta arasında iletişim sağlanmış ve bilimsel gelişmeler hız kazanmıştır. Bilim insanları üniversitelerde oluşan bilgi birikimini paylaşmak ve hız kazandırmak için sosyal ağ denebilecek bir sistemler geliştirmişlerdir. Bu sistemler sosyal ağların temelini oluşturmaktadır. Elektronik cihazlardaki gelişmelerde zamanla sosyal ağların oluşmasına desteklemiştir. Günümüzde çok fazla sayıda sosyal ağ sitesi bulunmaktadır. Sosyal ağ sitelerine örnek vermek gerekirse Facebook, Twitter, Instagram, Reddit vb. sitelerdir.

1.3. Twitter Nedir?

2006 yılında yayına başlamış bir sosyal medya sitesidir. Twitter'ın kelime anlamı İngilizce dilinde "Cıvıldamak" anlamına gelir ve bu siteyi kullanan kullanıcıların birbirleri ile konuşmalarını kuşların cıvıldamalarını anımsatmasından dolayı bu kelime siteye anlamını vermiştir. Twitter; İnsanların ne hissettikleri, insanların bulunduğu konumları ve gezdikleri yerler hakkında bilgiler paylaşabildiği, insanların ne gibi yayınlar okuyup, ne yazdıkları vs. gibi bilgileri çevrimiçi olarak paylaşabildikleri bir web ve mobil uygulamasıdır. İnsanlar Twitter üzerinden ne yaptıklarını paylaşabildikleri gibi merak ettikleri kişilerin durumunu da takip etme fırsatı bulurlar. Bu da Twitter gibi sosyal ağların kullanımını insanlar arasında yaygınlaştırıcı rol oynar. Twitter'ın kullanımın kolay olması ve diğer sosyal ağlara nazaran paylaşım tarzı bakımından kısıtlamalarının olması onu diğer sosyal ağlardan ayıran özelliklerindendir. Twitter ile birlikte Sosyal ağ kullanıcılarının hayatına yeni kavramlar girmiştir. Bu kavramlar aşağıda ayrı başlıklar altında incelenmiştir.

1.3.1.Tweet Nedir?

Twitter sosyal ağı içerisinde yazılan mesajlara verilen gelen isimdir. Tweet'lerin içeriğinde 280 karakteri geçmeyecek şekilde metin, video, fotoğraf gibi içerikler bulunur.

1.3.2.Retweet Nedir?

Bir kullanıcı tarafından paylaşılan Tweet'in, başka bir kullanıcının kendi hesabındaki kişilere aynı şekilde iletmesidir. Retweet, herhangi bir Tweet'in popüler hale getirilmesi birçok kişinin bu Tweet'i takip etmesi için kullanılır.

1.3.3.Mention Nedir?

Mesaj başına @ işareti konularak oluşturulan mesajlara Mention denir. Bir mesaja @ işareti konulduğunda bu mesaj yalnız @ ile belirtilen kişi veya kişilere gider. Hesaplardaki Mention ile gelmiş iletileri görmek için Twitter hesabımızda "Bahsedilenler" alanına bakılması gerekmektedir.

1.3.4.DM (Direct Message) Nedir?

Twitter üzerinden özel mesaj gönderme işlemine DM denir. Gizli bir şekilde mesaj gönderebilmek için her iki tarafın bir birini takip etmesi gerekir.

1.3.5.Hashtag Nedir?

Bir Tweet içerinde başına # işareti konulan sözcükleri ifade eder. Tweet içerisinde kullanıcının özellikle ilgilendiği konuları ana başlık ile ifade edilebilecek şekilde kullanabildiği bir terimdir. Hızlı bilgi akışı ve iletilen mesajın doğru kitleye ulaşması için hedef niteliği taşır. Tweet üzerinde bir etiket niteliğindedir. Hashtag içerisinde bulunan kelimeler arasında boşluk bırakılmaz.

1.3.6.TT (Trending Topic) Nedir?

Belli bir konum veya konumlarda en çok hangi konuların konuşulduğunu gösteren, bir nevi popular konu listesidir. Bu alan, atılan Tweet sayısına göre anlık olarak değişiklik göstermektedir.

2. GENEL KISIMLAR

Bu başlık altında Twitter spam tespiti üzerine bugüne kadar yapılmış bazı çalışmalar verilmiştir.

2.1. Twitter'da Spam Faaliyetlerini Derin Öğrenme Tekniğine Göre Tespiti

Deakin Üniversitesi (Avustralya) ile King Saud Üniversitesi'nin (Suudi Arabistan) ortak çalışmasıdır. 2016 yılında gerçekleştirilmiştir. 600 milyondan fazla Tweet içeren 10-day veri seti üzerinde çalışılmıştır. Ham haldeki Tweet'leri spam ve spam olmayan olarak ayırmak için Tweet içeriğindeki web bağlantıları (URL) Trend Micro's Web ile taranmıştır. 6.5 milyondan fazla Tweet spam olarak tespit edilmiştir. Testleri sonucunda %87 başarı elde etmişlerdir [1].

2.2. Spam İçerik Gönderen Hesapların Tespiti

Edge Hill Üniversitesi(İngiltere) Bilgisayar Bilimleri Bölümü çalışmasıdır. 2018 Yılında gerçekleştirilmiştir. SPDautomated ve Honeypot veri setleri birleştirilmiş 10 kat çaprazlama yapılarak Random Forest algoritmasında %94.7 gibi başarılı bir sonuç elde etmişlerdir. Bu veri seti 10.531 tane spam Tweet olmak üzere toplamda 61.845 Tweet bulundurmaktadır. Bir çok algoritma ile test sonucunda maksimum başarı Maksimum Entropi Sınıflandırıcısı ile %98.7'dir [2].

2.3. Farklı Yaklaşımlar İle Sosyal Medyada Spammer Tespti

Dalian Üniversitesi(Çin), Bilgisayar Bilimi ve Teknoloji Bölümü çalışmasıdır. 2016 yılında gerçekleştirilmiştir. Honeypot ve Kwak veri setleri kullanılmıştır. Toplamda 4.414'ü spam kabul edilen 10.080 kullanıcı elde edilmiştir. Tweet içeriği, kullanılan url ve hashtag'lere göre spam tespiti yapılmıştır. Support Vector Machine Algoritması ile test sonrası %84 lük başarı elde etmişlerdir [3].

2.4. Duygu Tabanlı Twitter Spam Tespiti

Islamia University Of Bahawalpur (Pakistan) ve Bahauddin Zakaria Üniversitesi'nin (Pakistan) ortak çalışmasıdır. 2016 yılında gerçekleştirilmiştir. Çalışmayı gerçekleştirenler kendi veri setlerini oluşturmuşlardır; 29 TT için toplamda 29.000 adet Tweet toplanmıştır. Spam tespiti için duygu analiz yöntemini kullanmışlardır. Toplamda Naive Bayes, Random Forest, J48 ve SVM olmak üzere 4 sınıflandırma algoritması kullanılmıştır. Aynı veri seti üzerinde test uygulanmıştır. 10 kat çaprazlama ile en iyi sonuç %92.34 oran ile J48 Sınıflandırıcısı olmuştur [4].

2.5. Veri Akışı Kümeleme İle Twitter Spammer Tespiti

Houghton Collage (ABD) ve Pennsylvannia State Üniversitesi'nin (ABD) ortak çalışmasıdır. 2013 yılında gerçekleştirilmiştir. Çalışmayı yürüten kişiler kendi veri setlerini oluşturmuşlardır. 3.239 kullanıcı toplanmış olup 208 kadarı spam hesaptır. Bu spam hesaplar daha önce yapılmış bir çalışmadaki veri setinden alınmıştır.[1] Naif Bayes Algoritması ile testlerde %91.7 başarı elde edilmiştir [5].

[1] A. Wang, Don't Follow Me: Spam Detection in Twitter, in: Int'l Conference on Security and Cryptography

2.6. Twitter'da Spammer Tespiti

Federal de Minas Gerais Üniversitesi'nin (Brezilya) çalışmasıdır. 2009 yılında gerçekleştirilmiştir. TT'lerden 1.7 milyar Tweet toplamışlardır. SVM Algoritması ile 5 kat çaprazlama yaparak oluşturdukları veri setindeki belirlenen spamları %70 başarı ile bulmuşlardır [6].

2.7. Geleneksel Sınıflandırma Algoritmaları İle Twitter'da Spam Tespiti

Lehigh Üniversitesi (ABD) çalışmasıdır. 2011 yılında gerçekleştirilmiştir. Trend Topic'lerden kendi oluşturdukları 1.000 kullanıcılı küçük veri seti üzerinde Random Forest, SVM, K-Nearest Neighbor, and Naïve Bayes gibi bir dizi makine öğrenmesi algoritmaları kullanılarak spam doğrulamasında Random Forest Algoritması ile %95 üzerinde başarı elde etmişlerdir. Ayrıca SVM ile de %93 başarı elde etmişlerdir [7].

2.8. Makine Öğrenmesi İle Çevrimiçi Sosyal Ağ Sitelerinde Spam Tespiti

Pennsylvania State Üniversitesi'nden Alex Hai Wang tarafından 2014 yılında gerçekleştirilen çalışmadır. 3 hafta boyunca, 25.847 Twitter hesabı, yaklaşık 500.000 Tweet ve yaklaşık 49 milyon takipçi toplamıştır. %1 oranında gerçek spammer tespiti yapılmıştır Fakat Twitter'ın bilinen gerçek spammer/non spammer oranına denk getirmek için %3 'e kadar başka veri setlerinden spammer eklenmiştir. Bu oran %3'tür.[2] Naive Bayes ile %91.7 oranında başarı elde etmiştir [8].

2.9. Twitter'da Spammer Toplulukların Tespiti

National Institute of Technology Karnataka (Hindistan) üniversitesinin çalışmasıdır. 2017 yılında gerçekleştirilmiştir. 23,869 bin Tweet içeren Honeypot isimli veri seti üzerinde çalışılmıştır. incelenen hesabın spam hesaplarının takip edilip edilmediği, spammer hesapların paylaştıkları Tweet'lerdeki içerik benzerliği, hesabın yaşı, davranışsal nitelikleri incelenmiştir.

[2]Analytics, I.P.: Twitter study. http://www.pearanalytics.com/wpcontent/uploads/2009/08/Twitter-Study-August-2009.pdf)

bu inceleme doğrultusunda spam analizi yapılmış ve graf yapısında gösterilmiştir. Testleri sonucunda %89 başarı elde etmişlerdir [9].

2.10. Sahte Takip Pazarlarında Mesafeye Dayalı Müşteri Tespiti

Seoul National University (Kore) üniversitesinin çalışmasıdır. 2019 yılında gerçekleştirilmiştir. Çalışmayı gerçekleştirenler kendi veri setlerini oluşturmuşlardır. Toplam 34.826 kullanıcı 24,552 sahte kullanıcıdan oluşmaktadır. Marketlerin belirli bir ücret karşılığında takipçi sayılarını arttırmalarını tespit etmek için coğrafi Özelliklerden faydalanarak marketten uzaklaştıkça gerçek kullanıcıların azaldığı ve sahte kullanıcıların artığı görülmüştür. Tespit için sahte hesaplar alınmıştır DetectVC ve CatchSync algoritmaları kullanılmıştır. Analiz sonucunda %98.1 başarı elde etmişlerdir [10].

2.11. Basamaklı Sosyal Bilgilere Dayalı Spam Tespiti

Seoul National University (Kore) ve Soongsil University (Kore) üniversitesi ortak çalışmasıdır. MPI-SWS tarafından sağlanan büyük ölçekli Twitter-Takip link veri seti kullanılmıştır. Bu veri Eylül 2009'da toplanmış, 1.963.263.821 yönlendirmeli sosyal link içermektedir ve ilgili kullanıcı sayısı 54.981.152 kişiden oluşmaktadır.. Ayrıca 41.362 spam göndericiyi içeren Follow Spammer veri seti kullanılmıştır. Collusionrank, yüksek hesaplama maliyetli TSP-filtreleme ve SS-filtreleme yöntemlerini, düşük bir hesaplama maliyeti olduğu ortaya çıkmıştır. Analiz sonucunda %94 başarı elde etmişlerdir [11].

2.12. Twitter Spam'la İlgili Aldatıcı Bilgilerin Araştırılması

Deakin University (Avusturya) ve King Saud University (Suudi Aarabistan) üniversitesi ortak çalışmasıdır. 2017 yılında gerçekleştirilmiştir. Veri setinde 600 milyon Tweet 33 milyon spam Tweet olmak üzere 633 milyon Tweet bulunmaktadır. Kara liste yönteminin doğruluğu incelenmiştir [12].

2.13. Twitter'daki Kısa URL'lere Dayalı Kötü Amaçlı Hesaplar Tespiti

National Institute of Technology Rourkela (Hindistan) üniversitesi çalışmasıdır. 2015 yılında gerçekleştirilmiştir. 4.820 kullanıcı ve 380 şüpheli kullanıcı incelenmiştir. Dataset'in içerisindeki verilerin %80 öğrenme %20 test olarak kullanılmıştır. Decision Tree, Nave Bayes, Random Forest algoritmaları kullanılarak kullanıcıların kısa linklerden yola çıkarak anormal davranışlarını kontrol edilmiş. Bu değerlendirmenin sonucunda kullanıcılara Eşik değeri verilmiştir. Bu eşik değeri 0,5 altında olanlar sahte kullanıcı kabul edilmiştir kısa linkler Google API yardımı ile analiz edilmiştir. .Bu analiz sonucunda spam tespiti yapılmıştır. Analiz sonucunda %94 başarı elde etmişlerdir [13].

2.14. Spam Kampanyaların Tespiti

The College of William and Mary (ABD) ve Bell Laboratories (ABD), kurumlarının ortak çalışmalarıdır. 2012 yılında gerçekleştirilmiştir. 22 milyon kullanıcı ve 50 milyon Tweet içeren kendi veri setlerini oluşturmuşlardır. Bireysel spam hesaplarının tespiti ve bunların kapatılmasından daha çok kampanya spamlarına engellemeye yönelik çalışmışlardır. Url bazlı araştırma yapmışlardır. Eğer URL kara listeye alınmışsa, kampanya spam olarak sınıflandırmışlardır. %82,3 ile %94,5 arasında başarı değeri elde etmişlerdir [14].

2.15. Dilin İstatiksel Analizi Kullanılarak TT 'ler İçerisinde Zararlı İçerik Tespiti

Nacional de Educación a Distancia Üniversitesi (İspanya) çalışmasıdır. 2013 yılında gerçekleşmiştir. Kendi veri setlerini, 1.5 hafta boyunca 34.000 TT ile 20.000.000 Tweet çekerek oluşturmuşlardır. İçlerindeki 6.000.000 Tweet Url içermektedir. Bu Tweetler için Google Safe Browsing Capture-HPC SURBL Project Honey Pot gibi araçlar kullanılarak, 168.000 Tweet'e Url bazlı spam teşhisi koyulmuştur. Tüm veri setinin %8'ine denk gelmektedir. %89.3 başarıyla spam tespiti yaparken %93.7 başarısıyla da spam olmayan hesap tespiti yapmışlardır [15].

3. KULLANILAN ARAÇ VE YÖNTEM

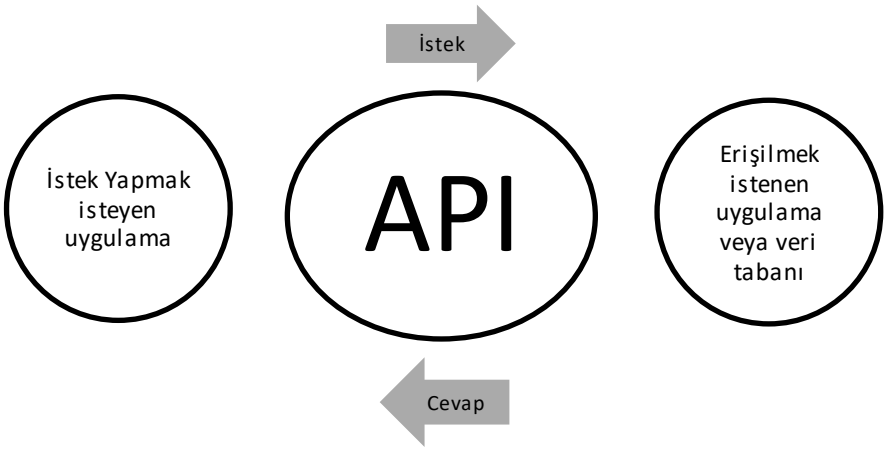
Bu başlık altında gerçekleştirilen çalışmada kullanılan teknolojiler ve bu teknolojilere ait teknik bilgilerin yanı sıra projeye ait teknik verilere değinilmiştir.

3.1. Kullanılan API'ler

Bu başlık altında API 'in teknik tanımı ve yapılan çalışma kapsamında kullanılan API'lere ait detay bilgiler verilmiştir.

3.1.1. API Nedir?

Twitter'ın sunmuş olduğu API'yi incelemeden önce bu başlık altında genel olarak API'lerin özellikleri üzerinde durulacaktır. API (Application Programming Interface) kelime anlamıyla Türkçeye çevrildiğinde uygulama programlama ara yüzü anlamını vermektedir. API, üzerinde çalıştığımız uygulama ile farklı bir uygulamanın haberleşmesini sağlayan bir yapıdır. Farklı bir biçimde açıklamak gerekirse bir API, bir uygulamanın tüm özelliklerini başka bir uygulamanın kullanabilmesini için oluşturulan bir servistir. Şekil 1'de API'lerin çalışması görsel olarak sunulmuştur.



Şekil 1: API Çalışma Mekanizması

3.1.2 Twitter API

API, Geliştiricilerin bir tür teknoloji ile etkileşim kurması için oluşturulmuş komut kümesidir. Twitter'da Twitter harici geliştiricilerin Twitter verilerine dayanan teknolojiyi geliştirmesine izin veren açık bir API oluşturmuştur. Twitter API diğer uygulamaların API'lerinde olduğu gibi bir uygulamanın özelliklerini farklı bir uygulamamızda kullanabilmesi

için oluşturulmuş bir servistir. Twitter API ile Twitter üzerinden yapabildiğimiz birçok işlemi kendi uygulamamıza kazandırabiliriz.

3.1.3.1.Twitter API Hakkında

Farklı kullanımlar için Twitter'ın farklı API'leri Mevcuttur. Bu API'ler barındırdıkları özelliklere göre farklılık göstermektedir.

- **Standart API (Ücretsiz) :** REST API ve Streaming API'lerinden oluşur. Bir entegrasyonu test etmek, bir konsepti doğrulamak veya ücretli ve kurumsal ürünlerle neler yapabileceğinizi tamamlayan çözümler oluşturmak için kullanılır. Bu API ile Twitter'dan yüksek hacimli veri elde edilememektedir. Bu API'nın tercih edilmesi durumunda bir izin mekanizması kurmak gerekir. Çünkü Twitter'ın bu API'si yapılan isteklere sınır getirmektedir. Bu izin mekanizması ilerleyen bölümlerde detaylı bir şekilde anlatılmıştır.

- **The enterprise API (Ücretli) :** Twitter verilerine bağımlı olanlara en yüksek düzeyde erişim ve güvenilirlik sunar. Twitter bu API ile teknik destek sağlar.

3.1.3.2. Twitter API'leri

Twitter Search API : Twitter'ın verilerini bir arama veya kullanıcı adıyla sorgulamayı içeren içerir. Arama API'si sayesinde, kullanıcılar bir çeşit "arama" kriteriyle eşleşen Tweet'leri ister. Kriterler anahtar kelimeler, kullanıcı adları, konumlar, adlandırılmış yerler, vb. olabilir. Twitter Arama API'sini düşünmenin iyi bir yolu, bir kullanıcının doğrudan Twitter'da bir arama yapmasını (search.twitter.com'a gitmek ve içeri girmek) gibi düşünülebilir.

Twitter Streaming API: Gerçek zamanlı olarak gerçekleşen tweet'ler ile ilişkili bir API'dir. API kullanıcısı anahtar kelimeler (Anahtar Kelimeler, Kullanıcı Adları, Lokasyonlar, İsimlendirmeler, Yer İsimler vb.) belirler ve Tweet'ler de bu anahtar kelimeler geçtikçe Twitter streaming API bu tweet'leri bize ulaştırır.

Twitter Firehose API: Twitter Firehose, Twitter'ın Streaming API'sine çok benzemektedir. Çünkü bu API'de verileri gerçek zamanlı olarak son kullanıcılara aktarmaktadır. Ancak, Twitter Firehose, kriterlere uyan Tweet'lerin %100'ünün getirileceğini garanti eder. Twitter

Firehose, Twitter Streaming API kullanımı ücretsizdir, ancak size sınırlı sonuçlar verir. Firehose'a erişim Twitter tarafından dayatılan çok sayıda kullanım kısıtlamasını kaldırır ancak tüm Tweet'lere erişim için oldukça maliyetlidir.[3]

Belirtildiği gibi Twitter'ın sunmuş olduğu birçok API bulunmaktadır. Bu API'lerin temel özelliklerine yukarıda değinilmiştir. Bu proje kapsamında kullanılan API ve API'lerin kullanılan metotları, proje dâhilinde sunucu taraflı yapılan çalışmaların anlatıldığı başlık altında ele alınacaktır.

3.1.3. Virus Total API

VT (Virus Total) API bir web ara yüzü gerektirmeksizin dosyaları veya URL'leri taramamıza imkân sağlayan bir Rest servisidir. VT API'yi kullanabilmek için VT Topluluğuna üye olunması gerekir. VT Topluluğuna üye olunduğunda kişisel API anahtarına sahip olursunuz ve VT API'sını kullanabilmek için gerekli tek koşul bu API anahtarına sahip olmaktır. VT topluluğuna kayıtlı kullanıcılar API'nin birçok özelliğine ücretsiz erişebilse de bazı özellikler Ücretli üyelik ile sınırlandırılmıştır. Ücretsiz API de dakika başına 4 sorgulama yapılabilir ve bu API'nin kullanımı durumda gerçekleştirilen iş hiçbir ticari ürün veya serviste kullanılamaz. Ücretli API de ise limit kavramı olmadığı gibi daha fazla kötü içerikli veri ve açıklama döndürür. Ücretli API geri dönüş garantisi verir. VT API de cevaplar JSON formatındadır. API de tanımlı 4 tür dönüş kodu mevcuttur. Bu kodlar Çizelge 1'de açıklanmıştır.[4]

Çizelge 1: VT API Geri dönüş kodları

Kod	Açıklama
200 OK	Eğer Yapılan istek API ye ulaşmış ve doğru bir şekilde işlem gerçekleşmiş ise 200 OK kodu ile dönülür.
204 NO CONTENT	204 kodu ile bir dönüş alınmış ise istek limit hakkı aşılmıştır. Limit hakkının tekrar yenilenmesi beklenmelidir.
400 BAD REQUEST	400 kodu ile bir olarak dönüş alınmış ise yapılan istek doğru bir formatta değildir.
403 FORBIDDEN	403 kodu ile bir dönüş alınmış ise isteği yapmak için yeterli ayrıcalıklara veya yetkilere sahip değilsinizdir.

[3] https://developer.twitter.com/en/docs
[4] https://developers.virustotal.com

Yapılan çalışmada VT API'nin ücretsiz versiyonu kullanılmıştır. Bu nedenle limitler ve geri dönüş tiplerine göre bir kontrol mekanizması oluşturulmuştur. VT API'nin kullanım amacı ve çalışma mekanizması ilerleyen kısımlarda detaylı olarak anlatılacaktır.

3.1.4. Weka API

Weka veri madenciliği için makine öğrenmesi algoritmalarını, veri hazırlama, sınıflandırma, regresyon ve kümeleme işlemlerine yönelik analiz ve görselleştirme araçları içeren açık kaynak kodlu bir yazılımdır. [5]

Yapılan çalışmada Weka API kullanılarak birçok makine öğrenmesi tekniği üzerini çalışılmıştır. Kullanılan teknikler ilerleyen bölümlerde daha detaylı olarak anlatılacaktır.

3.2. Veri Yönetimi İle İlgili Yapılan Çalışmalar ve Kullanılan Teknolojiler

3.2.1. Veri Modeli

Twitter API üzerine yapılan incelemelerin ve gerçekleştirilen analizler ile yapılan araştırma ve çalışma sonucunda tutulacak olan veriler ve yapısı belirlenmiştir. Tez çalışması kapsamında aşağıda çizelgeler halinde verilen veri yapısını içeren bir model oluşturulmuştur.

MTweet tablosu bir arama sonucu veya analiz sonucunda veri tabanında kayıt altına alınan Tweet'lere ait bilgileri tutmak için oluşturulmuştur. Diğer bir değişle Twitter tarafından gelen Tweet'lerde bizim tutacağımız alanların kayıt alındığı yerdir.

Çizelge 2: MTweet Tablosu

MTweet Tablosu		
Alan (Field)	**Açıklama**	**Veri Tipi**
id	Tweet id aynı zamanda Document id	String
text	Tweet içeriği	String
userScreenName	Tweet paylaşan kişi id	String
createdDate	Tweet'in oluşma tarihi	Date
Device	Tweet'in atıldığı cihaz	String
statusType	Tweet'in tipi (tweet,retweet, alıntı , cevap)	Enum
relatedStatusId	Eğer Tweet ,Tetweet , Alıntı veya Cevap ise ilişkili Tweet'in Id'sidir.	String
favoriteCount	Tweet'in beğeni sayısı	Integer
retweetCount	Tweet'in Retweet sayısı	Integer
Location	Tweet'in atıldığı konum	String
Language	Tweet'in dili	String
Hashtags	Tweet'in içeriğinde bulunan Hashtag'ler	List<String>
aramaId	Hangi arama ile ilişkili	String

[5] https://www.cs.waikato.ac.nz/ml/weka/

MUser tablosu bir arama sonucu veya analiz sonucunda veri tabanda kayıt altında alınacak olan kullanıcıların bilgilerinin tutulduğu alandır. Diğer bir değişle Twitter tarafından gelen kullanıcı bilgilerine karşı bizim tutacağımız alanların kayıt altın alındığı yerdir.

Çizelge 3: MUser Tablosu

MUser Tablosu		
Alan (Field)	**Açıklama**	**Veri Tipi**
id	Kullanıcı id aynı zamanda Document id	String
name	Kullanıcı adı ve soyadı	String
web	Kullanıcı web adresi	String
screenName	Kullanıcı adı	String
description	Kullanıcı profil tanımı	String
location	Kullanıcı konumu	String
profileImageUrl	Kullanıcı profil resmi adresi	String
followerCount	Kullanıcı takipçi sayısı	Integer
friendsCount	Kullanıcının takip ettiği kişi sayısı	Integer
createdTime	Kullanıcı profil oluşturma tarihi	Date
favouritesCount	Kullanıcı beğendiği tweet sayısı	Integer
language	Kullanıcı dili	String
statusCount	Kullanıcı tweet sayısı	Integer
isVerified	Kullanıcı hesabı onaylı bir hesap mı	Boolean
aramaId	Hangi arama ile ilişkili	String

MArama tablosu sistem üzerinden yapılan aramaların kayıtlarının tutulduğu alandır. Bu tablo içerisinde yalnızca yapılan bir aramaya ait detay bilgiler tutulur.

Çizelge 4: MArama Tablosu

MSearch Tablosu		
Alan (Field)	**Açıklama**	**Veri Tipi**
İd	Arama id ve Document id	String
content	Aranan kelime	String
startTime	Arama başlama tarihi	Date
endTime	Arama bitiş tarihi	Date

MTrainingTweet tablosu gerçekleştirilen çalışmanda eğitim amaçlı veri tabanlarının toplanan Tweet'lerin tutulduğu tablodur. Burada tutulan alanlar bu çalışmadaki önceki spam tespit çalışmalarında dikkat edilen nitelikler göz önündü bulundurularak oluşturulmuştur.

Çizelge 5: MTrainingTweet Tablosu

MTrainingTweet Tablosu		
Alan (Field)	**Açıklama**	**Veri Tipi**
id	Tweet numarası	Integer
accountAge	Kullanıcı hesap yaşı	Integer
noFollower	Kullanıcının takip ettiği kişi sayısı	Integer
noFollowing	Kullanıcının takipçi sayısı	Integer
noUserFavourites	Kullanıcının favori tweet sayısı	Integer
noLists	Kullanıcının üye olduğu liste sayısı	Integer
noTweets	Kullanıcının tweet sayısı	Integer
noRetweets	Kullanıcının retweet sayısı	Integer
noHashtag	Tweet içerisindeki hashtag sayısı	Integer
noUserMention	Tweet içerisindeki mention kullanıcı sayısı	Integer
noUrls	Tweet içerisindeki url sayısı	Integer
noChar	Tweet içerisindeki karakter sayısı	Integer
noDigits	Tweet içerisindeki numerik karakter sayısı	Integer
class	spammer , non-spammer durumu	String
dbFlag	Hangi veri tabanından alındığına dair bilgi	String

MTestTweet tablosu TA kullanılarak çekilen gerçek Tweet'lerin istediğimiz formatta tutulduğu tablodur. Twitter'dan çekilen Tweet'ler aynı anda MTweet ve MUser Tablosuna kayıt edilir.

Çizelge 6: MTestTweet Tablosu

MTestTweet Tablosu		
Alan (Field)	**Açıklama**	**Veri Tipi**
id	Tweet numarısı	Integer
accountAge	Kullanıcı hesap yaşı	Integer
noFollower	Kullanıcının takip ettiği kişi sayısı	Integer
noFollowing	Kullanıcının takipçi sayısı	Integer
noUserFavourites	Kullanıcının favori tweet sayısı	Integer
noLists	Kullanıcının üye olduğu liste sayısı	Integer
noTweets	Kullanıcının tweet sayısı	Integer
noRetweets	Kullanıcının retweet sayısı	Integer
noHashtag	Tweet içerisindeki hashtag sayısı	Integer
noUserMention	Tweet içerisindeki mention kullanıcı sayısı	Integer
noUrls	Tweet içerisindeki url sayısı	Integer
noChar	Tweet içerisindeki karakter sayısı	Integer
noDigits	Tweet içerisindeki numerik karakter sayısı	Integer
class	spammer , non-spammer durumu	String
dbFlag	Hangi veri tabanından alındığına dair bilgi	String

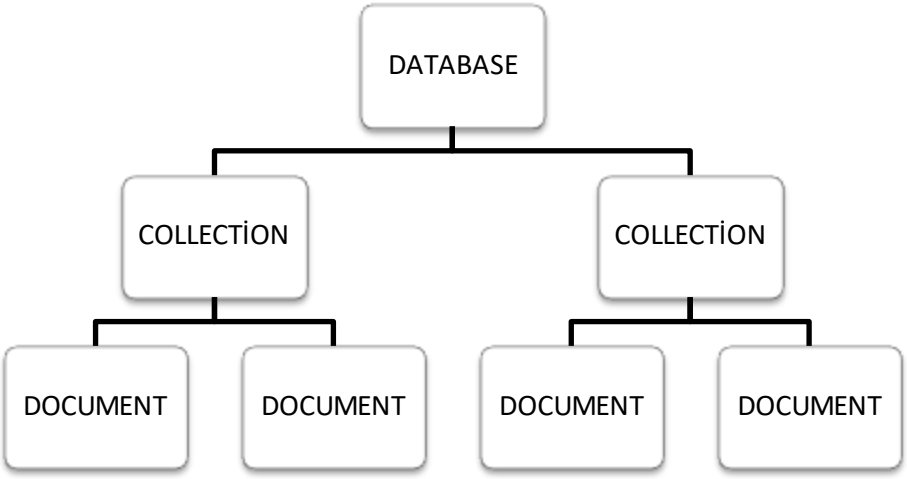
3.2.2. Veri Yönetimi İle İlgili Kullanılan Teknolojiler

3.2.2.1 MongoDB

Tutulacak olan verilerin ve bu verilere ait özellikler yukarıda her bir tablo için ayrı ayrı verilmiştir. Twitter üzerinden gelen verilerin analizinden önce bu verilerin kayıt altına alınması için MongoDB veri tabanı teknolojisinden yararlanmıştır.

NoSQL veri tabanı sistemleri ilişkisel veri tabanı istemlerine göre yüksek erişilebilirlik imkânı sunarlar. Okuma ve yazma performansları göreceli olarak ilişkisel veri tabanlarından daha hızlıdır. Büyük veriler üzerinde işlem yapabilir ve esnek yapılarından dolayı programlama ve bakım anlamında kolaylık sağlarlar.[6] NoSQL veri tabanlarının sağladığı bu avantajlardan dolayı ve tutulacak olan kayıtların alanları değişiklik gösterebileceğinden dolayı NoSQL bir yapıya sahip olan MongoDB kullanımı tercih edilmiştir. MongoDB, Twitter üzerinden tarafımıza gelen veriler üzerinde ek işlemler gerektirmeden bu verileri kayıt altına alabilmenin yanı sıra veri tabanında kayıt altına aldığımız verileri de proje dâhilinde esnek bir şekilde kullanılmasını sağlamıştır. MongoDB'nin veriler üzerinde sağladığı esnekliğin nedeni belgeye dayılı modeli kullanmasından gelmektedir. Belgeye dayalı modelde ilişkisel modellerde kullanılan terimlerine benzer terimler bulunmaktadır. İlişkisel modellerde "Satır" (row) kavramı yerine belgeye dayalı modellerde "Belge" (Document) kavramı kullanılmaktadır. Bir belge içirişinde dizi ve gömülü belge tutulabilmesi özellikleri ile belgeye dayalı modellerde birçok yönden hiyerarşik özellikteki verileri tek kayıt altında tutulabilmesi sağlanmıştır. Böyle bir durumun sağlanması ile nesneye yönelik programlama ile veri tabanı sistemleri üzerinde çalışma da kolaylaşmıştır. Belgeye dayalı modellerde veriler üzerinde önceden belirlenmiş bir şema bulunmamaktadır. Bu nedenle belge içerisindeki alanlar önceden belirlenmiş bir modele sahip olması gerekmemektedir. Böylece belge içerisindeki alanların bir şemaya sahip olmadan eklenebilmesi veya çıkarılabilmesi sağlanmıştır. MongoDB'nin sağlamış olduğu bu esneklikler sayesinde yazılım geliştirmelerde hızlı iterasyonlar yapılarak yazılım tamamlanma süreci kısalmakta ve yazılım projesi için en uygun veri modeli denenerek seçilebilmektedir.[7]

[6] https://kodcu.com/2014/03/nosql-nedir-avantajlari-ve-dezavantajlari-hakkinda-bilgi/
[7] https://www.mongodb.com/what-is-mongodb



Şekil 2: MongoDB Yapısı

Şekil 2'te MongoDB'nin yapısı temel olarak açıklanmıştır. MongoDB Veri tabanında ilişkisel veri tabanlarında bulunan "Tablo" kavramı yerine "Collection" kavramı bulunmaktadır. Yine aynı şekilde ilişkisel veri tabanlarında bulunan "kayıt" kavramı yerine "Belge" kavramı bulunmaktadır.

3.2.2.2 Java İle MongoDB'ye Bağlanma

Sunucu tarafında genel olarak Java programlama dilinden faydalanılarak işlemler yapılmaktadır. Yapılan araştırmalardan Java programlama dili ile mongoDB'ye bağlanma yolları arasında Mongo Driver ve Spring Data yöntemleri üzerine denemelerde bulunulmuştur. Her iki bağlantı yolu için yapılan araştırmalar ve denemeler sonucunda yapılan çalışma için Spring Data framework'unun daha uygun olduğu belirlenmiştir. Spring Data'nın tercih sebebi aşağıda da ayrı başlıklar altında açıklandığı gibi veriler üzerinde işlem yapabilmede sağladığı kolaylık olmuştur.

MongoDB Driver

MongoDB ile hem senkron hem de asenkron etkileşim sağlayan resmi MongoDB Java Sürücüsüdür. MongoDB Inc. Tarafından Java için oluşturulmuş kütüphane sayesinde mongoDB'ye bağlantı ve ekleme, okuma, güncelleme ve silme İşlemleri gibi birçok fonksiyonu barındırır. Collection üzerine ekleme, okuma, güncelleme ve silme işlemlerinde

esnek bir yapı sağlar. Nesneye dayalı projelerde nesneler üzerine yapılan ekleme, okuma, güncelleme ve silme işlemleri karmaşıklaşabilmektedir.[8]

Spring Data Mongo

Spring Data MongoDB veri tabanı ile entegrasyon sağlar. Spring Data'nın temel hedefi, Kaydını tutacağımız verinin özelliklerini korurken, Bu veri için tutarlı bir Spring tabanlı programlama modeli sağlamaktır. Spring data bir veri erişim teknolojisidir., ilişkisel ve ilişkisel olmayan veri tabanlarını kullanmayı kolaylaştırır. Spring Data ekleme, okume, güncelleme ve silme işlemlerinde de kolaylık sağlamaktadır.

3.3. Geliştirme Ortamı Ve Harici Kütüphaneler

3.3.1. Java Programlama Dili

Proje geliştirme dili olarak Java programlama dili tercih edilmiştir. Java dilinin birçok platformda çalışması, zengin geliştirici desteğinin olması, nesne yönelimli bir programlama dili olması ve Twitter API için yazılmış kütüphane desteklerinin olması tercih sebebi olmuştur. Gerçekleştirilen proje Java Spring Boot MVC Framework üzerine oluşturulmuş ve ilerleyen zamanlarda web üzerinden çalışması için zemin hazırlanmıştır.

3.3.2. Twitter4j Kütüphanesi

Twitter4j, Twitter tarafından resmi olmayan fakat önerdiği, Twitter API alışverişi yapan Java kütüphanesidir. Twitter4j'yi Java uygulamamıza Twitter servisiyle kolayca entegre edilebilmektedir. En az Java 5 versiyonunda çalışmaktadır.

Projede Twitter4j kütüphanesinin 4.0.7 versiyonu kullanılmaktadır. Kompleks ve ilkel şekilde HTTP alışverişi yapmamız yerine bu kütüphane ile gerekli OAuth destekli oturum açımı sonrasında hazır metotlar ile istenilen verilere kolayca ulaşılabilmektedir. OAuth, İnternet kullanıcılarının web sitelere veya mobil uygulamalara basit ve standart bir şekilde uygulamalar ile güvenli bir şekilde yetkilendirme için oluşturulmuş açık bir protokoldür. Bu mekanizma ile kullanıcıların hesapları hakkındaki bilgileri üçüncü taraf uygulamaları veya web siteleriyle paylaşmalarına izin vermek için kullanılır [16]. Twitter4j kütüphanesinde kullanılan metotlar çizelge 7'de verilmiştir.

[8] https://docs.mongodb.com/ecosystem/drivers/

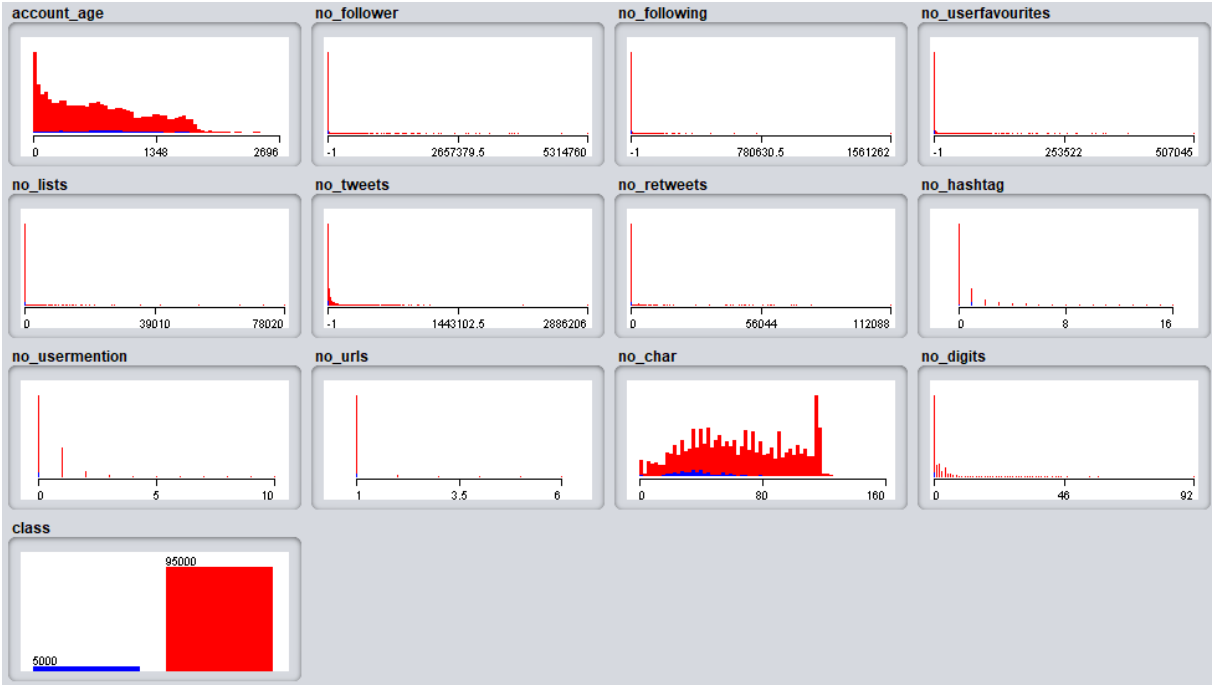
Çizelge 7: Twitter Api'de kullanılan methotlar

Geri Dönüş Tipi	Metot İsmi	Açıklaması
Map<String,RateLimitStatus>	getRateLimitStatus()	Kalan tüm haklar getirilir.
List<Status>	getHomeTimeline()	Oturum açılan kişinin zaman tünelindeki tweetleri getirilir.
List<Status>	getUserTimeline(screenName)	Screen name'i verilen kullanıcının tweetleri getirilir.
Status	showStatus(id)	Id'si verilen tweet getirilir.
ResponseList<Status>	getRetweets(id)	Id'si verilen tweet'in retweet'lerini getirir.
String	getSource(id)	Tweet atılan platformu getirir.
String	getPlace()	Tweet'in konumu varsa getirilir.
ResponseList<User>	searchUsers(query,page)	Arama kelimesine göre kullanıcılar getirilir.
ResponseList<User>	lookupUsers(screenName)	Aratılan kelimeye göre kullanıcıları detaylı bilgileriyle getirir.
List<Status>	(new Query()).getTweets()	Arama nesnesine göre Tweet'leri getirir.
User	showUser(screenName)	Screen Name'i verilen kullanıcı getirilir.
PagableResponseList<User>	getFriendsList(screenName)	Screen Name'i verilen kullanıcının takip ettiği kullanıcıları getirir.
PagableResponseList<User>	getFollowersList(screenName)	Screen Name'i verilen kullanıcıyı takip eden kullanıcıları getirir.
Relationship	showFriendship(sourceUser , TargetUser)	Screen Name'leri verilen kullanıcıların birbirleri aralarındaki ilişkiyi gösterir.

3.4. Spam Tespitinde Eğitim Amaçlı Kullanılan Veri Tabanları

3.4.1. ICC

Veri seti içerisinde 100.000 Tweet bulunmaktadır. Bu Tweet'lerin 95.000 tanesi normal Twitter kullanıcısı ve 5.000 tanesi Spammer olarak adlandırılan kötü içerik üreticisidir. Spam hesaplar spammer ile spam olmayan hesaplar non-spammer olarak etiketlenmiştir. Veri setindeki özniteliklere ait detay bilgiler şekil 3 ' te verilmiştir.



Şekil 3: ICC Veri tabanı nitelik dağılımı

Veri seti daha önce spam tespitinde yapılan çalışmalarda kullanılmıştır.[9] Buradaki öznitelikler ve açıklamaları çizelge 8'de verilmiştir.

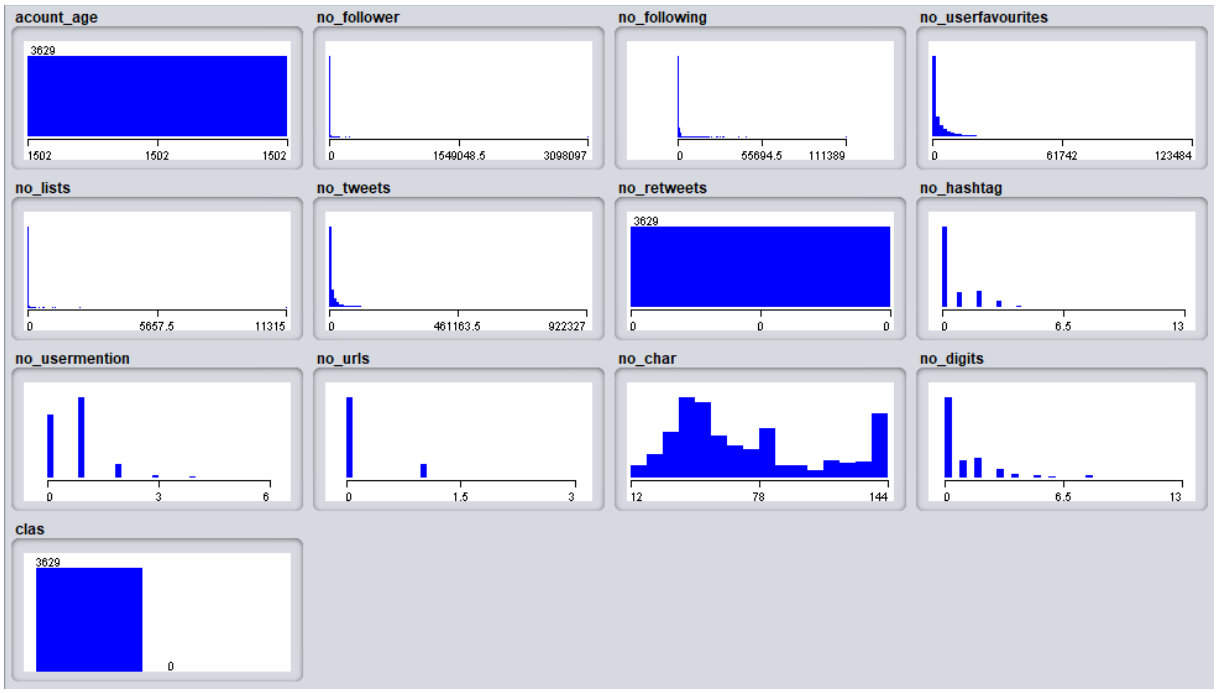
[9] 6 Million Spam Tweets: A Large Ground Truth for Timely Twitter Spam Detection , Chao Chen, Jun Zhang Xiao Chen, Yang Xiang and Wan lei Zhou School of Information Technology Deakin University, Victoria 3125, Australia

Çizelge 8: ICC Dataset Öznitelik Açıklamaları

ICC Öznitelik Tablosu		
Öznitelik	**Açıklama**	**Veri Tipi**
account_age	Kullanıcı hesap yaşı	Integer
no_follower	Kullanıcının takip ettiği kişi sayısı	Integer
no_following	Kullanıcının takipçi sayısı	Integer
no_userfavourites	Kullanıcının favori tweet sayısı	Integer
no_lists	Kullanıcının üye olduğu liste sayısı	Integer
no_tweets	Kullanıcının tweet sayısı	Integer
no_retweets	Kullanıcının retweet sayısı	Integer
no_hashtag	Tweet içerisindeki hashtag sayısı	Integer
no_usermention	Tweet içerisindeki mention kullanıcı sayısı	Integer
no_urls	Tweet içerisindeki url sayısı	Integer
no_char	Tweet içerisindeki karakter sayısı	Integer
no_digits	Tweet içerisindeki numerik karakter sayısı	Integer
class	spammer , non-spammer durumu	String

3.4.2. SuperBowl

Bu veri setinde toplamda 3.629 adet Tweet bulunmakta ve bu Tweet'lerin tamamı spam olarak bilinmektedir. Dataset içerisindeki veriler JSON formatındadır. JSON formatındaki veriler bir önceki tablodaki öznitelikleri alındığında özniteliklerin dağılımları şekil 4' teki gibidir.



Şekil 4: SuperBowl Veri tabanı nitelik dağılımı

ICC ve SuperBowl veri setlerindeki veriler Çizelge 7' deki öznitelikler göz önünde bulundurularak MongoDB veri tabanında MTrainingTweet Collaction'nunda kayıt altına alınmıştır. Sonuç olarak 8.629 spammer ve 95.000 non-spammer olmak üzere 103.629 Tweet verisi elde edilmiştir. Veri seti içerisinde 505 non-spammer ve 130 spammer olmak üzeri Tweet test veri seti olarak işaretlenmiştir. Kullanılan makine öğrenmesi tekniklerini eğitim ve test amaçlı olarak verilen veriler çizelge 10' de gösterilmiştir.

Çizelge 9: ICC Eğitim ve Test Dataseti spammer ve non-spammer dağılımları

Veri Seti	**spammer**	**non-spammer**
Eğitim Seti	8.629	95.000
Test Seti	130	505

3.5. Kullanılan Makine Öğrenmesi Teknikleri Ve Veri Setleri Üzerine Analiz Sonuçları

Elde edilen Tweet verileri üzerine 6 farklı makine öğrenmesi tekniği denenmiştir. Bu Algoritmalar; Naive Bayes, SVM, Logistic Regression, KNN, Random Forest ve Random Tree Algoritmalarıdır. Her bir algoritmanın veri seti üzerine başarısı ilerleyen bölümlerde verilmiştir.

3.5.1. Naive Bayes Algoritması

1960 yıllarından itibaren üzerinde çalışılan koşullu olasılık prensibine dayalı bir sınıflandırma algoritmasıdır [17]. İstatistik ve Bilgisayar Bilimlerinde farklı isimler ile basit Naive Bayes ve Bağımsız Naive Bayes olarak adlandırılabilmektedir [18].

3.5.2. SVM Algoritması

Makine öğrenmesinde, Destek Vektörü Makineleri, sınıflandırma ve regresyon analizi için kullanılan verileri analiz eden ilişkili öğrenme algoritmalarıyla denetlenen öğrenme modelidir [19]. Hava Siegelmann ve Vladimir Vapnik tarafından oluşturulan destek vektörü kümeleme algoritması, etiketleme verilerini kategorize etmek için destek vektör makineleri algoritmasında geliştirilen destek vektörlerinin istatistiklerini uygular ve endüstriyel alanda en yaygın kullanılan kümeleme algoritmalarından biridir [20].

3.5.3. Logistic Regression

Lojistik regresyon, bağımlı değişkenin kategorik bir değişken olduğu regresyon problemi gibidir. Doğrusal sınıflandırma problemlerinde yaygın bir biçimde kullanılır. Lojistik regresyon, bir sonucu belirleyen bir veya daha fazla bağımsız değişken bulunan bir veri kümesini analiz etmek için kullanılan istatistiksel bir yöntemdir. [10]

3.5.4. KNN

En Yakın Komşular Algoritması (k-nearest neighbors algorithm) , sınıflandırma ve regresyon için kullanılan parametrik olmayan bir yöntemdir [21]. Eğitim örnekleri, her biri bir sınıf etiketine sahip çok boyutlu bir özellik alanındaki vektörlerdir. Algoritmanın eğitim aşaması, sadece özellik vektörlerini ve eğitim örneklerinin sınıf etiketlerini saklamaktan ibarettir.

3.5.5. Random Forest ve Random Tree

Random Forest veya Random Decision Forests, sınıflandırma, regresyon, veya ortalama tahmin (regresyon) çıktısı veren bir öğrenme yöntemidir. Random Forest algoritması Tin Kam Ho tarafından geliştirilmiştir [22][23][24].

3.5.6. Veri Seti üzerinde algoritmaların sınıflandırma başarısı

Weka API ile yapılan analizler sonucu belirlenen sınıflandırma algoritmalarının veri seti üzerinde algoritmaların sınıflandırma başarısı çizelge 10'da verilmiştir.

Çizelge 10: Veri Seti üzerinde algoritmaların sınıflandırma başarısı

Algoritma	Toplam Örnek Sayısı	Doğru Sınıflandırılan örnek sayısı	Başarı Yüzdesi
Naive Bayes	635	481	%75.7480
SVM	635	505	%79.5276
Logistic Regression	635	505	%79.5276
KNN	635	635	%100
Random Forest	635	635	%100
Random Tree	635	635	%100

[10] https://veribilimcisi.com/2017/07/18/lojistik-regresyon/

Çizelge 11: Veri Seti üzerinde algoritmalara ait karmaşıklık matrisleri

Naive Bayes			**SVM**		
a	b		a	b	
13	117	a = spammer	0	130	a = spammer
37	468	b = non-Spammer	0	505	b = non-Spammer

Logistic Regression			**KNN**		
a	b		a	b	
0	130	a = spammer	130	0	a = spammer
0	505	b = non-Spammer	0	505	b = non-Spammer

Random Forest			**Random Tree**		
a	b		a	b	
130	0	a = spammer	130	0	a = spammer
0	505	b = non-Spammer	0	505	b = non-Spammer

Cross Validation işlemi yapılarak Algoritmaların veri seti üzerindeki başarısı ölçülmüştür. Çizelge 12 ve çizelge 13 de 20 katlamalı çapraz doğrulama (Cross Validation) gerçekleştirildiğinde veri seti üzerinde algoritmaların sınıflandırma başarısı ve karmaşıklık matrisleri verilmiştir.

Çizelge 12: Veri Seti üzerinde algoritmaların 20 katlamalı çapraz doğrulama (Cross Validation) sınıflandırma başarısı

Algoritma	**Toplam Örnek Sayısı**	**Doğru Sınıflandırılan örnek sayısı**	**Başarı Yüzdesi**
Naive Bayes	103629	91206	%88.0120
SVM	103629	98088	%94.6530
Logistic Regression	103629	97988	%94.5565
KNN	103629	97674	%94.2535
Random Forest	103629	100801	%97.2710
Random Tree	103629	98923	%95.4588

Çizelge 13: Veri Seti üzerinde algoritmalara ait karmaşıklık matrisleri – 20 katlamalı çapraz doğrulama (Cross Validation)

Naive Bayes			**SVM**		
a	b		a	b	
3534	5095	a = spammer	3088	5541	a = spammer
7328	87672	b = non-Spammer	0	95000	b = non-Spammer

Logistic Regression			**KNN**		
a	b		a	b	
3086	5543	a = spammer	5752	2877	a = spammer
98	94902	b = non-Spammer	3078	91922	b = non-Spammer

Random Forest			**Random Tree**		
a	b		a	b	
6058	2571	a = spammer	6295	2334	a = spammer
257	94743	b = non-Spammer	2372	92628	b = non-Spammer

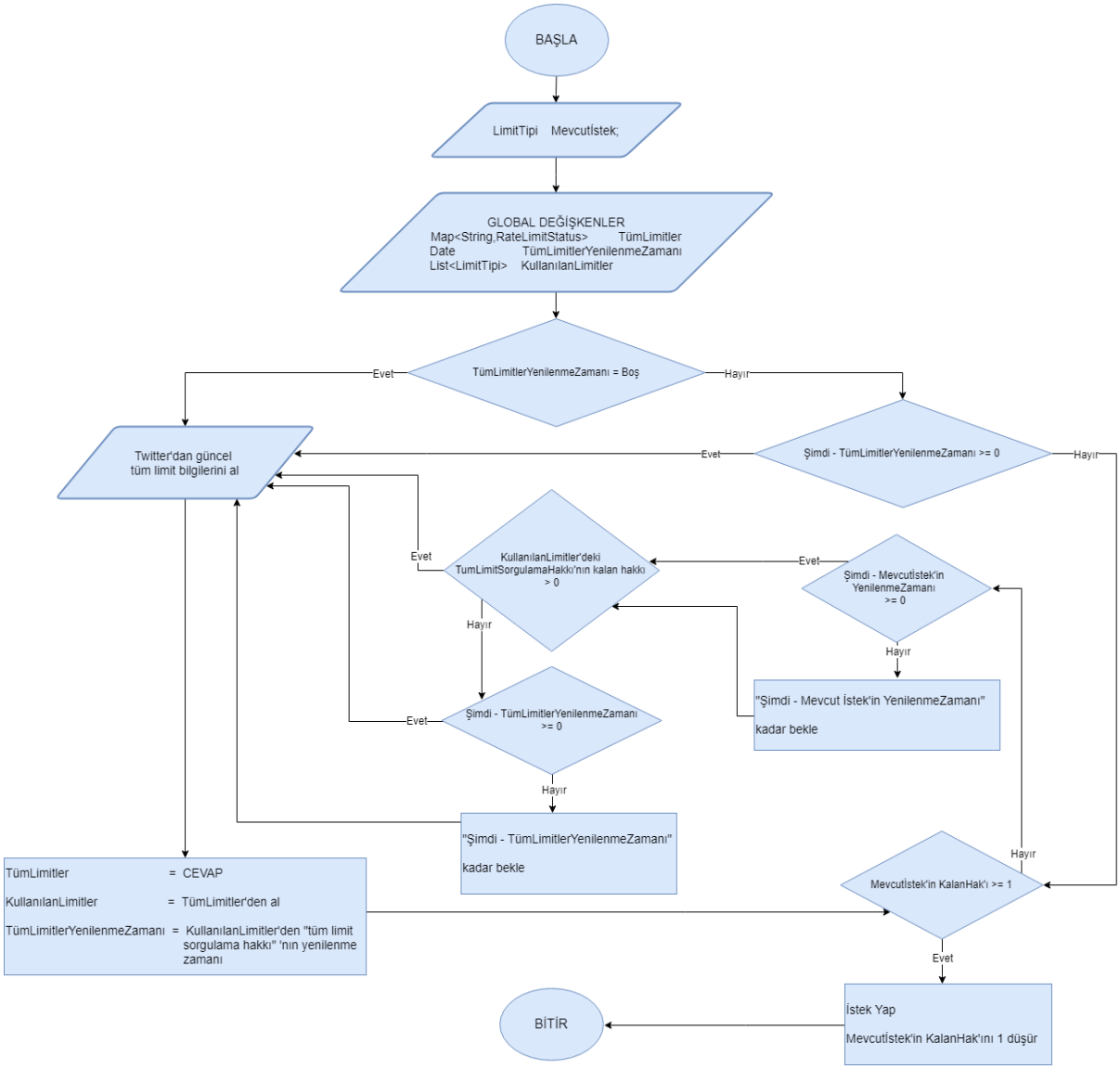
Görüldüğü gibi sınıflandırma başarısı algoritmadan algoritmaya değişmekte ve tercih edilen algoritmalar, veri seti üzerine yüksek bir yüzde ile sınıflandırma işlemini gerçekleştirmişlerdir. Bu işlemi gerçekleştirmek için yazılmış fonksiyon Ek-1' de verilmiştir.

3.6. Projenin Çalışma Mekanizması

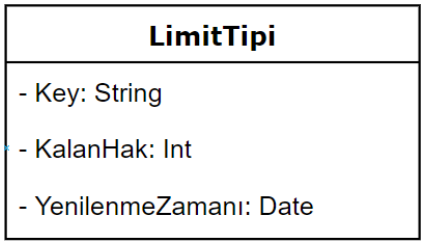
Yapılan çalışmada önceki bölümlerde de anlatıldığı üzere birçok teknoloji kullanılmıştır. Bu teknolojilerin bir araya gelmesi ile yürütülen çalışmada spam tespiti gerçekleştiren bir sistem ortaya çıkmıştır. Sistem üzerinde Twitter üzerinden veri akışı için TA kullanılmıştır. TA'nın geliştiriciler için sunmuş olduğu bu API' da, geliştirici belli kurallar çerçevesinde kalmak zorundadır. Bu kuralların en önemlisi yapılan isteklerde bir limitin olmasıdır. TA'nın bize sunmuş olduğu limitlerin dışında kasıtlı bir şekilde veya her hangi bir önlem alınmadığı için yapılan fazla isteklerde TA' da geliştirici BlackList' e düşebilir.[11] Bu durumun önüne geçilmesi için yapılan çalışmada bir izin mekanizması kurulmuş ve yapılan istekler kontrol altına alınmıştır. Bu işlemi gerçekleştiren mekanizma şekil 5 'te verilmiştir.

Twitter4j kütüphanesinde bulunan RateLimitStatus nesnelerine TwitterAPI'den kalan istek haklarımız çekilebilmektedir. Bu RateLimitStatus sınıfında sadece kapsülleme yapısının alıcı metodları (getter) mevcuttur. Limit kontrolünü başarılı bir şekilde yapabilmek için bu sınıfin alanlarından olan, kalan hak alanına müdahale etmemiz gerekmektedir. Bu sınıfin kopyası sayılabilecek bir sınıf oluşturulmuştur. LimitTipi adına olan bu sınıf şekil 6'da verilmiştir. Şekilde görüldüğü üzere; Key ifadesi ilgili limitin anahtar sözcüğünü, KalanHak ifadesi bu limit için kalan hakkı, YenilenmeZamanı ifadesi bu limitin tekrardan yenilenecek zamanı Date tipinde tutulmaktadır. TwitterAPI'den her genel limitler için kalan haklar sorgulandığında, RateLimitStatus sınıfından bu LimitTipi sınıfına veriler kopyalanmaktadır. Dolayısıyla her yapılan istek için buradaki hak 1 düşürülmekte, isteği yapmadan önce de kalan hakkın olup olmadığı kontrol edilmektedir. Eğer hak kalmamışsa, nesne içindeki YenilenmeZamanı alanının tuttuğu tarihe kadar beklenilmektedir.

[11] https://developer.twitter.com/en/docs/basics/rate-limiting



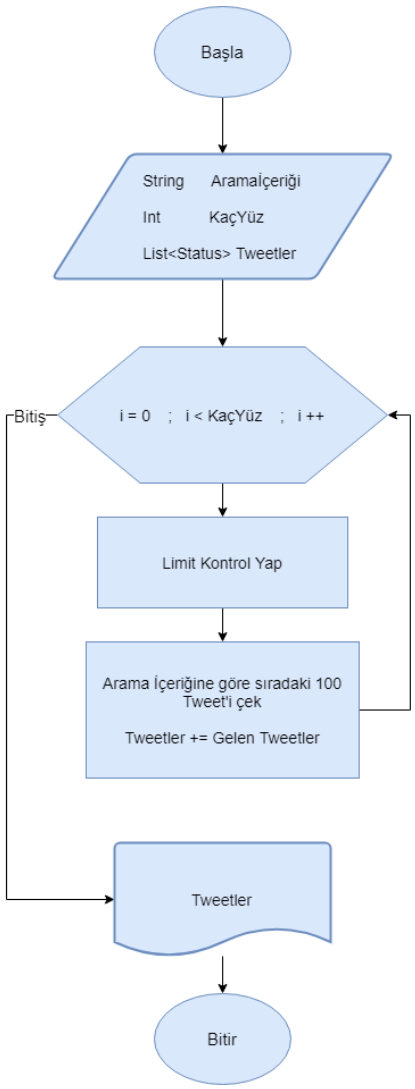
Şekil 5: TA Limit Kontrolü



Şekil 6: LimitTipi Sınıfı

Limit kontrolünden sonra TA üzerinden Tweet arama işlemine geçilebilir. Tweet arama için geliştirilen algoritmanın akış diyagramı ise şekil 7'de verilmiştir. Bu algoritmaya göre; İlgili

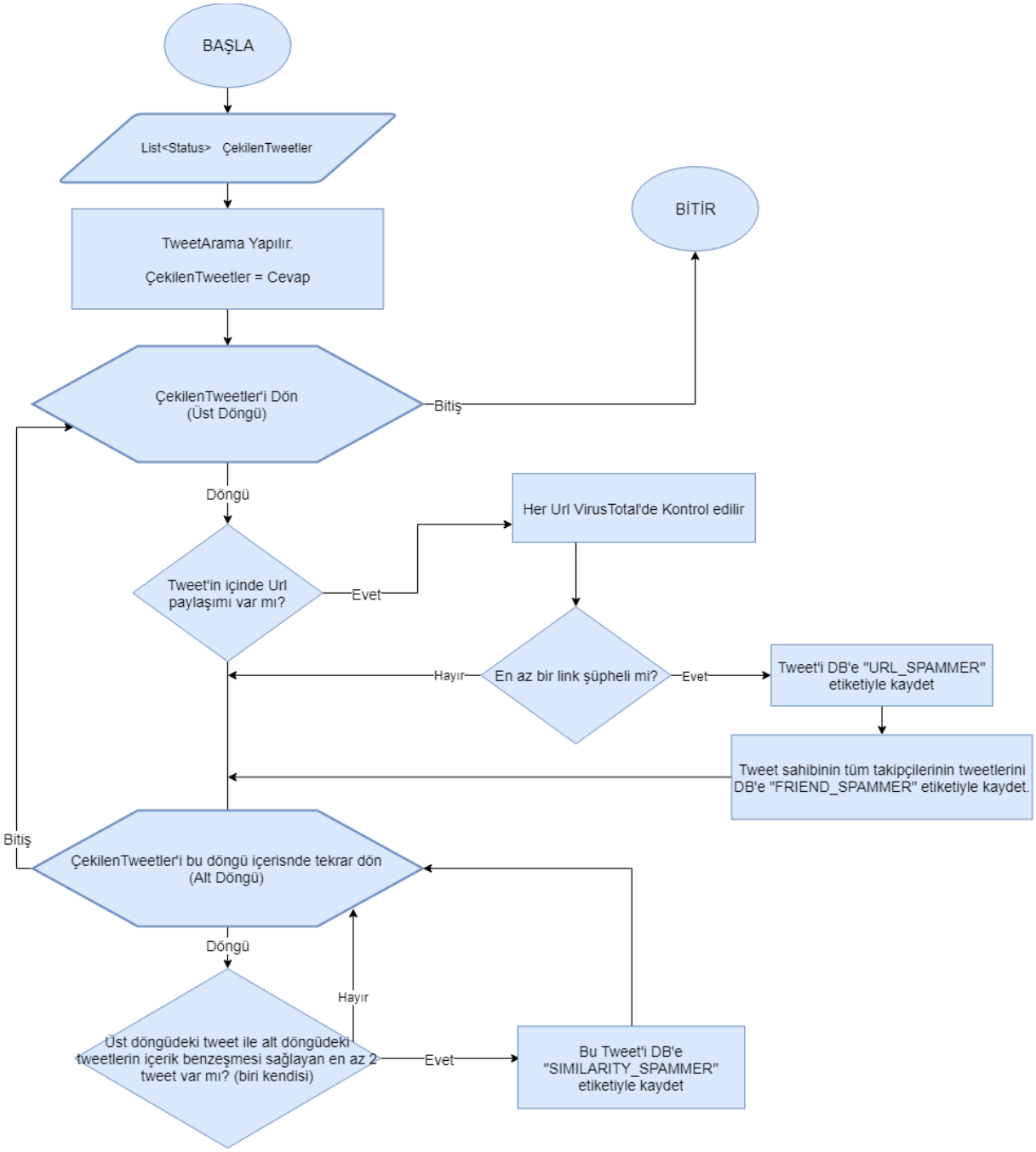
Tweet'leri çekmesi için istenilen arama sözcüğü AramaİçeriğiNe verilmektedir. Burada TA'da bulunan Sayfa mekanizmasına göre, çekilmek istenen sayfa sayısı girilir. TA'da her sayfada 100 Tweet gönderilmektedir. Kaç sayfa Tweet çekilmesi isteniyorsa(sayfa başına 100 Tweet) KaçYüz değişkenine verilmekte ve çekilen Tweet'ler bir listede toplanmaktadır. Her sayfa isteğinden sonra limit kontrolü yapılmakta ve en son kalınan sayfadan devam edilmektedir.



Şekil 7: Tweet Arama Algoritması

Tweet arama için geliştiren bu algoritma sistemin giriş kapısı olarak düşünülebilir. Buraya girilen arama kelimesine göre Twitter üzerinden limit kontrolü algoritması ile Tweet'ler çekilmektedir. Tweet arama işleminin devamı olarak geliştirdiğimiz spam arama algoritması çalışmaktadır. Burada Temel niteliği taşıyan fonksiyonların tamamlanması ardından(İzinler, Url Kontrol, Tweet Çekme), Teste tabi tutmak amacıyla spam Tweet arama işlemi gerçekleştirmek için şekil 8'deki algoritma geliştirilmiştir. Bu algoritmaya göre

TweetArama Algoritması ile istenilen kelimeye göre istenilen miktarda Tweet çekimi gerçekleştirilir. Tüm bu Tweet'ler ÇekilenTweetler adlı listede tutulur. ÇekilenTweetler adlı listedeki her bir Tweet için aşağıdaki maddelerdeki işlemler uygulanır. Sıradaki Tweet'in içinde Url varsa, hepsi VirusTotal Algoritmasına Gönderilir. En az bir url şüpheli ise Bu Tweet Ve sahibinin bilgileri test edilmeye uygun formata getirilip veri tabanına "URL_SPAMMER" etiketiyle kaydedilir. Bu Tweet sahibini takip edenler de spam kabul edilir ve onların da paylaştığı Tweet'ler sahiplerinin bilgileriyle teste uygun formata getirilip, veri tabanına "FRIEND_SPAMMER" etiketiyle kaydedilir. Sıradaki Tweet içeriği, diğer ÇekilenTweetler içindeki Tweet'lerin içeriği ile karşılaştırılır. Bu karşılaştırma Levenshtein Distance Algoritması ile yapılmıştır. En az 20 karakter olup %90 ve üzeri benzeşme sağlayan başka bir Tweet varsa, bu Tweet'ler sahiplerinin bilgileriyle teste uygun formata getirildikten sonra, veri tabanına "SIMILARITY_SPAMMER" etiketiyle kaydedilir.



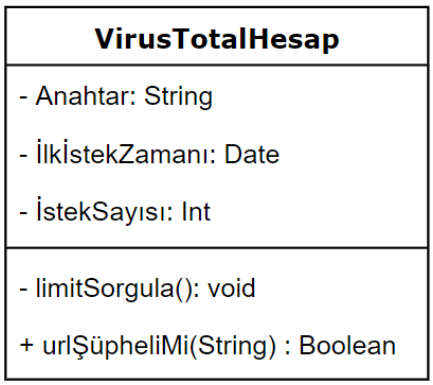
Şekil 8: Spam Tweet Arama Algoritması

Spam Tweet Arama Algoritmasından da görüleceği üzere spam olarak kabul edilen Tweet'lerin temel çıkış noktası zararlı url bulundurmalarıdır. Zararlı url bulundurma tespiti de VT API ile gerçekleştirilmiştir. Bu işlem için geliştirilen algoritma şekil 10 'da verilmiştir. VirusTotal bünyesindeki üyelere API hizmetini vermek için API anahtarı (Token) sağlamaktadır. Verilen API anahtarı ile standart üyeler için dakikada 4 veri sorgulama hakkına tabi tutmaktadır. İhtiyaç üzerine, VirusTotal adlı sınıf oluşturulmuştur içeriği şekil 9'da gösterilmiştir; Anahtar ifadesi Hesabın sahip olduğu API anahtarıdır. SonİstekZamanı

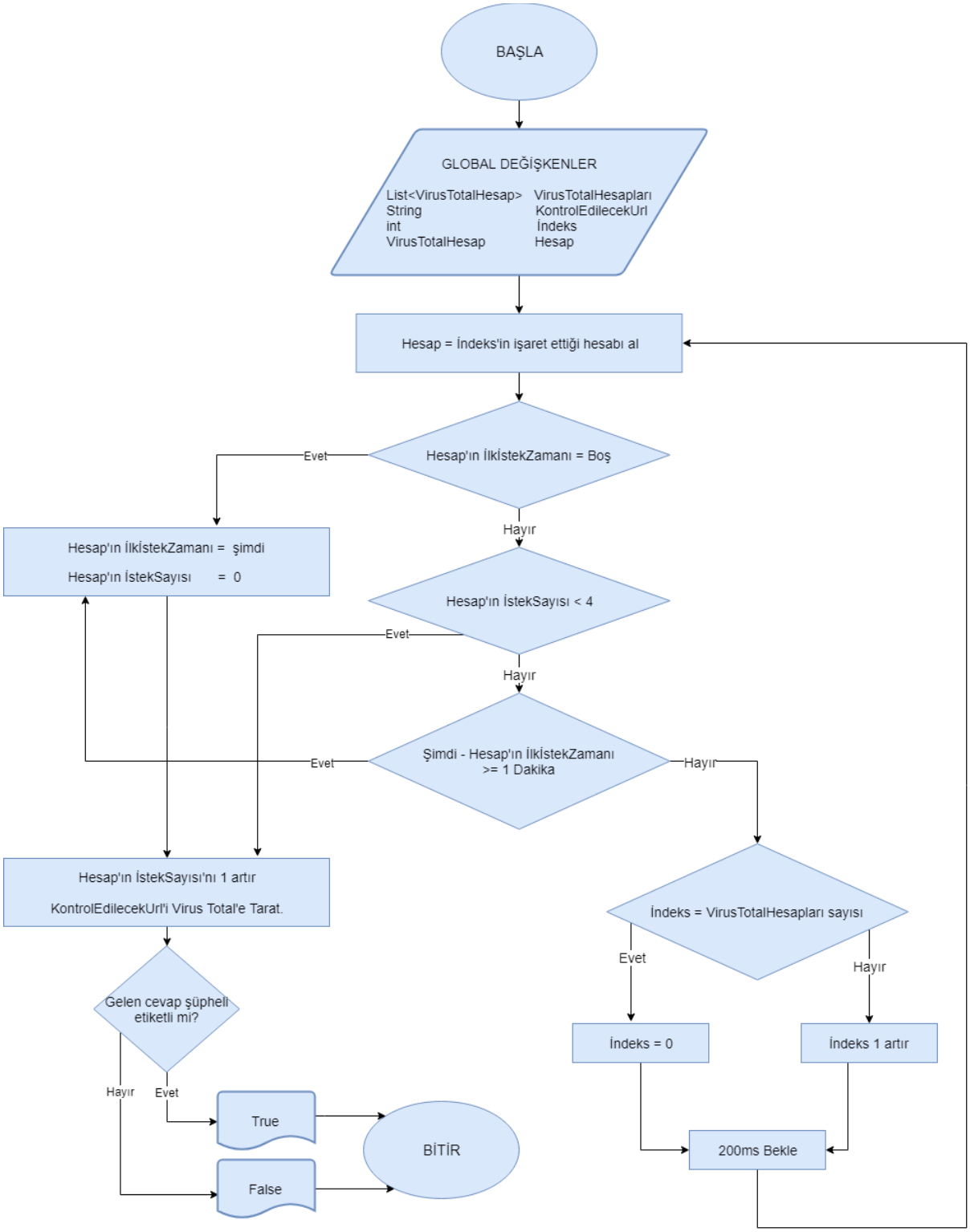
ifadesi son yapılan isteğin tarihi Date tipinde tutulmaktadır. İstekSayısı ifadesi SonİstekZamanı ifadesinde tutulan tarihinden sonra yapılan isteklerin toplamıdır Integer veri tipinde veri tutulmaktadır. Algoritmada, VirusTotalHesapları adlı dizide tüm VirusTotal hesapları tutulmaktadır. Ve bu hesaplar arasında iterasyonu sağlamak için Integer tipinde bir indeks tutucu oluşturulmuştur. Bu algoritmaya, kontrol edilmesi istenilen web bağlantısı (URL) gönderildiğinde, sıradaki VT hesabı üzerinde şu işlemler gerçekleştirilir;

- Eğer SonİstekZamanı boş ise daha önce istek yapılmamış demektir, dolayısıyla ilk istek başka kontrole gerek kalmadan sorgu istek yapılabilir. Ve ardından SonİstekZamanı güncellenirken İstekSayısı 1 yapılır.
- Eğer SonİstekZamanı dolu ise, İstekSayısı'na bakılır 4 sınırı aşılmamış ise istek yapılır ve İstekSayısı 1 artırılır.
- Eğer SonİstekZamanı dolu ve İstekSayısı 4'e erişmiş ve SonİstekZamanı üzerinden 1 dakika geçmişse, istek yapılması ardından, SonİstekZamanı güncellenirken İstekSayısı 1 yapılır.
- Eğer SonİstekZamanı dolu ve İstekSayısı 4'e erişmiş ve SonİstekZamanı üzerinden 1 dakika geçmemişse, 200ms Beklendikten sonra sıradaki VirusTotal hesabına geçiş yapılıp aynı işlemler gerçekleşir. Bu işlem istek gerçekleşene kadar devam eder.

Bu algoritma projede özyineli fonksiyon olarak kodlanmıştır. Hiçbir hesapta 1 dakika sürenin dolmaması ve 4 sınırına erişmiş olması durumunda, algoritma akışı gereği, tekrar tekrar sıradaki hesaba geçiş yapılıp Stack'in şişmesine neden olacaktır. Bu sorunun çözümü için son maddede görülen 200ms bekleme işlemi yapılmıştır.



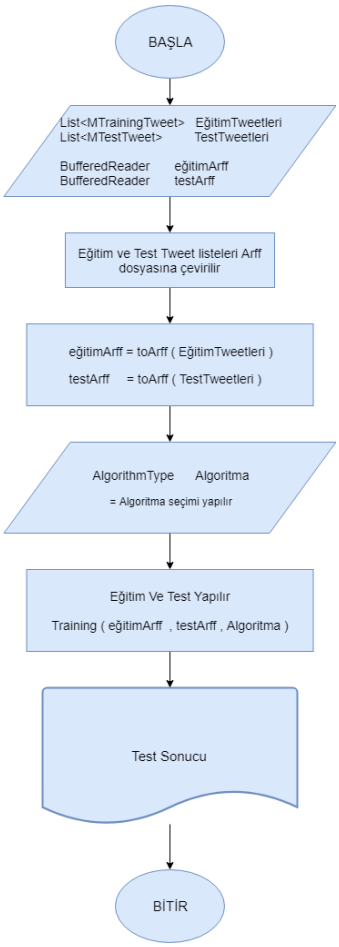
Şekil 9: VT Hesap Sınıfı



Şekil 10: VT Şüpheli Url tespiti Algoritması

4. TARTIŞMA VE SONUÇ

Kullanılan Araç ve Yöntem başlığı altında yapılan spam tespiti çalışmasında kullanılan teknolojiler, kayıt altına alınan Tweet'lere ait veri modeli, sistem eğitimi için toplanan veri setleri ve bu veri setleri üzerine uygulanan makine öğrenmesi yöntemleri verilmiştir. Yapılan çalışma sonucu bir araya gelen bu teknolojiler ve oluşturulan algoritmalar ile Twitter Spam Analizi gerçekleştiren bir sistem ortaya çıkmıştır. 3.Bölümde oluşturulan algoritmalar ve gerçekleştirdikleri işler verilmiştir. Bu kapsamda Tweet Arama Algoritması ile başlayıp, limitlerin kontrolü ile Spam Tespiti Algoritması ve VT API için oluşturulan algoritmalarının oluşturduğu bütün bize analiz edilebilir bir veri seti vermektedir. Daha önce de anlatıldığı gibi veri tabanımıza Spam şüphesiyle alınan ve "URL_SPAMMER", "FRIEND_SPAMMER" ve "SIMILARITY_SPAMMER" olarak işaretlenen bu Tweet'lerin eğitilmiş veri seti üzerine test işlemleri bu bölümde anlatılacaktır. Test işlemini gerçekleştiren algoritmaya ait akış diyagramı şekil 11'de verilmiştir.



Şekil 11: Spam Test Algoritması

Oluşturulan bu algoritmanın çalışma yapısı şöyledir; Bir önceki konuda spam teşhisi koyulan Tweet'ler veri tabanına kaydedilmişti. Bu Tweet'ler test edilme amacıyla programa veri tabanından çekilip TestTweetleri adlı listeye eklenmektedir. Eğitim için ise veri tabanında eğitim verileri olduğu kabul edilerek, EğitimTweetleri listesine eklenmektedir. Bu eğitim ve test işlemi Weka API kullanılarak gerçekleştirilmektedir. Dolayısıyla elde edilen eğitim ve test Tweet'leri Arff dosyasına, hazırlanan özel metot ile çevrilmektedir. Bu metot, oluşturulan Arff dosyasını BufferReader nesnesi olarak geri döndürmektedir. Ardından eğitim ve testin hangi algoritma ile yapılacağı seçilmektedir. Eğitim ve test ardından karmaşıklık matrisi ve algoritma başarısı ekrana yazdırılmaktadır.

Yapılan Spam Araması sonucu elimizde 3 farklı veri seti geçmektedir. Bu veri setleri "URL_SPAMMER" , "SIMILARITY_SPAMMER" ve "FRIEND_SPAMMER" olarak işaretlenen Tweetler'dir. Çeşitli anahtar kelimeler ile spam Tweet aramaları gerçekleştirilmiştir. Toplanan Tweet'ler üzerine sistemin başarısı ayrı ayrı aşağıda verilmiştir.

4.1. URL_SPAMMER Veri Seti

www, bit.ly , t.co , .net , .com , ow.ly ve tiny.cc anahtar kelimeleri ile TA üzerinden arama gerçekleştirilmiş arama sonucunda çekilen Tweet'lerde url var ve bu url VT API sonucunda zararları içerik içeriyorsa bu Tweet spam kabul edilmiştir. Arama sonucunda 118 Tweet spam olarak kabul edilmiştir. Bu Tweet'ler ile test işlemi yapıldığında Çizelge 13' teki ve Çizelge 14'teki sonuçlar elde edilmiştir.

Çizelge 14: URL_SPAMMER Veri Seti Üzerine Sistemin Başarısı

Algoritma	Toplam Örnek Sayısı	Doğru Sınıflandırılan örnek sayısı	Başarı Yüzdesi
Naive Bayes	118	17	%14.4068
Random Forest	118	11	%9.3220

Çizelge 15: URL_SPAMMER Veri Seti Üzerine Sistemin Başarısı – karmaşıklık matrisi

Naive Bayes			Random Forest		
a	b		a	b	
17	101	a = spammer	11	107	a = spammer
0	0	b = non-Spammer	0	0	b = non-Spammer

4.2. FRIEND_SPAMMER Veri Seti

URL_SPAMMER Veri Setinde; www, bit.ly , t.co , .net , .com , ow.ly ve tiny.cc anahtar kelimeleri ile TA üzerinden arama gerçekleştirilmiş arama sonucunda çekilen Tweet'lerde url var ve bu url VT API sonucunda zararları içerik içeriyorsa bu Tweet spam kabul edilmişti. Arama sonucunda spam olarak kabul edilen Tweet'lerin sahibinin takip etikleri hesaplar ve bu hesaplardaki Tweet'lerinde spam olarak kabul edildiği veri setidir . Bu Tweet'ler ile test işlemi yapıldığında Çizelge 15' teki ve Çizelge 16'daki sonuçlar elde edilmiştir.

Çizelge 16: FRIEND_SPAMMER Veri Seti Üzerine Sistemin Başarısı

Algoritma	Toplam Örnek Sayısı	Doğru Sınıflandırılan örnek sayısı	Başarı Yüzdesi
Naive Bayes	48456	28186	%58.1682
Random Forest	48456	32992	%68.0865

Çizelge 17: FRIEND _ SPAMMER Veri Seti Üzerine Sistemin Başarısı – karmaşıklık matrisi

Naive Bayes			Random Forest		
a	b		a	b	
28186	20270	a = spammer	32991	15624	a = spammer
0	0	b = non-Spammer	0	0	b = non-Spammer

4.3. SIMILARITY_SPAMMER Veri Seti

www, bit.ly , t.co , .net , .com , ow.ly ve tiny.cc anahtar kelimeleri ile TA üzerinden arama gerçekleştirilmiş arama sonucunda çekilen Tweet'lerde metin benzerliği %90'nın üzerinde ve metin uzunluğu 20 karakterin üzerinde olan Tweet'lerden oluşan veri setidir. Burada arama kelimelerinin linklerden oluşması bir önceki iki veri setini oluşturma aşamasında bu veri setini de oluşturmak amacıyla girilmiştir. Bu Tweet'ler ile test işlemi yapıldığında Çizelge 17' teki ve Çizelge 18'deki sonuçlar elde edilmiştir.

Çizelge 18: SIMILARITY _SPAMMER Veri Seti Üzerine Sistemin Başarısı

Algoritma	Toplam Örnek Sayısı	Doğru Sınıflandırılan örnek sayısı	Başarı Yüzdesi
Naive Bayes	1520	630	%41.4474
Random Forest	1520	897	%59.0132

Çizelge 19: SIMILARITY_ SPAMMER Veri Seti Üzerine Sistemin Başarısı – karmaşıklık matrisi

Naive Bayes			Random Forest		
a	b		a	b	
630	890	a = spammer	897	623	a = spammer
0	0	b = non-Spammer	0	0	b = non-Spammer

Yapılan çalışma sonucu elde edilen sonuçlar ve daha önce yapılmış spam tespiti çalışmaları Çizelge 20'de verilmiştir.

Çizelge 20: Yapılmış Çalışmalar Ve Sonuçları

Çalışma Kaynakça Numarası	Çalışma Yılı	Kullanılan Yöntem	Veri Seti	Veri Seti Boyutu	Başarı Oranı
-	2019	Random Forest, Naive Bayes	URL_SPAMMER	118	%14.40 - %9.32
-	2019	Random Forest, Naive Bayes	FRIEND_SPAMMER	48.456	%58.16 - %68.08
-	2019	Random Forest, Naive Bayes	SIMILARITY_SPAMMER	1.520	%41.44 - %59.01
1	2016	Url bazlı Tweet	10-day Veri Seti	600 Milyon	%87
2	2018	Random Forest	SPDautomated ve Honeypot veri setleri	61.845	%94.7
3	2016	SVM	Honeypot ve Kwak	-	%84
4	2016	Naive Bayes, Random Forest, J48 and SVM	Kendi veri setlerini oluşturmuşlardır	29.000	%92.34
5	2013	Naif Bayes	Kendi veri setlerini oluşturmuşlardır	3.239	%91.7
6	2009	SVM	Kendi veri setlerini oluşturmuşlardır	1.7 milyar	%70
7	2011	Random Forest, SVM, KNN, and Naïve Bayes	Kendi veri setlerini oluşturmuşlardır	1000	%93 - %95
8	2014	Naive Bayes	Kendi veri setlerini oluşturmuşlardır	500.000	%91.7
9	2017	içerik benzerligi	Honeypot	23.869	%89
10	2019	DetectVC ve CatchSync	Kendi veri setlerini oluşturmuşlardır	34.826	%98.1
11	2009	Collusionrank, TSP-filtreleme, SS-filtreleme	Twitter-takip link	54.981.152	%94
13	2015	Decision Tree,Nave Bayes,Random Forest	Kendi veri setlerini oluşturmuşlardır	4.820	%94
14	2012	Url bazlı Tweet	Kendi veri setlerini oluşturmuşlardır	50 Milyon	% 82,3 - %94,5
15	2013	Url bazlı Tweet	Kendi veri setlerini oluşturmuşlardır	20 milyon	%89.3

5. KAPANIŞ

Test amaçlı üç farklı veri seti olarak toplanan gerçek Tweet'lere ait analizler beşinci bölümde ayrı başlıklar ve çizelgelerde verilmiştir. Bu sonuçlara göre url bazlı spam analizinin düşük çıktığı görülmüştür. Burada sistemin başarısının düşük olması ICC ve SuperBowl Veri Setlerinin eski tarihli veriler olmasından kaynaklanması muhtemeldir. Çünkü yıllar içerisinde spammer olarak adlandırılan kötü içerik üreticilerinin gösterdiği davranışlar değişmektedir. Yapılan çalışmada spam tespiti için toplanan örnek veri kriterleri arasında spam olarak kabul edilen kişilerin takip ettiği hesaplar ve metin benzerliğine dayalı spam tespiti de bulunmaktadır. Yapılan çalışma sonucunda da bu iki yaklaşım için sistemin başarısı yüksek çıkmıştır. Burada yıllar içerisinde spamların değişen davranışları üzerine eski tarihli veri setleri ile yapılan çalışmalar doğru sonuç göstermeyebileceği görülmüştür. Bu nedenle yapılan çoğu çalışmada toplanan bilgiler ışığında kendi veri setlerini oluşturmuşlar ve bu veri seti üzerine eğitim işlemi gerçekleştirmişlerdir. Yapılan analizler sonucunda Tweet'ler üzerinde benzerlik oranına dayalı spam tespit, zararlı içerikli url içeren Tweet'ler ve spam hesapların takip ettiği hesapların spam tespiti araştırmalarında kullanılması durumda dikkat edilmesi gereken sonuçlar bulunmaktadır. Örneğin metin benzerliğine dayalı bir analiz sonucunda paylaşılan Tweet içeri bir özlü söz veya yanlış yazılmış bir TT hashtag'i ise spam olmayan hesapların da spam olarak kabul edileceğinden dolayı sistemin başarısı düşecektir. Yine aynı şekilde spam hesapların takip ettiği hesapları da spam kabul etmek sistemin başarısını etkilemektedir. url bazlı spam tespitinde de kullanılan virüs tarama teknolojisi önem kazanmaktadır. Çünkü bu hizmeti sunan her teknolojinin kullandığı veri tabanları farklı ve dolayısıyla farklı sonuçlar verebilmektedir.

KAYNAKÇA

[1] Tingmin Wu[a], Sheng Wen[a] , Shigang Liu[a], Jun Zhang[a], Yang Xiang[a], Majed Alrubaian[b], Mohammad Mehedi Hassan[b], 2017, Detecting spamming activities in twitter based on deep-learning technique, [a]School of Information Technology, Deakin University, Victoria 3125, Australia,[b] College of Computer and Information Sciences King Saud University, Riyadh 11543, Saudi Arabia

[2] Isa Inuwa-Dutse, Mark Liptrott, Ioannis Korkontzelos, 2018, Detection of spam-posting accounts on Twitter, Department of Computer Science, Edge Hill University, Ormskirk, Lancashire, UK

[3] Hua Shenaad, Fenglong Macc, Xianchao Zhangb, Linlin Zongaab, Xinyue Liub, Wenxin Liangb, 2016, Discovering social spammers from multiple views, aSchool of Computer Science and Technology, Dalian University of Technology, Dalian 116024, Chinab School of Software, Dalian University of Technology, Dalian 116620, Chinac SUNY at Buffalo, Buffalo, NY 14221, USA d College of Mathematics and Information Science, Anshan Normal University, Anshan 114007, China

[4] Nasira Perveena, Qaisar Rasoola, Malik M. Saad Missenb, Nadeem Akhtar b,2016, Sentiment Based Twitter Spam Detection, a Dept. of Comp. Science Bahauddin Zakaria Univ. Multan Pakistan, b Dept. of Comp. Science and IT The Islamia University of Bahawalpur Pakistan

[5] Miller Zacharya, Dickinson Briana, Deitrick Williama, Hu Weia, Wang Alex Haib, 2014, Twitter spammer Detection using data stream clustrering, aDepartment of Computer Science, Houghton College, Houghton, NY, USA, bCollege of Information Sciences and Technology, The Pennsylvania State University, Dunmore, PA, USA

[6] Fabr´ıcio Benevenuto, Gabriel Magno, Tiago Rodrigues, Virg´ılio Almeida,2010, Detecting Spammers on Twitter, Computer Science Department, Universidade Federal de Minas GeraisBelo Horizonte Brazil,

[7] McCort M., Chuah M., 2011, Spam Detection on Twitter Using Traditional Classifiers, Computer Science & Engineering Department, Lehigh University, Bethlehem, PA 18015, USA

[8] Wang Alex Hai, 2014, Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach, College of Information Sciences and Technology, The Pennsylvania State University, Dunmore, PA 18512, USA

[9] P. V. Bindu ·,Rahul Mishra, · P. Santhi Thilagam,2017, Discovering spammer communities in twitter, Department of Computer Science & Engineering, National Institute of Technology Karnataka, Surathkal, India

[10] Boyeon Jang, Sihyun Jeong, Chong-kwon Kim, 2018, Distance-based customer detection in fake follower markets, Department of Computer Science and Engineering, Seoul National University, Gwanak-gu, Seoul 08826, Republic of Korea

[11] Sihyun Jeongaa , Giseop Nohaa , Hayoung Ohb, Chong-kwon Kimb,2016, Follow spam detection based on cascaded social information, aDept. of Computer Science and Engineering, Seoul National University Gwanak-gu Seoul 151–744, bRepublic of Korea b School of Electronic and Engineering, Soongsil University, Dongjak-gu, Seoul 156–743 Republic of Korea

[12] Chao Chenaa , Sheng Wenaa, Jun Zhanga , Yang Xianga , Jonathan Oliverb, Abdulhameed Alelaiwic , Mohammad Mehedi Hassanc, 2016,aSchool of Information Technology, Deakin University, 221 BurwoodHwy, Burwood, Vic 3125, Australia b Trend Micro, Australia, 606 St Kilda Road, Melbourne, Vic 3004, Australia CCollege of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

[13] Rasula Venkatesh , 2015, Malicious Accounts Detection based on Short URLs in Twitter, Department of Computer Science and Engineering National Institute of Technology Rourkela Rourkela – 769 008, India

[14] Zi Chu, Indra Widjaja, Haining Wang, Murray Hill, 2012, Detecting Social Spam Campaigns on Twitter, Department of Computer Science, The College of William and Mary,Williamsburg, VA 23187 USA, Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ 07974, USA

[15] Martinez-Romo Juan, Araujo Lourdes, 2013, Detecting malicious tweets in trending topics using a statistical analysis of language, Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia (UNED), Madrid 28040, Spain

[16] Whitson Gordon. "Understanding OAuth: What Happens When You Log Into a Site with Google, Twitter, or Facebook". Retrieved 15 May 2016.

[17] Rennie Jason D. M., Shih Lawrence, Teevan Jaime, Karger David R., 2003, Tackling the Poor Assumptions of Naive Bayes Text Classifiers, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

[18] Hand David J., Yu Kemingb, 2001, Idiot's Bayes-Not So Stupid After All, Department of Mathematics, Imperial College, London, UK, bUniversity of Plymouth, UK

[19] Cortes Corinna, Vapnik Vladimir, 1995, Support-Vector Networks, AT&T Bell Labs., Holmdel, NJ 07733, USA

[20] Ben-Hur Asa[a], Horn David[b], Siegelmann Hava T.[c], Vapnik Vladimird, 2001, Support Vector Clustering, [a]BIOwulf Technologies, Berkeley, CA 94704, USA, [b]School of Physics and Astronomy, Tel Aviv University, Tel Aviv 69978, Israel, [c]Lab for Information and Decision Systems MIT Cambridge, MA 02139, USA, [d]AT&T Labs Research, Red Bank, NJ 07701, USA

[21] Altman N. S., 1992, An Introduction to Kernel and Nearest Neighbor Nonparametric Regression, Biometrics Unit, Cornell University, Ithaca, NY 14853, USA

[22] Ho Tin Kam, 1995, Random Decision Forests, AT&T Bell Labs., Holmdel, NJ 07733, USA

[23] Ho Tin Kam, 1998, The Random Subspace Method for Constructing Decision Forests, AT&T Bell Labs., Holmdel, NJ 07733, USA

[24] Hastie Trevor, Tibshirani Robert, Friedman Jerome, 2008, Stanford University, California, CA 94305, USA

EKLER

Ek-1

```
public static void Training(BufferedReader trainingDataset, BufferedReader
testDataset , AlgorithmType algorithmType , Integer... crossNumFolds )
throws Exception {
    Instances datasetTrainig = new Instances(trainingDataset);
    datasetTrainig.setClassIndex(datasetTrainig.numAttributes()-1);

    Instances datasetTest = new Instances(testDataset);
    datasetTest.setClassIndex(datasetTest.numAttributes()-1);

    Classifier trainingClassifier = null;

    switch (algorithmType)
    {
        case NAIVE_BAYES:
            trainingClassifier = new NaiveBayes();
            break;
        case LOGISTIC_REGRESSION:
            trainingClassifier = new Logistic();
            break;
        case SUPPORT_VECTOR_MACHINE:
            trainingClassifier =  new SMO();
            break;
        case RANDOM_FOREST:
            trainingClassifier =  new RandomForest();
            break;
        case RANDOM_TREE:
            trainingClassifier =  new RandomTree();
            break;
        case K_NEAREST_NEIGHBOURS:
            trainingClassifier =  new IBk();
            break;
    }

    trainingClassifier.buildClassifier(datasetTrainig);

    Evaluation eval = new Evaluation(datasetTrainig);
    if (crossNumFolds.length>0)
        eval.crossValidateModel(trainingClassifier, datasetTrainig,
crossNumFolds[0], new Random(1));
    else
        eval.evaluateModel(trainingClassifier,datasetTest);

    System.out.println(eval.toSummaryString());
    System.out.print(" Algoritmasına göre giriş verilerinin ifadesi ");
    System.out.println(trainingClassifier);
    System.out.println(eval.toMatrixString());
    System.out.println(eval.toClassDetailsString());
}
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyad	Abdullah CANGUL
Doğum Tarihi Ve Yeri	01.05.1996 - Kelkit/GÜMÜŞHANE
Yabancı Dil	İngilizce
E-Posta	abdullahcangul@gmail.com
Telefon	+9 0553 791 29 51

EĞİTİM BİLGİLERİ

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Bilgisayar Mühendisliği	Düzce Üniversitesi	2019
Lise	Fen Bilimleri	Orhangazi Anadolu Lisesi	2014

KİŞİSEL BİLGİLER

Adı Soyadı	Hamza TAŞ
Doğum Tarihi Ve Yeri	29.04.1996 – Merkez/BATMAN
Yabancı Dil	İngilizce
E-Posta	hamza.tas@yahoo.com
Telefon	+9 0534 668 27 48

EĞİTİM BİLGİLERİ

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Bilgisayar Mühendisliği	Düzce Üniversitesi	2019
Lise	Fen Bilimleri	İbrahim Turhan Anadolu Lisesi	2015

KİŞİSEL BİLGİLER

Adı soyadı	Muhammed Erdem SİYAM
Doğum Tarihi Ve Yeri	25.08.1996 – Merkez/DÜZCE
Yabancı Dil	İngilizce
E-Posta	erdemsiyam@gmail.com
Telefon	+9 0551 132 75 35

EĞİTİM BİLGİLERİ

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Bilgisayar Mühendisliği	Düzce Üniversitesi	2019
Lise	Fen Bilimleri	Akçakoca Barbaros Anadolu Lisesi	2014