

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 409 BİTİRME ÇALIŞMASI

AĞ İZLEME VE ANORMALLİK TESPİTİ

G1409.10041 - Burak DÜZ
G1512.10071 - Enes KOCAMAN

Bölüm : **BİLGİSAYAR MÜHENDİSLİĞİ**
Danışman : **Öğr. Üyesi Dr. Mustafa AKPINAR**

2018-2019 Bahar Dönemi

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

AĞ İZLEME VE ANORMALLİK TESPİTİ

BSM 498 - BİTİRME ÇALIŞMASI

**Burak DÜZ
Enes KOCAMAN**

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

.....
Jüri Başkanı

.....
Üye

.....
Üye

ÖNSÖZ

Günümüzde birçok kurumsal ağ sistemi ve bu ağlara izleme hizmet veren firmalar bulunmaktadır. Ancak bu firmaların alarm sistemleri, eşik değerleri üzerine dayanmaktadır. Bu sebeple özel günlerde ve günün bazı saatlerinde sistemler aşırı derecede fazla uyarı vermesi durumunda bilgi kirliliği kaçınılmaz oluyor. Ayrıca özünde önemli olan birçok hata insan gücü kullanılmasından dolayı kaçırılıyor.

Bitirme projemizdeki amacımız, SNMP protokolü ile aldığımız verileri, makine öğrenmesi algoritmaları yardımı ile sisteme öğretmek ve ortaya çıkan anormallikleri raporlamak. Bu sayede yukarıda bahsettiğimiz problemin önüne geçmek istedik. Bu çalışma sırasında bize destek olan değerli hocamız Dr. Mustafa AKPINAR'a teşekkür ederiz.

İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
TABLOLAR LİSTESİ.....	viii
ÖZET.....	ix
BÖLÜM 1.	
GİRİŞ.....	1
1.1.SNMP.....	2
1.1.1. SNMP verilerinin alınması.....	2
1.2.ELK-Stack Yapısı.....	2
1.2.1. Elasticsearch.....	3
1.2.2. Logstash.....	4
1.2.3. Kibana.....	4
1.3.Makine Öğrenmesi.....	4
BÖLÜM 2.	
SİSTEM MİMARİSİ.....	5
2.2. Donanım Mimarisi.....	5
2.3. Yazılım Mimarisi.....	6
BÖLÜM 3.	
MODEL.....	7
3.1. SNMP Verilerinin Alınması.....	8
3.2. Verilerin Hazırlanması Ve Makinenin Eğitilmesi.....	9

BÖLÜM 4.	
SONUÇLAR VE ÖNERİLER.....	11
KAYNAKLAR.....	12
ÖZGEÇMİŞ.....	13
BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	14

SİMGELER VE KISALTMALAR LİSTESİ

SNMP	: Simple Network Management Protocol
MIB	: Management Information Base
ELK-Stack	: Elasticsearch Logstash Kibana Stack
JSON	: JavaScript Object Notation
WCSS	: Within-cluster Sums Of Squares

ŞEKİLLER LİSTESİ

Şekil 1.1.	Yıllara Göre İnternet Kullanımı	01
Şekil 1.2.	ELK-Stack Yapısı	03
Şekil 1.2.1.	Elasticsearch Veri Indexleme	03
Şekil 2.1.	Sistemin Çalışma Şekli	05
Şekil 2.2.1	K-means++ Örnek Kümeleme	07
Şekil 3.2.	WCSS Değer Grafiği	10

TABLolar LİSTESİ

Tablo 3.1.	SNMP OIDs ve İsimleri	08
------------	-----------------------	----

ÖZET

Anahtar kelimeler: SNMP, Ağ İzleme ve Makine Öğrenmesi

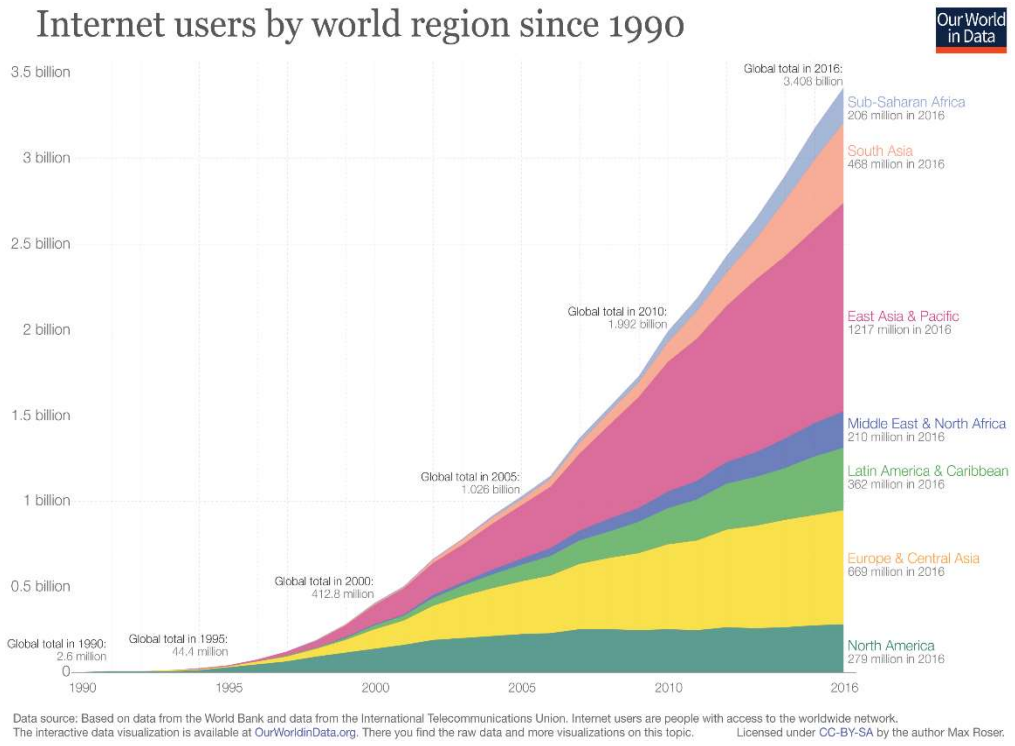
Her geçen gün dünya genelinde internet kullanımı artmaktadır. Bununla beraber şirketlerin internet trafiğini kontrol etmesi zorlaşmaktadır.

Projemizde, kullanıcıların yönlendiricilerin'deki SNMP(Basit Ağ Yönetim Protokolü) ile aldığımız verileri makine öğrenmesi algoritmaları yardımı sayesinde sisteme öğretip alarm sisteminin doğru bir şekilde çalışması amaçlanmıştır.

Bu bitirme çalışması ile SNMP protokolü, ELK-stack yapısı ve makine öğrenmesinin temelleri araştırılmıştır. Sonuç olarak SNMP ile verilerin alınması tamamlanmış ve bu verilerin doğru bir şekilde ELK-stack yapısına aktarılması sağlanmıştır.

BÖLÜM 1. GİRİŞ

İnternet kullanımını her geçen gün artmaktadır ve kontrol etmesi zorlaşmaktadır. Dolayısıyla ağ üzerindeki sorunların anlaşılması da gün geçtikçe zorlaşmaktadır. Projedeki amacımız, ağ üzerindeki veri akışını makinaya öğretmek için bir anormallik tespit sistemi oluşturarak bu zorlukların önüne geçebilmektir.



Şekil 1.1. Yıllara Göre İnternet Kullanımı[7]

1.1. SNMP

SNMP, bir ağ yönetim protokolüdür, SNMP kullanarak internete bağlanan herhangi bir cihaza(bilgisayar, yönlendirici, anahtarlama cihazı vb.) ait bilgileri elde edebilirsiniz. Bu bilgilerden bazıları cpu kullanımı, cihaz sıcaklığı, veri akışı ve üretim bilgileridir.

SNMP kullanılan sistemde 3 temel bileşen bulunur bunlar: SNMP yöneticisi, SNMP ajanı ve SNMP MIB.

SNMP yöneticisi ağ yöneticisi tarafından kullanılan yönetim yazılımıdır bu yazılım ile SNMP ile veriler elde edilir.

SNMP ajanı, bilgilere erişilmek istenilen cihazda bulunan yazılımdır. Bu yazılım sayesinde gönderdiğimiz sorgu sonucu dönen verileri alırız.

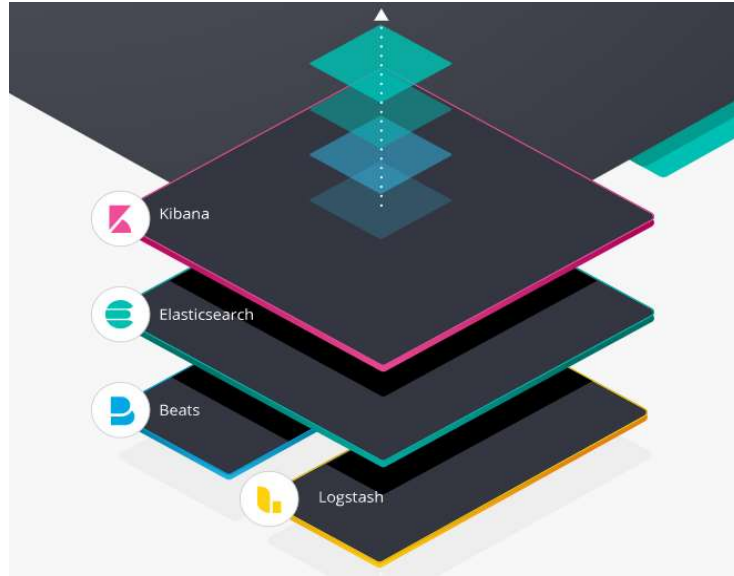
SNMP MIB Yönetim Bilgi Birimleri(Management Information Base) ağ yöneticisi ile cihaz arasında iletilecek bilgilerin yapılarını içeren bileşendir.[1]

1.1.1. SNMP verilerinin alınması

SNMP verilerinin alınması için C# dili ve SntpSharpNet kütüphanesi kullanılmıştır. SntpSharpNet kütüphanesi opensource bir kütüphanedir. SNMP version 1, 2 ve 3 Get, Get-Next, Get-Bulk ve Set request desteği sunmaktadır. Aynı zamanda Trap verilerini almamıza da olanak tanır.[2]

1.2. ELK-Stack Yapısı

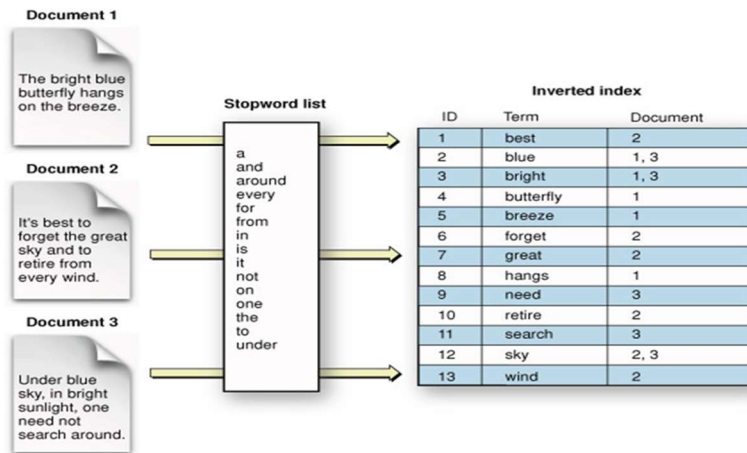
Elastic Stack, açık kaynak kodlu log toplama arama ve analiz bileşenleri için kullanılan, Elastic firmasının çıkardığı ürünlerin kullanımından oluşan yapıya denir. Günümüzdeki log izleme ve analiz konusunda hem firmaların ürünleriyle olan kolay entegrasi hem de kendi içinde durmadan yenilenen yapısı ile popüler olmuştur.



Şekil 1.2. ELK-Stack yapısı

1.2.1. Elasticsearch

Elasticsearch, Apache Lucene üzerinde geliştirilmiş açık kaynak kodlu, Restful, dağıtılmış arama ve analitik altyapısıdır[3]. Elasticsearch JSON türünde veri alır ve bu veriyi depolar, işler ve aranabilir belge başvurusuna ekler. Elasticsearch ücretsiz, büyük hacimli verileri paralel işleyerek sizin istediğiniz sonucu en kısa sürede elde etmenizi sağladığı ve diğer uygulamalara, yazılan projelere kolay entegre edilebildiği için popüler bir arama motoru ve NoSql veritabanıdır.



Şekil 1.2.1 Elasticsearch veri indexleme[4]

1.2.2. Logstash

Logstash server tarafında çalışan data processing pipeline'dır. Logstash birden fazla kaynaktan gelen veriyi bizim verdiğimiz konfigürasyonlara göre değiştirir ve bizim istediğimiz veri saklama noktasına gönderir[5]. Bizim kullandığımız veri saklama noktası Elasticsearch. Logstash birçok sistemden gelen verileri alabilir bunları kendi sayfasındaki input bölümünden bulabilirsiniz aynı zamanda open source olduğu için Logstash tarafından yayınlanmayan sistemler için kendi input kodlarımızı yazabiliriz.

1.2.3. Kibana

Kibana open-source analiz ve görselleştirme platformudur. Elasticsearch ile birlikte çalışır, Elasticsearch için hazırlanmış kendi içinde bir arayüz yoktur. Bu ihtiyacı Kibana karşılar. Kibana ile verileriniz arasında istediğiniz şekilde dolaşabilirsiniz. Kibana ile kendi görselleştirme işlerinizi yapabilir bunları birden fazla dashboard içinde tutabilir ve hazırlanan grafikler üzerinde isteğiniz sorguları yapabilirsiniz.

1.3 Makine Öğrenmesi

Makine öğrenmesi(Machine Learning), matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyene dair tahminlerde bulunan yöntem paradigmasıdır[6]. Çoğu insanın market alışverişi ve online satış sebebiyle aşina olduğu bu sistem, çalıştırdığı tavsiye motorlarının makine öğrenmesi algoritmalarını kullanması sayesinde verilen reklamları daha fazla kişiselleştirebilir. Kullanım amaçlarına değinecek olursak; elimizdeki ham veriye ait yorumlamalar yapabilme, var olan veri girişlerini kullanarak bir sonraki veri çıkışlarını tahmin edebilme, hali hazırda var olan bir sistemin çalışma evresini izleyip sonrasında

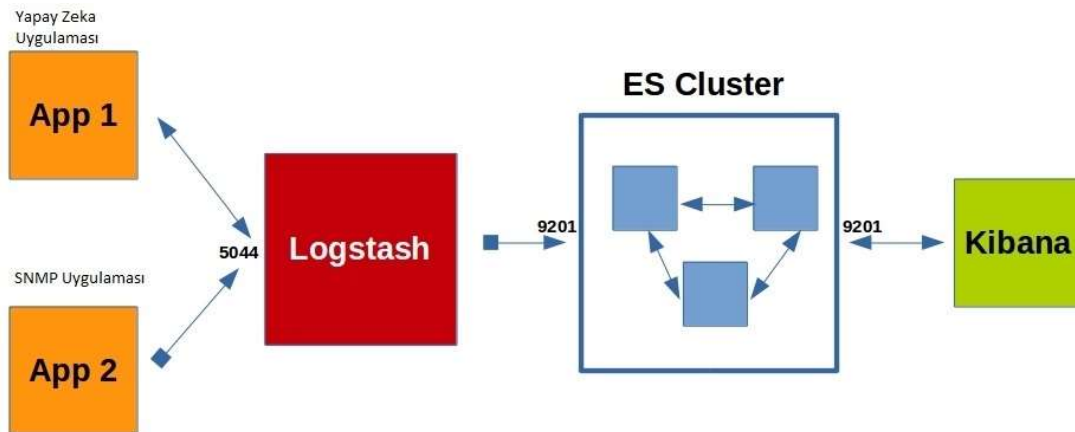
öğrenip sistemin aksine bir durum tespit edildiğinde durumu raporlama ve gerekirse işlem yapabilme vb. birçok özelliği sebebiyle tercih edilebilir bir yöntemdir.

BÖLÜM 2. SİSTEM MİMARİSİ

Sistemimizi olabildikçe maliyetsiz olması için tasarladık bunun için visual studio'nun ücretsiz versiyonunu, veritabanı, pipeline ve görselleştirme için açık kaynak ücretsiz programları kullandık. İşletim sistemini üniversitenin bize sağladığı microsoft hesabından ücretsiz olarak alabileceğimiz windows işletim sistemi üzerine kurduk.

2.1. Donanım Mimarisi

Bizim projemiz iki adet program ve ELK-stack yapısından oluşacak. Bu iki programdan biri SNMP protokolü ile ağ cihazlarından verileri çekip ELK-stack yapısına gönderecek, diğer program ise makine öğrenmesini yaptığımız yer olacak. Bu program da ELK-Stack yapısından belli aralıklarla eklenen verileri çekecek, bunları makineye öğretecek ve olası alarm durumlarında alarm üretme işini bu program yapacak.



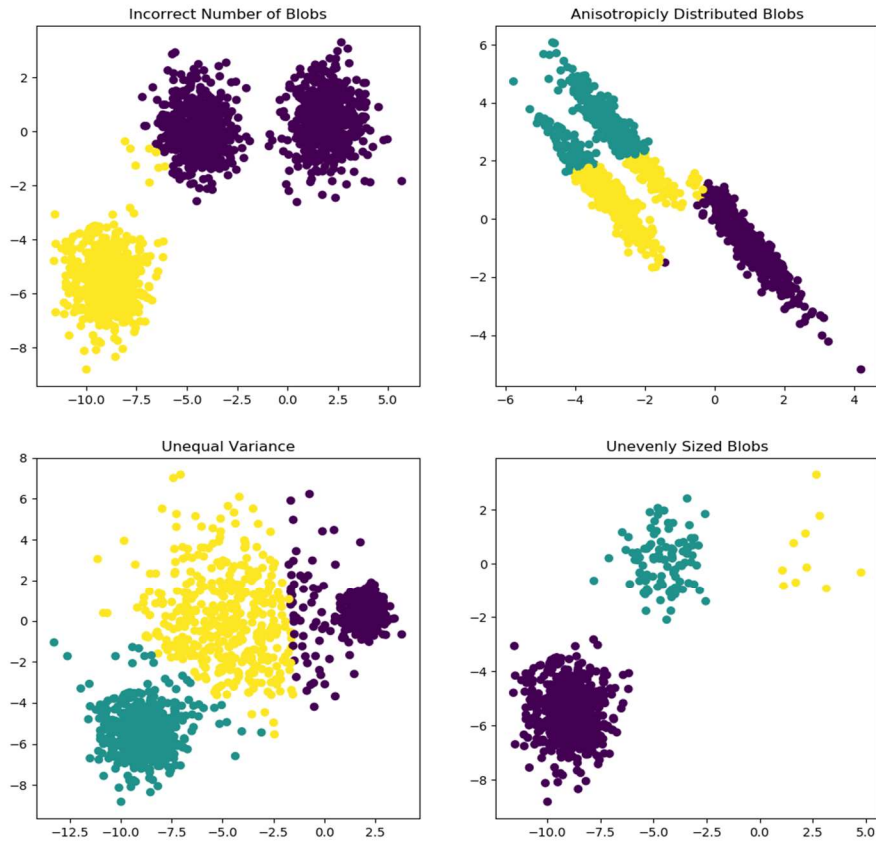
Şekil 2.1. Sistemin Çalışma Şekli

Şekilde gördüğünüz numaralar bizim iletişim için kullandığımız port numaralarıdır, bu port numaralarından bazıları güvenlik nedeniyle default değerlerinden değiştirilmiştir bu değişiklikler Logstash, Kibana ve Elasticsearch programlarını indirdiğimiz konfigürasyon dosyaları içerisinde yapılmaktadır. Bu dosyalar, içerisinde çalıştığı serverın ip adresi gibi birçok bilgiyi verebiliriz. Biz localhost kullandığımız için localhost adresini verdik.

2.2. Yazılım Mimarisi

SNMP bilgilerini almak için Spyder ide'sinde python dilini kullanarak PySNMP kütüphanesi aracılığı ile SNMP verilerini yönlendirici üzerinden aldık.

Makine öğrenmesi modelimizi Spyder ide'sinde python dilini kullanarak clustering metodlarından K-means algoritmasını kullandık. K-means algoritması veriler içerisinde bazı küme merkezleri belirler ve bu küme merkezlerinin çevresindeki veriler ile bir küme oluşturur kaç adet küme merkezi (bu aynı zamanda kaç adet kümemiz olacağı anlamına gelir) biz belirliyoruz ve bu algoritma merkez noktaları bizim verdiğimiz K-means++ yöntem ile yapmaktadır. **K-Means++**'a göre rastgele başlangıç noktası seçildikten sonra diğer tüm veriler ile arasındaki mesafe hesaplanır. Bu mesafenin karesi alınıp belli hesaplamalar yaparak yeni başlangıç noktaları seçilir.[9]



Şekil 2.2.1. K-means++ Örnek Kümeleme[10]

BÖLÜM 3. MODEL

Verileri topladığımız sistemi Sakarya Üniversitesi Bilgisayar Müh. Fakültesi içerisinde bulunan Siber Güvenlik Lab. İçerisinde kurduğumuz sistem için 3 adet router kullandık bunlar cisco marka 2500 ve 2600 serisi routerları kullandık. Routerlar günümüzdeki sistemlere göre eski oldukları için SNMP protokolünün son versiyonunu kullanmadık ve bazı bilgileri alamadık bu yüzden aldığımız bilgiler içerisinden bazı kısımları çıkardık, SNMP verilerini toplamak ve veri akışını sağlamak için iki adet dizüstü bilgisayar kullandık. Verileri ilk önce c# dilini

kullanarak aldık ancak istediğimiz şekilde sağlıklı çalışmadı bu yüzden python dilini kullanarak tekrar bir program yazdık.

3.1. SNMP Verilerinin Alınması

Python içerisindeki pysnmp kütüphanesi aracılığı ile verileri router üzerinden çektik ilk başta tablo 3.1 üzerinde gözüken bütün bilgileri almaya çalıştık ancak bazı bilgileri alırken sıkıntı yaşadığımız için eğitim kısmında bazılarını kullanmadık.

Tablo 3.1. SNMP OIDs ve İsimleri[8]

OIDs	NAME
1.3.6.1.2.1.31.1.1.1.6	ifHCInOctets
1.3.6.1.2.1.31.1.1.1.10	ifHCOctets
1.3.6.1.2.1.31.1.1.1.7	ifHCInUcastPkts
1.3.6.1.2.1.31.1.1.1.11	ifHCOctetsUcastPkts
1.3.6.1.2.1.31.1.1.1.8	ifHCInMulticastPkts
1.3.6.1.2.1.31.1.1.1.12	ifHCOctetsMulticastPkts
1.3.6.1.2.1.31.1.1.1.9	ifHCInBroadcastPkts
1.3.6.1.2.1.31.1.1.1.13	ifHCOctetsBroadcastPkts
1.3.6.1.2.1.2.2.1.13	ifInDiscards
1.3.6.1.2.1.2.2.1.19	ifOutDiscards
1.3.6.1.2.1.6.7	tcpAttemptFails
1.3.6.1.2.1.6.17	tcpHCInSegs
1.3.6.1.2.1.6.18	tcpHCOctetsSegs
1.3.6.1.2.1.4.3	ipInReceives
1.3.6.1.2.1.4.9	ipInDelivers
1.3.6.1.2.1.4.8	ipInDiscards
1.3.6.1.2.1.5.1	icmpInMsgs
1.3.6.1.2.1.5.14	icmpOutMsgs
1.3.6.1.2.1.7.8	udpHCInDatagrams
1.3.6.1.2.1.7.9	udpHCOctetsDatagrams
1.3.6.1.2.1.7.3	udpInErrors

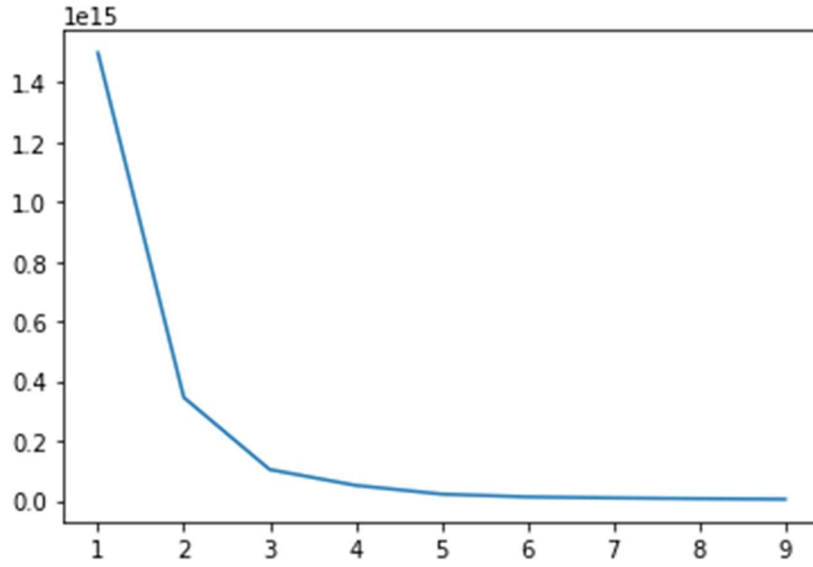
Verileri 6 ile 12 saniye aralığında aldık bunun nedeni kullandığımız bilgisayarın işlem gücü iyi olmadığı için belli bir bekleme süresi vermedik ve işlem bittimi diğeri başla şeklinde oluşturduk bunun nedeni çektiğimiz veriler arasında zaman farkının fazla olmasını istemememiz. Aldığımız verileri Json tipine getirdik ve urllib.request kütüphanesi ile belirttiğimiz adresin portuna Json verimizi yolladık bizim sistemimizde localhost:31311 adresi ve portunu kullandık. Bu adres ve portun diğeri tarafında Logstash servisi çalışıyor aldığı verileri bizim oluşturduğumuz konfigürasyon dosyası ile Elasticsearch veri tabanına atıyor. Logstash kullanmamızın nedeni veri toplama ve uygulama kısımlarında istediğimiz şekilde yönlendirebildiğimiz için örneğin tek adrese yollanan veriyi hem elasticsearch veritabanına kaydedebiliriz hemde makina öğrenmesi algoritmamızın çalıştığı uygulamaya gönderebiliriz.

Makina öğrenmesi algoritmasının tahmin edeceği verileri almak için tcp socket yazdık buraya gelen verileri tahmin metoduna yolladık ve sonucu konsol ekranına yazdırdık.

3.2. Verilerin Hazırlanması Ve Makinenin Eğitilmesi

Verilerimizin bulunduğu elasticsearch veritabanından pyelasticsearch kütüphanesini kullanarak verilerimizi çektik ve bunları bir liste veri yapısının içine attık. Aldığımız veriler MIB ağacında bulunduğu şekilde aldığımız için üstünde işlem yapmamız gerekti. Veri işleminden geçtikten sonra veriyi bir data frame içerisine aldık.

K-means++ algoritması kaç adet küme oluşturacağını bize sorar bizde bunun için WCSS değerlerini kullanırız WCSS değeri bizim her kümenin elemanlarının merkeze olan uzaklığının mesafesinin karesi bu şekilde verilerimiz merkezden uzaklaştıkça karesel olarak mesafe artıyor .1 ile 10 arasındaki küme sayısı ile WCSS değerlerini hesaplattık ve bunu şekil 3.2 de görüldüğü gibi bir grafik elde ettik.



Şekli 3.2. WCSS Değer Grafiği

Bu grafikten görüldüğü gibi verimiz 2 yerde kırılıyor bunun anlamı temel olarak verimizin kaç gruba ayrıldığını gösteriyor. Grafik yardımı ile küme sayımızı 3 olarak belirledik ve python sklearn.cluster kütüphanesi yardımı ile makinamızı eğittik sonuçlara baktığımızda verimiz normal iletim , saldırı ve bozuk veri olarak 3 parçaya ayrıldığını gördük ancak bizim amacımız normal iletim verilerini düşük, orta ve yüksek olarak 3 parçaya ayırmak istiyorduk bu yüzden 3 küme kullanımı bizim ihtiyacımızı karşılamadı. Grafiğe tekrar baktığımızda 5 küme sayısında hafif bir kırılma olduğunu gördük ve küme sayımızı 5'e çıkardık bunun sonucunda istediğimize ulaştık normal iletim verilerimiz hızlarına göre gruplara ayrıldı. Bundan sonra saat aralıkları belirledik kolay olması için 24:00 – 08:00 arasını ve 08:00 – 24:00 aralıklarını seçtik ve bu aralıklar içerisindeki verilerin hangi gruplara üye olduğunu bulduk.

BÖLÜM 4. SONUÇLAR VE ÖNERİLER

Bitirme projemizdeki amacımız SNMP protokolü ile aldığımız verileri makine öğrenmesi algoritmaları yardımı ile sisteme öğretmek ve anormallikleri raporlamak idi.

Verileri aldık ve makinayı eğittik, makinayı eğitirken verileri test ve eğitim olarak bölmedik verilerin hepsi ile makinayı eğittik ve testleri elimizle yaptık sonuçlar olumlu çıktı saat aralıklarına göre anormallik tespiti yapabildik.

Sistemimiz genel kullanım kütlesi hesaplayamıyor örneğin her gün 30 gigabaytlık bir aktarım yapılıyor ancak bir gün yapılmadı bu bir anormalliktir ancak bizim sistemimiz verileri hangi gruba dahil olduğuna baktığı için toplam iletim kütlesi ile ilgili anormallikleri şu an tespit edemiyor.

Projemizi gelecekte toplam iletim kütlesi sorununu çözeceğiz aynı zamanda bizim projemiz sadece bizim kurduğumuz sistem üzerinde çalışıyor bunu genel çalışabilir duruma getireceğiz.

KAYNAKLAR

- [1] <http://kod5.org/basit-ag-yonetim-protokolu-snmp-nedir/>
- [2] <https://www.snmpsharpnet.com/>
- [3] <https://aws.amazon.com/tr/elasticsearch-service/what-is-elasticsearch/>
- [4] <https://stackoverflow.com/questions/47003336/need-elasticsearch-index-sharding-explanation?rq=1>.
- [5] <https://www.elastic.co/products/logstash>
- [6] <https://www.emrealadag.com/makine-ogrenmesi-nedir.html>
- [7] <https://ourworldindata.org/internet>
- [8] <http://cric.grenoble.cnrs.fr/Administrateurs/Outils/MIBS/?oid=1.3.6.1.2.1>
- [9] <https://medium.com/@ekrem.hatipoglu/machine-learning-clustering-k%C3%BCmeleme-k-means-algorithm-part-13-be33aeef4fc8>
- [10] <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans>

ÖZGEÇMİŞ

Burak Düz, 22.03.1996 de İstanbul'da doğdu. İlk ve orta eğitimini Gebzede tamamladı. 2014 yılında Deniz Yıldızları Teknik Ve Endüstri Meslek Lisesi, Bilgisayar Bölümünden mezun oldu. 2014 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümünü kazandı. 2018 yılında Logo Yazılım Şirketinde yazılım ve donanım stajını yapmıştır. SAÜ Bilgisayar Mühendisliği Bölümünden 2019 yılında mezun olmuştur.

Enes Kocaman, 17.10.1997 tarihinde İstanbul'da doğdu. İlk ve orta öğrenimini Beyoğlu ilçesinde tamamladı. 2015 yılında Güner Akın Lisesinden mezun oldu. 2015 yılı Eylül ayında Sakarya Üniversitesi, Bilgisayar Mühendisliği bölümünü kazandı. 2016 yılının yaz döneminde Kariyer.net Ar-Ge departmanında 3 ay süreyle, 2017 yılında BBS und Montage Industrietechnik GmbH şirketinde 1 ay süreyle, aynı yıl Kariyer.net Ar-Ge bünyesinde 2 ay süreyle ve 2018 yılında 28Apps Software GmbH firmasında 3 ay süreyle yazılım stajyeri olarak çalışmıştır. Eğitimini sürdürdüğü Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünden 2019 yılında mezun olmuştur.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU :

ÖĞRENCİLER (Öğrenci No/AD/SOYAD):

Değerlendirme Konusu	İstenenler	Not Aralığı 1	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN :

DANIŞMAN IMZASI: