
GAZİ ÜNİVERSİTESİ
Gazi
1926

AKILLI ÇİFTLİKLER İÇİN BÜYÜK VERİ ANALİZİ

Duygu Nazife ZARALI

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ŞUBAT 2018

Duygu Nazife ZARALI tarafından hazırlanan "AKILLI ÇİFTLİKLER İÇİN BÜYÜK VERİ ANALİZİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Hacer KARACAN

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……

Başkan: Prof. Dr. M. Ali AKCAYOL

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ……...

Üye: Doç. Dr. Osman ABUL

Bilgisayar Mühendisliği Anabilim Dalı, TOBB ETÜ

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ……...

Tez Savunma Tarihi: 08/02/2018

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..…….

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Duygu Nazife ZARALI

08/02/2018

AKILLI ÇİFTLİKLER İÇİN BÜYÜK VERİ ANALİZİ

(Yüksek Lisans Tezi)

Duygu Nazife ZARALI

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2018

ÖZET

Akıllı sistemler genel olarak, verileri toplama, analiz etme ve diğer sistemler ile iletişim kurma kapasitesine, belirli bir derecede zekaya sahip olan gömülü bilgisayar veya kontrol edilen sistemleri, makineleri ve cihazları ifade eder. Çiftçilerin çiftliklerini kolaylıkla yönetmelerine olanak tanıyan akıllı teknolojilerdeki gelişmeler, akıllı çiftlik kavramını ortaya çıkarmaktadır. Bu teknolojiler, süt çiftçiliğine harcanan emeğe ve insan-hayvan etkileşimine olan ihtiyacı azaltır. Bu teknolojilerin çiftliklerde uygulanmasıyla çiftlik verilerinin miktarı ve kapsamı artmakta ve çiftçilik işlemleri giderek veri odaklı ve veri etkin hale gelmektedir. Bu nedenle büyük veri analizi, çiftlik işlemlerinde tahmine dayalı bilgiler sunmak, gerçek zamanlı operasyonel kararlar vermek için kullanılmaktadır. Büyük verilerin analizi, depolama, analiz ve görselleştirme güçlüklerini beraberinde getiren daha değişken ve karmaşık yapıları işlemek için algoritmaların kullanılmasını gerektirir. Büyük hacimli ve giderek büyüyen veri kümelerinin işlenmesi gereksinimini karşılamak için farklı depolama ve işleme yöntemleri uygulanmaktadır. Apache Spark, büyük veri işlemeyi daha kolay ve hızlı hale getiren etkili bir dağıtık veri işleme motorudur. Spark, ölçeklenebilir makine öğrenmesi, grafik analizi, akan ve yapısal veri işleme için geliştirilen bellek içi programlama modeline ve üst düzey kütüphanelere sahiptir. Bu tezde akıllı çiftliklerden elde edilen büyük ölçekli veriler analiz edilmiş ve sıralı örüntü madenciliği algoritması olan PrefixSpan, Apache Spark makine öğrenme kütüphanesi kullanılarak uygulanmıştır. Geçmiş verilerinin analiz edilmesiyle, sorunların ana kaynakları tahmin edilebilir ve ortaya çıkabilecek muhtemel problemler ortadan kaldırılabilir durumda olacaktır. Bu analizle, sistemlerde meydana gelebilecek arızaların erken tespit edilmesi ve bakım işlemlerinin buna göre yönetilmesiyle önemli maliyetlerin en aza indirgenmesi mümkün olabilir.

Bilim Kodu : 92414

Anahtar Kelimeler : Büyük veri analizi, Apache Spark, sıralı örüntü madenciliği, PrefixSpan algoritması, akıllı çiftlik

Sayfa Adedi : 69

Danışman : Doç. Dr. Hacer KARACAN

BIG DATA ANALYSIS FOR SMART FARMING

(M. Sc. Thesis)

Duygu Nazife ZARALI

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

February 2018

ABSTRACT

Intelligent systems refer broadly to computer embedded or controlled systems, machines and devices that possess a certain degree of intelligence with the capacity to gather and analyze data and communicate with other systems. Developments of intelligent technologies that allow farmers to manage farms with ease has introduced the smart farming concept. These technologies take over the labor of dairy farming and reduce the need for human-animal interactions. As implementation of these technologies on farms, farm data grow in quantity and scope and farming processes become increasingly data-driven and data-enabled. Therefore, big data analysis is being used to provide predictive insights in farming operations, drive real-time operational decisions. The analysis of big data requires algorithms to handle more varied and complex structures with difficulties in storing, analyzing and visualizing. Different storage and processing methods are implemented to meet the requirement of processing large-volume, growing datasets. Apache Spark is an effective distributed data processing engine that makes big data processing easier and faster. Spark has advanced in-memory programming model and upper-level libraries for scalable machine learning, graph analysis, streaming and structured data processing. In this thesis, large scale data obtained from intelligent farms, are analyzed and PrefixSpan, a sequential pattern mining algorithm, is implemented using Apache Spark machine learning library. Based on the analysis of the past data, the main sources of the problems could be predicted and the possible problems that may arise can be eliminated. With this analysis, it can be possible to minimize significant costs by early detection of failures that may occur in systems and management of maintenance processes accordingly.

Science Code : 92414

Key Words : Big data analysis, Apache Spark, sequential pattern mining, PrefixSpan algorithm, smart farming

Page Number : 69

Supervisor : Prof. Dr. Hacer KARACAN

TEŞEKKÜR

Çalışmalarım boyunca yardım ve katkılarıyla beni yönlendiren ve şahsi olarak örnek aldığım değerli hocam Doç. Dr. Hacer KARACAN'a, Triodor Ar-Ge ekibine, Gazi Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümü öğretim üyelerimize, araştırma görevlisi arkadaşlarıma ve idari kadromuza, Gazi Üniversitesi Fen Bilimleri Enstitüsü bünyesindeki idari ve akademik kadromuza, her alandaki destekleriyle beni hiç yalnız bırakmayan eşim Gökay ZARALI'ya ve kıymetli aileme yürekten bir teşekkürü borç bilirim.

İÇİNDEKİLER

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
DAG	Directed Acyclic Graph
HDFS	Hadoop Distributed File System
MLlib	Machine Learning Library
SQL	Structured Query Language
RDD	Resilient Distributed Dataset
YARN	Yet Another Resource Negotiator

1. GİRİŞ

Akıllı çiftlik, çiftlik yönetiminde bilgi ve iletişim teknolojisinin kullanımını vurgulayan bir gelişmedir ve Nesnelerin İnterneti, Bulut Bilişim gibi yeni teknolojiler bu gelişimi güçlendirmektedir [53].

Akıllı çiftlik, robotik veya otomatik sağım, yemleme, temizlik vb. sistemlerinden oluşur ve bu sistemler süt çiftçiliğine harcanan emeğe ve insan-hayvan etkileşimine olan ihtiyacı azaltacak yeni teknolojilerdir [20]. Robotik veya otomatik sağım, yemleme, temizlik sistemlerinin geliştirilmesinin başlıca nedeni birçok süt ülkesinde artan emek maliyetlerine bağlı olarak emek verimi geliştirilmesine ihtiyaç duyulmasıdır [12].

Akıllı makineler ve algılayıcılar çiftliklerde kullanılır hale geldikçe ve çiftlik verileri miktarı ve kapsamı artacaktır. Nesnelerin İnterneti ve Bulut Bilişim'deki hızlı gelişmeler, Akıllı çiftlik adı verilen olguyu yönlendirmektedir [49].

Çiftçilikte robot ve yapay zekanın daha fazla kullanılması beklenmektedir. Büyük veri teknolojileri bu gelişmede karşılıklı ve önemli bir rol oynamaktadır. Büyük veri, çiftlik işlemlerinde öngörücü bilgiler sunmak, gerçek zamanlı operasyonel kararlar almak ve iş süreçlerini yeniden tasarlamak için kullanılmaktadır [53]. Verilerden anlamlı çıkarımlar elde etmek ve elde edilen sonuçlar doğrultusunda kararlar almak için akış hızı ve boyutu yüksek olan büyük verinin işlenmesi gerekir.

Bu tez çalışmasında akıllı çiftliklerden elde edilen büyük ölçekli veriler, büyük veri işleme teknolojisi olan Apache Spark platformu kullanılarak sayısal ve oransal bazda analiz edilmiş ve bu veriler üzerinde sıralı örüntü madenciliği (sequential pattern mining) algoritması olan PrefixSpan uygulanmıştır. Yapılan çalışmada akıllı çiftliklerde çalışan robotların çeşitli sebeplerle verdiği alarmlar analiz edilmiştir. Geçmiş alarm bilgilerinin işlenmesiyle alarmlar arasındaki ilişkilerin ve örüntülerin bulunarak alarmların temel kaynaklarına inilmesi çalışmanın temel amacıdır. Bu çalışmada Nesnelerin İnterneti teknolojilerini kullanan birçok akıllı sistemde olduğu gibi akıllı çiftliklerde de büyük veri kavramının yaygınlaştığı, bu verilerin işlenmesiyle akıllı çiftliklerin üretim ve maliyet gibi durumları üzerinde değişiklik sağlanabileceği, ayrıca sıralı örüntü madenciliği

algoritmalarının da büyük veri teknolojileri ile büyük veri işlemede kullanılabileceği ifade edilmiştir.

Tez çalışması, tezin konusunun ve planlamasının özetlendiği giriş bölümü ile birlikte altı bölümden oluşmaktadır. Büyük veri kavramı ve büyük verinin özellikleri, kaynakları, zorlukları, analizi, teknolojileri ikinci bölümde açıklanmıştır. Veri analizi ve çalışmada kullanılan sıralı örüntü madenciliği ve algoritmaları üçüncü bölümde verilmiştir. Yapılan çalışma ile benzer olan akademik çalışmalar dördüncü bölümde verilmiş ve yapılan çalışmada kullanılan veri setinin detayları, problem tanımı ve çalışmanın adımları ile sonuçları beşinci bölümde yer almıştır. Son olarak altıncı bölümde çalışmanın kısa bir özeti ve sonuçlarının yorumları bulunmaktadır.

2. BÜYÜK VERİ

Büyük veri kavramı, geleneksel veri tabanı yöntemleri ve araçları kullanılarak verimli bir şekilde işlenemeyen yüksek hacimli veriyi ifade eder [37]. Daha detaylı bir tanımla büyük veri; karar alma, anlam çıkarma ve işlem optimizasyonunu artırma için yeni işleme yöntemleri gerektiren yüksek hacimli, yüksek hızlı ve / veya yüksek çeşitlilikli bilgi varlıklarıdır [23]. Bir veri kümesinin mevcut teknolojilerle yakalanması, toplanması, iyileştirilmesi, analizi ve görselleştirilmesi zor olduğunda bu veri kümesi büyük veri olarak adlandırılabilir. Genel anlamıyla ise çok sayıda çeşitlilik gösteren çok büyük veri kümeleri topluluğudur [10].

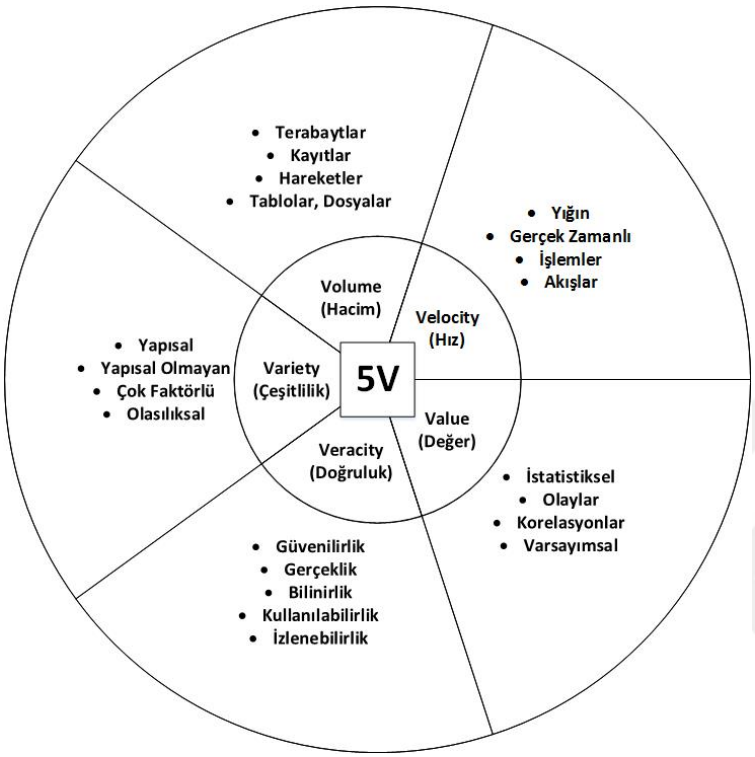
Kurum ve kuruluşlarda, analog teknolojiler yerine dijital teknolojilerin kullanımının artması, internete bağlı cihazların ve sistemlerin artması, sosyal medya kullanıcılarının ve kullanımının artması büyük veri kavramını ortaya çıkarmıştır [37].

Büyük veri, verilerden anlam çıkarmaya ve elde edilen sonuçlar doğrultusunda karar almaya yardımcı olması sebebiyle değerlidir. Dolayısıyla anlamlı çıkarımlar elde edilmesi için akış hızı ve boyutu yüksek olan büyük verinin işlenmesi gerekir. Büyük veri işleme yöntemleri temel olarak veri yönetimi ve veri analizi olmak üzere ikiye ayrılır. Büyük veri yönetimi verinin toplanması, kaydedilmesi, çıkarılması, temizlenmesi, açıklanması, bütünleştirilmesi, kümelenmesi aşamalarından oluşur ve veri analize hazır hale getirilir. Büyük veri analizi verinin modellenmesi, analiz edilmesi ve yorumlanması aşamalarından oluşur. Büyük veri analizi, karar odaklı ve eylem odaklı olarak ayrılabilir. Karar odaklı analiz, büyük veri kaynaklarının belirli bir kısmının analiz edildiği ve elde edilen sonuçların iş konularında kararlar almada kullanıldığı durumları içerir. Eylem odaklı analiz, verinin belirli bir türü veya örüntüsü tespit edildiğinde hızlı cevap verilmesini ve bir işlem yapılmasını, bir eylemde bulunulmasını gerektiren durumları içerir [51].

2.1. Büyük Verinin Özellikleri

Büyük verinin özellikleri, geleneksel olarak 3V (volume, velocity, variety) ile ifade edilirken [23] güncel kaynaklarda genellikle 5V (volume, velocity, variety, veracity, value) ile verilmektedir [48]. Bu özelliklerden hacim (volume) verinin boyutunu, hız (velocity)

verilerin akış hızını, çeşitlilik (variety) heterojen veri tiplerini, doğruluk (veracity) verilerin tutarlılığını ve güvenilirliğini, değer (value) veri kümelerinden elde edilen kazanımları ifade eder [49].



Şekil 2.1. Büyük verinin özellikleri [13]

2.1.1. Hacim/büyüklük (volume)

Büyük verinin büyük olarak adlandırılması bu verilerin ne kadar yüksek hacimlere sahip olduğu konusuna açıklık getirmektedir. Büyük veri hacmi; verinin büyüklük, boyut, ölçek ve miktar özelliklerini kapsar. Terabayt (ondalık gösterimde 10^{12}), eksabayt (ondalık gösterimde 10^{18}) boyutlarında veriler kaydedilmekte ve tüm bu verilerin erişilebilir, aranabilir, işlenmiş ve yönetilebilir olması gerekir. Günümüzdeki veriler petabayt (ondalık gösterimde 10^{15}) boyutunda iken yakın gelecekte verilerin zettabayt (ondalık gösterimde 10^{21}) boyutuna çıkması beklenmektedir [13,36].

2.1.2. Hız (velocity)

Büyük verinin hızı, çeşitli kaynaklardan gelen verilerin hızıyla ilgili bir kavramdır. Bu özellik, verilerin üretim hızını veya iletim hızını ifade eder. Veriler üretildiği hızda

yakalanmazsa çok sayıda veri kaybı yaşanır. Yüksek hızda veri üreten kaynaklara örnek olarak log dosyaları, sosyal medya, bilgi işlem cihazları, çevrim içi oyunlar verilebilir. Yüksek hızdaki veriler zamanla büyük veri kavramını oluşturur. Örneğin algılayıcı cihazlardan gelen veriler sürekli olarak veri tabanına taşınacak ve bu miktar küçük olmayacaktır. Dolayısıyla geleneksel sistemler, gerçek zamanlı verilere dair analitik bilgi verme konusunda yeterli değildir [36, 40].

2.1.3. Çeşitlilik (variety)

Veriler yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış olarak sınıflandırılır. Web sayfaları, log dosyaları, sosyal medya siteleri, e-posta, dokümanlar, algılayıcı cihazlar gibi çeşitli kaynaklardan gelen veriler metin, görüntü, ses, video vb. olabilir. Tüm bu veriler, işlenebilmeden önce ortak bir biçime dönüştürülmelidir. Bu kadar çok çeşitli verinin mevcut analitik sistemler ve yöntemler kullanılarak işlenmesi zordur [36, 40].

2.1.4. Doğruluk (veracity)

Büyük verinin doğruluk boyutu, veri tutarlılığı (veya kesinliği) ve veri güvenilirliği olmak üzere iki yönden incelenir. Doğruluk özelliği verinin kaynağının, toplanma ve işlenme yöntemlerinin, depolanmasının da güvenilir olmasını gerektirir. Kullanılan verilerin yetkisiz erişime ve değiştirilmeye karşı korunmasını sağlar. Verinin doğruluğunun mümkün olması için;

- Verilerin ve bağlı verilerin bütünlüğü,
- Veri orijinalliği ve kaynağının güvenilirliği,
- Hem verilerin hem de kaynağın tanımlanması,
- Bilgisayar ve depolama platformunun güvenilirliği,
- Kullanılabilirlik ve zamanlamanın doğruluğu

sağlanmalıdır [13].

2.1.5. Değer (value)

Veri işleme, büyük miktarda çeşitli veri toplayan organizasyonların temel kaygılarından biridir. İşlenmemiş veriler nispeten sınırlı bir değere sahiptir. Veri sahipleri için birikmiş verilerden ek değerler çıkarılması ve türetilmesi önemlidir. Depolanan veriler üzerinde

belirli sorguların çalıştırılmasıyla elde edilen filtrelenmiş verilerden önemli sonuçlar çıkarılabilir. Bilgi birikimi ve kullanılabilir bilgi elde edilmesi için verilerin analiz edilmesi gerekir [16, 36].

2.2. Büyük Verinin Kaynakları

Veri üretimi büyük verinin ilk adımıdır. Günümüzde büyük verilerin ana kaynakları, işletmelerdeki işlem ve ticaret bilgileri, Nesnelerin İnterneti'ndeki lojistik ve algılama bilgileri, İnternet dünyasındaki insan etkileşimi bilgileri, konum bilgisi ve bilimsel araştırmalarda üretilen verilerdir. Doğada yaygın olan algılama ve bilgi işlem, ticari, İnternet, hükümet ve sosyal ortamlar benzeri görülmemiş karmaşıklıkla heterojen veri üretmektedir. Bilimsel uygulamalar arttıkça, veri kümesinin ölçeği giderek genişlemekte ve bazı disiplinlerin geliştirilmesi büyük ölçüde veri yığınlarının analizine dayanmaktadır [9].

Çizelge 2.1. Büyük Veri kaynakları ve kullanım alanları [14]

Büyük Veri Kaynağı	Kullanım Alanları
1. Bilim	a. Bilimsel Keşifler
2. Telekom	b. Yeni Teknolojiler
3. Endüstri	c. İmalat, Süreç Kontrolü, Nakliye
4. İşletme	d. Kişisel Hizmetler
5. Yaşam Çevresi, Şehirler	e. Yaşam Çevresi Desteği
6. Sağlık Hizmetleri	f. Sağlık Hizmetleri Desteği
7. Sosyal Medya ve Sosyal Ağlar	

İnternete bağlanan fiziksel nesnelerin sayısının giderek artması Nesnelerin İnterneti kavramını ortaya çıkarmaktadır. Nesnelerin İnterneti, her yerde bulunan ve yaygın olan bilgisayar, gömülü aygıtlar, iletişim teknolojileri, algılayıcı ağları, İnternet protokolleri ve uygulamaları gibi temel teknolojileri kullanarak bu fiziksel nesneleri birbirleriyle haberleşebilen ve organize olarak çalışabilen akıllı nesnelere dönüştürür [5].

Nesnelerin İnterneti dünyasında sürekli olarak çok miktarda ham veri toplanmaktadır ve bu verilerin kullanılabilir bilgiye dönüştürülmesi gerekir [47]. Algılayıcılarla donatılmış çok sayıda fiziksel nesnenin internete bağlanması, "büyük veri" kavramını üretir. Birbirine

bağlı cihazların ürettiği bu verilerin toplanması, depolanması ve işlenmesi için bir mekanizma gereklidir. Ancak yaygın olarak kullanılan donanım ve yazılım araçları, kabul edilebilir bir zaman diliminde verilerin yakalanması, yönetilmesi ve işlenmesi için yeterli değildir. Büyük veri analizi için Apache, Hadoop ve SciDB gibi platformlar bulunmaktadır. Nesnelerin İnterneti verilerinin miktarı mevcut araçlar tarafından beslenip işlenemeyecek kadar büyüktür ve bu platformlar, kullanıcılara verimli bir şekilde hizmet vermek için gerçek zamanlı olarak çalışmalıdır [5].

2.3. Büyük Verinin Zorlukları

Büyük veri çağında hızla artan veri yoğunluğu; veri toplama, depolama, yönetim ve analiz konusunda büyük zorluklar doğuruyor. Geleneksel veri yönetimi ve analiz sistemleri, ilişkisel veri tabanı yönetim sistemine dayanmaktadır. Bununla birlikte, ilişkisel veri tabanı yönetim sistemleri yalnızca yapılandırılmış veriler için geçerlidir. Geleneksel ilişkisel veri tabanı yönetim sistemleri büyük verilerin büyük hacim ve heterojenliğini idare edememektedir [9].

Büyük verinin sahip olduğu özellikler ve hassasiyet, teknik, hukuksal vb. açılardan önem arz etmesi nedeniyle birçok yönden zorlukları bulunmaktadır. Verilerin büyük hacimlerde olması erişim, depolama, arama yapma, işleme, analiz aşamalarını; verilerin üretim ve akış hızının yüksek olması yakalama ve işleme aşamalarını; çeşitlilik özelliği işleme ve analiz aşamalarını zorlaştırmaktadır [41]. Bu zorluklar, veri yaşam döngüsüne dayalı olarak üç temel kategoriye ayrılabilir [4]:

- Veriden kaynaklanan zorluklar; verilerin hacim (volume), hız (velocity), çeşitlilik (variety), doğruluk (veracity), değer (value), değişkenlik (variability), görselleştirme (visualisation) gibi özellikleri sebebiyledir.
- Veri işleme sürecindeki zorluklar; verilerin toplanması ve depolanması, temizlenmesi ve ön işlemeden geçirilmesi, birleştirilmesi ve bütünleştirilmesi, analiz için doğru modelin seçilmesi ve sonuçların yorumlanması aşamalarıyla ilişkilidir.
- Yönetimsel zorluklar; gizlilik, güvenlik, veri mülkiyeti, veri ve bilgi paylaşımı, maliyet gibi konuları kapsar.

2.4. Büyük Veri Analizi Teknolojileri

Büyük veri kavramı verilerin işlenmesi, analizi ve depolanması için yeni nesil araçlara ve veri tabanlarına ihtiyaç duymaktadır. Bu ihtiyaç doğrultusunda dağıtık sistemler üzerinde çalışan yeni analiz yazılımları ve veri tabanı programları ortaya çıkmıştır. HDFS dosyalama sistemini ortaya çıkaran ve üzerinde MapReduce programlama modeli kullanılmasına olanak sağlayan Apache Hadoop [24], Apache Spark [25] gibi teknolojileridir. Bu teknolojileri kullanan Mahout [26] gibi veri madenciliği yazılımları ve sorgulama yapılmasını mümkün kılan Hive [27], Pig [28], Drill [29] gibi yazılımlar ortaya çıkmıştır. Bunların dışında büyük verilerin depolanması ve performanslı sorgulama yapılmasını sağlayan MongoDB [30], HBase [31], Cassandra [32] gibi NoSQL [33] veri tabanları büyük veri dünyasına kazandırılmıştır.

2.4.1. Apache Hadoop

Hadoop, Apache Software Foundation tarafından sağlanan bir açık kaynak projesidir. Hadoop, sıradan sunuculardan (commodity servers) oluşan küme (cluster) üzerinde büyük verileri işlemek amaçlı uygulamaları çalıştıran, Java ile geliştirilmiş açık kaynaklı bir kütüphanedir. Hadoop mimarisi dört bileşenden oluşur [36, 41]:

- Hadoop Distributed File System (HDFS): Küme genelinde çok yüksek bant genişliği sağlamak için verileri birden çok sunucu üzerinde depolayan dağıtık dosya sistemidir.
- MapReduce: Büyük ölçekli verilerin paralel olarak işlenmesini sağlar.
- Hadoop YARN (Yet Another Resource Negotiator): Kümedeki kaynakların yönetimini ve iş planlamasını gerçekleştirir.
- Hadoop Common: Diğer Hadoop modüllerinin gerektirdiği kütüphanelerin ve araçların bulunduğu bir araç kutusudur.

<u>Hadoop dağıtık dosya sistemi</u>

Hadoop dağıtık dosya sistemi (Hadoop Distributed File System - HDFS), sıradan sunuculardaki kümeler üzerinde çalışan ve çok büyük dosyaları depolamak için tasarlanmış bir dosya sistemidir. HDFS blok boyutu, normal dosya sisteminkinden çok

daha büyüktür. Bu büyük blok boyutunun nedeni, disk araması sayısını azaltmaktır. HDFS kümelerinde NameNode ve DataNode olmak üzere iki tip düğüm vardır [36].

NameNode ana (master) süreç olarak blokların sunucular üzerindeki dağılımından, yaratılmasından, silinmesinden, bir blokta sorun meydana geldiğinde yeniden oluşturulmasından ve her türlü dosya erişiminden sorumludur. NameNode olmadan dosyaya erişmek mümkün değildir. Bu nedenle, hatalara karşı dayanıklı olması çok önemlidir. HDFS üzerindeki tüm dosyalar hakkındaki bilgiler (metadata) NameNode tarafından saklanır ve yönetilir. Her kümede yalnızca bir adet NameNode olabilir. HDFS ad alanı (namespace), dosya ve dizinlerin bir hiyerarşisidir. Dosyalar ve dizinler, izin, değiştirme ve erişim zamanı, ad alanı ve disk alanı kotaları gibi nitelikleri kaydeden NameNode'da temsil edilir. Dosya içeriği büyük bloklara ayrılır (Genellikle 128 megabayttır, ancak kullanıcı tarafından belirlenebilir.) ve dosyanın her bloğu birden çok DataNode'da (Genellikle üçtür, ancak kullanıcı tarafından belirlenebilir.) çoğaltılır [36, 45].

DataNode ise işlevi blokları saklamak olan işçi (worker) süreçtir. Her DataNode kendi yerel diskindeki veriden sorumludur. Ayrıca diğer DataNode'lardaki verilerin yedeklerini de barındırır. DataNode'lar küme içerisinde birden fazla olabilir. Bir DataNode üzerindeki her blok kopyası ana sunucu üzerinde iki dosya ile gösterilir. İlk dosya verinin kendisini içerir, ikinci dosya ise bloğun meta verisini içerir [36, 45].

MapReduce

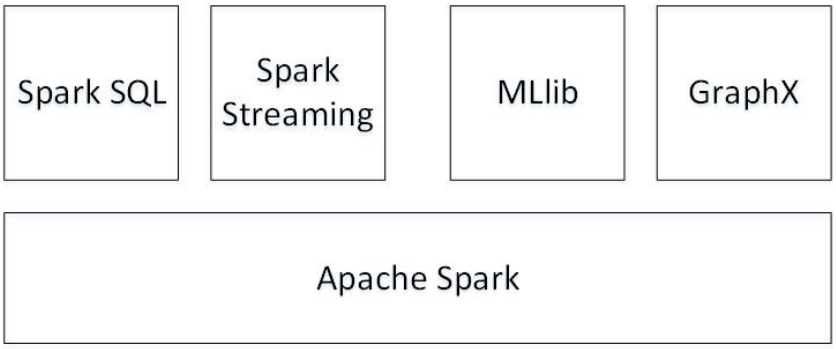
MapReduce, büyük ölçeklenebilirlik sağlayan bir programlama yaklaşımıdır. Hadoop kümelerindeki verileri işleyebilmek amacıyla kullanılır. Temel olarak Map görevi ve Reduce görevi olmak üzere iki işlemi gerçekleştirir. Map görevleri dağıtık dosya sisteminden girdiler alır ve bu girdilerden anahtar-değer (key-value) çiftleri üretir. Bu işlem Map fonksiyonu için yazılan koda göre gerçekleştirilir. Üretilen değer ana kontrolcü tarafından toplanır, anahtar ile sıralanır ve Reduce görevlerine göre bölünür. Sıralama temelde aynı anahtar-değerlerin aynı Reduce görevleriyle bitmesini sağlar. Reduce görevleri bir anahtarla ilişkili olan tüm değerleri bir anahtarda birleştirir. Birleştirme işlemi de Reduce görevi için yazılan koda göre gerçekleştirilir [36].

2.4.2. Apache Spark

Apache Spark, hız, kullanım kolaylığı ve karmaşık analitik üzerine kurulan ve büyük ölçekli verilerin paralel olarak işlenmesini sağlayan açık kaynak kodlu kütüphanedir. Spark metin verisi, grafik verisi gibi çok çeşitli veri setleri içeren, toplu veya gerçek zamanlı veri kaynağına sahip olan büyük ölçekli verilerin işlenmesi sürecinin yönetilmesini sağlayan geniş kapsamlı ve birleşik bir yapı sunmaktadır [44].

Spark, Scala programlama dili ile yazılmıştır ve Java Virtual Machine ortamında çalışır. Spark üzerinde Java, Python, Clojure, Scala ve R programlama dilleriyle uygulama geliştirilebilir. Çok sayıda üst düzey operatör içermesi, etkileşimli olarak çalışabilen paralel uygulamaların geliştirilmesini kolaylaştırmaktadır. Spark, programcıların Directed Acyclic Graph (DAG) örüntüsü kullanarak karmaşık, çok adımlı veri hatları geliştirmelerine olanak tanır ve farklı işlerin aynı veri ile çalışabilmesi için bellek içi veri paylaşımını destekler. Apache Spark teknolojisinde programlama modeline verilen isim Resilient Distributed Dataset (RDD)'dir. RDD kavramı verinin bellek içerisinde tutulan ve paralel olarak işlenebilen halini ifade eder. Bellek içi veri depolama ve gerçek zamanlı işleme gibi yetenekleri nedeniyle Spark, diğer büyük veri teknolojilerinden birkaç kat daha hızlı performans sergileyebilmektedir [25, 44].

Spark ekosisteminin bir parçası olan ve büyük veri analitiği ve makine öğrenmesi alanlarında ilave yetenekler sağlayan ek kütüphaneler bulunmaktadır.



Şekil 2.2. Apache Spark kütüphaneleri [25]

- Spark Streaming: Gerçek zamanlı akış verilerini işlemek için kullanılabilir. Ölçeklenebilir hataya dayanıklı akış uygulamaları oluşturmayı kolaylaştırır.

- Spark SQL: Veri setlerini Java Database Connectivity ara yüzü üzerinde ortaya çıkarır ve görselleştirme araçları kullanarak Spark verisi üzerinde SQL (Structured Query Language) sorgularını çalıştırma olanağı sağlar. Sorguları hızlı hale getirmek için bir maliyet tabanlı iyileştirme, dikey depolama ve kod üretimi içerir.
- Spark MLlib (Machine Learning Library): Sınıflandırma, regresyon, kümeleme, ilişkilendirme kuralları, karar ağaçları, filtreleme, boyut azaltma gibi işlemleri içeren makine öğrenmesi algoritmalarından ve özellik dönüşümü, model değerlendirmesi gibi iş akışı yardımcı programlarından oluşur.
- Spark GraphX: Graflar (çizge) ve graf paralel hesaplama için kullanılan ara yüzdür. Graf analitiği görevlerini basitleştiren graf algoritmaları içerir [6, 25].

3. SIRALI ÖRÜNTÜ MADENCİLİĞİ

Agrawal ve Sirikant tarafından sunulan sıralı örüntü madenciliği (sequential pattern mining), bilgi keşfi ve veri madenciliği alanındaki model keşfi yaklaşımlarından biridir. Bu kavram, sıralı bir dizide sık geçen tüm alt dizilerin bulunması işlemidir. Sıralı bir dizi, elemanlardan oluşur; elemanlar da ögelerden oluşur. Sıralı örüntü madenciliği algoritmaları, kullanıcının belirlediği minimum destek değeri üzerinden çalışır. Sıralı örüntü madenciliği modeli şu şekilde ifade edilebilir [1]:

$I = \{i_1, i_2, \ldots, i_k\}$ kümesi ögelerden (item) oluşur ve I'nın bir alt kümesine (subset), öge seti (itemset) denir. Sıralı bir liste olan $s = \langle s_1, s_2, \ldots, s_m \rangle$ ($s_i \subseteq I$) ise sekans (sequence) veya sıralı dizi olarak adlandırılır. Burada s_n, eleman ismini alır ve bir öge setidir. Bu sıralı dizideki elemanların sayısı, sıralı dizinin boyutunu verir. Sıralı dizinin uzunluğu ise sıralı dizideki ögelerin (item) toplam sayısıdır. Eğer, $a = \langle a_1, a_2, \ldots, a_x \rangle$ ve $b = \langle b_1, b_2, \ldots, b_v \rangle$ şeklinde iki sıralı dizi (sequence) için $1 \leq j_1 < j_2 < \ldots < j_x\text{-}1 < j_x \leq v$ şeklindeki sayılar için $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}$, …, $a_x \subseteq b_{jx}$ olursa, b sıralı dizisi a sıralı dizisini kapsar (b super-sequence, a subsequence).

Sıralı örüntü madenciliği, önceden belirlenen bir minimum destek değeri eşiğini aşan belirli sıralı örüntüleri çıkarma sürecidir. Sıralı dizilerin sayısı çok büyük olabildiğinden ve kullanıcıların en ilginç sıralı örüntüleri elde etmek için farklı çıkar ve gereksinimleri olduğundan genellikle minimum bir destek değeri kullanıcı tarafından önceden belirlenir. Minimum destek değerinin kullanılmasıyla bu ilgisiz örüntüler ilgi çekici hale getirilebilir, dolayısıyla madencilik işlemi daha verimli hale getirilebilir [58].

Sıralı örüntü madenciliğinin kullanımı; biyolojide farklı amino asitlerin modellerinin analizi, DNA dizilerindeki örüntülerin keşfedilmesi ve biyoinformatik [74, 75, 76, 77, 78, 79, 80, 81], ticari organizasyonlarda müşteri davranışlarının incelenmesi [82, 83, 84, 85], web servislerinde kullanıcı erişim örüntülerinin keşfedilmesi [86, 87, 88], sistem performans analizi [89], telekomünikasyon ağı analizi [90, 91, 92], metin analizi [93, 94, 95] gibi çeşitli alanlara yayılmış durumdadır [66].

3.1. Sıralı Örüntü Madenciliği Algoritmaları

Sıralı örüntü madenciliği üzerinde yoğun şekilde çalışılması sonucu çeşitli algoritmalar ortaya çıkmıştır. Literatürdeki sıralı örüntü madenciliği algoritmaları çizelge 3.3'te verildiği gibidir. Bu algoritmalar dört farklı kategoriye ayrılmıştır:

- Apriori Tabanlı Algoritmalar (Apriori Based Algorithms)
- Dikey Projeksiyon Tabanlı Algoritmalar (Vertical Projection Based Algorithms)
- Örüntü-Büyüme Algoritmalar (Pattern-Growth Algorithms)
- Erken Budamalı Algoritmalar (Early Prunning Algorithms).

Çizelge 3.1. Sıralı örüntü madenciliği algoritmaları

Yıl	Algoritma	Kategori
1995	AprioriAll [1]	Apriori Based
1996	GSP [2]	Apriori Based
2000	FreeSpan [18]	Pattern Growth
2000	PrefixSpan [42]	Pattern Growth
2001	SPADE [56]	Vertical Projection
2002	SPAM [7]	Vertical Projection
2004	DISC-all [11]	Early Prunning
2005	HVSM [46]	Vertical Projection
2005	LAPIN-SPAM [54]	Early Prunning
2006	LAPIN-LCI [55]	Early Prunning
2006	LAPIN-SUFFIX [55]	Early Prunning
2007	PRISM [17]	Vertical Projection

İlk olarak, Apriori-All ve GSP, prunning arama alanı için basit ve sezgisel ancak etkili bir fikir olan apriori prensibine dayanmaktadır [1, 2]. Apriori ilkesi sık bir dizinin tüm alt dizilerinin sık olması gerektiğini belirtmektedir. Algoritmalar, bir dizinin alt dizinleri üzerindeki kararını temel alarak arama alanını budamak için bu gerçeği kullanır.

İkincisi, SPADE algoritması dizi veri tabanları için dikey projeksiyon gösterimini kullanmaktadır [56]. SPADE algoritmaları, apriori prensibinden yararlanıp Apriori-All ve GSP'nin aday ve test yaklaşımını üretmeyi takip etseler de dikey izdüşüm nedeniyle Apriori-All ve GSP'den çok daha etkili saymayı desteklemektedir.

Üçüncü olarak, örüntü büyüme algoritmaları FreeSpan ve PrefixSpan önceki algoritmalardan farklı olarak, çok sayıda aday oluşturmadan, öngörülen veri tabanları kavramını getirmiştir [18, 42]. Örüntü büyüme algoritmaları, desen büyüdükçe yansıtılan şekilde projekte edilen veri tabanları üzerinde madencilik yaparlar. Başka bir deyişle, örüntü büyüme yaklaşımı, p her prefix (önek) olarak çıktı desenlerine çıktığında, bu tahmin edilen veri tabanındaki kalıpları bulan ve bulunan her sık geçen örüntü için öngörülen veri tabanı oluşturur. Tahmini veri tabanları, örüntüler büyüdükçe öngörülen veri tabanları küçültüldüğünden, veri tabanı taramalarının boyutunu azaltır. PrefixSpan algoritması, hem GSP hem de FreeSpan'den çok daha iyi performans gösterir [39].

Son olarak, model büyüme algoritmaları için arama alanını budamak için yeni bir basit LAPIN (Last Position Index) fikri ortaya atılmış ve LAPIN fikrinin, yoğun veri tabanlarında önceki yaklaşımlardan daha iyi performans gösterdiği görülmüştür [54, 55].

3.1.1. PrefixSpan algoritması

PrefixSpan (prefix-projected sequential pattern mining), sıralı bir dizide (sequence) önekleri (prefix) belirleyerek alt dizileri (subsequences) bulan bir sıralı örüntü madenciliği algoritmasıdır. Önekleri belirleyerek alt dizilerin çıkarılması, aday alt dizilerin oluşturulması sırasında harcanan çabayı azaltmaktadır. Genel fikri, yalnızca önek alt dizilerini (prefix subsequences) incelemek ve yalnızca projekte edilen veri tabanlarındaki sonek sıralı alt dizilerini (postfix subsequences) projelendirmektir. Projekte edilen her veri tabanında sıralı örüntüler yalnızca yerel sık geçen örüntüler keşfedilerek büyütülür [19].

Madencilik verimliliğini daha da artırmak için seviye bazlı (level-by-level) projeksiyon ve iki seviyeli (bi-level) projeksiyon olmak üzere iki tür veri tabanı projeksiyonu keşfedilmiştir. Projeksiyon maliyetini azaltmak ve öngörülen (alt) veri tabanı ve onun eşleştirilmiş psuedo projeksiyon işleme yapısı ana hafızaya sığabildiğinde işlemeyi hızlandırmak için bir ana bellek tabanlı psuedo projeksiyon tekniği geliştirilmiştir. Bu çalışma, veri tabanı büyük olduğunda iki seviyeli projeksiyonun daha iyi performans gösterdiğini ve tahmin edilen veri tabanları belleğe sığdığında işlemeyi büyük oranda hızlandırdığını göstermektedir. PrefixSpan tüm örüntü setini keşfetmekte ve Apriori tabanlı GSP algoritması ve FreeSpan'ten önemli ölçüde daha etkili ve daha hızlı çalışmaktadır [19].

$\alpha = <e_1, e_2, \dots, e_n>$ ve $\beta = < e_1', e_2', \dots, e_m'>$ $(m \leq n)$ şeklinde iki sıralı dizi (sequence) için; eğer (1) $i \leq m-1$ için $e_i' = e_i$, (2) $e_m' \subseteq e_m$, (3) $e_m' - e_m$ kümesindeki bütün ögeler e_m' 'den sonra sıralı olarak geliyorsa β, α'nın önekidir (prefix) [19].

Verilen α ve β sıralı dizilerinden β, α'nın alt sıralı dizisi (subsequence) olacak şekilde; eğer (1) α' 'de β öneki var ve (2) α' 'nün uygun süper sıralı dizisi (super-sequence) α'' yoksa, α'nın alt sıralı dizisi olan α' , β önekine göre α'nın bir projeksiyonudur [19].

$\alpha' = <e_1, e_2, \dots, e_n>$, α'nın $\beta = < e_1, e_2, \dots, e_{m-1}, e_m'>$ $(m \leq n)$ önekine göre projeksiyonu ve $e_m'' = (e_m - e_m')$ olacak şekilde; $\gamma = <e_m'', e_{m+1}, \dots, e_n>$ sıralı dizisi, α'nın β önekine dayanan soneki (postfix) olur ve $\alpha = \beta.\ \gamma$ şeklinde gösterilir [19].

- Eğer β, α'nın sıralı alt dizisi değilse α'nın β'ya göre projeksiyonu ve soneki yoktur.
- α, bir sıralı veri tabanı olan S'nin sıralı örüntüsü olacak şekilde; α-projekte veri tabanı, S'deki α önekine göre sıralı dizilerin soneklerinin bütünüdür ve $S|_\alpha$ şeklinde gösterilir.
- α, S sıralı veri tabanındaki bir sıralı örüntü ve β, α önekiyle başlayan bir sıralı dizi olacak şekilde; $S|_\alpha$ α-projekte veri tabanında β'nın destek (support) değeri, $S|_\alpha$ 'daki γ sıralı dizilerinin sayısıdır ve $support_{S|a}$ şeklinde gösterilir. Genellikle $support_{S|\alpha}(\beta) \leq support_{S|\alpha}(\beta / \alpha)$'dır [19].

Algoritmanın girdisi, bir sıralı veri tabanı olan S ve minimum destek eşiği olan min_sup değeridir. Çıktısı ise sıralı örüntü setlerinin tamamıdır. Parametreleri; sıralı örüntüler olan α, α'nın uzunluğu olan L, α-projekte veri tabanı olan $S|_\alpha$'dan oluşmaktadır. Yöntem basamaklarında önce $S|_\alpha$ bir kez taranır ve F sık geçen öge setleri bulunur; (1) F, sıralı bir örüntü oluşturmak üzere α'nın son elemanına eklenebilir veya (2) sıralı bir örüntü oluşturmak üzere α'ya eklenebilir. Sonraki adımda bir α' sıralı örüntü oluşturmak ve α' çıktısını bulmak için her bir F sık geçen ögesini α'ya eklenir. Daha sonra her bir α' için $S|_\alpha'$ α'-projekte veri tabanı oluşturulur ve PrefixSpan (α', L+1, $S|_\alpha'$) çağrılır [19].

PrefixSpan, veri kümesine birden fazla tarama yapmaya gerek kalmadan, bir veri kümesinden sıralı örüntüler bulmayı amaçlar. Bu algoritma, her turda birer uzun öge bulabilir ve daha sonra örüntüleri oluşturmak için tüm yüksek frekanstaki ögeleri bulmak için art arda uygulanır. Tekrarlama sırasında, veri kümesi daima bulunulan sık ögelere dayanılarak temizlenir; bu da veri kümesini büyük oranda azaltır ve araştırma süresini

kısaltır. PrefixSpan, sıralı veri tabanının yalnızca bir taranması ile desenler bulabilen bir desen büyüme yöntemi kullanır. PrefixSpan'e üç temel adım eklenmiştir:

1. Veri tabanını tarayarak ve işlemlerdeki öğeleri sayarak uzunluk-1 sıralı örüntüleri bulur.
2. Proje veri tabanını arama alanını azaltmak ve adım 1'de bulunan her örüntü için proje veri tabanını oluşturmak, yani her işlemde desenin ilk görünüşünden önce tüm ögeleri kaldırmaktır. Ardından, örüntünün kendisi öngörülen verilerin önüne gelecektir.
3. Öngörülen veri tabanında sıralı örüntüler bulur. Adım 1'deki prosedürü izleyerek, öngörülen veri tabanında uzunluk-2, uzunluk-3 gibi uzunlukları olan sık kullanılan örüntüler de bulunacaktır.

Son olarak, sıralı örüntülerin tümü, yukarıdaki 3 adımın ardışık olarak uygulanmasıyla bulunabilir [15].

PrefixSpan tarafından hiçbir sıralı aday dizinin üretilmesi gerekmez. Apriori benzeri algoritmaların aksine, PrefixSpan yalnızca uzunluğu kısa olan sık geçen ögeler üzerinden uzunluğu daha fazla olan sıralı örüntüleri üretir. Öngörülen bir veri tabanında var olan herhangi bir aday dizisi üretmez veya test etmez. Önemli sayıda aday dizisi üreten ve test eden GSP ile karşılaştırıldığında, PrefixSpan çok daha küçük bir alanı arar. PrefixSpan'in başlıca maliyeti projekte veri tabanlarının oluşturulmasıdır. En kötü ihtimalle, PrefixSpan her sıralı örüntü için projekte bir veri tabanı oluşturur. Sıralı örüntülerin sayısı iyi ise maliyet önemsizdir [19].

Apache Spark platformunda PrefixSpan algoritması

Makine öğrenmesi algoritmalarının çalıştırılabildiği Apache Spark platformunun MLlib kütüphanesi üzerinde PrefixSpan programı iki aşamada gerçekleştirilir. Frekans hesaplama kısmı kavramı, her işlemdeki ögeleri saymak ve oluşum sayısını minimum destekle karşılaştırmaktır. RDD'de flat map ve map ögeleri saymak için kullanılabilir ve her bir madde bir tekilde saklanır ve daha fazla karşılaştırma yapılır. Proje veri tabanı kısmı harita ve filtre fonksiyonlarıyla uygulanabilir. Map işlevi, öneki kaldırıldıktan sonra her bir işlemi geri getirecek ve filtre, ögeler olmadan işlemleri kaldırmak için kullanılacaktır. PrefixSpan programını yazmak için RDD'yi kullanarak işlem modeli basitleştirilmiştir.

Dahası, ara sonuçların tümü, işlemi hızlandırmak için bir başka gelişme olacak sabit disk yerine belleğe kaydedilebilir [15].

4. LİTERATÜR ARAŞTIRMASI

Son yıllarda sıralı örüntü madenciliği önemli bir veri madenciliği tekniği haline gelmiş ve pek çok alanda uygulanmıştır. Sıralı örüntü madenciliğinin temel amacı, bir işlemsel veri tabanı içerisinde geçen sık dizileri keşfetmektir. Sıralı örüntü madenciliği, sıralı olayların aralarındaki ilişkileri bulmaya olayların belirli bir sıralaması olup olmadığını belirlemeye çalışır [58].

Akbar ve Saptawati, 2016 yılında yaptıkları çalışmada Spark platformunda uygulanan PrefixSpan'i dağıtık bir sistem olarak kullanarak veriden daha fazla bilgi edinmek için ölçeklenebilir sıralı örüntü çıkarmayı önermişlerdir. Çalışmanın amacı, karmaşık ve yüksek boyutlu verilerin ölçeklenebilirliğinin etkin ve hızlı bir şekilde gerçekleştirilmesidir. Çalışmada yapılan deneyler, bu yöntemin madencilik işlemini hızlandırmak için kümeleme kaynaklarını tam olarak kullanabildiğini, veri tabanı tarama süresini azalttığını ve Spark platformunda artan sayıda işçi ile öngörülen veri tabanını oluşturabileceğini göstermektedir. PrefixSpan geniş ve yüksek boyutlu verileri işlemek üzere genişletilmiştir. İşçi sayısının artırılması, özellikle yüksek boyutlu veriler için zaman performansı üzerinde yüksek etkiye sahiptir. Bununla birlikte, deney sonucu, özellikle işçi sayısı üç veya daha fazla olduğunda, performansın işçi sayısıyla doğrusal olarak artmadığını göstermiştir. Deney ayrıca performansın yürütücü hafızanın sayısıyla doğrusal bir şekilde ölçeklenmediğini, bu nedenle Spark'ın optimum düzeyde çalışması için çok fazla yapılandırma gerektirdiğini açığa çıkarmıştır [3].

Wei ve diğerleri, 2012 yılında yaptıkları çalışmada büyük veri kümesindeki sıralı örüntülerin incelenmesi için, MapReduce programlama modeline ve PrefixSpan'e dayalı dağıtık ardışık örüntü araştırma algoritmasını önermişlerdir. Madencilik görevleri birçok küçük göreve ayrılmıştır; map fonksiyonu her prefix-projected ardışık örüntüyü keşfetmek için kullanılır ve öngörülen veri tabanları paralel olarak oluşturulmuştur. Arama alanını basitleştirir ve daha yüksek bir maden verimliliği elde eder. Ardından ara değerler, muhtemelen daha küçük bir değerler seti üretmek için tüm bu değerleri bir araya getiren bir reduce işlevine geçirilir. Hem teorik analizler hem de deney sonuçları, MapReduce-PrefixSpan'in veri tabanını tarama süresini azalttığını göstermiştir. Hadoop platformunda artan sayıdaki işlemci ile büyük veri madenciliği sorununu etkin bir şekilde çözer, önemli

hızlanma ve ölçeklendirme performanslarına sahiptir [52].

Deng ve diğerleri, 2014 yılında yaptıkları çalışmada ardışık bir örüntü araştırma algoritmasını hızlı büyük ölçekli bir veri işleme motoru olan Spark ile entegre etmek için yeni bir yöntem, büyük verilerdeki örüntü madenciliğini önermişlerdir. Bu yöntemin hızlı ve rahat bir şekilde muazzam verileri nasıl işlediğini göstermek için iyi bilinen bir algoritma PrefixSpan örnek olarak kullanılmıştır. Deneyler, bu yöntemin ortak platform Hadoop'tan daha iyi bir performans ile madencilik işlemini hızlandırmak için kümeleme kaynaklarının tam olarak kullanılabileceğini göstermiştir [15].

Huang ve diğerleri, 2010 yılında yaptıkları çalışmada sıralı örüntü madenciliğinin ölçeklenebilirlik problemini ele alan bir dağıtık madencilik algoritması tasarlamışlardır. Önerilen DPSP algoritması, Hadoop platformu üzerinde uygulanmıştır. Kullanılmayan ögeleri silmek, mevcut aday sıralı örüntüleri için DPSP'de MapReduce görevleri önerilmiştir. Deneysel sonuçlar, DPSP'nin büyük ölçeklenebilirliğe sahip olduğunu ve sonuç olarak madencilik algoritmalarının performansını ve uygulanabilirliğini artırdığını göstermiştir [22].

Honarvar ve Sami 2016 yılında yaptıkları çalışmada PrefixSpan algoritmasını kullanarak gerçek cihazların güç kullanımına ait veri setinden değerli sıralı dizi örüntüleri çıkarmışlardır. Bu araştırmada dağıtık ve paralel büyük veri işleme platformu olan Spark kullanılarak iki farklı küme üzerinde gerçekleştirilen deneylerin sonucunda elde edilen bulgular, elektrik kullanımının azaltılarak karbondioksit ve sera gazı emisyonunu azaltmak gibi çeşitli uygulamalar için sıralı dizi örüntülerinin çıkarılmasının önemini göstermiştir [21].

Wei ve diğerleri, 2012 yılında sıralı örüntü madenciliği üzerine yaptıkları çalışmada Bulut Bilişim kümesi ortamında büyük ölçekli veri madenciliği problemini çözme konusunda iyi performans gösteren MR-PrefixSpan algoritmasını önermiş ve uygulamışlardır. Büyük ölçekli veri setleri üzerinde sıralı örüntü madenciliği yapılabilmesi için önerilen algoritma, PrefixSpan ve MapReduce programlama modeline dayanan dağıtık sıralı örüntü madenciliği algoritmasıdır. Bu araştırmada gerçekleştirilen hem teorik analizler hem de deney sonuçları, MR-PrefixSpan'in veri tabanını tarama süresini azalttığını göstermektedir [52].

Sık geçen öge seti madenciliği, birliktelik kuralları madenciliği sürecinde önemli bir adımdır. Büyük veri çağında sık geçen öge seti madenciliği için geleneksel yaklaşımlar, bilgi işlem gücü ve bellek alanı sınırlı olduğunda önemli zorluklarla karşılaşır. Zhang ve diğerleri, 2015 yılında yaptıkları çalışmada, matris temelli bir budama yaklaşımı uygulayarak aday öge seti miktarını önemli ölçüde azaltabilen etkili bir şekilde dağıtık sık geçen öge seti madenciliği algoritmasını (Distributed Frequent Itemset Mining Algorithm - DFIMA) önermişlerdir. Önerilen algoritma, iteratif hesaplamanın verimliliğini daha da artırmak için Spark kullanılarak uygulanmıştır. Önerilen algoritma, paralel FP büyüme (parallel FP growth) algoritması ile karşılaştırılmış ve standart karşılaştırmalı değerlendirme veri kümelerini kullanan sayısal deney sonuçları, DFIMA'nın daha iyi verimlilik ve ölçeklenebilirliğe sahip olduğunu göstermiştir [57].

Chen ve diğerleri, 2013 yılında yaptıkları çalışmada, buluttaki MapReduce modeline (SPAMC olarak kısaltılır) dayanan sıralı örüntü madenciliği algoritmasını önermişlerdir. Önceki SPAM algoritmasından türetilmiştir, sözcük dizisi ağacını oluştururken aday örüntüleri etkin bir şekilde üretmek ve budamak için yinelemeli bir MapReduce framework tasarlanmıştır. Bu, yalnızca ağaç yapımının alt görevlerini bağımsız mapper'lara paralel olarak dağıtmakla kalmaz aynı zamanda destek sayımının paralel olarak işlenmesini de sağlar. Deneysel sonuçlar, SPAMC'nin büyük verilerle maden zamanını önemli ölçüde azaltabileceğini, aşırı derecede yüksek ölçeklenebilirlik elde edebildiğini ve bulut kümesinde mükemmel bir yük dengelemesi sağlayabileceğini göstermiştir [8].

Büyük veriler, ölçeklenebilirlik, uyarlanabilirlik ve kullanışlılık açısından geleneksel makine öğrenmesi için sayısız zorluklar yaratırken diğer yandan birçok teknik zorluğu gidermek, gerçek dünya etkileri yaratmak ve yeni makine öğrenmesi çözümlerine ilham vermek için yeni fırsatlar sunar. Yapılan bazı çalışmalarda büyük veri ile makine öğrenmesinin fırsatları ve zorlukları derlenmiş, sistematik bir şekilde özetlenmiş ve büyük verinin özelliklerine (boyut, hız, çeşitlilik, doğruluk, değer) göre kategorize edilmiştir. Makine öğrenmesi yaklaşımlarına genel bir bakış sunulmuş ve bu tekniklerle tanımlanan çeşitli zorlukların üstesinden gelinme şekli tartışılmıştır. Makine öğrenmesi yaklaşımları ve teknikleri, uygulayıcıların kullanım durumları için uygun çözümleri seçmesine yardımcı olmak için nihai amacı ile çeşitli zorlukları nasıl ele alabilecekleri konusunda tartışılmıştır. Bu çalışmaların amacı araştırmacılara, Büyük veri ile makine öğrenmesi konusunda daha kolay ve daha iyi seçim yapılmasına yönelik güçlü bir temel sağlamaktır. Bu hedef, çeşitli

zorluklar ve makine öğrenmesi yaklaşımları arasındaki ilişkileri ortaya koyan kapsamlı bir matris geliştirerek ve böylece bir dizi koşulda verilen en iyi seçimleri vurgulayarak gerçekleştirilmiştir [59, 60].

Qiu ve diğerleri, 2016 yılında yaptıkları çalışmada büyük boyutlu veri işleme için makine öğrenmesi üzerine yapılan çalışmalardaki son gelişmelerin araştırması sunmuşlardır. Makine öğrenmesi teknikleri gözden geçirilmiş ve temsil çalışmaları, derin öğrenme, dağıtık ve paralel öğrenme, aktarma öğrenme, aktif öğrenme ve çekirdek tabanlı öğrenme gibi son çalışmaların bazı umut verici öğrenme yöntemleri vurgulanmıştır. Büyük veri için makine öğrenmesindeki zorluklar ve olası çözümler hakkında analiz ve tartışmalara odaklanılmıştır. Büyük veri işleme için sinyal işleme teknikleriyle makine öğrenmesinin yakın bağlantıları araştırılmıştır [43].

Singh ve Reddy'nin 2015 yılında yaptıkları çalışmanın amacı, büyük veri analizi gerçekleştirmek için kullanılabilen farklı platformların derinlemesine bir analizini gerçekleştirmektir. Bu çalışma, büyük veri analizi için farklı donanım platformlarını araştırmış ve ölçeklenebilirlik, veri hızı, hata toleransı, gerçek zamanlı işleme, desteklenen veri boyutu ve yinelemeli görev gibi çeşitli metriklere dayanan bu platformların avantajlarını ve dezavantajlarını değerlendirmiştir. Donanıma ek olarak, bu platformların her birinde kullanılan yazılım çerçevelerinin ayrıntılı bir açıklaması da güçlü yanları ve dezavantajları ile birlikte ele alınmıştır. Burada açıklanan kritik özelliklerden bazıları, okuyucuların hesaplama ihtiyaçlarına bağlı olarak doğru platform seçimi konusunda bilinçli bir karar vermeleri için potansiyel olarak yardımcı olabilir. Büyük veri analizi bağlamında platformların her birinin etkinliğine daha fazla bilgi sağlamak için, yaygın olarak kullanılan k-means kümeleme algoritmasının çeşitli platformlardaki belirli uygulama seviyesi ayrıntıları aynı zamanda kaba kod şeklinde tanımlanmıştır [61].

Yapılan bazı çalışmalarda Apache Spark kullanılarak büyük veri analizleri üzerine teknik bir inceleme sunulmuş ve Apache Spark'ın temel bileşenleri, soyutlamalar ve temel özellikleri üzerine odaklanılmıştır. Apache Spark projesinin, büyük veri analizinin temel zorluklarını çözmek için çok önemli bir katkı sağladığı belirtilmiş ve bir makine öğrenmesi örneği çalıştırılmıştır. Doğrusal regresyon modelinin çalışmasından elde edilen sonuçlara dayanılarak Spark'ın paralel hesaplama ve iterasyon uygulamalarında iyi olduğu kanısına varılmıştır [62, 63].

Hafez ve diğerleri, 2016 yılında yaptıkları çalışmada büyük veri madenciliği aracı olan Apache Spark tarafından farklı türde ve boyutta veri kümeleri üzerinde farklı makine öğrenmesi algoritmaları kullanmışlardır. Algoritmalar pazarlama, paketleme ve istatistik, güvenlik veri setleri olmak üzere üç farklı veri kümesinde uygulanmıştır. Algoritmalar esas olarak doğruluk ve eğitim süresi olmak üzere birkaç parametre temel alınarak karşılaştırılmıştır. Modelin oluşturulmasında yürütme süresi ile veri kayıtlarının hacmi ve veri kümelerinin nitelik büyüklüğü arasında doğrudan bir ilişki olduğu gözlemlenmiştir. Bu deneyin bulgusu, karar ağacı algoritmasının pazarlama ve güvenlik veri seti için en uygun çözüm olduğunu göstermiştir. Ayrıca, paketleme ve istatistik veri setinde lojistik regresyon algoritması en yüksek doğruluğa sahip olmuştur [64].

Büyük miktarlardaki verilerin işlenmesi için yeni platformlara ihtiyaç duyulmasıyla birlikte MapReduce modelini temel alan ve ana özelliği bellek içi hesaplama olan Apache Spark ile dağıtık akan veri ve yığın veri işleme üzerine odaklanan Apache Flink ortaya çıkmıştır. Büyük veriyi işlemek ve depolamak için kullanılan bu iki platformun ölçeklenebilirliğinin yığın veri işlemesi üzerine karşılaştırmalı bir çalışması yapılmıştır. Bu iki platform, kendi makine öğrenmesi kütüphanelerinde bulunan öğrenme algoritmaları olan destek vektör makineleri ve doğrusal regresyon kullanılarak aynı veri kümesi üzerinde test edilmiştir. Deneysel sonuçlar, Spark MLlib'in Flink'ten daha iyi performansa ve genel olarak daha düşük çalışma zamanlarına sahip olduğunu göstermiştir [65].

Beykent Üniversitesinde, Kayım tarafından 2015 yılında yapılan “K-Means ile DBSCAN Algoritmasının Paralelleştirmesi ve Hadoop Üzerinde Büyük Veri Analizinde Kullanılması, Performans ve Yeterlilik Karşılaştırması” isimli yüksek lisans tezinde veri madenciliği kümeleme algoritmalarından olan k-means ve DBSCAN algoritmasının paralelleştirmesinin büyük veriler üzerindeki etkileri incelenmiş ve paralelleştirmenin performans ve hız kriterlerini olumlu etkilediği görülmüştür. K-means ve DBSCAN algoritmalarının değişik düğümlerde paralelleştirmesi incelendiğinde düğüm sayısının artması ile kümeleme hızının da arttığı görülmüştür. Çalışma kapsamında yapılan uygulamalar neticesinde Hadoop üzerinde veri madenciliği algoritmalarının Mahout ile çalıştırılarak büyük veriler üzerinde iyi performanslar elde edilebileceği görülmüştür [67].

İstanbul Üniversitesi bünyesinde, Çetinkaya tarafından 2016 yılında yapılan “Hadoop / MapReduce Teknolojisi Kullanılarak Hızlı Tüketim Sektöründe Büyük Veri Analizi”

isimli yüksek lisans tezinde büyük verinin analizi için Hadoop platformunda MapReduce ve HDFS ile pazar sepeti analizi ve k-means ile kümeleme algoritmaları üzerinde çalışılmıştır. Bu çalışmalar neticesinde müşteri kanalları miktara ve karlılığa göre yeniden oluşturulmuştur. Ayrıca pazar sepeti analizi, MapReduce tekniği ile yapılarak ürün birliktelikleri bulunmuştur. Bu analiz ile ürünlerin birliktelikleri belirlenerek müşterilere ürün önerme, müşterilerin alışkanlıkları hakkında fikir edinme, raf dizilimi, kampanya modellerinin belirlenmesi ve müşteri memnuniyeti amaçlanmıştır. Hadoop üzerinde MapReduce tekniği ile paralel olarak yapılan analizlerin veri miktarına bağlı olarak belirli bir düğüm sayısına kadar kazanım sağladığı ve veri miktarı arttırıldıkça en optimum sürede analiz etmek için kullanılacak düğüm sayısını arttırmak gerektiği gözlemlenmiştir. Ayrıca veri miktarına bağlı olarak belirli bir düğüm sonrasında düğüm sayısını arttırmanın kazanım sağlamadığı aksine daha uzun zamanda analiz edildiği gözlemlenmiştir. Bu çalışma ile paralel işlemenin ve Hadoop platformunun kullanılması ile etkin sonuçlar alındığı belirlenmiştir [68].

İstanbul Teknik Üniversitesinde, Akgün tarafından 2016 yılında yapılan "Apache Spark Tabanlı Destek Vektör Makineleri İle Akan Büyük Veri Sınıflandırma" isimli yüksek lisans tezinde kullanılan sahtecilik veri kümesi için lojistik regresyon ve destek vektör makineleri yöntemlerinin başarımlarını analiz etmek adına birikmiş veri üzerinden SAS ürünü kullanılarak deneyler yapılmıştır. Sınıflandırma probleminin çözümünde destek vektör makineleri yönteminin başarılı olduğu gözlemlendikten sonra ilgili destek vektör makineleri yöntemi akan veri üzerine uyarlanmaya çalışılmıştır. Akan büyük veri teknolojisi olarak Apache Spark Streaming seçilmiştir. Apache Spark Streaming ve Apache Spark MLlib teknolojileri kullanılarak destek vektör makineleri yöntemi akan veri üzerine uyarlanmıştır. Apache Spark Streaming teknolojisi ile beraber gelen mevcut akan lojistik regresyon yöntemi ile sonuçlar karşılaştırılmış ve geliştirilen destek vektör makineleri yönteminin kullanılan veri kümeleri üzerinde daha başarılı sonuçlandığı gözlemlenmiştir. Öte yandan Apache Spark teknolojisi paralel programlama modeli ile çalışmaktadır. Geliştirilen akan destek vektör makineleri yöntemin dağıtık bilgisayarlara getirdiği yük incelendiğinde mevcut akan lojistik regresyon yöntemi ile benzer sonuçlar çıktığı gözlemlenmiştir [69].

Fırat Üniversitesi bünyesinde, Hallaç tarafından 2014 yılında yapılan "Büyük Veri Analizinde Dağıtık Makine Öğrenmesi Algoritmalarının Kullanılması" isimli yüksek lisans

tezinde oluşturulan bulut sistemi üzerinde açık kaynaklı büyük veri işleme teknolojileri olan Hadoop ve Spark teknolojileriyle uygulamalar geliştirilmiştir. Büyük verilerden otomatik olarak faydalı bilgi çıkarımında kullanılan makine öğrenmesi teknikleri ve bu yöntemlerin dağıtık bir sistem üzerinde uygulanışı uygulamalar geliştirilerek incelenmiştir. İnternet üzerinden otomatik bir şekilde ekonomi, spor, kültür, politika ve dünya kategorilerinden haberler toplanıp bu haberlerin otomatik bir şekilde sınıflandırılması Naive Bayes makine öğrenmesi algoritmasının paralel bir şekilde çalıştırılması ile gerçekleştirilmiştir. Yapılan sınıflandırma, hem kategorilerin doğru olarak tahmin edilmesi açısından başarı göstermiş hem de büyük miktarda verinin yüksek performanslı bir şekilde paralel olarak sınıflandırılabildiğini göstermiştir. Makine öğrenmesi tekniklerinin incelenmesinin yanında, bu tekniklerinin uygulanması aşamasına kadar olan sürecin de paralel bir şekilde gerçekleştirilmesi incelenmiş ve bu amaçla log analizi, dağıtık doğal dil işleme çalışmaları yapılmıştır. Log analizi ile büyük miktarda log verisinin Hadoop ile analizi yapılmıştır. Birden fazla Hadoop işinin zincirleme olarak çalıştırılması gösterilmiştir. Aynı işlemler Spark ile gerçekleştirilip Hadoop ve Spark teknolojileri için performans karşılaştırılması yapılmıştır. Yapılan karşılaştırmalar sonucunda Spark'ın Hadoop'a göre daha hızlı sonuç verdiği görülmüştür [70].

Sabancı Üniversitesinde, Selçuk tarafından 2015 yılında yapılan "Büyük Veri Üzerinde Dağıtık Dosya Sistemi ve Paralel İşleme Kullanarak Mahremiyet Korumalı Arama" isimli yüksek lisans tezinde belli bir büyük veri uygulaması ile ilişkili güvenlik ve mahremiyet sorunları adreslenmektedir. Şifreli bulut verisi üzerinde güvenli kelime-tabanlı aram işleminin Büyük Veri ortamında zor olduğunu vurgulanıp teknik zorluklar belirtilmiştir. Ayrıca, sadece devasa değil aynı zamanda değişen ve çok hızlı biriken büyük veri ortamı için, şifreli veriler üzerinde uygulanabilir temel işlemlerden biri olan mahremiyet korumalı kelime arama işlemi üzerinde var olan bir çalışma uyarlanmıştır. Geliştirilen çözümler, büyük veri ortamında, şifreli veriler üzerinde aramaya olanak veren güvenli bir endeks yapısını makul bir hız ile inşa edebilmeli, ayrıca verimli ve etkili bir kelime arama işlemi yöntemi için çok hızlı güncelleyebilmelidir. Önerilen çözümlerin, çok büyük veri kümeleri ile çalışacak şekilde ölçeklendirilebilmesi için, Hadoop Dağıtık Dosya Sistemi (HDFS) ve MapReduce programlama modeli gibi paralel programlama teknikleri ve dağıtık dosya sistemleri kullanılmaktadır. Gerçek veriler üzerinde gerçekleştirilen kapsamlı deneyler vasıtasıyla önerilen yöntemin etkinliği ve doğruluğu deneysel olarak gösterilmiştir [71].

Gazi Üniversitesi bünyesinde, Aydemir tarafından 2016 yılında yapılan “Sınır Güvenliği için Büyük Veri Teknik ve Teknolojileri ile Boru Hattı Tasarımı” isimli yüksek lisans tezinde sınır güvenliği için gerçek zamanlı akan veri ve yığın veri üzerinde analiz yapmayı sağlayan tutarlı, dayanıklı, dağıtık ve ölçeklendirilebilir bir sistem ilk örneği geliştirilmiştir. Sistem akan veri ve yığın veri üzerinde işlem yapmayı sağlayan lambda mimarisi temel alınarak hem gerçek zamanlı hem de geçmişe dönük veriler üzerinde analiz yapılmasına olanak sağlayan bir yapıda oluşturulmuştur. Geliştirilen sistemin test edilebilmesi için örnek senaryolar oluşturulmuş elde edilen sonuçlar gözlemlenmiştir. Mevcut sistemlerin sahip oldukları veri işleme yetenekleri ile kıyaslandığında önerilen sistemin yeni yaklaşımlar ve çözümler geliştirilmesine olanak sağlayan erken uyarı sistemi olarak kullanılabilecek bir tasarıma sahip olduğu görülmüştür [72].

Pamukkale Üniversitesinde, Salur tarafından 2016 yılında yapılan “Büyük Veri Araçlarından Hadoop Kullanarak Veri Madenciliği” isimli yüksek lisans tezinde Büyük Veri işleme araçlarından olan Hadoop üzerinde veri madenciliği yapmak için özelleştirilmiş olan Mahout aracı kullanılmıştır. Metin madenciliğinde kullanılan veri kümesi için Türkiye'deki 15 günlük gazetenin Twitter'da paylaşmış oldukları haber başlıkları kullanılmıştır. Bu haber başlıkları Türkçe doğal dil işleme için geliştirilen Zemberek kütüphanesi yardımıyla ön işlemlerden geçirilmiştir. Bu haber başlıkları olumlu veya olumsuz olarak sınıflandırılmıştır. Sınıflandırma işlemi için Mahout aracıyla birlikte Naive Bayes istatistik tabanlı sınıflandırma algoritması kullanılmıştır. Sınıflandırma işleminde %80'e yakın başarı elde edilmiştir [73].

5. UYGULAMA

5.1. Veri Seti ve Problem Tanımı

1948 yılında Hollanda'da kurulan ve dünya genelinde birçok ülkeye çiftlik, tarım arazisi, enerji alanlarında çeşitli hizmetler veren bir şirket, yine Hollanda'da kurulan ve İstanbul'da şubesi bulunan bir yazılım geliştirme firmasıyla ortaklık kurarak tarım ve hayvancılık sektöründe hasat, süt sağımı, besleme ve üremeye ilişkin mevcut olan bütün verilerin birbiriyle bağlantısını yaparak çiftçilere destek olmayı hedeflemiştir [34].

Bu Ar-Ge firmasında tarım ve hayvancılık sektörüne yönelik oldukça kapsamlı robotik ve gömülü yazılım projeleri yürütülmektedir. Süt Üretim Çiftliği Otomasyonu projesi, modern mandıra çiftliklerinde kullanılan süt sağma, yem verme, temizlik, vb. robotlarının otomasyonunu sağlamak amacıyla başlatılmıştır. Bu robotlar kullanıcıya ihtiyaç duymayan, tamamen otonom süt sağım, yemleme ve temizlik sistemleri kapsamında geliştirilen robotlardır. Karşılıklı mesajlaşma, periyodik tarama vb. yöntemlerle koordineli çalışan robotlar çiftliklerde süt sağım, yemleme ve temizlik süreçlerini tamamen otomatik hale getirmek suretiyle çiftçilerin yeterince ağır olan işlerini azaltmakta ve daha verimli bir süt üretimi, iş gücünde verimlilik, hayvanların düzenli ve sağlıklı beslenmesi, ölçüm ve raporlama avantajları sağlamaktadır [35].

Resim 5.1. Süt sağım robotu [35]

Resim 5.2. Yemleme robotu [35]

Çiftlik hayvanlarını izlemek amacıyla kurulan sisteme, dünya genelinde bulunan yaklaşık 30000 robottan sürekli veri akışı olmaktadır. Çiftliklerde çalışan robotlar kendi aralarında mesajlaşmakta, aynı zamanda kullanıcıyı uyarmak ve bilgilendirmek amacıyla çeşitli alarmlar vermektedir. Merkezi bir sistemde toplanan bu alarmlar, robotun çalışmasını ve

çiftlikteki önemli bir süreci durduran kritik alarmlar olabileceği gibi aciliyet seviyesi düşük basit uyarı mesajları da olabilir.

Alarmların aralarındaki ilişkileri tanımlayabilecek akıllı bir mekanizma bulunmaması sebebiyle bazen robotların ürettiği aynı alarmlar çiftçiye tekrar tekrar bildirim olarak gönderilebilmektedir. Dolayısıyla büyük boyutlardaki bu veri trafiği, sistemi ve çiftçiyi yormaktadır. Yapılan çalışmada geçmiş alarm bilgileri analiz edilerek bu problemlere çözüm getirilmesi amaçlanmıştır.

Mevcut veriden bilgi keşfi süreci, veri madenciliği aşamaları göz önünde bulundurularak gerçekleştirilmiştir. Süreç sırasıyla; problemin belirlenmesi, veri setinin ön işleme tabi tutulması, sıralı örüntülerin çıkarılması ve sonuçların değerlendirilmesi olarak gerçekleştirilmiştir.

Süt sağım sistemi; süt sağım robotu, süt depolama tankı, kontrol sistemi ve yönetim sistemi olmak üzere dört temel parçadan oluşmaktadır. Bu cihazlardan toplanan verilerden oluşan veri tabanı; cihazların bilgileri, alarm türleri, alarm tanımları, alarm bilgileri gibi tabloları içermektedir. Çiftliklerde bulunan bütün cihazların isimleri, id'leri, versiyonları, seri numaraları, konum bilgileri gibi veriler cihaz bilgileri tablosunda; cihazların verdiği alarmların kategorileri ve aciliyet seviyeleri alarm türleri tablosunda; alarmların id'leri ve tanımları alarm tanımları tablosunda; verilen alarmların başlangıç ve bitiş zamanları tarih ve saat biçiminde, alarmları veren cihazların id'leri ve alarm id'leri alarm bilgileri tablosunda bulunmaktadır. SQL Server aracılığıyla veri tabanı üzerinde SQL sorguları yazılarak tablolardan kullanılacak veri türleri çekilmiştir. Bu şekilde elde edilen veri setinin bir örneği şekil 5.1'de gösterildiği gibidir.

Kullanılan veri seti; robotların bulunduğu konumlar, robotların id'leri, robotların isimleri, üretilen alarmların oluştuğu tarihi ve saati, alarmların id'leri, alarmların tanımlamaları, alarmların aciliyet seviyesi gibi alanlardan oluşmaktadır. Çiftliklerde aynı id'ye sahip robotlar birden fazla bulunabilir ancak robotların konum bilgisini veren adres numarası tektir. Alarmların aciliyet seviyeleri ise "1 (critical alarm), 2 (alarm), 3 (attention), 4 (report), 5 (pulse alarm)" şeklinde beş kategoriden oluşmaktadır. Bu kategorilere dahil olan alarm türleri 691 adet farklı alarm tanımlamalarından oluşmaktadır. Aciliyet seviyesi 1 (critical alarm) olan 315 adet, 2 (alarm) olan 208 adet, 3 (attention) olan 106 adet, 4

(report) olan 61 adet ve 5 (pulse alarm) olan 1 adet alarm tanımlaması mevcuttur.

Bu tanımlamalardan kritik alarm seviyesinde olan alarm tanımlarına gücün kesilmesi, alarm seviyesinde olan alarm tanımlarına hava basıncının düşmesi, uyarı seviyesinde olan alarm tanımlarına id tanıma hatası, rapor seviyesinde olan alarm tanımlarına temizliğin başlangıç-bitiş bildirimleri örnek olarak verilebilir ancak darbeli alarm seviyesinde olan alarm yalnızca hayvanların denetlenmesi gerektiğinin bildirimidir. Kritik alarmlar, derhal müdahale gerektiren bir durum olduğunda verilir. Bu durumlarda cihazın çalışması durmuştur veya birkaç dakika içinde duracaktır. Cihazın yeniden çalıştırılabilmesi için kullanıcı müdahalesi gerekir. İkinci seviyedeki alarmlar, müdahale gerektiren ancak acil olmayan bir durum olduğunda verilir. Bu durumlarda cihaz çalışmaya devam etmektedir.

DeviceAddress	IndicationCreationTime	IndicationDescription	IndicationName	DeviceId	DeviceName	IndicationId
101	2017-05-17 12:47:13.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
101	2017-05-17 12:16:25.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
201	2017-05-17 12:16:24.000	End of main cleaning <P1>	Report	24	CRS M3 A	107
101	2017-05-17 12:14:05.000	Exceeding conduct. value after cleaning	Attention	30	Astronaut A4 101	13
201	2017-05-17 12:00:31.000	Start of main cleaning (<P1>) <P2>	Report	24	CRS M3 A	106
101	2017-05-17 09:59:43.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
101	2017-05-17 08:38:31.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
101	2017-05-17 08:07:11.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
101	2017-05-17 07:56:37.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
101	2017-05-17 07:27:30.000	Milk filter has been exchanged	Report	30	Astronaut A4 101	128
101	2017-05-17 06:16:27.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
201	2017-05-17 06:16:25.000	End of main cleaning <P1>	Report	24	CRS M3 A	107
201	2017-05-17 06:00:32.000	Start of main cleaning (<P1>) <P2>	Report	24	CRS M3 A	106
101	2017-05-17 05:24:18.000	Failure led <P1> <P2>	Attention	30	Astronaut A4 101	242
101	2017-05-17 02:44:53.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
101	2017-05-16 22:23:07.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
201	2017-05-16 22:23:05.000	End of main cleaning <P1>	Report	24	CRS M3 A	107
101	2017-05-16 22:20:38.000	Exceeding conduct. value after cleaning	Attention	30	Astronaut A4 101	13
201	2017-05-16 22:00:31.000	Start of main cleaning (<P1>) <P2>	Report	24	CRS M3 A	106
101	2017-05-16 21:25:49.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97
101	2017-05-16 20:40:29.000	Cleaning: <P1>, <P3>	Report	30	Astronaut A4 101	97

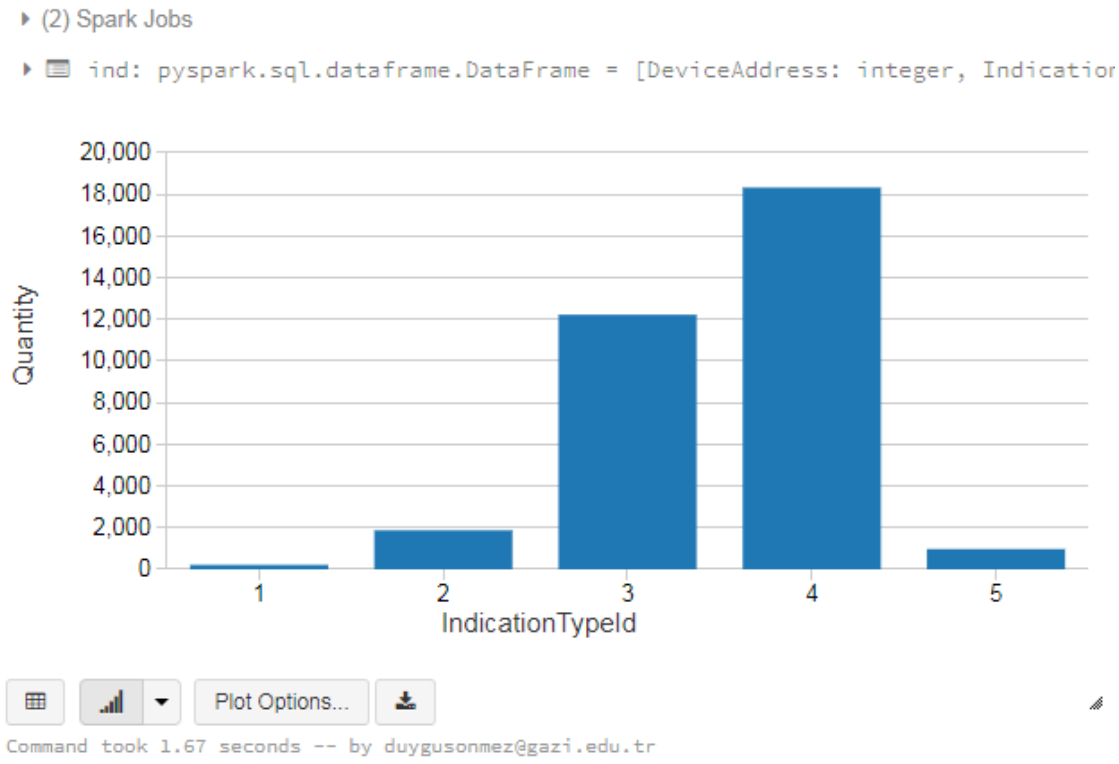
Şekil 5.1. Veri setinden bir kesit

5.2. Veri Setinin Analizi

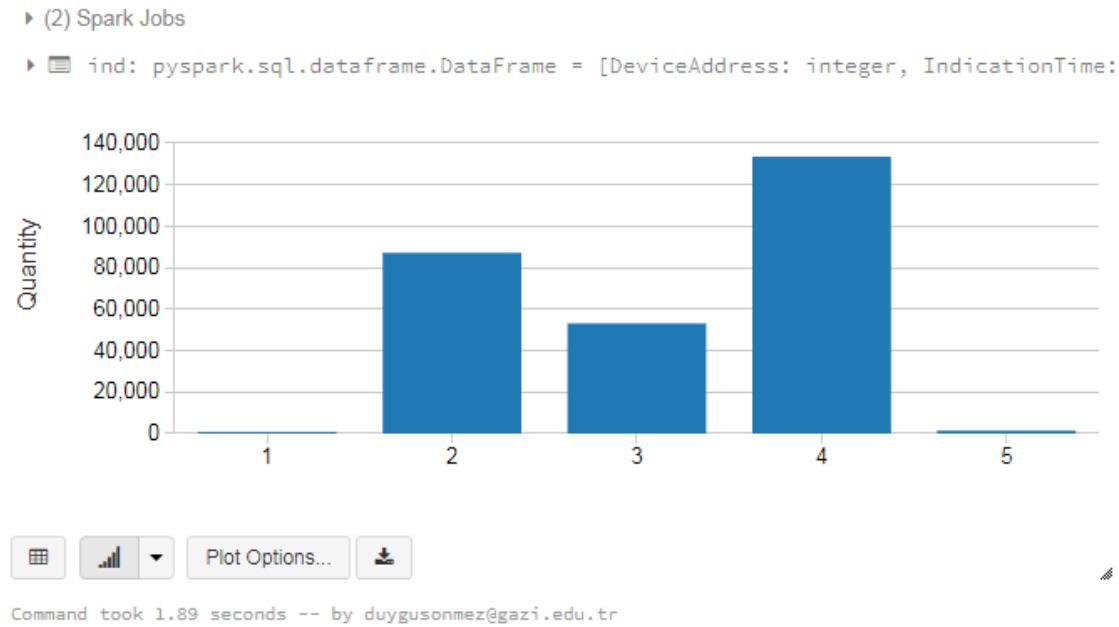
Yapılan çalışmada akıllı çiftliklerde bulunan robotlardan hangi robotun hangi alarmları verdiğini bulabilmek amacı ile her bir robotun verdiği alarmlar ayrı ayrı analiz edilmiştir. Bu analizler Apache Spark platformunda Python dili kullanılarak yapılmıştır. Bir çiftlikte aynı robottan birden fazla bulunabilmektedir, dolayısıyla DeviceId yani cihaz numaraları aynı olacaktır. Ancak bu robotların bulundukları konumu gösteren DeviceAddress numaraları tektir. Bu nedenle robot ayrımları DeviceAddress (robot konumu) bilgisi ile sağlanmıştır.

DeviceAddress (robot konumu) numarası 101 olan robotun 834 günlük 33625 adet alarm verisi IndicationTypeId yani alarmın aciliyet seviyesine göre incelendiğinde şekil 5.2'de gösterildiği üzere yaklaşık 18000 adet aciliyet seviyesi 4 (report) olan alarmlar, yaklaşık 12000 adet aciliyet seviyesi 3 (attention) olan alarmlar, yaklaşık 2000 adet aciliyet seviyesi 2 (alarm) olan alarmlar, yaklaşık 1000 adet aciliyet seviyesi 5 (pulse alarm) olan alamlar ve son olarak yaklaşık 200 adet aciliyet seviyesi 1 (critical alarm) olan alarmlar mevcuttur. Bu analiz var olan toplu veri setinin kategorize edilerek görülmesini ve hangi oranlarda hangi alarm tipi ile karşılaşıldığının tespitini yapmaktadır. Detaylı incelendiğinde ise DeviceAddress (robot konumu) numarası 101 olan robotun 834 günde yaklaşık 200 defa çalışmasının kesildiği ve duruş yaşandığı görülmektedir.

Aynı analiz DeviceAddress (robot konumu) numarası 102 olan robotun 3011 günlük 276173 adet alarm verisi alarmın aciliyet seviyesine göre incelendiğinde şekil 5.3'te gösterildiği üzere yaklaşık 133000 adet aciliyet seviyesi 4 (report) olan alarmlar, yaklaşık 87000 adet aciliyet seviyesi 2 (alarm) olan alarmlar, yaklaşık 53000 adet aciliyet seviyesi 3 (attention) olan alarmlar, yaklaşık 1500 adet aciliyet seviyesi 5 (pulse alarm) olan alamlar ve son olarak yaklaşık 850 adet aciliyet seviyesi 1 (critical alarm) olan alarmlar mevcuttur. Detaylı incelendiğinde ise DeviceAddress (robot konumu) numarası 102 olan robotun 3011 günde 850 defa çalışmasının kesildiği ve duruş yaşandığı görülmektedir.



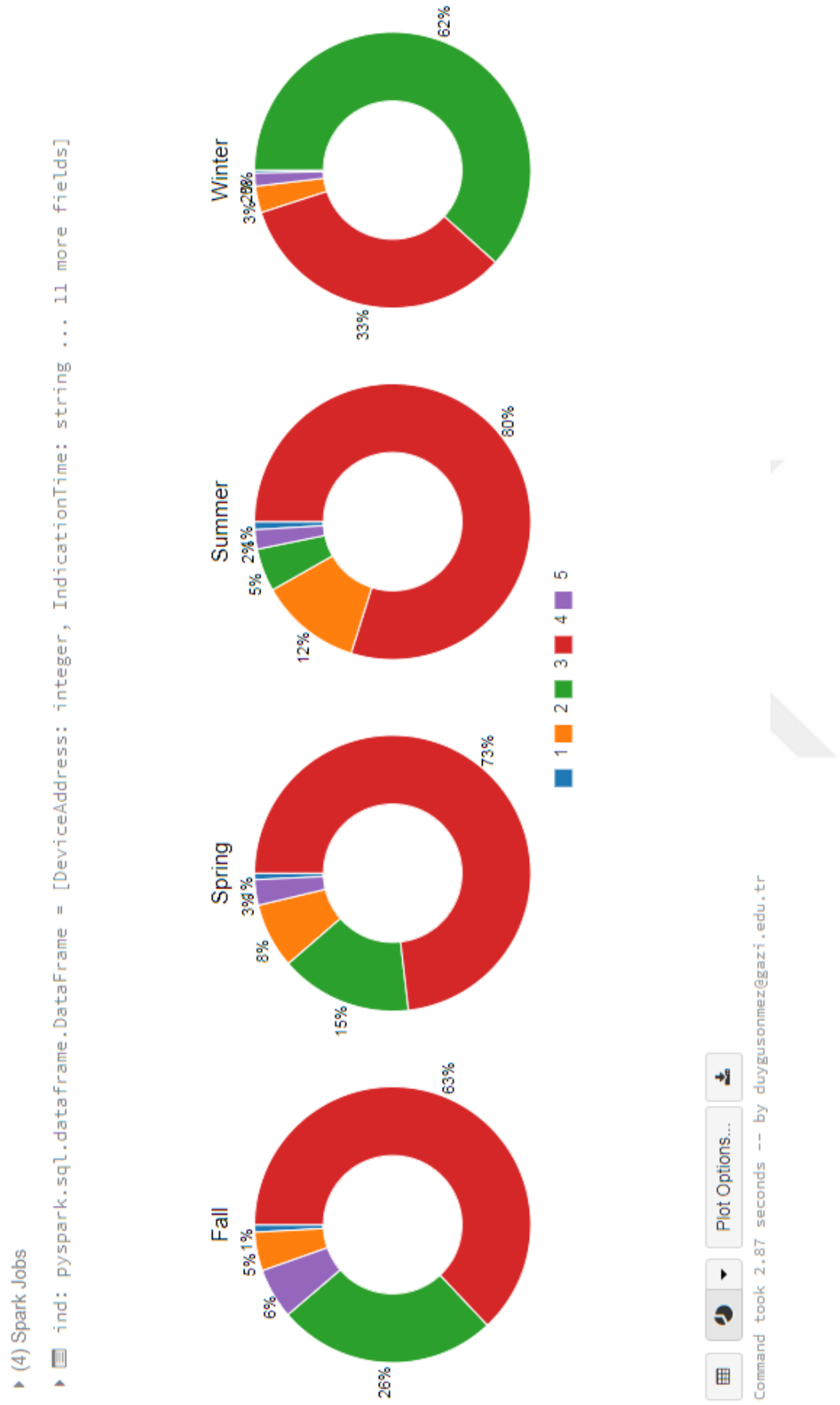
Şekil 5.2. Alarmların aciliyet seviyesine göre gerçekleşme miktarları (101)



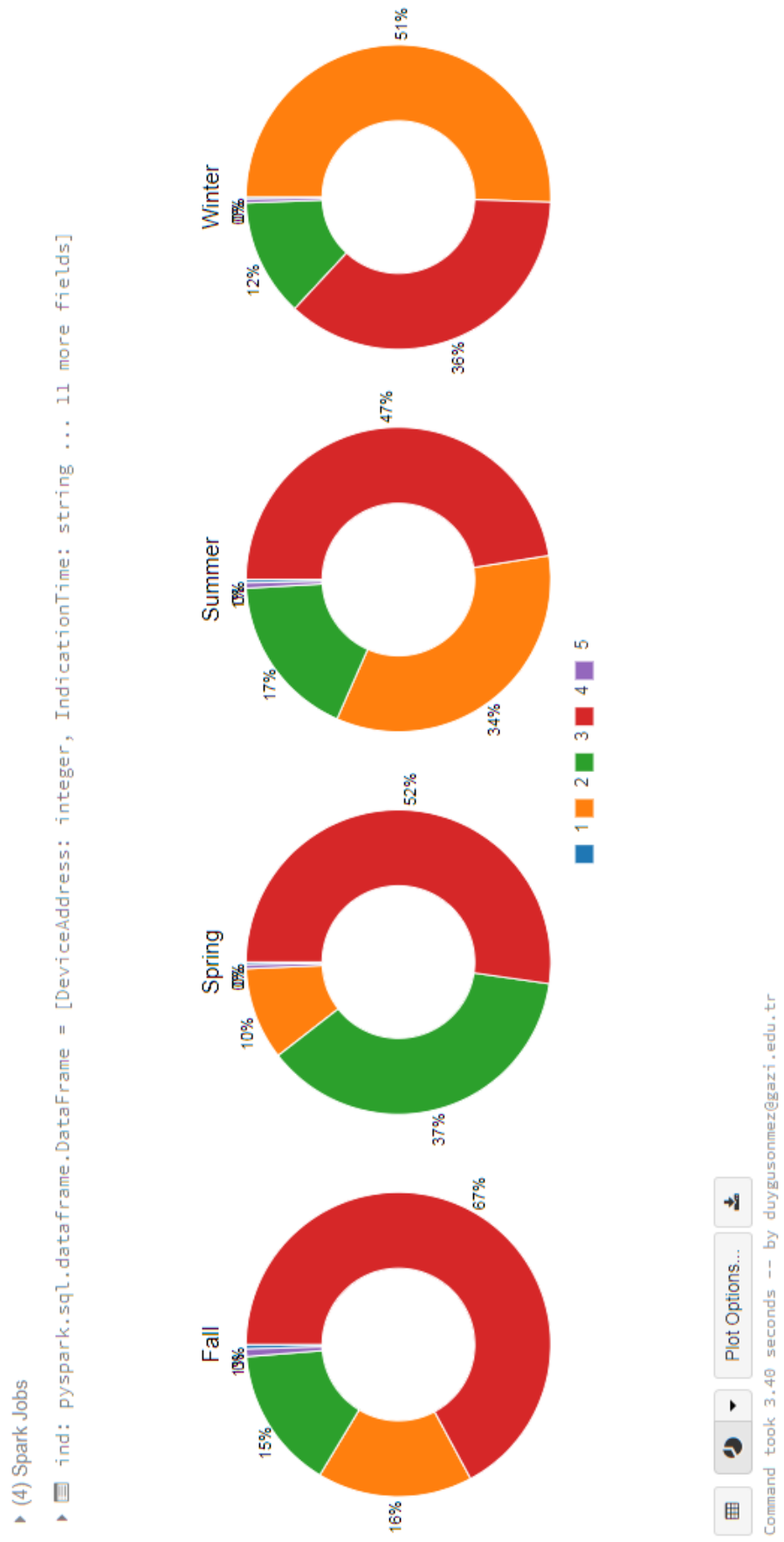
Şekil 5.3. Alarmların aciliyet seviyesine göre gerçekleşme miktarları (102)

DeviceAddress (robot konumu) numarası 101 olan robotun 834 günlük ve DeviceAddress (robot konumu) numarası 102 olan robotun 3011 günlük verisine ait alarmların grafiksel analizlerinin detayına inildiğinde hangi alarm tipinin hangi mevsimde daha çok oluştuğu

şekil 5.4'te ve şekil 5.5'te gösterildiği gibidir. Şekil 5.4 incelendiğinde sonbahar, ilkbahar ve yaz mevsimlerinde aciliyet seviyesi 4 (report) olan alarmlar daha çok görülürken kış mevsiminde aciliyet seviyesi 3 (attention) olan alarmların daha fazla olduğu görülmektedir. Şekil 5.5 incelendiğinde sonbahar, ilkbahar ve yaz mevsimlerinde aciliyet seviyesi 4 (report) olan alarmlar daha çok görülürken kış mevsiminde aciliyet seviyesi 2 (alarm) olan alarmların daha fazla olduğu görülmektedir. Dolayısıyla mevsimlerin alarm oluşumunda etkiye sahip olduğu ve kış aylarında görülen alarmların önem seviyesinin daha yüksek olduğu anlaşılmaktadır. Kış aylarında daha fazla gerçekleşen alarmların diğer mevsimlerden farklı olmasının sebebinin araştırılması ve problemin kaynağına inilmesi amacıyla alarm numaralarının karşılıklarının incelenmesi akıllı çiftlikler açısından önemlidir. Bu sebeple alarm kayıtları mevsimsel bazda gruplandırılarak analiz edilebilmektedir.



Şekil 5.4. Alarm tiplerinin mevsimsel bazda gerçekleşme oranları (101)



Şekil 5.5. Alarm tiplerinin mevsimsel bazda gerçekleşme oranları (102)

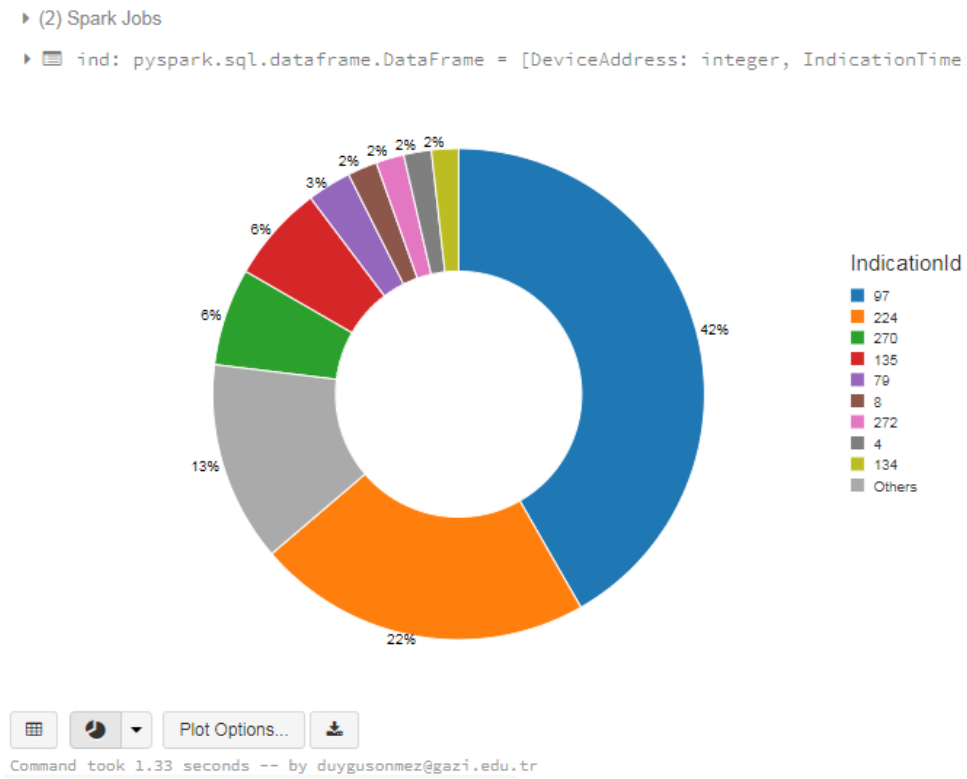
Alarm tiplerinin detayına inildiğinde 691 farklı alarm tanımlaması mevcuttur. DeviceAddress (robot konumu) numarası 101 olan robotun 33625 adet alarm verisine ait alarm tanımları analiz edildiğinde şekil 5.6'da gösterildiği üzere %42 oranında IndicationId yani alarm numarası 97 olan alarmlar, %22 oranında alarm numarası 224 olan alarmlar, %6 oranında alarm numarası 270 ve 135 olan alarmlar, %3 oranında alarm numarası 79 olan alarmlar, %2 oranında alarm numarası 8, 272, 4, 134 olan alarmlar ve %13 oranında diğer alarmlar mevcuttur.

Aynı şekilde DeviceAddress (robot konumu) numarası 102 olan robotun 276173 adet alarm verisine ait alarm tanımları analiz edildiğinde şekil 5.7'de gösterildiği üzere %19 oranında IndicationId yani alarm numarası 209 olan alarmlar, %17 oranında alarm numarası 97 olan alarmlar, %8 oranında alarm numarası 60 ve 227 olan alarmlar, %5 oranında alarm numarası 135 olan alarmlar, %4 oranında alarm numarası 4, 249 olan alarmlar, %1 oranında alarm numarası 88 olan alarmlar ve %12 oranında diğer alarmlar mevcuttur.

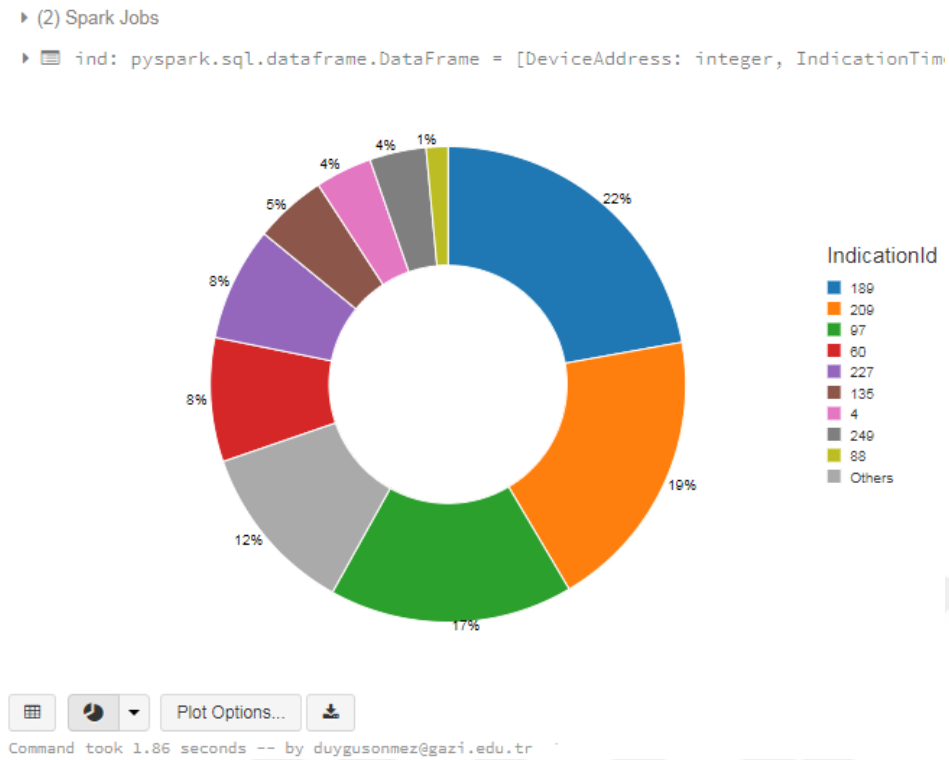
Aciliyet seviyesine göre kategorize edilmiş alarmlar ve gerçekleşme miktarları ise şekil 5.8 ve şekil 5.9'da gösterilmiştir. Oransal ve sayısal bazda analiz edilen bu alarm id'lerinin bir kısmının ne anlama geldiği çizelge 5.1'de belirtildiği gibidir. Bu analizlerin detayına inildiğinde özellikle aciliyet seviyesi yüksek olan ısı sensör hatası veya bütün süt kazanlarının dolduğu gibi alarmların büyük oranlarda görülmeleri, çiftliklerin bu konular üzerine yoğunlaşarak bu tip alarmların azalması için gerekli önlemleri almaları gerektiğini göstermektedir.

Çizelge 5.1. Alarm tiplerine ve tanımlarına karşılık gelen id'ler

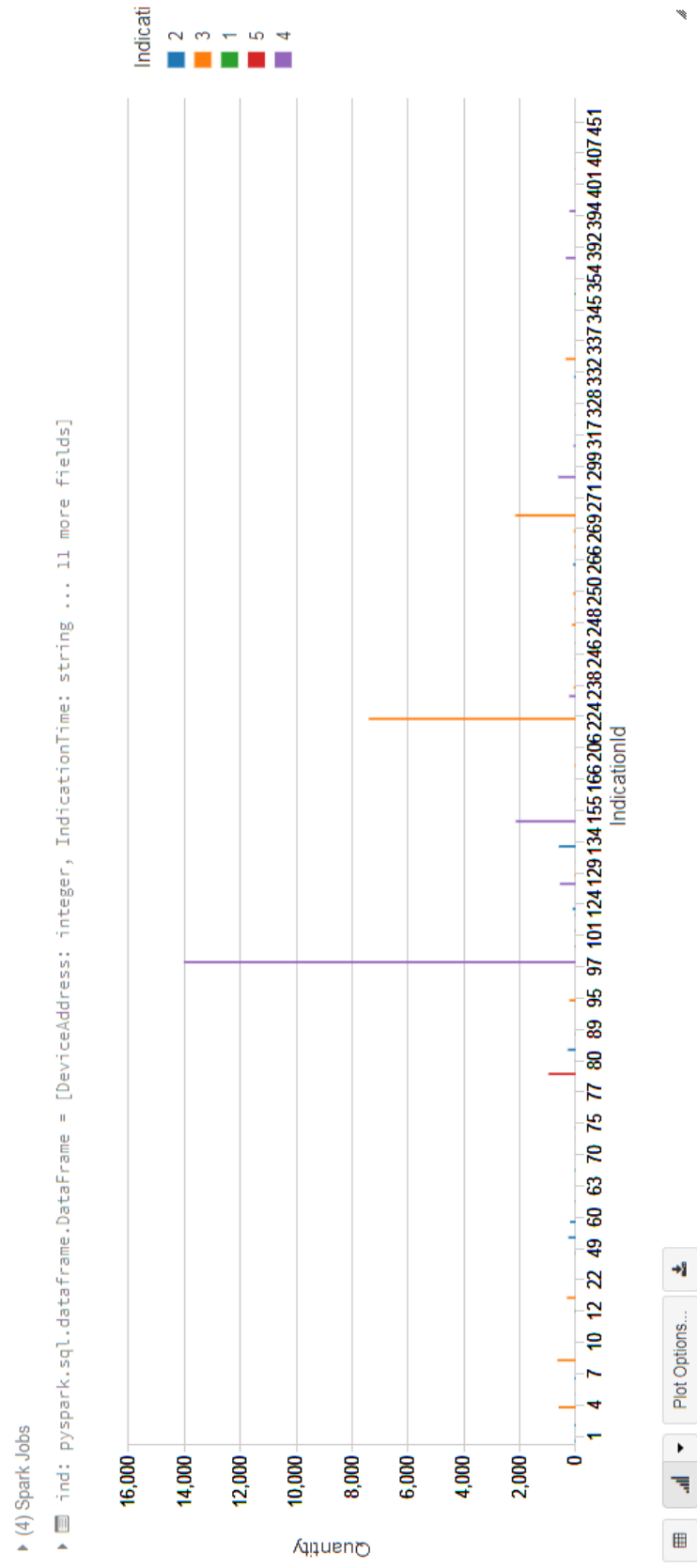
Alarm Tipi Numarası (IndicationTypeId)	Alarm Tipi (IndicationType)	Alarm Numarası (IndicationId)	Alarm Tanımı (IndicationDescription)
4	Report	97	Cleaning
3	Attention	224	Local database error
3	Attention	270	MWS max offset correction exceeded
4	Report	135	Bucket M4USE filled, tagnr.
5	Pulse Alarm	79	Visit cow needs supervision
3	Attention	8	User not present (cow, tag under supervision)
4	Report	272	Downtime ended
3	Attention	4	Cow detected not identified
2	Alarm	134	All buckets M4USE filled
4	Report	189	Boiler
2	Alarm	209	MQC-C: Temp sensor fault
3	Attention	227	Function not configured
2	Alarm	60	Overdue heating boiler time Astronaut
3	Attention	249	Timeout heating Pura
2	Alarm	88	Time out between visit cows



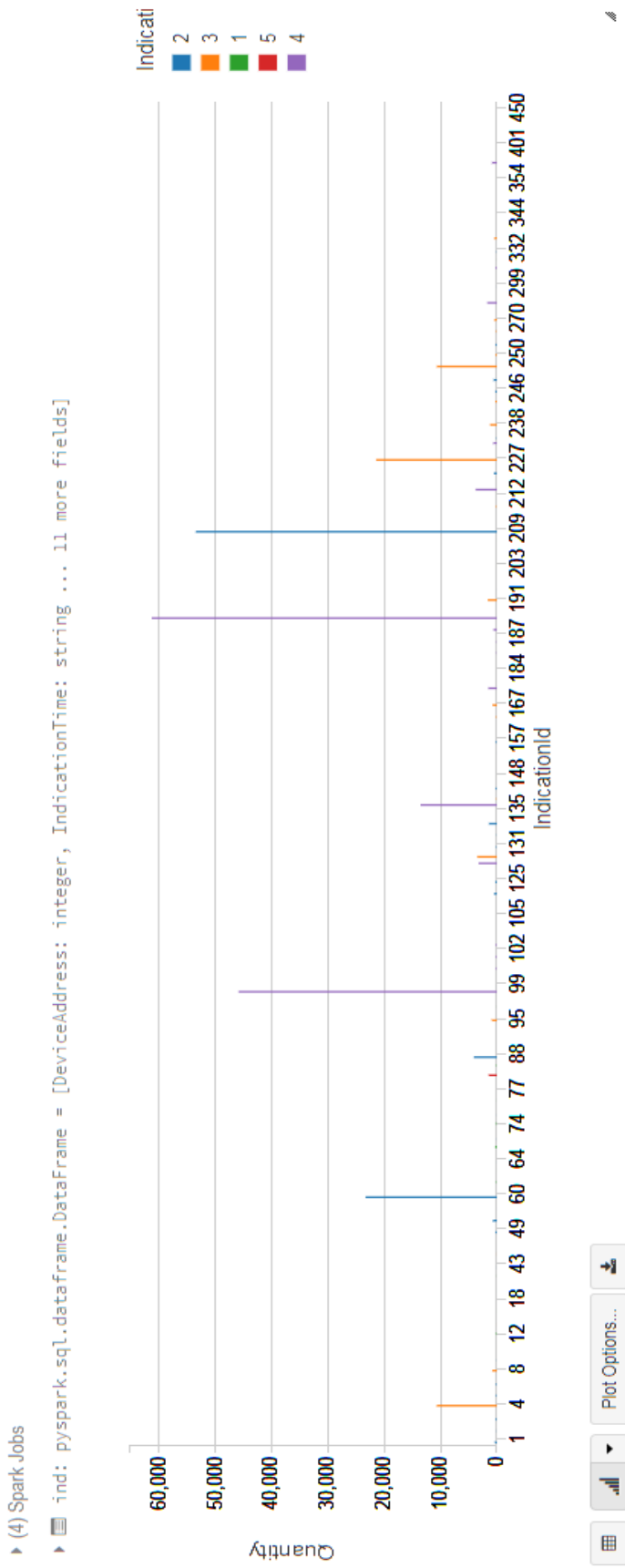
Şekil 5.6. Alarmların gerçekleşme oranları (101)



Şekil 5.7. Alarmların gerçekleşme oranları (102)



Şekil 5.8. Aciliyet seviyesine göre kategorize edilmiş alarmlar ve gerçekleşme miktarları (101)



Şekil 5.9. Aciliyet seviyesine göre kategorize edilmiş alarmlar ve gerçekleşme miktarları (102)

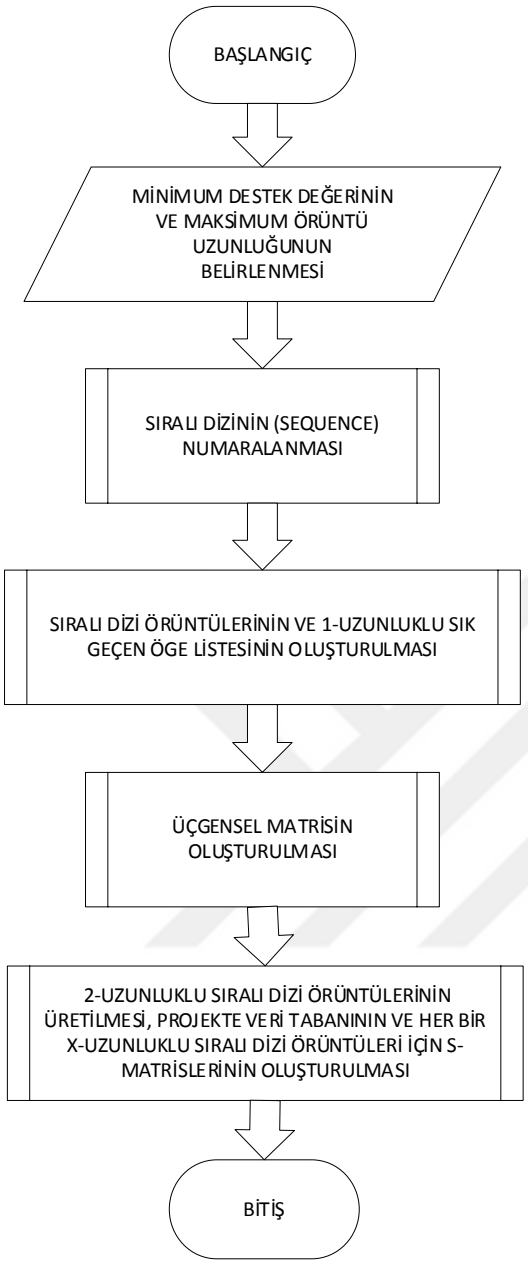
5.3. Algoritmanın Uygulanması

Geçmiş alarm bilgilerinin PrefixSpan algoritması ile analiz edilebilmesi için alarmların gün bazında sıralamasını elde etmek amacı ile veri seti üzerinde SQL sorguları yapılarak alarmların tarihleri ve hangi sırada gerçekleştikleri belirlenmiştir. PrefixSpan algoritması uygulanan verilerin bir kısmı şekil 5.10'da belirtildiği gibidir.

Dates	Indications
18.08.2008	83, 189, 120, 120, 189, 120, 120, 120, 120, 120, 120, 120, 120, 189, 120, 120, 120, 120, 97, 189, 120, 120, 189, 106, 189, 97, 107, 189, 189, 5, 135,
19.08.2008	189, 106, 189, 97, 107, 189, 189, 186, 189, 4, 4, 4, 189, 97, 189, 189, 106, 189, 97, 107, 4, 189, 135, 186, 128, 135, 4, 4, 135, 4, 189, 4, 189, 189, 4,
20.08.2008	97, 189, 106, 189, 97, 107, 189, 189, 191, 189, 189, 189, 97, 189, 189, 106, 189, 97, 107, 4, 4, 4, 189, 135, 135, 135, 128, 189, 108, 109, 97, 97, 189
21.08.2008	189, 106, 189, 97, 107, 189, 189, 189, 189, 97, 189, 189, 106, 189, 97, 107, 189, 135, 128, 135, 135, 79, 189, 135, 189, 189, 189, 146, 189, 189, 97,
22.08.2008	185, 185, 189, 106, 189, 97, 107, 189, 189, 4, 4, 186, 189, 189, 97, 189, 189, 106, 189, 97, 107, 189, 128, 135, 135, 189, 135, 189, 185, 189, 189, 18
23.08.2008	189, 106, 189, 97, 107, 189, 189, 189, 189, 186, 97, 97, 189, 4, 189, 106, 97, 189, 97, 107, 189, 4, 4, 189, 135, 135, 189, 135, 189, 189, 189, 189, 4,
24.08.2008	189, 106, 189, 97, 107, 189, 189, 186, 189, 189, 189, 97, 189, 189, 106, 189, 97, 107, 189, 128, 135, 135, 135, 189, 189, 189, 189, 189, 189, 189, 4,
25.08.2008	189, 106, 97, 189, 97, 107, 189, 97, 189, 189, 189, 97, 189, 189, 106, 189, 97, 107, 189, 135, 128, 135, 135, 189, 189, 189, 189, 189, 189, 189, 97, 1
26.08.2008	4, 189, 106, 189, 97, 107, 189, 189, 189, 189, 186, 186, 189, 97, 189, 189, 106, 189, 97, 107, 189, 135, 128, 135, 189, 189, 4, 4, 189, 189, 186, 189,
27.08.2008	189, 106, 189, 97, 107, 189, 189, 4, 4, 4, 191, 189, 186, 4, 189, 4, 4, 4, 189, 4, 4, 97, 189, 4, 189, 4, 106, 189, 97, 107, 189, 135, 128, 135, 135, 189,
28.08.2008	189, 106, 189, 97, 107, 189, 189, 189, 186, 189, 189, 97, 189, 129, 129, 189, 106, 189, 97, 107, 135, 189, 135, 135, 189, 189, 189, 189, 189, 189, 18
29.08.2008	189, 106, 189, 97, 107, 189, 189, 189, 97, 186, 189, 97, 189, 189, 106, 189, 97, 107, 128, 135, 189, 135, 189, 189, 189, 129, 189, 189, 189, 189, 97,
30.08.2008	189, 106, 189, 97, 107, 4, 185, 5, 189, 5, 5, 5, 5, 79, 135, 189, 189, 4, 97, 189, 170, 4, 189, 106, 189, 97, 107, 97, 69, 120, 135, 135, 83, 120, 97, 189
31.08.2008	189, 106, 189, 97, 107, 97, 189, 189, 185, 5, 5, 97, 97, 189, 191, 189, 106, 189, 97, 107, 79, 135, 134, 97, 189, 135, 97, 186, 135, 97, 135, 97, 128, 1
01.09.2008	189, 106, 189, 97, 107, 4, 189, 186, 189, 4, 191, 189, 146, 4, 97, 189, 4, 189, 106, 189, 97, 107, 4, 4, 4, 189, 146, 135, 146, 135, 146, 135, 97, 184, 1
02.09.2008	189, 97, 107, 189, 97, 189, 4, 185, 189, 106, 189, 97, 107, 191, 189, 185, 185, 189, 97, 4, 4, 95, 97, 189, 4, 189, 106, 189, 97, 107, 135, 128, 189, 13
03.09.2008	185, 189, 106, 211, 189, 97, 107, 185, 189, 189, 189, 189, 97, 189, 97, 189, 106, 186, 189, 97, 107, 186, 189, 135, 135, 135, 97, 189, 4, 189, 108, 10
04.09.2008	189, 106, 52, 189, 97, 107, 4, 189, 189, 189, 189, 97, 189, 4, 189, 106, 97, 189, 97, 107, 135, 189, 135, 135, 135, 128, 189, 189, 189, 189, 189, 189,
05.09.2008	139, 189, 189, 97, 189, 4, 4, 189, 189, 128, 135, 189, 135, 135, 135, 189, 189, 189, 189, 189, 97, 189, 189, 135, 135, 189, 189, 189, 108, 83, 60, 120
06.09.2008	189, 106, 189, 97, 107, 189, 186, 189, 4, 189, 97, 189, 189, 106, 189, 97, 107, 189, 135, 97, 128, 135, 97, 135, 135, 97, 189, 191, 191, 189, 122, 124
07.09.2008	186, 189, 106, 189, 97, 107, 189, 4, 189, 189, 97, 189, 185, 189, 106, 189, 97, 107, 189, 135, 135, 185, 135, 135, 97, 189, 128, 189, 189, 189, 189, 9

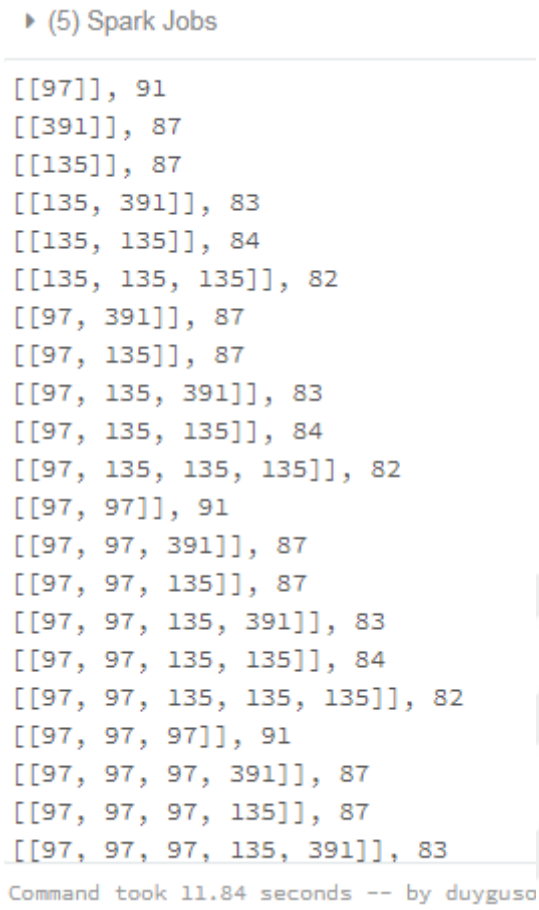
Şekil 5.10. Ön işlemeden geçirilmiş veri

Geçmiş alarm bilgilerinin analiz edilmesi amacı ile PrefixSpan algoritması ile Spark platformu üzerinde Scala programlama diliyle kodlanmıştır. Alarmların analizi günlük olarak sıralanarak mevsimsel bazda yapılmıştır. Minimum destek değeri 0,9 ve maksimum örüntü uzunluğu 10 olarak belirlenerek algoritma veri seti üzerinde uygulanmıştır. PrefixSpan algoritmasının akış şeması şekil 5.11'de belirtildiği gibidir.



Şekil 5.11. Algoritma akış şeması

Cihazların ürettiği alarmların mevsimsel olarak analiz edilmesi amacı ile DeviceAddress (robot konumu) numarası 101 ve 102 olan robotların alarm verileri günlük olarak sıralanmış ve 4 mevsimlik alarm verileri 3'er aylık alınarak işleme tabi tutulmuştur. PrefixSpan algoritmasının çalışma adımlarında anlatıldığı gibi şekil 5.12'de gösterildiği üzere algoritma öncelikle uzunluğu bir olan örüntüleri bulur ve bu sayı kullanıcının belirlediği maksimum örüntü uzunluğuna kadar sürer.

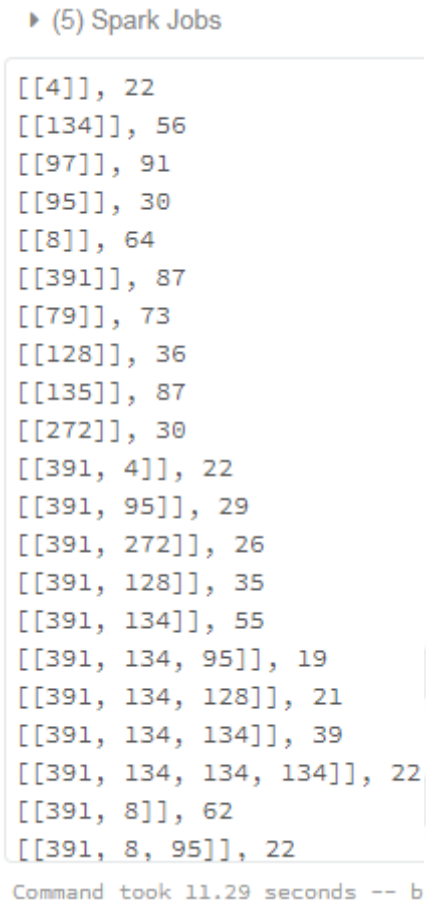


Şekil 5.12. Algoritma sonucu (sonbahar mevsimi)

DeviceAddress (robot konumu) numarası 101 olan robota ait veriler üzerinde algoritma çalıştırıldıktan sonra çıkan 01.09.2015 – 30.11.2015 tarihleri arasındaki sonbahar mevsimi için elde edilen sonuçlar incelendiğinde uzunluğu 1 olan sıralı alt dizilerden 97 (temizlik) numaralı alarmla 91 günde de karşılaşıldığı görülmektedir. Uzunluğu 2 olan sıralı alt dizilerden 135 (kazanın dolması) ve 391 (pompa çalışma saati) şeklinde sıralanan alarmlarla 83 günde de karşılaşıldığı görülmektedir. Benzer şekilde kullanıcının belirlediği maksimum örüntü uzunluğuna kadar sıralı alt diziler bulunmaktadır. Ancak bulunan bu alarm dizilerinin aciliyet seviyesi 4 (report)'tür, yani kullanıcıyı bilgilendirme amacıyla verilen bildirim şeklinde alarmlardır. Yaklaşık olarak her gün verilen bu alarmların aciliyet seviyesinin düşük olması çiftlikler açısından olumlu bir sonuçtur. Aksi takdirde aciliyet seviyesi yüksek olan alarmların bu denli sık görülmesi çiftliklerin çalışmasını büyük oranda aksatırdı.

Bu çalışmanın amacının aciliyet seviyesi yüksek olan alarmların tespiti ve bu alarmların diğer alarmlar ile aralarındaki ilişkilerin bulunması olması sebebiyle PrefixSpan

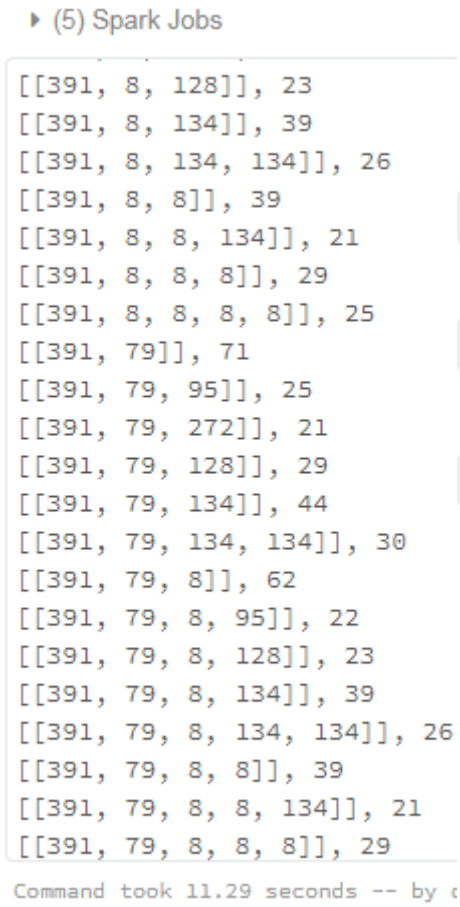
algoritması, minimum destek değeri 0,2 ve maksimum örüntü uzunluğu 10 olarak belirlenerek veri seti üzerinde yeniden uygulanmıştır. 01.09.2015 – 30.11.2015 tarihleri arasındaki sonbahar mevsimi için elde edilen yeni sonuçlar şekil 5.13'te gösterildiği gibidir. Bu sonuçlar incelendiğinde uzunluğu 1 olan sıralı alt dizilerden 4 (algılanan ineğin etiket numarasının belirlenememesi) numaralı alarmla 22 günde de karşılaşıldığı görülmektedir. Uzunluğu 2 olan sıralı alt dizilerden 391 (pompa çalışma saati) ve 4 (algılanan ineğin etiket numarasının belirlenememesi) şeklinde sıralanan alarmlarla 22 günde de karşılaşıldığı görülmektedir. Benzer şekilde kullanıcının belirlediği maksimum örüntü uzunluğuna kadar sıralı alt diziler bulunmaktadır. Birçok günde karşılaşılan 4 (algılanan ineğin etiket numarasının belirlenememesi), 8 (etiket numarası belirli olan bir ineğin denetlenmesi gerektiği halde bir kullanıcının mevcut olmaması), 95 (RCS alarmı), 134 (bütün kazanların dolması) gibi alarmların aciliyet seviyeleri 2 (alarm) ve 3 (attention)'tür, yani daha sık görülen aciliyet seviyesi düşük olan alarmlara göre daha yüksektir. Dolayısıyla aciliyet seviyeleri yüksek olan alarmların tespiti ve bu alarmların diğer alarmlar ile aralarındaki ilişkilerin bulunması amacıyla sonraki analizler, minimum destek değeri 0,2 ve maksimum örüntü uzunluğu 10 olarak belirlenerek veri seti üzerinde uygulanmaya devam edilmiştir.



Şekil 5.13. Algoritma sonucu (sonbahar mevsimi)

Sonuçlarda görülen örüntü uzunluğu, belirlenen maksimum örüntü uzunluğuna kadar devam etmektedir. Şekil 5.13'te verilen sonucun devamının bir kısmı şekil 5.14'te verildiği gibidir. Bu sonuçlardan bir örnek olan "391, 79, 8, 8, 8" şeklinde sıralanan alarmlarla 29 günde de karşılaşıldığı görülmektedir. Bu alarmlardan IndicationId yani alarm numarası 391 olan alarmın aciliyet seviyesi 4 (report), alarm numarası 79 olan alarmın aciliyet seviyesi 5 (pulse alarm) iken alarm numarası 8 olan alarmın aciliyet seviyesi 3 (attention)'tür. Bu alarmların tanımlarına bakıldığında 391 numaralı alarm pompanın çalışma saatini, 79 numaralı alarm ineklerin denetlenmesi gerektiğini ve 8 numaralı alarm etiket numarası belirli olan bir ineğin denetlenmesi gerektiği halde bir kullanıcının mevcut olmadığını belirtmektedir. Bu sıralı diziden çıkarılan sonuç dolayısıyla 79 numaralı alarmdan sonra 8 numaralı alarmın sık sık verilmesi, gerekli denetimlerin yapılmadığının göstergesidir. Elde edilen örüntülerin incelemesine devam edildiğinde "135, 135, 134, 134, 134" şeklinde sıralanan alarmlarla 22 günde de karşılaşıldığı görülmektedir. Alarm numarası 135 olan alarmın aciliyet seviyesi 4 (report) iken alarm numarası 134 olan alarmın aciliyet seviyesi 2 (alarm)'dır. Bu alarmların tanımlarına bakıldığında 135

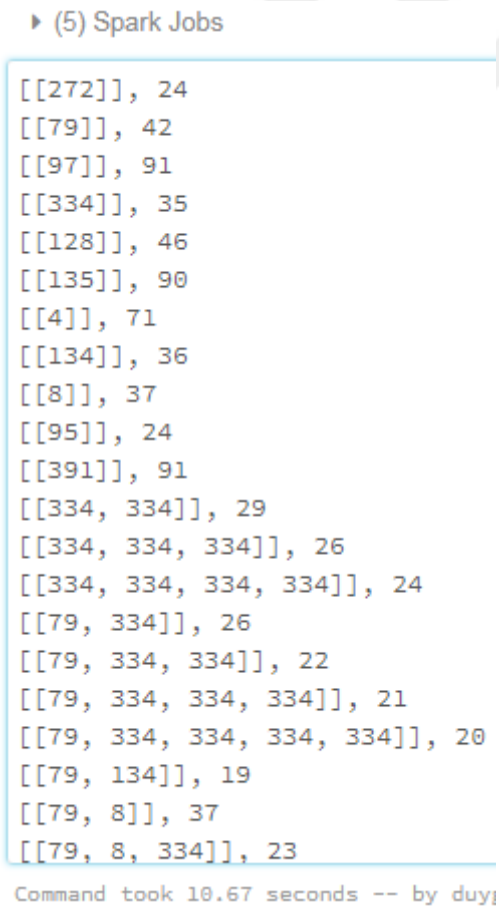
numaralı alarm süt kazanının dolduğunun, 134 numaralı alarm ise bütün kazanların dolduğunun göstergesidir. 135 numaralı alarmın arka arkaya verilmesinden sonra 134 numaralı alarmın da sık sık verilmesi, 135 numaralı alarmın daha fazla dikkate alınarak bütün süt kazanları dolmadan yani 134 numaralı alarmın sıkça tekrarlanmasına sebebiyet vermeyecek şekilde önlem alınması gerektiğini gösterir. Bu alarm dizilerinden çıkarılan sonuçların incelenmesiyle çiftliklerde görülen problemlerin ve işleyiş aksamalarının temel kaynaklarına inilebilecektir.



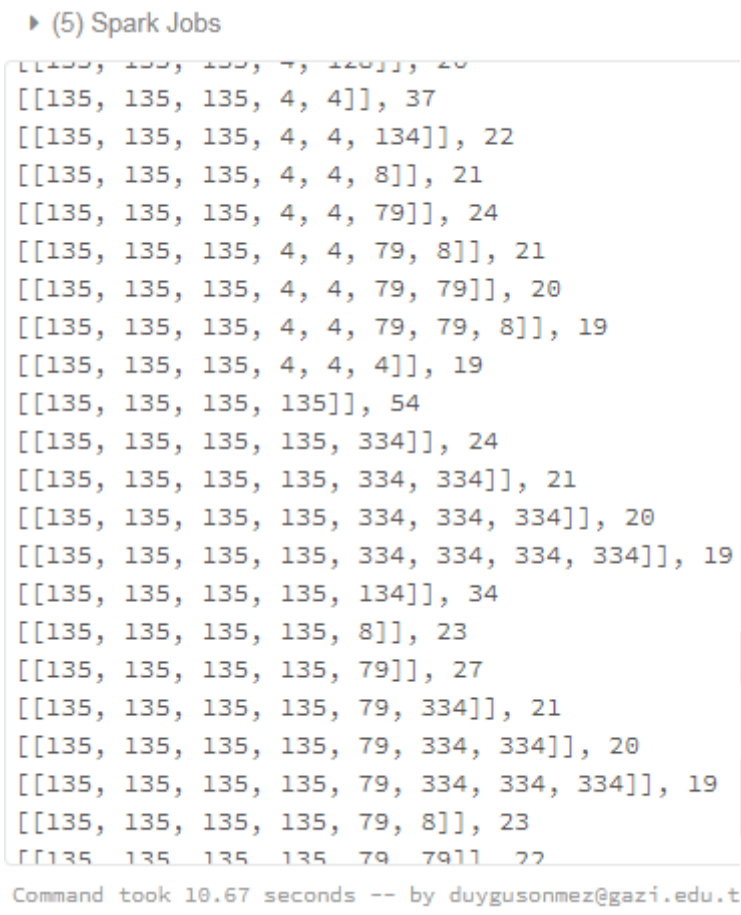
Şekil 5.14. Algoritma sonucu (sonbahar mevsimi-devam)

01.12.2015 – 29.02.2016 tarihleri arasındaki kış mevsimi için elde edilen sonuçların bir kısmı şekil 5.15'te ve şekil 5.16'da verilmiştir. Bir önceki sonbahar mevsiminden farklı olarak görülen aciliyet seviyesi 3 (attention) olan 334 numaralı alarmın tanımı MQC-C cihazının çalışmayı durdurmasıdır ve bu alarmla 35 günde de karşılaşılmıştır. Bu cihazın özellikle kış mevsiminde hangi sebeple bu kadar fazla çalışmayı durdurmasının sebepleri araştırılmalıdır. Mevsimsel bazda yapılan oransal analizlerde kış mevsiminde aciliyet seviyesi 3 (attention) olan alarmların daha fazla verilmesinin sebeplerine bu gibi

incelemelerle ulaşılabilecektir. Bir başka örnek olan ve 21 günde de görülen “135, 135, 135, 4, 4, 79, 8” şeklinde sıralanan alarm dizisi incelendiğinde ise 135 numaralı alarmın aciliyet seviyesi 4 (report), 4 numaralı alarmın aciliyet seviyesi 3 (attention), 79 numaralı alarmın aciliyet seviyesi 5 (pulse alarm), 8 numaralı alarmın aciliyet seviyesi 3 (attention)’tür. Bu alarmların tanımlamalarına bakıldığında 4 numaralı alarm sağım için gelen bir inek olduğunun algılandığı ancak bu ineğin etiket numarasının belirlenemediğini ifade eder. Bu alarmın ardından verilen 79 numaralı alarm ineğin denetlenmesi gerektiğini ve 8 numaralı alarm ineğin denetlenmesi gerektiği halde bir kullanıcı bulunmadığını ifade eder. Bu sonuçlar ineklerin etiketlerinin kontrol edilmesi, inek denetimlerinin sıklaştırılması gerektiği ortaya çıkarmaktadır. Bu mevsimde de 135 (süt kazanının dolması) numaralı alarmın arka arkaya verilmesinden sonra 134 (bütün süt kazanlarının dolması) numaralı alarmın da sık sık verilmesi, 135 numaralı alarmın daha fazla dikkate alınarak bütün süt kazanları dolmadan yani 134 numaralı alarmın sıkça tekrarlanmasına sebebiyet vermeyecek şekilde önlem alınması gerektiğini gösterir.



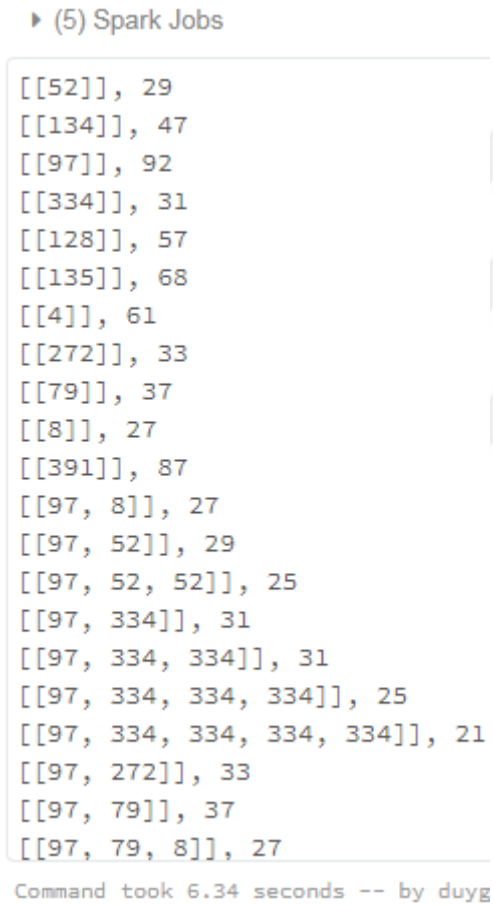
Şekil 5.15. Algoritma sonucu (kış mevsimi)



Şekil 5.16. Algoritma sonucu (kış mevsimi-devam)

01.03.2016 - 31.05.2016 tarihleri arasındaki ilkbahar mevsimi için elde edilen sonuçların bir kısmı şekil 5.17'de verilmiştir. Önceki mevsimlerden farklı olarak aciliyet seviyesi 2 (alarm) olan 52 numaralı alarmla 29 günde de karşılaşıldığı görülmektedir. Bu alarmın tanımı temizlik deterjanının bittiğini ifade etmektedir. Bu alarmla bu kadar fazla karşılaşılmaması için deterjan miktarının yenilenmesi sıklığı artırılması veya deterjanın bulunduğu kısmın boyutunun büyütülmesi gerekebilir. Sonraki örüntüler incelendiğinde "97, 97, 97, 97, 52, 52" şeklinde sıralanan alarm dizisinin 25 günde de görüldüğü belirlenmiştir. Temizlik yapıldığını ifade eden 97 numaralı alarmın aciliyet seviyesi 4 (report) iken temizlik deterjanının bittiğini ifade eden 52 numaralı alarmın aciliyet seviyesi 2 (alarm)'dir. Arka arkaya çok fazla temizlik yapıldığını ifade eden alarmın verilmesinden sonra temizlik deterjanının bittiğini ifade eden alarmın tekrarlanarak verilmesi, belirli bir temizlik miktarından sonra deterjan miktarının yenilenmesi veya deterjanın bulunduğu kısmın boyutunun büyütülmesi gerekebileceğinin göstergesidir. Bu mevsimde de 135 (süt kazanının dolması) numaralı alarmın arka arkaya verilmesinden sonra 134 (bütün süt

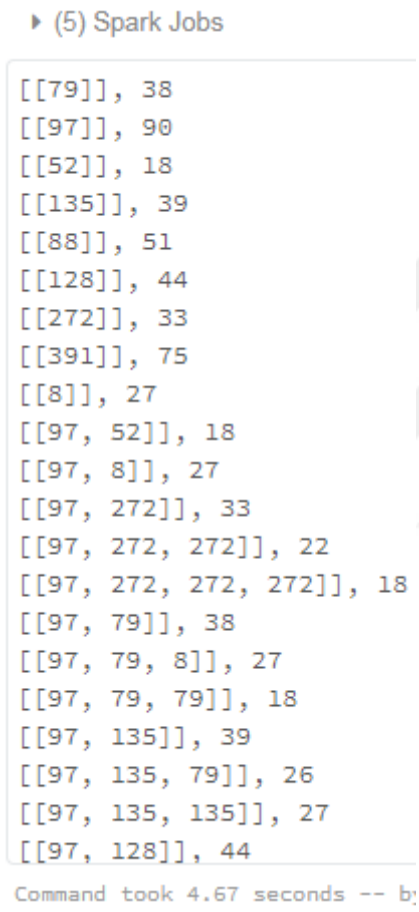
kazanlarının dolması) numaralı alarmın da sık sık verilmesi, 135 numaralı alarmın daha fazla dikkate alınarak bütün süt kazanları dolmadan yani 134 numaralı alarmın sıkça tekrarlanmasına sebebiyet vermeyecek şekilde önlem alınması gerektiğini gösterir. Önceki mevsimde olduğu gibi 4 (algılanan ineğin etiket numarasının belirlenememesi) numaralı alarmın ardından 79 (ineğin denetlenmesi gerektiği) numaralı alarmın ve 8 (ineğin denetlenmesi gerektiği halde bir kullanıcı bulunmadığı) numaralı alarmın sıralı dizisinin görülmesi, ineklerin etiketlerinin kontrol edilmesi ve inek denetimlerinin sıklaştırılması gerektiğini ortaya çıkarmaktadır.



Şekil 5.17. Algoritma sonucu (ilkbahar mevsimi)

01.06.2016 - 30.08.2016 tarihleri arasındaki yaz mevsimi için elde edilen sonuçların bir kısmı şekil 5.18'de verilmiştir. Önceki mevsimlerden farklı olarak aciliyet seviyesi 2 (alarm) olan 88 numaralı alarmla 51 günde de karşılaşıldığı görülmektedir. Alarm tanımına bakıldığında 88 numaralı alarm, ineklerin ziyaret süreleri arasında zaman aşımı yaşandığını ifade etmektedir. Yaz aylarında bu alarmla hangi nedenden dolayı bu kadar fazla karşılaşıldığı belirlenmeli ve önlem alınmalıdır. Önceki mevsimde olduğu gibi bu mevsimde de 97 (temizlik yapılması) numaralı alarmın tekrarlamasının ardından 52

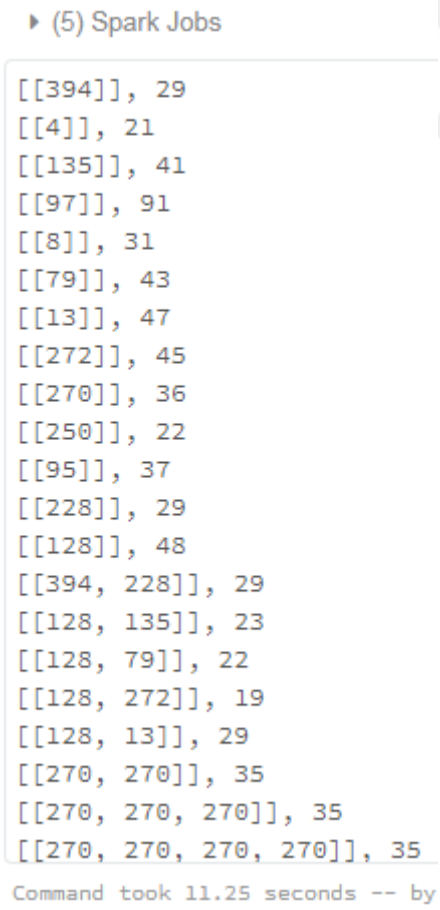
(temizlik deterjanının bitmesi) numaralı alarmın verilmesi, belirli bir temizlik miktarından sonra deterjan miktarının yenilenmesi veya deterjanın bulunduğu kısmın boyutunun büyütülmesi gerekebileceğinin göstergesidir. Önceki mevsimlerde görülen örüntülere benzer olarak 79 (ineğin denetlenmesi gerektiği) numaralı alarmın ardından 8 (ineğin denetlenmesi gerektiği halde bir kullanıcı bulunmadığı) numaralı alarmın görülmesi, ineklerin etiketlerinin kontrol edilmesi ve inek denetimlerinin sıklaştırılması gerektiğini ortaya çıkarmaktadır.



Şekil 5.18. Algoritma sonucu (yaz mevsimi)

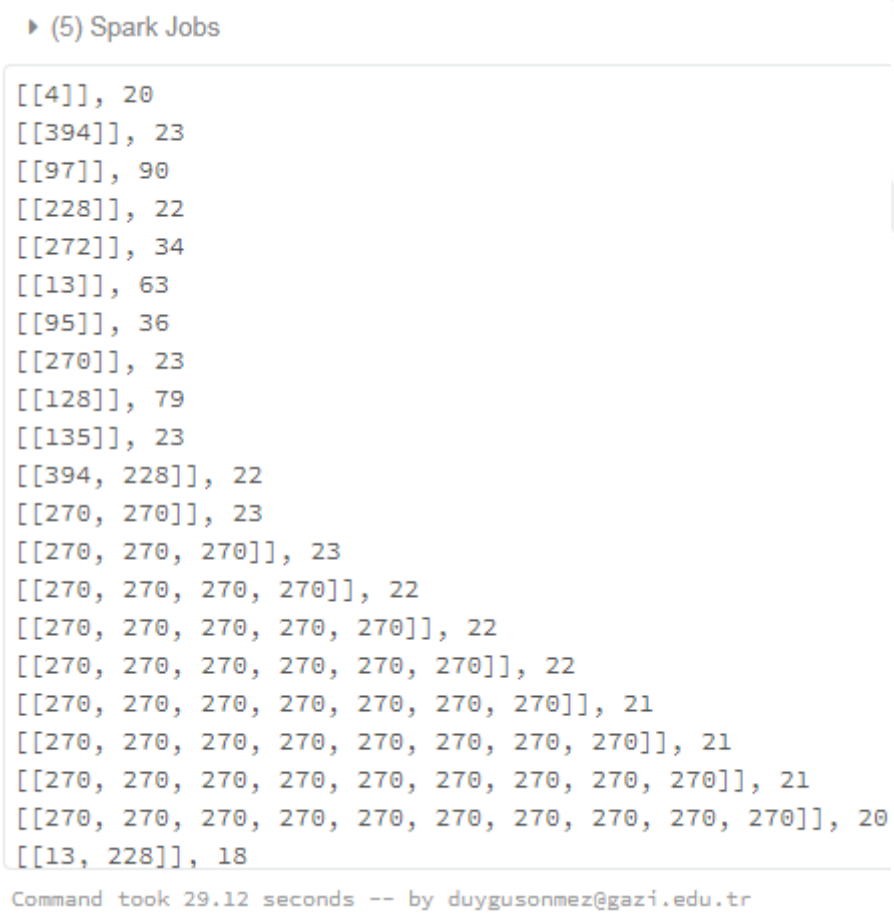
01.09.2016 - 30.11.2016 tarihleri arasındaki sonbahar mevsimi için elde edilen sonuçların bir kısmı şekil 5.19'da verilmiştir. Önceki mevsimlerden farklı olarak aciliyet seviyesi 3 (attention) olan 13 (temizlikten sonra iletkenlik değerinin aşılması), 250 (milkcup/liner cihazının temiz olmaması), 270 (MWS cihazının maksimum offset düzeltmesinin aşılması) numaralı alarmlarla 47, 22 ve 36 günde de karşılaşıldığı görülmektedir. Bu alarmlarla diğer zamanlardan farklı olarak bu kadar fazla karşılaşılmasının sebepleri araştırılarak önlemler alınmalıdır. Sonrasındaki örüntüler incelendiğinde "128, 13" sıralı alarm dizisi ile 29 günde de karşılaşıldığı görülmektedir. Aciliyet seviyesi 4 (report) olan 128 numaralı

alarmın tanımı süt filtresinin değiştirilmesi ve aciliyet seviyesi 3 (attention) olan 13 numaralı alarmın tanımı temizlikten sonra iletkenlik değerinin aşılmasıdır. Süt filtresinin değiştirilmesinden sonra iletkenlik değerinin aşılmasının bu kadar sık görülmesi ve temizlikle ilgili bu durumun alarm olarak verilmesinin sebebi araştırılmalı ve probleme yönelik önlem alınmalıdır. Aynı şekilde 97 (temizlik yapılması) numaralı alarmın ardından 13 (temizlikten sonra iletkenlik değerinin aşılması) numaralı alarmla 47 günde de karşılaşılması temizlik konusunda düzeltmeler yapılması gerektiğinin göstergesi olabilir. Benzer bir durum olarak 13 (temizlikten sonra iletkenlik değerinin aşılması) numaralı alarmın ardından 270 (MWS cihazının maksimum offset düzeltmesinin aşılması) numaralı alarmla çok sık karşılaşılmasının sebebi araştırılmalıdır. Önceki mevsimlerde olduğu gibi 79 (ineğin denetlenmesi gerektiği) numaralı alarmın ardından 8 (ineğin denetlenmesi gerektiği halde bir kullanıcı bulunmadığı) numaralı alarmın tekrarlanarak görülmesi, ineklerin etiketlerinin kontrol edilmesi ve inek denetimlerinin sıklaştırılması gerektiğini ortaya çıkarmaktadır.



Şekil 5.19. Algoritma sonucu (sonbahar mevsimi)

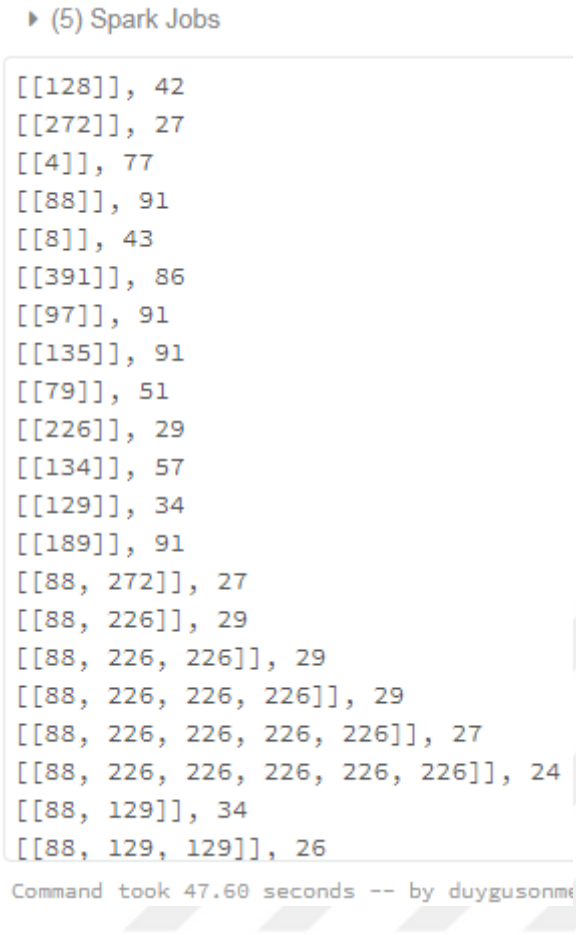
01.12.2016 - 28.02.2017 tarihleri arasındaki kış mevsimi için elde edilen sonuçların bir kısmı şekil 5.20'de verilmiştir. Bir önceki mevsimde olduğu gibi aciliyet seviyesi 3 (attention) olan 13 (temizlikten sonra iletkenlik değerinin aşılması) ve 270 (MWS cihazının maksimum offset düzeltmesinin aşılması) numaralı alarmlarla sıkça karşılaşılmasının sebepleri araştırılarak görülen problemlere yönelik önlemler alınmalıdır. Önceki mevsimlerle aynı şekilde 128 (süt filtresinin değiştirilmesi) numaralı alarmın ardından 13 (temizlikten sonra iletkenlik değerinin aşılması) numaralı alarmın sıkça görülmesi ve 97 (temizlik yapılması) numaralı alarmın ardından 13 (temizlikten sonra iletkenlik değerinin aşılması) numaralı alarmla sıkça karşılaşılması, ayrıca "97, 128, 13" şeklinde sıralı olan örüntünün sıkça görülmesi temizlik konusunda düzeltmeler yapılması gerektiğinin göstergesi olabilir.



Şekil 5.20. Algoritma sonucu (kış mevsimi)

DeviceAddress (robot konumu) numarası 102 olan robota ait veriler üzerinde algoritma çalıştırıldıktan sonra çıkan 01.09.2015 – 30.11.2015 tarihleri arasındaki sonbahar mevsimi için elde edilen sonuçların bir kısmı şekil 5.21'de verilmiştir. Önceki mevsimlerden farklı olarak aciliyet seviyesi 2 (alarm) olan 226 numaralı alarmla 29 günde de karşılaşıldığı

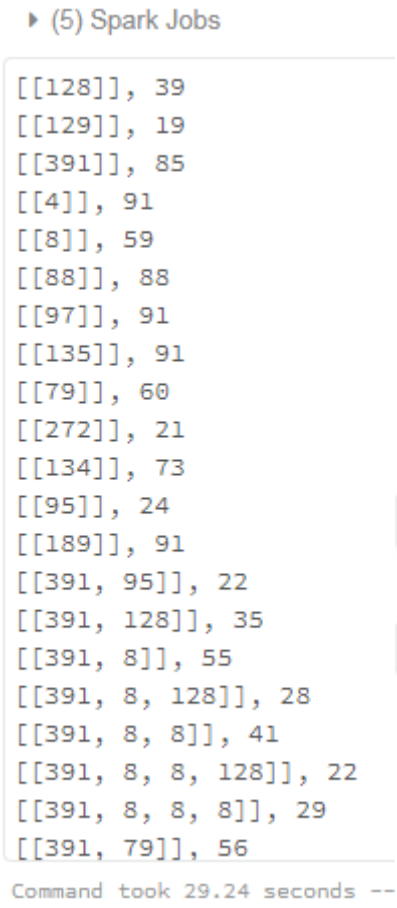
görülmektedir. Bu alarmın tanımı, sağım robotundaki temizlik fırçasıyla ilgili bir döngünün çok sık gerçekleştiğini ifade etmektedir. Bu alarmla bir mevsimde bu kadar fazla karşılaşılması fırçayla ilgili problemin uzun süredir giderilmediğinin ve bu cihazların kontrolünün daha sık yapılması gerektiğinin göstergesidir. Aciliyet seviyesi 3 (attention) olan ve 34 günde de görülen 129 numaralı alarmın tanımı, sağıma gelen ineğin etiket numarasının sağım süresince farklı farklı okunmasıdır. Bu durum, ineklerin etiketlerinin kontrolünün sıklaştırılması gerektiğinin göstergesi olabilir. Sonraki örüntüler incelendiğinde alarm numarası 4 (algılanan ineğin etiket numarasının belirlenememesi) ve 129 (sağıma gelen ineğin etiket numarasının sağım süresince farklı farklı okunması) şeklinde sıralanan alarmların görülme sıklığının yüksek olması, bu gerekliliği desteklemektedir. Önceki analizlerde belirlendiği üzere yaz mevsiminde sık görülen, aciliyet seviyesi 2 (alarm) olan 88 (ineklerin ziyaret süreleri arasında zaman aşımı yaşanması) numaralı alarmla 91 günde de karşılaşıldığı görülmektedir. Bu alarmın sonbahar mevsiminde de hangi nedenden dolayı bu kadar fazla görüldüğü belirlenmeli ve önlem alınmalıdır. Ayrıca 88 (ineklerin ziyaret süreleri arasında zaman aşımı yaşanması) numaralı alarmdan sonra 129 (sağıma gelen ineğin etiket numarasının sağım süresince farklı farklı okunması) veya 226 (sağım robotundaki temizlik fırçasıyla ilgili bir döngünün çok sık gerçekleşmesi) numaralı alarmın çok sık verildiği günler bulunmaktadır. Aciliyet seviyeleri kritik olan bu alarmlar arasındaki ilişkinin belirlenmesiyle yaşanan probleme daha köklü çözümler bulunabilir. Bu mevsimde de 135 (süt kazanının dolması) numaralı alarmın arka arkaya verilmesinden sonra 134 (bütün süt kazanlarının dolması) numaralı alarmın da sık sık verilmesi, 135 numaralı alarmın daha fazla dikkate alınarak bütün süt kazanları dolmadan yani 134 numaralı alarmın sıkça tekrarlanmasına sebebiyet vermeyecek şekilde önlem alınması gerektiğini gösterir. Önceki mevsimlerde olduğu gibi 4 (algılanan ineğin etiket numarasının belirlenememesi) numaralı alarmın ardından 79 (ineğin denetlenmesi gerektiği) numaralı alarmın ve 8 (ineğin denetlenmesi gerektiği halde bir kullanıcı bulunmadığı) numaralı alarmın sıralı dizisinin görülmesi, ineklerin etiketlerinin kontrol edilmesi ve inek denetimlerinin sıklaştırılması gerektiğini ortaya çıkarmaktadır.



Şekil 5.21. Algoritma sonucu (sonbahar mevsimi)

01.12.2015 – 29.02.2016 tarihleri arasındaki kış mevsimi için elde edilen sonuçların bir kısmı şekil 5.22'de verilmiştir. Önceki analizlerde belirlendiği üzere yaz ve sonbahar mevsimlerinde sık görülen, aciliyet seviyesi 2 (alarm) olan 88 (ineklerin ziyaret süreleri arasında zaman aşımı yaşanması) numaralı alarmla 88 günde de karşılaşıldığı görülmektedir. Bu alarmın mevsimden bağımsız olarak hangi nedenden dolayı bu kadar fazla görüldüğü belirlenmeli ve önlem alınmalıdır. Aynı şekilde aciliyet seviyesi 3 (attention) olan 129 (sağıma gelen ineğin etiket numarasının sağım süresince farklı farklı okunması) numaralı alarmın sık görülmesi, ineklerin etiketlerinin kontrolünün hala gerçekleştirilmediğinin göstergesi olabilir. Bu mevsimde de 135 (süt kazanının dolması) numaralı alarmın arka arkaya verilmesinden sonra 134 (bütün süt kazanlarının dolması) numaralı alarmın da sık sık verilmesi, 135 numaralı alarmın daha fazla dikkate alınarak bütün süt kazanları dolmadan yani 134 numaralı alarmın sıkça tekrarlanmasına sebebiyet vermeyecek şekilde önlem alınması gerektiğini gösterir. Önceki mevsimlerde olduğu gibi 4 (algılanan ineğin etiket numarasının belirlenememesi) numaralı alarmın ardından 79 (ineğin denetlenmesi gerektiği) numaralı alarmın ve 8 (ineğin denetlenmesi gerektiği halde

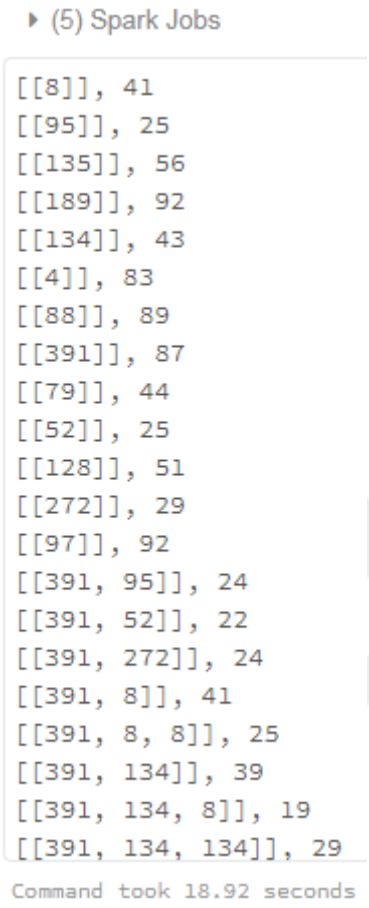
bir kullanıcı bulunmadığı) numaralı alarmın sıralı dizisinin görülmesi, ineklerin etiketlerinin kontrol edilmesi ve inek denetimlerinin sıklaştırılması gerektiğini ortaya çıkarmaktadır.



Şekil 5.22. Algoritma sonucu (kış mevsimi)

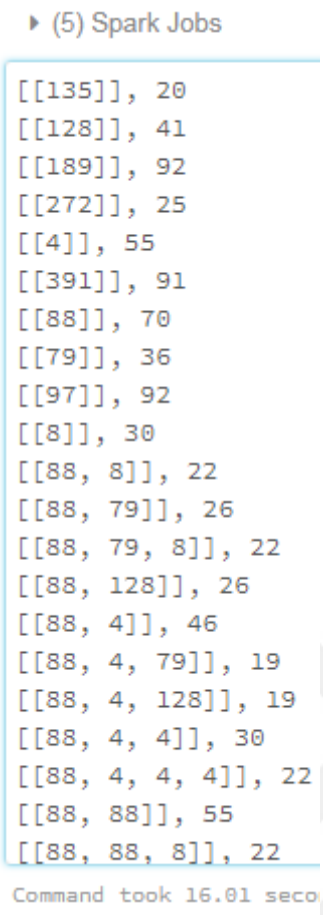
01.03.2016 - 31.05.2016 tarihleri arasındaki ilkbahar mevsimi için elde edilen sonuçların bir kısmı şekil 5.23'te verilmiştir. Önceki mevsimlerde olduğu gibi aciliyet seviyesi 2 (alarm) olan 52 (temizlik deterjanının bitmesi) numaralı alarmla 25 günde de karşılaşıldığı görülmektedir. Bu alarmla bu kadar fazla karşılaşılmaması için deterjan miktarının yenilenmesi sıklığı artırılması veya deterjanın bulunduğu kısmın boyutunun büyütülmesi gerekebilir. Bu mevsimde de 135 (süt kazanının dolması) numaralı alarmın arka arkaya verilmesinden sonra 134 (bütün süt kazanlarının dolması) numaralı alarmın da sık sık verilmesi, 135 numaralı alarmın daha fazla dikkate alınarak bütün süt kazanları dolmadan yani 134 numaralı alarmın sıkça tekrarlanmasına sebebiyet vermeyecek şekilde önlem alınması gerektiğini gösterir. Önceki mevsimlerde olduğu gibi 4 (algılanan ineğin etiket numarasının belirlenememesi) numaralı alarmın ardından 79 (ineğin denetlenmesi

gerektiği) numaralı alarmın ve 8 (ineğin denetlenmesi gerektiği halde bir kullanıcı bulunmadığı) numaralı alarmın sıralı dizisinin görülmesi, ineklerin etiketlerinin kontrol edilmesi ve inek denetimlerinin sıklaştırılması gerektiğini ortaya çıkarmaktadır.

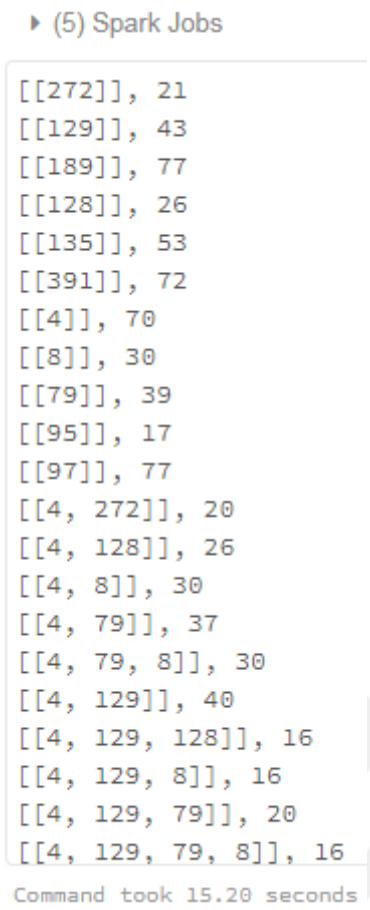


Şekil 5.23. Algoritma sonucu (ilkbahar mevsimi)

01.06.2016 - 30.08.2016 tarihleri arasındaki yaz mevsimi ve 01.09.2016 - 16.11.2016 tarihleri arasındaki sonbahar mevsimi için elde edilen sonuçların bir kısmı şekil 5.24 ve 5.25'te verilmiştir. Bu mevsimlerde de önceki mevsimlerle aynı sonuçlara ulaşılmıştır.



Şekil 5.24. Algoritma sonucu (yaz mevsimi)



Şekil 5.25. Algoritma sonucu (sonbahar mevsimi)

6. SONUÇ

Bu tez çalışmasında büyük veri konusu ele alınmış, büyük verilerin yönetimi ve büyük veriden otomatik olarak faydalı bilgilerin çıkarımı konusundaki yeni teknoloji ve yöntemler incelenmiştir. Sıralı örüntü madenciliği ve algoritmaları anlatılarak kullanılan PrefixSpan algoritması açıklanmıştır.

Bu tez çalışmasında akıllı çiftliklerden elde edilen büyük ölçekli alarm verileri, veriler sayısal ve oransal bazda Apache Spark platformunda Python programlama dili kullanılarak analiz edilmiştir. Büyük veri üzerinde sıralı örüntü madenciliği problemini ele alabilecek PrefixSpan algoritması Apache Spark MLlib teknolojisi kullanılarak Scala programlama diliyle bu veriler üzerinde uygulanmıştır.

Yapılan analizlerin sonucunda mevsimlerin verilen alarmlar üzerinde bir etkisi olabileceği tespit edilmiştir. Özellikle kış mevsimlerinde elde edilen sonuçlarda aciliyet seviyesi yüksek olan alarmların daha fazla gerçekleştiği görülmüştür. Örnek olarak MQC-C cihazının çalışmayı durdurduğu günlerin özellikle kış aylarında diğer mevsimlere oranla çok daha fazla görülmesi verilebilir, bunun sonucunda bu aksamanın sebepleri araştırılmalı ve gerekli önlemler alınmalıdır. İlkbahar mevsiminde temizlik deterjanının bittiğini ifade eden alarmın sıkça görüldüğü tespit edilmiştir. Bu alarmla çok fazla karşılaşılmaması için deterjan miktarının yenilenmesi sıklığı artırılması veya deterjanın bulunduğu kısmın boyutunun büyütülmesi gerekebilir. Mevsimlerin geneline bakıldığında çeşitli örüntüler ortak olarak görülmüştür. Örnek olarak sağım için gelen bir inek olduğunun algılandığı ancak bu ineğin etiket numarasının belirlenemediğini ifade eden alarmın ardından verilen alarmlar ineğin denetlenmesi gerektiğini ve ineğin denetlenmesi gerektiği halde bir kullanıcı bulunmadığını belirtmektedir. Başka bir örneğe bakıldığında algılanan ineğin etiket numarasının belirlenemediğini ifade eden alarmın ardından sağıma gelen ineğin etiket numarasının sağım süresince farklı farklı okunduğunu ifade eden alarmın görülme sıklığının yüksek olduğu görülmektedir. Bu sonuçlardan dolayı ineklerin etiketlerinin kontrol edilmesi, inek denetimlerinin sıklaştırılması gerektiği ortaya çıkmaktadır. Süt kazanının dolması alarmının arka arkaya verilmesinden sonra bütün süt kazanlarının dolması alarmının da sık sık verilmesi, bu alarmların daha fazla dikkate alınarak bütün süt kazanları dolmadan önlem alınması gerektiğini veya kullanılan süt kazanlarının hacminin

genişletilmesi gerektiğini gösterir. Arka arkaya çok fazla temizlik yapıldığını ifade eden alarmın verilmesinden sonra temizlik deterjanının bittiğini ifade eden alarmın tekrarlanarak verilmesi, belirli bir temizlik miktarından sonra deterjan miktarının yenilenmesi veya deterjanın bulunduğu kısmın boyutunun büyütülmesi gerekebileceğinin göstergesidir. Benzer şekilde temizlik yapılması alarmının ardından süt filtresinin değiştirilmesi ve temizlikten sonra iletkenlik değerinin aşılması alarmlarının sıkça görülmesi temizlik konusunda düzeltmeler yapılması gerektiğinin göstergesi olabilir. PrefixSpan algoritması sonucu elde edilen örüntülerin bu şekilde incelenmesi aracılığıyla bu alarmların temel sebeplerine inilebildiği, alarmların verildiği sıraların tespit edilmesi sonucu alarmlar arasında ilişkiler bulunduğu ve mevcut problemlere getirilebilecek çözüm önerileri sunulabileceği gösterilmiştir.

Akıllı çiftliklerdeki geçmiş alarm bilgileri analiz edilerek yapılan çalışmanın temel amacı alarmlar arasındaki ilişkilerin ve örüntülerin bulunarak alarmların temel kaynaklarına inilmesidir. Birbiriyle ilişkili alarmların ve örüntülerin belirlenmesi, alarmların gerçek kaynaklarına inilmesini kolaylaştırmaktadır. Verilerin analiz edilmesi ile geçmiş alarm verilerine dayanılarak oluşması muhtemel sorunların önceden tahmin edilebilmesiyle sistemlerde oluşabilecek arızlar erken tespit edilerek bakım süreçlerinin buna göre yönetilmesi sonucu maliyetlerde önemli düşüşler gerçekleştirilebilecektir.

KAYNAKLAR

1. Agrawal, R. and Srikant, R. (1995, March 6-10). Mining sequential patterns. *Eleventh International Conference on Data Engineering, IEEE,* 3-14.

2. Agrawal, R. and Srikant, R. (1996, March 25-29). Mining sequential patterns: Generalizations and performance improvements. *Advances in Database Technology, International Conference on Extending Database Technology, Springer*, 1-17.

3. Akbar, M. N. and Saptawati, G. P. (2016, October 26-27). Scalable sequential pattern mining based on PrefixSpan for high dimensional data. *International Conference on Data and Software Engineering, IEEE*, 1-6.

4. Zicari, R.V. (2014). Big Data: Challenges and Opportunities. In R. Akerkar (Ed.), *Big data computing*. CRC Press. Florida, USA, 103-128.

5. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376.

6. Armbrust, M., Xin, R. S., Lian, C., Huai, Y., Liu, D., Bradley, J. K., Kaftan, T., Franklin, M. J., Ghodsi, A. and Zaharia, M. (2015, May 31 - June 04). Spark SQL: Relational data processing in Spark. *ACM SIGMOD International Conference on Management of Data*, 1383-1394.

7. Ayres, J., Flannick, J., Gehrke, J. and Yiu, T. (2002, July 23-26). Sequential pattern mining using a bitmap representation. *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 429-435.

8. Chen, C. C., Tseng, C. Y. and Chen, M. S. (2013, June 27 - July 2). Highly scalable sequential pattern mining based on MapReduce model on the cloud. *IEEE International Congress on Big Data*, 310-317.

9. Chen, C. P. and Zhang, C. Y. (2014, August 10). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275, 314-347.

10. Chen, M., Mao, S. and Liu, Y. (2014, April). Big data: A survey. *Mobile Networks and Applications*, 19(2), 171-209.

11. Chiu, D. Y., Wu, Y. H. and Chen, A. L. P. (2004, March 30 - April 2). An efficient algorithm for mining frequent sequences by a new strategy without support counting. *20th International Conference on Data Engineering*, 375–386.

12. De Koning, C. J. A. M. (2011). Robotic milking, *Wageningen UR Livestock Research, Elsevier*, 952-958.

13. Demchenko, Y., Grosso, P., De Laat, C., and Membrey, P. (2013, May 20-24). Addressing big data issues in scientific data infrastructure. *International Conference on*

Collaboration Technologies and Systems, IEEE, 48-55.

14. Demchenko, Y., De Laat, C., and Membrey, P. (2014, May 19-23). Defining architecture components of the Big Data Ecosystem. *International Conference on Collaboration Technologies and Systems, IEEE*, 104-112.

15. Deng, J., Qu, Z., Zhu, Y., Muntean, G. M., and Wang, X. (2014, May 15-17). Towards efficient and scalable data mining using spark. *International Conference on Information and Communications Technologies, IET*, 2.075(1).

16. Géczy, P. (2014). Big data characteristics. *The Macrotheme Review*, 3(6), 94-104.

17. Gouda, K., Hassaan, M. and Zaki, M. J. (2010, February). Prism: An effective approach for frequent sequence mining via prime-block encoding. *Journal of Computer and System Sciences, Elsevier*, 76(1), 88–102.

18. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U. and Hsu, M. C. (2000, August, 20-23). Freespan: frequent pattern-projected sequential pattern mining. *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 355-359.

19. Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. and Hsu, M. C. (2001, April 2-6). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. *17th International Conference on Data Engineering, IEEE*, 215-224.

20. Holloway, L., Bear, C. and Wilkinson, K. (2014, June). Robotic milking technologies and renegotiating situated ethical relationships on UK dairy farms. *Agriculture and human values, Springer*, 31(2), 185-199.

21. Honarvar, A. R. ve Sami, A. (2016). Extracting usage patterns from power usage data of homes' appliances in smart home using big data platform. *International Journal of Information Technology and Web Engineering*, 11(2), 39-50.

22. Huang, J.-W., Lin, S.-C. and Chen, M.-S. (2010, June 21-24). DPSP: Distributed progressive sequential pattern mining on the cloud. *Advances in Knowledge Discovery and Data Mining, Springer*, 27-34.

23. İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fwww.gartner.com%2Fit-glossary%2Fbig-data&date=2018-01-21 . Son Erişim Tarihi: 21.01.2018.

24. İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fhadoop.apache.org&date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

25. İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fspark.apache.org%2F&date=2018-01-21 . Son Erişim Tarihi: 21.01.2018.

26. İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fmahout.apache.org&date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

27. İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fhive.apache.org

&date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

28.İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fpig.apache.org &date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

29.İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fdrill.apache.org &date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

30.İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fwww.mongodb. com&date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

31.İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fhbase.apache.or g&date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

32.İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fcassandra.apach e.org&date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

33.İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fnosql-database.org&date=2018-02-06 . Son Erişim Tarihi: 06.02.2018.

34.İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fwww.lely.com %2Ftr&date=2018-01-21 . Son Erişim Tarihi: 21.01.2018.

35.İnternet:URL:http://www.webcitation.org/query?url=http%3A%2F%2Fwww.triodorarg e.com%2F&date=2018-01-21 . Son Erişim Tarihi: 21.01.2018.

36.Katal, A., Wazid, M. and Goudar, R. (2013, August 8-10). Big data: issues, challenges, tools and good practices. *Sixth International Conference on Contemporary Computing, IEEE*, 404-409.

37.Kaur, P. D. (2015, October 8-10). A survey on Big Data storage strategies. *International Conference on Green Computing and Internet of Things, IEEE*, 280-284.

38.L'Heureux, A., Grolinger, K., ElYamany, H. F. and Capretz, M. (2017, April 20). Machine learning with big data: Challenges and Approaches. *IEEE Access*, 5, 7776-7797.

39.Mabroukeh, N. R. and Ezeife, C. I. (2010, November). A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys*, 43(1), 3.

40.Menon, S. P. and Hegde, N. P. (2015, January 9-10). A survey of tools and applications in big data. 9th *International Conference on Intelligent Systems and Control, IEEE*, 1-7.

41.Nasser, T. and Tariq, R. S. (2015, October). Big data challenges. *Journal of Computer Engineering & Information Technology*, 4:3, 9307(2).

42.Pei, J., Han, J., Mortazavi-Asl, B. and Zhu, H. (2000, April). Mining access patterns efficiently from web logs. *Pacific-Asia Conference on Knowledge Discovery and Data Mining. Current Issues and New Applications, Springer*, 1805, 396–407.

43.Qiu, J., Wu, Q., Ding, G., Xu, Y. and Feng, S. (2016, December). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing, Springer*, 2016(1), 67.

44.Shoro, A. G. and Soomro, T. R. (2015, February 21). Big data analysis: Apache Spark perspective. *Global Journal of Computer Science and Technology*, 15(1-C).

45.Shvachko, K., Kuang, H., Radia, S. and Chansler, R. (2010, May 3-7). The hadoop distributed file system. *26th Symposium on Mass Storage Systems and Technologies, IEEE*, 1-10.

46.Song, S., Hu, H. and Jin, S. (2005, July 22-24). Hvsm: A new sequential pattern mining algorithm using bitmap representation. *International Conference on Advanced Data Mining and Applications, Springer*, 731-732.

47.Stankovic, J. A. (2014, February). Research directions for the internet of things. *IEEE Internet of Things Journal*, 1(1), 3-9.

48.Storey, V. C. and Song, I. Y. (2017, March). Big data technologies and management: What conceptual modeling can do. *Data & Knowledge Engineering, Elsevier*, 108, 50-67.

49.Sundmaeker, H., Verdouw, C., Wolfert, S. and Pérez Freire, L. (2016). Internet of food and farm 2020. In: Vermesan, O., Friess, P. (2016). *Digitising the Industry - Internet of Things Connecting Physical, Digital and Virtual Worlds*. Netherlands: River Publishers, 129-51.

50.Terzi, D. S., Terzi, R. and Sagiroglu, S. (2015, December 14-16). A survey on security and privacy issues in big data. *10th International Conference for Internet Technology and Secured Transactions, IEEE*, 202-207.

51.Vashisht, P. and Gupta, V. (2015, October 8-10). Big data analytics techniques: A survey. *International Conference on Green Computing and Internet of Things, IEEE*, 264-269.

52.Wei, Y. Q., Liu, D. and Duan, L. S. (2012, August 3-5). Distributed PrefixSpan algorithm based on MapReduce. *International Symposium on Information Technology in Medicine and Education, IEEE*, 2, 901-904.

53.Wolfert, S., Ge, L., Verdouw, C. and Bogaardt, M. J. (2017, May). Big Data in smart farming-a review. *Agricultural Systems, Elsevier*, 153, 69-80.

54.Yang, Z., Wang, Y. and Kitsuregawa, M. (2005). LAPIN: Effective sequential pattern mining algorithms by last position induction.

55.Yang, Z., Wang, Y. and Kitsuregawa, M. (2006, January 16-18). An effective system for mining web log. *Frontiers of WWW Research and Development - APWeb 2006, Springer*, 40-52.

56.Zaki, M. J. (2001, January). SPADE: An efficient algorithm for mining frequent

sequences. *Machine Learning, Springer*, 42(1-2), 31-60.

57.Zhang, F., Liu, M., Gui, F., Shen, W., Shami, A. and Ma, Y. (2015, December). A distributed frequent itemset mining algorithm using Spark for Big Data analytics. *Cluster Computing, Springer*, 18(4), 1493-1501.

58.Zhao, Q. and Bhowmick, S. S. (2003). Sequential pattern mining: A survey. *Technical Report, CAIS Nanyang Technological University*, 1, 26.

59.Zhou, L., Pan, S., Wang, J. and Vasilakos, A. V. (2017, May 10). Machine learning on big data: Opportunities and challenges. *Neurocomputing, Elsevier*, 237, 350-361.

60.L'heureux, A., Grolinger, K., Elyamany, H. F. and Capretz, M. A. (2017). Machine learning with big data: Challenges and approaches. *IEEE Access*, 5, 7776-7797.

61.Singh, D. and Reddy, C. K. (2015, December). A survey on platforms for big data analytics. *Journal of Big Data, Springer*, 2(1), 8.

62.Salloum, S., Dautov, R., Chen, X., Peng, P. X. and Huang, J. Z. (2016, November). Big data analytics on Apache Spark. *International Journal of Data Science and Analytics, Springer*, 1(3-4), 145-164.

63.Fu, J., Sun, J. and Wang, K. (2016, December 3-4). Spark–a big data processing platform for machine learning. *International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration, IEEE*, 48-51.

64.Hafez, M. M., Shehab, M. E., El Fakharany, E. and Hegazy, A. E. F. A. G. (2016, October 18). Effective selection of machine learning algorithms for big data analytics using apache spark. *International Conference on Advanced Intelligent Systems and Informatics, Springer*, 692-704.

65.García-Gil, D., Ramírez-Gallego, S., García, S. and Herrera, F. (2017). A comparison on scalability for batch big data processing on Apache Spark and Apache Flink. *Big Data Analytics*, 2(1), 1.

66.Li, T. R., Xu, Y., Ruan, D. and Pan, W. M. (2005). Sequential pattern mining. In *Intelligent Data Mining*. Springer, Berlin, Heidelberg, 103-122.

67.Kayım, F. (2015*). K-Means ile DBSCAN Algoritmasının Paralelleştirmesi ve Hadoop Üzerinde Büyük Veri Analizinde Kullanılması, Performans ve Yeterlilik Karşılaştırması*, Yüksek Lisans Tezi, Beykent Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 1-66.

68.Çetinkaya, S. (2016). *Hadoop / MapReduce Teknolojisi Kullanılarak Hızlı Tüketim Sektöründe Büyük Veri Analizi*, Yüksek Lisans Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 1-70.

69.Akgün, B. (2016). *Apache Spark Tabanlı Destek Vektör Makineleri İle Akan Büyük Veri Sınıflandırma*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 1-49.

70.Hallaç, İ.R. (2014). *Büyük Veri Analizinde Dağıtık Makine Öğrenmesi Algoritmalarının Kullanılması*, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, 1-75.

71.Selçuk, A. (2015). *Büyük Veri Üzerinde Dağıtık Dosya Sistemi ve Paralel İşleme Kullanarak Mahremiyet Korumalı Arama*, Yüksek Lisans Tezi, Sabancı Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 1-50.

72.Aydemir, F. (2016). *Sınır Güvenliği için Büyük Veri Teknik ve Teknolojileri ile Boru Hattı Tasarımı*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 1-54.

73.Salur, M. U. (2016). *Büyük Veri Araçlarından Hadoop Kullanarak Veri Madenciliği*, Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli, 1-102.

74.Exarchos, T. P., Papaloukas, C., Lampros, C. and Fotiadis, D. I. (2008, February). Mining sequential patterns for protein fold recognition. *Journal of Biomedical Informatics, Elsevier,* 41(1), 165-179.

75.Bellazzi, R., Ferrazzi, F. and Sacchi, L. (2011). Predictive data mining in clinical medicine: a focus on selected methods and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(5), 416-430.

76.Exarchos, T. P., Papaloukas, C., Lampros, C. and Fotiadis, D. I. (2006, August 30 - September 3). Protein classification using sequential pattern mining. *International Conference of the IEEE Engineering in Medicine and Biology Society,* 5814-5817.

77.Ho, J., Lukov, L. and Chawla, S. (2005, December 20-22). Sequential pattern mining with constraints on large protein databases. *International Conference on Management of Data*, 89-100.

78.Wang, M., Shang, X. Q. and Li, Z. H. (2008, October 8-10). Sequential pattern mining for protein function prediction. *International Conference on Advanced Data Mining and Applications, Springer*, 5139, 652-658.

79.Wang, K., Xu, Y. and Yu, J. X. (2004, November 8-13). Scalable sequential pattern mining for biological sequences. *ACM International Conference on Information and Knowledge Management*, 178-187.

80.Wright, A. P., Wright, A. T., McCoy, A. B. and Sittig, D. F. (2015, February). The use of sequential pattern mining to predict next prescribed medications. *Journal of Biomedical Informatics, Elsevier*, 53, 73-80.

81.Zabihi, F., Ramezan, M., Pedram, M. M. and Memariani, A. (2011, January). Rule Extraction for Blood donators with fuzzy sequential pattern mining. *The Journal of Mathematics and Computer Science*, 2(1), 37-43.

82.Pitman, A. and Zanker, M. (2010, December 13). Insights from applying sequential pattern mining to e-commerce click stream data. *IEEE International Conference on*

Data Mining Workshops, 967-975.

83.Yun, C. H. and Chen, M. S. (2007, March). Mining mobile sequential patterns in a mobile commerce environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(2), 278-295.

84.Hsueh, S. C., Lin, M. Y. and Chen, C. L. (2008, December 9-12). Mining negative sequential patterns for e-commerce recommendations. *IEEE Asia-Pacific Services Computing Conference*, 1213-1218.

85.Cho, Y. B., Cho, Y. H. and Kim, S. H. (2005, February). Mining changes in customer buying behavior for collaborative recommendations. *Expert Systems with Applications, Elsevier*, 28(2), 359-369.

86.Chen, J. and Cook, T. (2007, May 8-12). Mining contiguous sequential patterns from web logs. *International Conference on World Wide Web*, 1177-1178.

87.Ezeife, C. I. and Lu, Y. (2005, January). Mining web log sequential patterns with position coded pre-order linked wap-tree. *Data Mining and Knowledge Discovery, Springer*, 10(1), 5-38.

88.Parmar, J. D. and Garg, S. (2007, June). Modified web access pattern (mWAP) approach for sequential pattern mining. *INFOCOMP*, 6(2), 46-54.

89.Safyallah, H. and Sartipi, K. (2006, June 14-16). Dynamic analysis of software systems using execution pattern mining. *IEEE International Conference on Program Comprehension*, 84-88.

90.Wu, P. H., Peng, W. C. and Chen, M. S. (2001, September 10). Mining sequential alarm patterns in a telecommunication database. *International Workshop on Databases in Telecommunications, Springer*, 2209, 37-51.

91.Kim, S. W., Park, S., Won, J. I. and Kim, S. W. (2008, February). Privacy preserving data mining of sequential patterns for network traffic data. *Information Sciences, Elsevier*, 178(3), 694-713.

92.Ouh, J. Z., Wu, P. H. and Chen, M. S. (2001, December 3-6). Experimental results on a constraint based sequential pattern mining for telecommunication alarm data. *International Conference on Web Information Systems Engineering, IEEE*, 2, 186-193).

93.Jaillet, S., Laurent, A. and Teisseire, M. (2006). Sequential patterns for text categorization. *Intelligent Data Analysis*, 10(3), 199-214.

94.Sun, G., Liu, X., Cong, G., Zhou, M., Xiong, Z., Lee, J. and Lin, C. Y. (2007, June). Detecting erroneous sentences using automatically mined sequential patterns. *Annual Meeting of the Association of Computational Linguistics*, 81-88.

95.Feng, J., Xie, F., Hu, X., Li, P., Cao, J. and Wu, X. (2011, August 5-7). Keyword extraction based on sequential pattern mining. *International Conference on Internet Multimedia Computing and Service,* 34-38.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ZARALI, Duygu Nazife
Uyruğu : T.C.
Doğum tarihi ve yeri : 02.03.1990, Kayseri
Medeni hali : Evli
Telefon :
E-mail : duygusonmez@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam ediyor
Lisans	Erciyes Üniversitesi / Bilgisayar Mühendisliği	2013
Lise	Nuh Mehmet Küçükçalık Anadolu Lisesi	2008

İş Deneyimi

Yıl	Yer	Görev
2015-Halen	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

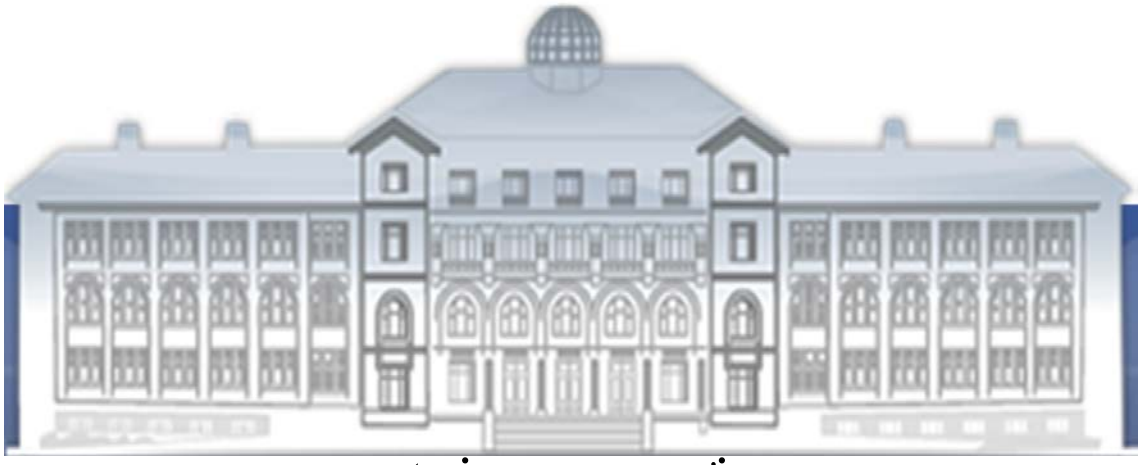
İngilizce, Almanca.

Yayınlar

Zaralı, D. N. and Karacan, H. (2017, December 6-8). *Mining sequential patterns in smart farming using Spark*. Paper presented at the10th International Statistics Congress, Ankara, Turkey.

Hobiler

Kitap okuma, dil bilgisi, doğa yürüyüşü.



GAZİ GELECEKTİR..