
GAZİ ÜNİVERSİTESİ
Gazi
1926

NESNELERİN İNTERNETİ TABANLI SİSTEMLERDE KARMAŞIK OLAY İŞLEME

Mehmet Ulvi ŞİMŞEK

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2021

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.

Mehmet Ulvi ŞİMŞEK

29/06/2021

NESNELERİN İNTERNETİ TABANLI SİSTEMLERDE KARMAŞIK OLAY İŞLEME

(Doktora Tezi)

Mehmet Ulvi ŞİMŞEK

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2021

ÖZET

Nesnelerin interneti (Internet of Things - IOT) ile birlikte sensörler tarafından üretilen ham veri miktarı artmıştır. Yüksek hacimli ve hızdaki bu veriden anlamlı bilgiler çıkarmak için verinin gerçek zamanlı olarak işlenmesi gerekmektedir. Bu kapsamda karmaşık olay işleme (Complex Event Processing- CEP) sistemler ham veriden değerli bilgileri çıkaran bir sistem olarak gerçek zamanlı veri işleme amacı ile kullanılmaktadır. Ancak CEP sistemlerinde kural tanımlama zorluğu ve tahmin yöntemlerinin entegrasyonu konularında zorluklar yaşanmaktadır. Bu kapsamda bu tez çalışmasında CEP sistemlerinin IoT verileri kullanarak derin öğrenme temelli çözümler önerilmiştir. Tez kapsamında ilk olarak IoT verilerinden genel CEP kurallarının çıkarılması amacı ile kümeleme ve kural madenciliği aşamalarını içeren bir model önerilmiştir. Önerilen model kapsamında kural madenciliği algoritmalarının belirlenen kümelere ilişkin kuralları çıkarmada başarılı sonuçlar verdiği ölçülmüştür. Bu tez çalışmasında ikinci olarak, derin öğrenme (deep learning - DL) yöntemleri ile etiketsiz IoT verilerinden CEP kurallarının otomatik çıkarılmasını sağlayan bir çerçeve model önerilmiştir. Önerilen çerçeve model yeniden çatılma hatası ve başarım performansları dikkate alınarak akıllı şehir uygulamasında yer alan hava kirlilik gaz verileri kullanılarak değerlendirilmiştir. Bu tez çalışmasında son olarak CEP tahmin probleminin çözümüne odaklanılmıştır. Bu kapsamda DL yöntemleri ve CEP işleme motoru yapısını içeren IoT sis hesaplama yaklaşımı ile bir sistem önerilmiştir. DL yöntemleri ile IoT zaman serileri tahmin edilerek başarım performansı geleneksel yöntemlerden Destek Vektöre Regresyonu (Support Vector Regression- SVR) ile karşılaştırılmıştır. Tahmin değerleri CEP işleme motorunda değerlendirilerek her bir algoritma için başarım performansları verilmiştir. Önerilen sistemde sis hesaplama dikkate alınarak uçtan uca ağ gecikmesi hesaplanmıştır. Bu bağlamda önerilen sistem modelinde tahmin edilen IoT verilerinin CEP işleme motorunda değerlendirilmesinin küçük ağ gecikmesi ile birlikte yüksek bir başarımla sağlandığı ortaya koyulmuştur.

Bilim Kodu	:	92432
Anahtar Kelimeler	:	Nesnelerin interneti, derin öğrenme, karmaşık olay işleme, sis hesaplama
Sayfa Adedi	:	116
Danışman	:	Prof. Dr. Suat ÖZDEMİR

COMPLEX EVENT PROCESSING IN INTERNET OF THINGS

(Ph. D. Thesis)

Mehmet Ulvi ŞİMŞEK

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2021

ABSTRACT

With the Internet of Things (IoT), the amount of raw data generated by sensors has increased. In order to extract meaningful information from this high volume and speed data, the data must be processed in real time. In this context, Complex event processing (CEP) is promising technology that used for a real time data processing with extracting valuable information from raw data. However, there are open problems about CEP system such as defining automatically CEP rules and CEP prediction. In this thesis, DL based solutions offered to solve these problems with using IoT data. Firstly, a model which has clustering and rule mining phase is proposed to extract CEP rules from IoT data. Experimental results show that the developed model shows excellent performance with determining the scope of the cluster. Secondly, we propose a generalized framework for automatic CEP rule extraction with using DL methods. The proposed framework has two phases which names labeling and automatic rule extraction. In this context, we compare several DL methods with each other and regression-based methods to evaluate the proposed framework in smart city scenario with using air pollution gases by reconstruction error and prediction metrics. The result shows that the Dl based framework success rate increase when the number of attributes increases as well. Finally, we focus on the predictive CEP problem. In this context, we propose a novel deep learning based predictive CEP system for air pollution dataset in IoT framework. DL methods and SVR are compared in terms of the prediction performance. The predicted data is processed in the CEP engine and the performance of the each algorithm in CEP engine is shown in detail. We examine proposed system in terms of end-to-end network delay and measure its impact on the network. The result demonstrate that predicted system with Dl methods show excellent performance while guaranteeing minimum end-to-end network delay.

Science Code	:	92432
Key Words	:	Internet of things, Deep Learning, complex event processing, fog computing
Page Number	:	116
Supervisor	:	Prof. Dr. Suat ÖZDEMİR

TEŞEKKÜR

Doktora çalışmalarım süresince yardımlarını esirgemeyen ve çalışmalarımda tıkandığım tüm noktalarda yönlendirmeleriyle ufkumu açan, değerli bilgi ve birikimleriyle akademik gelişimime katkı sağlayan, danışman hocam Prof. Dr. Suat ÖZDEMİR'e en içten teşekkürlerimi sunarım. Tez çalışması boyunca desteklerini esirgemeyen ve tezin geliştirilmesine yönelik olumlu katkılarda bulunan tez izleme komitemde yer alan değerli hocalarım Doç. Dr. Hacer KARACAN'a ve Dr. Öğr. Üyesi Bülent Tuğrul'a teşekkürlerimi sunarım. Doktora çalışmaları süresince beraber çalışma fırsatı bulduğum değerli arkadaşlarım Dr. İbrahim KÖK'e ve Dr. Feyza YILDIRIM OKAY'a teşekkür ederim.

Eğitim hayatım boyunca desteklerini esirgemeyen canım annem Aysel ŞİMŞEK ve babam Ali ŞİMŞEK başta olmak üzere tüm aileme sonsuz teşekkür ederim. Doktora sürecinin yoğun ve tempolu çalışma süresince sabır ve anlayış gösteren ve her zaman yanımda olan sevgili eşim Tuğba ŞİMŞEK'e, hayatımıza tat katan canım oğlum Ali Yağız ŞİMŞEK'e teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
s	saniye
ms	milisaniye

Kısaltmalar	**Açıklamalar**
AQI	Air Quality Index (Hava Kalite İndeksi)
CEP	Complex Event Processing (Karmaşık Olay İşleme)
CNN	Convolutional Neural Network (Evrişimli Sinir Ağı)
DBN	Deep Belief Network (Derin İnanç Ağları)
GRU	Gated Recurrent Unit (Kapılı Yineleyen Birim)
IoT	Internet of Things (Nesnelerin İnterneti)
KNN	K-Nearest Neighbor (K-En Yakın Komşu)
LSTM	Long Short Term Memory (Uzun Kısa Süreli Bellek)
MAE	Mean Absolute Error (Ortalama Mutlak Hata)
RBM	Restricted Boltzmann Machine (Sınırlı Boltzmann Makinesi)
RMSE	Root Mean Squared Error (Ortalama Karesel Hata)
RNN	Recurrent Neural Network (Yinelemeli Sinir Ağı)
SGD	Stochastic gradient descent
SVM	Destek vektör makineleri
SVR	Support Vector Regressor (Destek Vektörü Regresörü)

1. GİRİŞ

Nesnelerin İnterneti (Internet of Things - IoT), her türlü nesne olarak adlandırılan cihazların internet üzerinden veya kendi aralarındaki iletişim ağları sayesinde iletişim kurabilip ham veri ürettiği bir paradigmadır. Cihazlar arası iletişimin artması ile birlikte IoT alanı çevre, sağlık, sosyal alanlar, endüstri, otomotiv, askeri, akıllı uygulamalar, taşıma / lojistik ve tarım gibi birçok alana yayılmıştır [1]. Donanım, yazılım ve iletişim teknolojilerindeki gelişmeler ile birlikte fiziksel dünyanın sensörler aracılığı ile gözlemlenmesini ve veri üretmesini hızlandırarak internet bağlantılı cihazların ortaya çıkmasını kolaylaştırmıştır.

IoT, sensörlerin ürettiği veriyi yöneten ve akıllıca yönetmek için kullanıcılara daha iyi hizmetler sunmak ve performansını geliştirmek için veriyi kullanmaktadır. Bu kapsamda IoT, fiziksel dünyadan gelen bilgi, gözlem ve veri ölçümlerine ilişkin gelişmiş sistemler sunabilmek amacı ile ağ üzerinden farklı kaynaklardan ham verilere erişebilme ve bu bilgileri analiz etme ihtiyacı duyar [2]. Sensörler tarafından üretilen veri değerli bilgiler içermekte ve analiz ile üretici, tüketici ve hizmet sağlayıcılarına değerli bilgi sağlamaktadır [3,4]. Değerli bilginin oluşma sürecinde ise verinin işlenmesi yer almaktadır. Bu kapsamda geleneksel veri madenciliği ve istatiksel yöntemler alana özgü olarak problem odaklı ve genel veri işleme amaçlı kullanılmaktadır. Özellikle IoT verisinin normal veriden farklı olarak heterojen, gürültülü, çeşitli ve hızlı büyüyen yapısı sebebi ile diğer verilerden ayrılmaktadır [5]. Bu sebeple gelişmiş makine öğrenmesi teknikleri IoT alanındaki veri analitiğini ve öğrenme süreçlerini kolaylaştırması sebebi ile yoğunlukla tercih edilmektedir [6].

Karmaşık olay işleme (Complex Event Processing / CEP), olayların oluştuğu anda tespit eden ve akış verilerini verimli şekilde işleyen bir sistemdir. CEP sistemi atomik olayları içeren akış verilerinin işlenmesi ile birlikte karmaşık olayları yakalayan bir sistemdir [7]. Önceden belirlenen kurallara göre akış verilerinin işlenmesi ile birlikte karmaşık olaylar tespit edilmektedir. IoT alanının genişlemesi ile birlikte sensörlerin oluşturduğu verilerin anlık olarak işlenmesi erken uyarı sistemleri veya karar destek sistemleri gibi vb. sistemler için önem kazanmıştır. Bu kapsamda CEP teknolojisi gerçek zamanlı olarak ileme yetenekleri sayesinde bu alanda kullanılmaya başlanmıştır. Ancak IoT alanındaki veri niteliklerinin çeşitli olması sebebi ile CEP uygulamalarının genişleme alanı

kısıtlanmaktadır. CEP sisteminin genişlemesinde ve farklı alanlarda kullanılmasında kural tanımlamalarının etkisi büyük olacaktır. Literatürde kural tanımlamaları alan uzmanları, kural madenciliği ve matematiksel optimizasyon teknikleri ile yapılmaktadır [8]. Bu bağlamda IoT alanındaki veriler için otomatik kural çıkarılması için taleplerin karşılanması amacı ile yeni yöntemlere ihtiyaç duyulmaktadır. Bununla birlikte CEP alanının mevcut veriyi işlemesi sebebi ile geçmiş veri analizi ile ilgilenmemektedir. Ancak günümüzde derin öğrenme tekniklerinin veri analizi konusunda başarısı ile birlikte her alanda tahmin ve veri analizine yönelik çalışmalar artmıştır. Bu kapsamda literatürde CEP mimarisi içerisinde kullanılan akış verilerinin geçmiş bilgileri dikkate alınarak önerilen CEP tahmin sistemi ile geleceğe dönük karmaşık olaylar çıkarılabilmektedir [9,10].

Bu tez çalışmasında, IoT alanında oluşan verilerin CEP sisteminde işlenmesi ile ilgili olarak üç problem adreslenerek çözümler üretilmiştir. Bu sayede CEP sistemlerinin IoT alanında kullanımı artarak, IoT alanında CEP sistemlerinin kullanılması ile birlikte işleme ve analiz süreçlerinde iyileştirme hedeflenmiştir. Bu kapsamda adreslenen problemler genel CEP kuralların kümeleme yaklaşımı ile otomatik çıkarılması, IoT verilerinden anomali belirleme yöntemi ile CEP kurallarının otomatik çıkarılması ve IoT verilerinde CEP tahmin sistemlerinin sis hesaplama ile gerçekleştirilmesi olarak ifade edilebilir. Tez çalışması kapsamında adreslenen problemlere ilişkin olarak özet bilgiler ve katkıları aşağıda verilmiştir.

Literatürde CEP kurallarının çıkarılmasında veri madenciliği ve optimizasyon teknikleri kullanılmaktadır. İncelenen çalışmaların genellikle alana özgü ve sadece önerilen alan için çözüm sunduğu gözükmektedir. Ayrıca mevcut çalışmalar genel olarak etiketli veri kullanmaktadır. Özellikle IoT verilerinin etiketsiz olması sebebi ile alan uzmanlarının müdahalesi ile CEP kuralları oluşturulmaktadır. Ancak sensör teknolojisinin gelişmesi ile birlikte birçok alanda alan uzmanlarının CEP kurallarını tanımlaması imkansız hale gelecektir. Özellikle farklı nitelik sayısı arttıkça kural tanımlama karmaşıklığı artacaktır. Bu kapsamda etiketsiz verilerden CEP kuralların çıkarılması özellikle IoT alanında CEP sistemlerinin genişlemesine katkı sağlayacağı düşünülmektedir. Bu kapsamda IoT verileri kullanılarak genel CEP kurallarının çıkarılması amacı ile kümeleme yaklaşımı ile otomatik olarak çıkarılmasına ilişkin model önerilmiştir. Önerilen modelin, tez çalışmasına katkıları aşağıda sunulmuştur.

- Literatürde alana özgü olan problemlerin çözümüne odaklanılmıştır. Bu kapsamda IoT verileri için genel bir mimari yaklaşımı önerilerek etiketsiz veriler için otomatik olarak genel CEP kuralları çıkarılmıştır.
- Önerilen model ile CEP sistemlerinin akıllı şehir uygulamaları içerisinde daha fazla yer alacağı ve erken uyarı sistemlerinin gelişmesine katkı sunacaktır.
- Kural çıkarımı aşamasında kullanılan algoritmalar karşılaştırmalı olarak analiz edilmiştir.

IoT alanında sensörlerin oluşturduğu veriler zaman serileri olarak ifade edilmektedir. Bu kapsamda zaman serilerinin tahmin, örüntü analizi gibi işlemlerde yoğunluklu olarak derin öğrenme teknikleri kullanılmaktadır. Derin öğrenme teknikleri yapısında kullanılan derin sinir ağları ile zaman serilerinin veri özelliklerini otomatik olarak çıkartabilmektedir. Ancak literatürde derin öğrenme yöntemleri kullanılarak CEP kurallarının otomatik olarak çıkarılmasına ilişkin ile yapılan çalışmalara rastlanılmamıştır. Bu kapsamda derin öğrenme teknikleri kullanılarak IoT verilerinden CEP kurallarının otomatik olarak çıkarılması amacı ile çerçeve bir model önerilmiştir. Bu kapsamda hazırlanan IoT verilerinden anomali belirleme yöntemi ile CEP kurallarının otomatik çıkarılması problemine ilişkin tez çalışmasının katkıları aşağıda sunulmuştur.

- CEP kural çıkarımı için IoT verileri kullanılarak genelleştirilmiş ARECEP isimli bir çerçeve önerilmiştir. Akıllı şehir senaryosu dikkate alınarak hava kirletici gazlarından otomatik olarak CEP kurallarının çıkarılması sağlanmıştır. Bu bağlamda akıllı şehir verileri dikkate alınarak literatürde ilk defa bir model geliştirilmiştir.
- Literatürde DL yöntemleri yaygın olarak tahmin sistemlerinde kullanılmaktadır. Otomatik kural çıkarımı alanında DL yöntemleri kullanılarak yapılan ilk çalışmadır.
- ARECEP çerçevesi içerisinde kullanılan DL yöntemleri karşılaştırmalı olarak değerlendirilmiştir. Bu kapsamda kullanılan yöntemlere ilişkin tahmin ve anomali belirleme performansı test edilerek karşılaştırılmıştır
- ARECEP çerçevesi içerisinde farklı kural madenciliği algoritmaları kullanılmıştır. CEP kural çıkarımı problemine ilişkin olarak kural madenciliği algoritmalarının etiketsiz IoT verileri üzerinde karşılaştırmalı performans analizleri sunulmuştur.
- Sensör veri nitelik sayısının artması durumunda kural çıkarım sonuçlarının karşılaştırmalı analizleri sunulmuştur. Bu bağlamda nitelik sayısı arttığında çıkarılan kuralların başarım oranı test edilmiştir.

CEP sistemleri akış verilerinden kurallar aracılığı ile analiz yapılarak karmaşık olayları çıkarmaktadır. Bir başka ifade ile CEP sistemleri makine öğrenmesi gibi tahmin yöntemlerinden yoksundur. Özellikle gerçek zamanlı sistemlerde anlık bilgi önemli olmakla birlikte geleceğe dönük bilgilerin çıkarılması önemli olmaktadır. Bu bağlamda makine öğrenmesi teknikleri CEP sistemlerine entegre edilerek çeşitli modeller oluşturulmuştur. Bu kapsamda IoT mimarisinin doğası gereği CEP tahmin sistemlerinin sis hesaplama yaklaşımı ile yapılması problemi ortaya çıkmaktadır. Bu bağlamda derin öğrenme teknikleri kullanılarak IoT verileri için CEP tahmin probleminin çözümüne yönelik bir sistem önerilmiştir. Bu kapsamda problemin çözümüne ilişkin tez çalışmasının katkıları aşağıda sunulmuştur.

- CEP tahmin problemlerine ilişkin hava kirletici gazları kullanılarak IoT alanında akıllı şehir uygulamalarında DeepFogCEP isimli bir sistem önerilmiştir. DeepFogCEP sistemi kirletici gaz konsantrasyonları dikkate alarak gelecekteki verileri tahmin eder ve CEP işleme motoru aracılığı ile işlenerek karmaşık olaylar belirlenmiştir.
- DeepFogCEP sistemi içerisinde yer alan tahmin ünitesinde CNN, LSTM, RNN ve SVR modelleri kullanılmıştır. Bu modellerin tahmin performansı karşılaştırmalı olarak ölçülmüştür.
- Önerilen sistem ile tahmin verileri CEP işleme motorunda kullanılarak gelecekteki hava kirliliğine ilişkin erken uyarı sistemi geliştirilmiştir. Bu kapsamda CEP işleme sisteminde belirlenen alarm seviyelerine göre kullanılan modellerin başarım performansı karşılaştırmalı olarak verilmiştir.
- Önerilen DeepFogCEP sistem tasarımında sis- bulut tabanlı mimari dikkate alınmıştır. Bu kapsamda sistem içerisindeki iletişim gecikmeleri, tahmin ve CEP ünitesi işleme gecikmeleri hesaplanarak değerlendirilmiştir. Uçtan uca ağ performansı önerilen sis hesaplama mimarisi açısından değerlendirilerek uygulanabilirliği gösterilmiştir.

Yukarıda ifade edilen problemlere ilişkin önerilen çözümler ve kullanılan yöntemler baz alınarak tez aşağıdaki şekilde organize edilmiştir.

Bölüm 2 kapsamında CEP ve IoT sistemleri ve kavramları hakkında detaylı olarak bilgiler sunulmuştur. Bu bölüm içerisinde CEP kavramı, bileşenleri, alanları ve mevcut problemleri sunulmuştur. Bununla birlikte IoT alanları, mimarisi ve IoT genel zorlukları hakkında

detaylı bilgiler verilmiştir. Sonrasında önerilen çözüm problemleri ışığında kullanılan DL ve geleneksel veri madenciliği modelleri ve kural madenciliği algoritmaları açıklanmıştır.

Bölüm 3'te, genel CEP kurallarının çıkarılması için bir model önerilmiştir. Model içerisinde kullanılan kümeleme yöntemi ve kural madenciliği algoritmalarına ilişkin bilgiler sunulmuştur. Bu kapsamda model içerisinde kullanılan algoritmaların başarım performansları karşılaştırmaları olarak sunularak önerilen modelin başarısı değerlendirilmiştir.

Bölüm 4'te CEP kurallarının DL yöntemleri ile otomatik olarak çıkarılması için bir çerçeve önerilmiştir. Bu kapsamda kullanılan DL yöntemleri ve parametreleri açıklanmıştır. Literatürde CEP kural çıkarımı alanında yapılan çalışmalar özetlenerek sunulmuştur. Sonrasında geliştirilen ARECEP çerçevesi açıklanmıştır. Kullanılan modellerden en iyi DL ve geleneksel yöntem sonuçları seçilerek kural çıkarım aşaması sonuçları verilmiştir. IoT verileri dikkate alınarak gerçekleştirilen testlere ilişkin kullanılan model ve algoritmaların başarım performansları detaylı olarak sunulmuştur.

Bölüm 5'te IoT alanında CEP tahmin sistemine ilişkin olarak bir sistem önerilmiştir. Sistem tasarımında sis işleme mimarisi dikkate alınarak bu kapsamda oluşan işlem gecikmeleri detaylı olarak sunulmuştur. Önerilen sistem içerisinde kullanılan DL ve geleneksel yöntemlerinin tahmin başarım performansları karşılaştırmalı olarak değerlendirilmiştir. Kullanılan DL ve geleneksel yöntemlere ilişkin CEP işleme motoru çıktıları başarım performansı analiz edilmiş olup karşılaştırmalı olarak sunulmuştur.

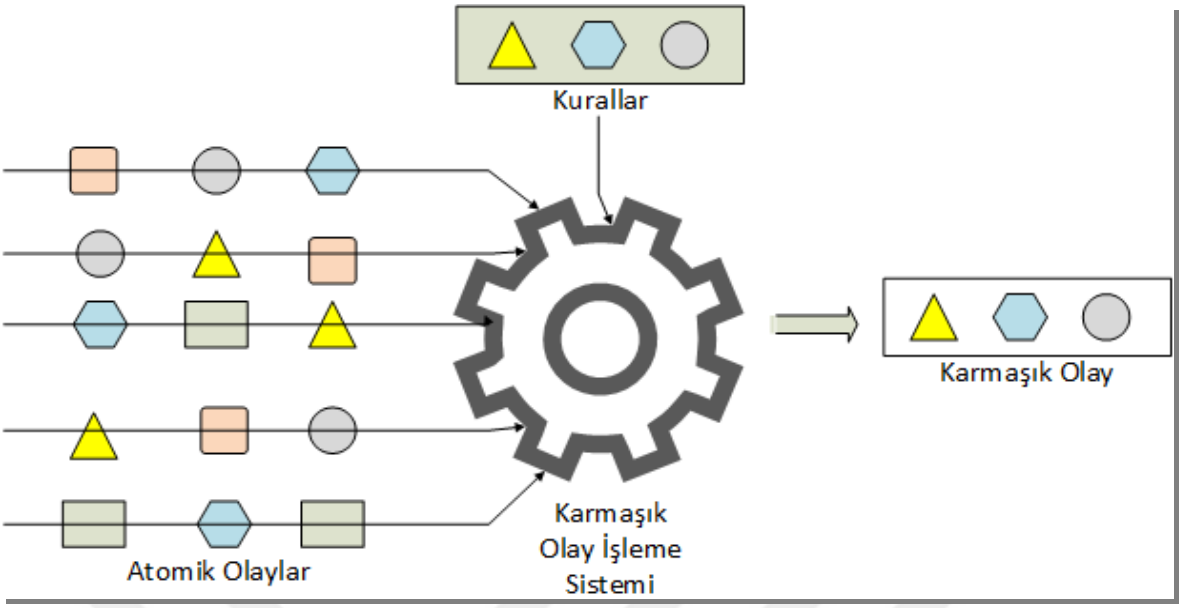
Bölüm 6 kapsamında tez çalışmasında önerilen çözümlere ilişkin bulgular değerlendirilmiş ve sonuçlar özetlenmiştir. Ayrıca tez çalışmasının literatüre ve bilime katkısı açıklanarak geleceğe yönelik yapılabilecek çalışmalar hakkında öneriler sunulmuştur.

2. KARMAŞIK OLAY İŞLEME ve NESNELERİN İNTERNETİ

2.1. Karmaşık Olay İşleme

Bilgi sistemlerinin gelişmesi ile birlikte üretilen veri miktarı artmıştır. Veri miktarının artması ile birlikte son kullanıcıların ihtiyaç duyduğu anlamlı bilgiler önemli olmaktadır. Bu sebeple gerçek zamanlı sistemlerden istenilen bilginin hızlı bir şekilde çıkarılması önemli olmaktadır. Gerçek zamanlı olarak üretilen verilerin boyutunun artması ile birlikte bunları kontrol eden ve bu verileri işleyen sistemlere ihtiyaç duyulmaktadır. Bu durum, anlık olarak gelen verinin işlenerek istenen bilginin hızlı şekilde elde edilmesi gerçek zamanlı sistemler için önemli olmaktadır. Özellikle düşük gecikme ile verilerin hızlı analizi sonucu istenen karmaşık bilgiye ihtiyaç duyulmaktadır. Bu kapsamda karmaşık olay işleme yaklaşımı gerçek zamanlı olarak gelen akış verilerin işlenmesi amacı ile ortaya çıkmıştır. Karmaşık olay işleme, gelen atomik bilginin işleme motorları ile işlenerek karmaşık olayların çıkarılması olarak ifade edilebilir [11].

Karmaşık olay işleme büyük akış verileri üzerinde atomik olarak nitelendirilen olayların zamana göre analiz edilerek karmaşık olayların belirlendiği sistemdir. Bir başka deyişle, akış verilerini atomik veri veya olay olarak nitelendirilerek son kullanıcıların istediği gözlemlenemeyen karmaşık olayları tespit ederek bu olayları birleştirerek ortaya çıkaran sistemdir [12]. Yani, karmaşık olay işleme sistemlerinin girdisi atomik olaylar, çıktısı ise karmaşık olaylar olarak ifade edilmektedir. Atomik olay akış verilerinin kaynağı olan sensör veya direk veri kaynaklarından gelen işlenmemiş veri olarak ifade edilmektedir. Karmaşık olay ise atomik olayların oluşturduğu birleşik olay olarak ifade edilmektedir. Karmaşık olay işleme sistemi ise dinamik bir sistemde bilgi sağlayıcıları ve bilgi tüketicilerini ayrıştırarak tüketicilerin ihtiyaç duyduğu bilgileri çıkaran sistemdir. Karmaşık olay işleme sisteminin temel anlamda iki katkısı olduğunu ifade edilmektedir. Olay kaynakları yani ham veriyi üretenler ile olay tüketicileri yani karmaşık olayların kullanıldığı son kullanıcılar arası bir ayrışma sağlamasıdır. Bir diğer katkı ise olay tüketicilerine iletilecek olan veri miktarı veya mesaj yükünün azaltılmasıdır. Bu sayede sistemin ölçeklenebilir olması sağlanacaktır [13].



Şekil 2. 1. Karmaşık Olay İşleme

Karmaşık olay işleme sistemlerinde verinin ham halinden karmaşık halinin elde edilmesi sürecinde aşağıda belirtilen tanımlardan faydalanılmaktadır. Bu tanımlar Çizelge 2.1 de ifade edilmiştir.

Çizelge 2. 1. Olay işleme tanımları

Olay	Olay tiplerine göre semantik olarak ve içerik olarak belirlenir [7]. Olay tanımı " bir şeylerin yâda kayda değer durumların oluşması" olarak ifade edilmiştir. Benzer olarak "atomik durumlar " olarak nitelendirilmiştir [14].
Başlangıç olayı/ Ham Olay	Verinin ham halini ifade eden ve parçalanamayan anlık bir olayı ifade eden olayı ifade eder [15]
Birleşik olay/olay/karmaşık olay	Atomik olayların içeriğini toplayarak ve birleştirilerek oluşturulur [16]. Olaylar dizisinin özetlenmesi ve birleştirilmesi sonucu oluşan olaylardır [14,17].
Zaman Serileri	Farklı kaynaklardan gelen verilerin üretim zamanına göre sıralanması sonucu oluşan serilerdir [7].
Olay Tipleri	Farklı veri kaynaklarından gelen farklı verileri ifade eder [3].

Çizelge 2. 2. (Devam) Olay işleme tanımları

Olay Nitelikleri	Olay yapısı için tanımlanan bileşenlerdir [14].
Olay Tarihi	Zamansal olarak sıralanmış olay örneklerinden çıkarılan karmaşık olayın oluştuğu tarihtir.
Olay Akışı	Lineer olarak sıralanmış olaylar dizisi olarak ifade edilmektedir [14,17].
Olay Örüntüsü	Eşleşen belli olay setlerini ifade etmektedir[18]. Yani akış verileri içerisinde olay verilerinin sıralanış biçimini içyapısını ve koşullarını açıklayan kavramdır [19,20].
Karmaşık Olay İşleme Sistemi	Karmaşık olay atomik olayların işlenmesi ile oluşan olaylardır. Zamansal olarak gelen akış verilerinin gerçek zamanlı olarak işlenmesini sağlayan sistemlerdir. Birleşik veya karmaşık olaylar önceden tanımlanan kural veya örüntüler aracılığı ile yakalandığı sistemin bütünüdür [21].

2.1.1. Olay

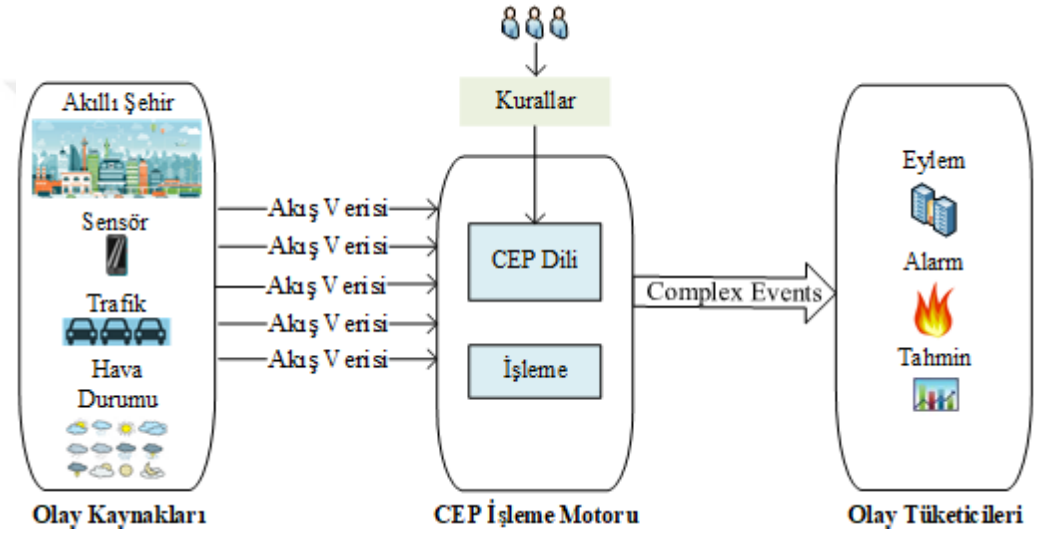
Olay, bir çevre veya spesifik bir alana özgü dikkate değer olarak gerçekleşen atomik durumları ifade eder. Her bir olay zaman, id, isim, tip vb. şeklinde akış verisi olarak ifade edilebilecek çeşitli özelliklere sahiptir. Her olay kendi içerisinde tutarlı veri gruplarından oluşur [22]. Örnek olarak atomik olay, sensör değerleri, RFID mesajları ve akış verisi şeklinde ifade edilebilen veriler olarak ifade edilebilir.

2.1.2. Karmaşık olay

Tek veya birden fazla atomik olayın belli bir zaman diliminde veya zaman bağımsız olarak birlikte oluşturdukları anlamlı verilere karmaşık olay denilmektedir.

2.1.3. Karmaşık olay işleme mimarisi

Karmaşık olay işleme sistemleri çoklu akış verileri üzerinden gerçek zamanlı olarak anlamlı veriler çıkartır [23]. Sistem, olay gözeticileri (kaynaklar), olay tüketicileri ve karmaşık olay işleme mimarisi bölümlerinden oluşmaktadır (Şekil 2.1). Karmaşık olay işleme mimarisi uygulamadan uygulamaya farklılıklar göstermektedir. Mimarilerin temel anlamda farklılıkları işleme motoru, işleme algoritması ve kullanılan kural tanımlama dilleri arasındaki farklılıklardan kaynaklanmaktadır [24].



Şekil 2. 2. Karmaşık olay işleme mimarisi

Karmaşık olay işleme sitemleri temel olarak dört ana kısımdan oluşmaktadır. Atomik olayın üretiminden başlayarak olay tüketicilerine sunulduğu aşamada son bulmaktadır. Yani bir karmaşık olay işleme sistemi birçok kaynaktan olay işleme aşamasına yönlendirilmektedir. Olay işleme aşamasında atomik olay olarak ifade edilen olaylardan, kurallar aracılığı ile karmaşık olaylar belirlenerek olay tüketicilerine sunulur.

Karmaşık olay işleme mimarisinde uygulama bazlı meydana gelen ihtiyaçlar sebebi ile literatürde farklı kavramlar ortaya çıkmıştır. [25] çalışmasında karmaşık olay işleme mimarisi bileşenleri olay kaynakları, işleme motoru, kurallar ve olay tüketicileri olarak ifade edilmiştir. Karmaşık olay işleme mimarisi her ne kadar giriş ve çıkış arasında ham olaylardan karmaşık olayların çıkarılmasına yönelik olsa da uygulama alanlarına göre farklılıklar gösterebilmektedir. Örneğin; sensörler için gerçekleştirilen sistem mimarisinde,

“server”, “sink” ve “node” tarafı düşünülerek bir mimari geliştirilmiştir. Sorgulamanın başlangıç yeri olarak belirlenen sunucular en yakın ara düğümlere sorgu göndererek ağ içerisindeki sonuç akışlarını toplamayı hedeflemektedir. Ara düğümler ise sunucu düğümlerle direk iletişim kuran ve diğer sensörlerden kendisine gelen verileri sunucu düğüme gönderen düğümlerdir. Sensör düğümler tarafında ise geleneksel uygulamalardan farklı olarak düğüm tarafında filtreleme, kümeleme ve karmaşık olay işleme operasyonları gibi işlemleri yapabilen sensörler olduğu ifade edilmiştir. Bu değişikliklerin iletişim maliyetini azaltarak aynı zamanda ham veriler üzerinden sadece faydalı bilgilerin elde edileceği söylenmiştir [26]. [13] çalışmasında ise diğer mimarilerden farklı karasız otomat oluşturulması, filtreleme ve sıralı tarama ve sonuç aşamalarını kullanılmıştır. [27] makalesinde karmaşık olay işleme mimarisi olay belirtimi, olay işleme kuralları, olay işleme motoru ve kurumsal entegrasyon omurgası kullanılmıştır. Olay belirtimi aşamasında tüm olası olay türlerini örnekleriyle birlikte (yani olay verileri), bağımlılıklarını, etkileşimlerini ve hiyerarşilerini açıklanmıştır. Olay işleme kuralları aşamasında olay belirtimine dayanarak, olayları işlemek için kurallar tanımlanmıştır. Bu işleme olayların diğer biçimlere dönüştürülmesi, olayların filtrelenmesi ve toplanması ve yeni olaylar oluşturulması olabilir. Olay işleme motoru sistemin çekirdeği olarak tanımlanarak verilerini yükler ve üzerinde kuralları uygulayarak karmaşık olayları elde eder. Kurumsal entegrasyon omurgası ise sistemin farklı bileşenleri arasındaki olayları değiştirmek için arabulucu olarak çalışan standart bir ara katman aracı olarak tanımlanmıştır [27]. Yukarıda anlatılan karmaşık olay işleme mimarilerinden yola çıkarak sistemin bileşenleri ve yapılan işlemler aşağıdaki şekilde ifade edilebilir [28].

Şekil 2. 3. Karmaşık olay işleme bileşenleri

Olay Kaynakları: Dış kaynaklardan gelen akış verilerini işleme motoruna gönderen ve bağlantıyı sağlayan kısımdır. Atomik olay veya ham veri olarak ifade edilen verinin toplandığı kısımdır.

Olay Modelleme: Alan uzmanları tarafından veya otomatik kural tanımlama yöntemleri ile kural veya örüntü olarak tanımlanarak işleme motoruna entegre edilen kısımdır.

Olay İşleme: Atomik verilerin işlenerek daha önceden tanımlı kurallara göre tespit edilmesi işlemidir.

Olay Tüketicileri: Karmaşık olayların dış sistemlere iletildiği kısımdır.

2.1.4. Olay işleme yöntemleri

Olay işleme yöntemleri, farklı karmaşık olay ifadeleri tarafından belirtilen karmaşık olayları ham ve basit olaylardan algılamak için kullanılan yaklaşımlardır. Literatürde kullanılan olay belirleme modelleri aşağıdaki gibidir.

- Sonlu Durum Makineleri [14,29]
- Sonlu Olmayan Durum Makineleri [30]

- Ağaçlar [31]
- Graflar [32]
- Kural Tabanlı Yaklaşımlar [14]
- RETE algoritması [14]
- Depolama tabanlı yaklaşımlar [14]

2.1.5. Olay işleme sorguları

Karmaşık olay işleme mimarisinin ana kısmını işleme motoru kısmı oluşmaktadır. Bu kısım daha önceden belirlenen kurallara yani olay sorgulama dillerine göre çalışmaktadır. Olay sorgulama dilleri kullanıcıların olay örüntüleri hakkında bilgi edinmesini sağlamaktadır [33]. Literatürde birbirinden farklı özelliklere sahip birçok sorgulama dili bulunmaktadır. Sorgulama dillerinin farklılık göstermesinin nedeni her birinin kendine ait özellikleri ve farklı sitilleri olmasından kaynaklanmaktadır. [34] çalışmasında sorgu dilleri sitillerine göre 3 kategori altında tanımlandığı ifade edilmiştir.

- Birleşim Operatörleri: Birleşim operatörleri, farklı birleşim operatörlerini (bağlanma, sıralama, olumsuzlama) ve ifadelerin yerleştirilmesini kullanarak tek bir olay oluşturarak karmaşık olayları ifade eder.
- Veri Akışı Sorgu Dili (Data Stream Query Language): Veri akışı sorgu dilleri SQL tabanlıdır; olayları demet olarak nitelendiren veri akışları, veritabanı ilişkilerine dönüştürülür. Daha sonra, SQL sorguları değerlendir ve son olarak ilişkiler tekrar veri akışına dönüştürülmektedir. Örneğin CQL, Coral8, StreamBase, Aleri, Esper, vb.
- Üretim kuralları: Belirli durumlar girildiğinde yapılacak eylemleri belirtir, bu yüzden daha dar anlamda bir olay sorgu dili oluşturmazlar. Karmaşık olay işleme mimarisi için uygun olduğu ifade edilmiştir.

Olay sorgulama dilleri yapılarına göre farklılık göstermesine karşın kullanılan operatörler, seçme işlemi ve sıralı dizi yani örüntü elde etme işlemi ile olay sorgulama kuralları oluşturulmaktadır [35]. Benzer özellikler taşıyan birçok kural tanımlama dili bulunmaktadır [36]. Genel olarak birçoğunun yapısı SQL diline benzemektedir. Aşağıda SQL diline benzer özellikler taşıyan kural tanımlama diline örnek verilmiştir.

SELECT * FROM (t S1 ; t S2 ; t L) WHERE FILTER(S1.acc = S2.acc), FILTER(S2.acc = L.acc), S1.amount < 100 AND S2.amount < 100 AND L.amount > 250 AND (S1, L) OCCURS WITHIN 12 hours

Olay sorgulama dilinin yapısında matematiksel operatörler, filtreleme, pencere boyutu, birleşme ve kümeleme yoğunluklu olarak kullanılmaktadır. Kullanılan bu yapılar aşağıdaki şekilde ifade edebilir.

- Matematiksel operatörler: İlişkisel cebir ifadelerini kapsayan büyüktür, küçüktür, eşittir ve benzeri ifadeleri içinde barındıran yapıdır [37]. (=, ≠, >, <, ≥, ≤).
- Filtreleme: Son kullanıcılar tarafından ihtiyaç duyulan veriyi veya veri aralığını tanımlayabilen yapıdır. Atomik olarak tüm verilere ihtiyaç duymayan kullanıcılar için sadece ihtiyaç duyduğu spesifik olaylar ile ilgilenirler [38,39].
- Pencere boyutu: Olay tüketicileri tarafından ihtiyaç duyulan herhangi bir zaman aralığına ait örüntülerin elde edilmesi amacı ile kullanılmaktadır [40].
- Birleşme: Birden fazla akış verisi arası ilişkisel bağlantıların kurulması için kullanılmaktadır [38].
- Kümeleme: Toplam, minimum, maksimum ve ortalama gibi fonksiyonlar ile verilerin bir araya getirilerek istatiksel bilgilerin alınabilmesi amacı ile kullanılır [41].
- Sıralama: İki olay veya daha fazla olay içerisinden birbirini takip eden olayların çıkarılması için kullanılmaktadır[42].
- Ayrılma: zamandan bağımsız olarak meydana gelen iki olaydan birisini seçmek için kullanılır [42].
- Olumsuzluk: Olayın meydana gelmemesini ifade etmektedir [42].
- Seçim: Nitelikleri belli zaman aralığı veya başka seçenekleri içeren olayları seçmek için kullanılır [42].
- Projeksiyon: Nitelikleri bir olayın alt kümesini ifade eden olaylardır [42].

Olay sorgulama dilleri yukarıda ifade edilen temel bileşenlerden oluşmaktadır. Olay sorgulama dilleri uygulamadan uygulamaya farklılık göstermekte olup kullanım şekillerine göre değişmektedir. Yani dilin yapısal özelliklerine göre karmaşık olayları elde etmek amacı ile farklı tipte kurallar yazılabilmektedir. Aşağıda karmaşık olay işleme sistemlerinde kullanılan bazı diller için açıklamalar verilmiştir.

- RAPIDE-EPL dili olay örüntüleri için tanımlanan bir dil olup adlandırılmış operatörler kullanılmaktadır. C ve Pascal dili ile benzerlik göstermektedir [43].
- SASE, olay tanımla dili ile filtreleme, korelasyon ve olayların dönüştürülmesi işlemlerini gerçekleştirmektedir [44].
- CAYUGA dilinde ise "olay cebiri" kullanmaktadır. SQL diline benzerlik göstermektedir [45]. CAYUGA buffer işlemi ile kararsız sonlu otomatları kullanarak olay örüntülerini eşleştiren sorgular üretir [46].
- ESPER dili SQL benzeri bir dil olup java ve. net kütüphanesi içerisinde kullanılmaktadır. Olay akışı sorguları, SQL benzeri pencere, toplama, birleştirme ve analiz işlevleri sağlamaktadır [47].
- NEEL dili ise esnek şekilde "and, or, negation ve seq" kullanarak iç içe sorgulama işlemini gerçekleştirmektedir [48].

2.1.6. Olay işleme sorguları oluşturma yöntemleri

Olay sorgusu/kuralları genel olarak alanında uzman kişiler tarafından tanımlanmaktadır. Sorgular ilişkisel olaylar hakkında bilgi sahibi olmayı, geçici bağımlılıkları bilmeyi ve olay nitelikleri hakkında bilgiye sahip olmayı gerektirmektedir [49]. Olay sorgulama dillerinin çeşitli olmasına karşın kural oluşturma yöntemleri [8] çalışmasında 3 kategoride ifade edilmiştir.

- Alan Uzmanları
- Veri Madenciliği
- Matematiksel Optimizasyon

Tanımlama yöntemleri verinin karmaşıklığına ve veri tabanı bulunup bulunmamasına göre değişmektedir. Karmaşıklık yönetilebilir ise alanında uzman kişiler tarafından kural tanımlaması daha kolay yapılabilmektedir. Ancak karmaşıklık arttığında madencilik ve optimizasyon teknikleri daha başarılı olacağı ifade edilmektedir. İnsan kapasitesinin ötesinde zorlukta olan karmaşıklığa sahip bir veride ise matematiksel optimizasyon kullanıldığı ifade edilmiştir. Verinin veritabanında saklanması ve veri içerisinde ilişkisel bilgiler mevcutsa madencilik algoritmaları ve optimizasyon teknikleri ile olay sorgusu tanımlamanın daha uygun olacağı ifade edilmektedir [8].

2.1.7. Uygulama alanları

Karmaşık olay işleme akış verisine konu olan birçok alanda uygulanabilmektedir. Özellikle sensör teknolojisinin gelişmesi ile birlikte uygulama alanları artmaktadır. Bu bağlamda iş izleme sistemleri, sensör ağlar, piyasa verileri [33] yoğunluklu olmakla birlikte sosyal medya izleme, akıllı ev, sağlık, akıllı ulaşım sistemleri gibi alanlarda uygulama alanı bulmuştur [50]. Bu alanların dışında sensörler ve uç sistem cihazları sayesinde birçok alanda veri toplanabilmektedir. Bu kapsamda olay işleme alanları astronomi, telekomünikasyon, elektrik şebekesi gibi alanlarda uygulama alanı bulmaktadır [51].

2.1.8. Karmaşık olay işleme ve nesnelerin interneti

Sensör teknolojisi gelişmesiyle birlikte birçok alanda sensörler kullanılmaya başlanmıştır. IoT cihazlarının günlük yaşantımızda yer alması ile birlikte IoT uygulama alanı oldukça genişlemiştir. Kullanılan IoT cihazları sayesinde büyük miktarda veri üretilmektedir. Akış verisi olarak nitelendirilen bu verinin gerçek zamanlı olarak analiz edilmesi bilgi çıkarımı ve işlemesi için önem arz etmektedir. Bu verilerin üretildiği anda analiz edilmesi için CEP sistemleri kullanılmaktadır [52]. CEP sistemleri verinin verimli şekilde işlenmesi ve oluşabilecek acil durumları yakalaması için kullanılmaktadır [7]. CEP'in gerçek zamanlı olarak anlamlı bilgileri yakalamaktaki başarısı IoT ve CEP alanını kesiştirmektedir. Özellikle IoT algılama kapsamında atomik veriler üretilmektedir. Bu atomik verilerden gerçek zamanlı olarak karmaşık olaylar CEP aracılığı ile çıkarılabilir.

2.1.9. Sınırlar ve araştırma alanları

Literatürde CEP alanında geliştirilen uygulamalarda CEP tahmin, otomatik kural çıkarımı, dağıtık mimari, semantik CEP, optimizasyon ve zaman senkronizasyonu, geçici semantik gibi zorluklar ile karşılaşılmaktadır[9,53-55]

<u>CEP tahmin</u>

CEP uygulamaları gerçek zamanlı olarak anlık akış verisini incelemektedir. Veriler arasındaki korelasyon ve zaman bilgileri gibi özellikleri kullanarak küçük gecikmeler ile karmaşık olayları yakalamaktadır. Yani CEP sistemleri geçmiş veriler ile ilgilenmemekte

sadece sorgunun içerdiği zaman aralığında karmaşık olayları tespit etmektedir. Makine öğrenmesi ise geçmiş verileri kullanarak kümeleme, tahmin, örüntü çıkarma vb. alanlarda kullanılmaktadır. Literatürde CEP ile makine öğrenmesi teknikleri aynı mimarilerde kullanılarak geleceğe dönük karmaşık olayların çıkarılması işlemi gerçekleştirilmektedir [9,56,57] CEP mimarisi içerisinde makine öğrenmesi tahmin yetenekleri kullanılarak atomik verinin tahmini gerçekleştirilmiştir. Tahmin edilen verilerin CEP işleme motorunda kullanılarak geleceğe dönük karmaşık olaylar çıkarılabilmektedir [9,10].

Otomatik kural çıkarımı

CEP mimarisinde karmaşık olay belirleme işlemi sorgulara göre yapılmaktadır. Bu sebeple sorgu belirleme işlemi oldukça önemli olmaktadır. Genel olarak sorgular alan uzmanları tarafından manuel tanımlanmaktadır. Bu sebeple veriler arasındaki ilişkilerin ve bağımlılıkların alan uzmanları tarafından bilinmesi gerekmektedir [49]. Sensör teknolojisinin gelişmesi ile birlikte üretilen veri miktarı ve çeşidi artmaktadır. Olay sorgularının tanımlama zorluğu giderek karmaşıklaşmaktadır. Bu sebeple alan uzmanlarının sorgu tanımlaması giderek zorlaşmaktadır [58]. Bununla birlikte değişen ortam şartlarına göre kuralların güncellenmesi gerekmektedir. Dinamik IoT uygulamalarında değişen çevre şartlarına göre kuralların güncellenmesi veya optimize edilmesi gerekmektedir [10].

Dağıtık mimari CEP

IoT ‘nin gelişmesi ile birlikte akıllı cihazlar geniş coğrafik bölgelere yerleştirilip akış verisi üretmektedir. Akış verilerinin dağıtık mimari üzerinde geçerlenmesi, sorgu dağıtımı veya sonuçlarının birleştirilmesi gibi hususlar dağıtık mimarinin yönetimi için gerekmektedir [38]. Dağıtık mimari içerisinde işleme motorunun dağıtık olarak gerçekleştirilmesinin merkezi çözüme göre ağ trafiğini azalttığı ifade edilmiştir [59]. Bu kapsamda dağıtık mimari içerisinde sorgu optimizasyonu, sorgu geçerlenmesi, sorgu çalıştırılması ve işleme kısımları karşılaşılan zorluklar olarak ifade edilebilir.

Optimizasyon ve zaman senkronizasyonu

IoT’nin dağıtık yapısı gereği heterojen bir yapı oluşmaktadır. Farklı kaynaklardan gelen akış verileri birbirlerinden farklı olması sebebi ile optimizasyon teknikleri kullanılması

gerekmektedir. Akış verilerinin farklı ortamlardan gelmesi sebebi ile oluştukları zaman dilimleri veya zamanlar arası fark oluşabilmektedir. CEP sistemi zamana duyarlı olarak çalışması sebebi ile bu fark çeşitli problemlere yol açabilmektedir [55].

Semantik CEP

Olayların zamansal boyutları, sırası ve bağımlılıkları başka bir olayın algılanmasında veya anlamlandırılmasında önemli olmaktadır [55]. Klasik CEP teknolojileri sadece akış verileri ile ilgilenir, arka plan bilgileri dikkate almamaktadır. Semantik olarak akış verilerini anlamlandırmak için ise arka plan bilgilerine ihtiyaç duyulmaktadır [60]. Olay işleme ile semantik teknolojisinin birleştirilmesi, olay işlemenin zenginleşmesine yol açacağı ifade edilmiştir. Semantik olarak zenginleştirilmiş bir olay işleme motoru olay reaksiyonları, süreçleri başlatacağı durumları ve yeni olayların belirlenmesinde fayda sağlayacağı ifade edilmiştir [61].

2.2. Nesnelerin İnterneti

IoT kavramının oluşması internetin 1989 yılında ortaya çıkmasından on sene sonra olmuştur. İnternetin ortaya çıkması ile birlikte, geçmişten günümüze internete bağlı cihaz sayısı artmaya başlamıştır. IoT kavramı ise Massachusetts Institute of Technology (MIT) Auto-ID merkezi tarafından 1999 yılında ortaya sürülmüştür [62]. Asıl ününü ise 2000 yılında yılın da LG firması tarafından akıllı buzdolabı ile duyurmuştur. 2000 ile 2011 yılları arasında ilgi duyulan bir alan olmasına karşın asıl IPV6 ‘nın 2011 yılında kullanılmaya başlaması ile birlikte herkesin dikkatini çekmeyi başarmıştır [63]. Başlangıçta RFID sistemleri ile geliştirmeler yoğunluklu olmasına karşın günümüzde her alanda yer almaktadır. Sensör teknolojisinin gelişmesi ile birlikte kullanım alanı giderek yaygınlaşmıştır.

2.2.1. IoT mimarisi, bileşenleri ve teknolojileri

IoT uygulamalarında nesnelerden gelen hizmetlerin insan etkileşimi olmadan katmanlar arası iletişimi ve işbirliği sağlanmaktadır. Bu işbirliğinin sağlanması amacı ile çeşitli bileşenler kullanılmaktadır. Bu temel bileşenler nesneler, ağ geçişleri, ağ ve bulut altyapısından oluşmaktadır [64].

- Nesneler: İnsan etkisi olmadan herhangi bir bölge veya özel bir alan içerisinden verilerin toplanması ve iletişim sağlayan cihazlar olarak ifade edilmektedir.
- Ağ geçişleri: Nesneler ve bulut altyapısı arası bağlantı sağlayan ara blok olarak tanımlanmaktadır.
- Ağ altyapısı: Verinin buluta doğru akışını kontrol eden altyapıdır.
- Bulut altyapısı: Analitik, mantıksal ve gelişmiş işleme yeteneklerinin sağlandığı altyapıdır.

Günümüzde IoT uygulamalarına artan ilgi ile birlikte IoT bileşenleri de farklılaşmaktadır. Bu kapsamda özellikle akıllı sistemlerin gelişmesi ile birlikte aşağıdaki IoT bileşenleri ortaya çıkmıştır [65].

- Tanımlama: Benzersiz ve açık kimlik yönetimi ayrım yapılarak hassas ve normal veriye erişim mekanizması sağlar. (EPC adlandırma, uCode, MAC adresi)
- Algılama: Fiziksel veya dijital cihazlar aracılığı bilgiyi toplayan ve bağlı cihazlara aktarma işlemidir. (sensörler, kameralar, kart okuyucular vb.
- İletişim: Protokoller aracılığı ile kısa ve uzun menzilli olarak verinin taşınmasıdır. (RFID, Bluetooth, NFC, UWB, , BLE, IEEE 802.15.4, WiFi, Z-wave, LTE, 4G, 5G)
- İşleme: IoT uygulamalarını çalıştıran donanım ürünleri bu aşamada yer almaktadır. IoT nin beyni olarak adlandırılır. (donanım ürünleri, yazılım ürünleri, bulut ve sis işleme ürünleri, büyük veri işleme)
- Servisler: Kimliklendirme, bilgi toplama, işbirliği ve tüm mimaride ihtiyaç duyulan servisler olarak ifade edilmektedir.
- Semantik: Bilgiyi mantıkla hale getirmek amacı ile verinin modellenmesi, analiz edilmesi ve sunulması işlemleridir.

Aynı fonksiyonel işlemleri yapan bileşenler gruplanarak IoT mimarisi kavramı ortaya çıkmıştır. Literatürde IoT mimarisinde üç, dört, beş katmanlı ve bulut ve sis tabanlı mimari yapıları karşımıza çıkmaktadır [66]. Bu katmanlara ilişkin açıklayıcı bilgiler aşağıda sunulmuştur.

Üç katmanlı mimari

IoT mimarisinde yer alan katmanları temel olarak üç seviyede sınıflandırmışlardır. Bunlar algılama, ağ ve uygulama katmanıdır [67-69].

Algılama katmanı: Bu katman sensörlerin çalıştığı katman olarak ifade edilmektedir. Yani veri toplayan veya üreten tüm akıllı cihazları kapsamaktadır. Temel olarak bu katmanda bilgiler toplanarak, veri işlemesi yapılır ve ağ katmanına gönderilir [67].

Ağ katmanı: Ağ katmanı içerisinde yer alan cihazlar aracılığı ile IoT katmanları arası iletişim sağlanan katmandır. Veri aktarımının sağlandığı katmandır [69].

Uygulama katmanı: Son kullanıcıların bulunduğu katmandır. Bu katmanda akıllı sistemler aracılığı ile son kullanıcıya hizmet sağlanır. Bununla birlikte veriye ait gerçeklik, bütünlük ve gizlilik gibi unsurlar bu katmanda sağlanır [67].

Dört katmanlı mimari

Dört katmanlı mimari yapısında akıllı cihaz/sensör, ağ, servis yönetim ve uygulama katmanından oluşmaktadır [70,71]. Bu katmanlara hangi işlemlerin yapıldığına ilişkin açıklamalar aşağıda sunulmuştur

Akıllı cihaz/sensör: Akıllı cihazlar ve sensörlerin bulunduğu en alt katmandır. Fiziksel cihazların konumlandırılıp dijital dünya ile iletişim kurduğu katmandır [70]. Dijital dünyadan algılanan veriler bir diğer katmana iletilmektedir.

Ağ katmanı: Sensörler tarafından üretilen büyük miktardaki verinin kablolu veya kablosuz ağlar aracılığı ile farklı teknolojiler kullanarak katmanlar arası iletim işleminin yapıldığı katmandır [70,71].

Servis yönetim katmanı: Ağ katmanından gelen bilgilerin süreç, yönetim, analitik ve güvenlik prosedürlerinin uygulandığı katmandır. Veri yönetimi ile birlikte sadece ihtiyaç duyulan bilginin gizlilik ve gerekli güvenlik önlemleri ile bir üst katmana sunulmasını sağlar [70].

Uygulama katmanı: Ulaşım, yaşam, sağlık, turizm, çevre ve benzeri alanlardaki son kullanıcıya sunulan akıllı uygulamaları kapsayan katmandır.

Beş katmanlı mimari

Uluslararası Telekomünikasyon Birliği tarafından IoT mimarisi beş katman olarak tanımlanmıştır. Bunlar algılama, taşıma, ağ, orta ve uygulama katmanı olarak ifade edilmiştir. Üç katmanlı mimariye ek olarak erişim ve orta katman kavramları eklenmiştir [72]. Bununla birlikte beş katmanlı mimari yaklaşımında algılama, taşım, işleme, uygulama ve iş katmanları yaklaşımı da bulunmaktadır [73].

Algılama katmanı: Nesnelerin algıladığı sinyallerin ağ katmanında iletebilecek nitelikte dijital veriye dönüştürüldüğü katmandır.

Erişim ve Taşıma katmanı: Ağ katmanı ile aynı işleve sahip olan taşıma katmanı algılama katmanından aldığı dijital sinyalleri ağlar arası iletiminden sorumludur. Bu katmanda FTTx, 3G,4G, kablosuz ağ, bluetooth, Zigbee, UMB, kablolu iletim gibi benzeri teknolojiler kullanılmaktadır [69].

İşleme katmanı: Üç katmanlı mimari içerisinde uygulama katmanında yer almaktadır. Soyut katman olarak ifade edilmektedir. Bu katmanda bulut ve veri analitiği işlemleri yapılmaktadır [73]. Başka bir ifade ile orta katman olarak ifade edilmektedir [74].

Uygulama katmanı: Farklı IoT teknolojilerin kullanılması ile birlikte IoT uygulamalarının gelişimini ve yayılımının sağlandığı katmandır. Bu katman günümüzde de popüler olan akıllı ev, akıllı bina, akıllı şehir ve benzeri uygulamaların olduğu katmandır [73].

İş katmanı: IoT sistemi içerisinde yer alan uygulamaların ve modellerin yönetildiği katmandır. Bu katman uygulama katmanından gelen verileri işleyerek geleceğe dönük eylemleri belirler [73].

IoT uygulamalarında alt katman cihazlarından toplanan verinin taşınması için internet ortamı kullanılmaktadır. Bu cihazların birbirleri ile bağlanması ve katmanlar arası iletişim farklı teknolojilerin birbirleri ile entegrasyonuna bağlı olmaktadır. IoT uygulamalarının gelişmesi

ile birlikte kullanılan teknoloji ve yöntemler giderek artmaya başlamıştır. Bunlar arasında RFID, Anten, WSN, IPV6, EPC, Barkod, Blueetooth ve ZigBee teknolojileri sayılabilir [75].

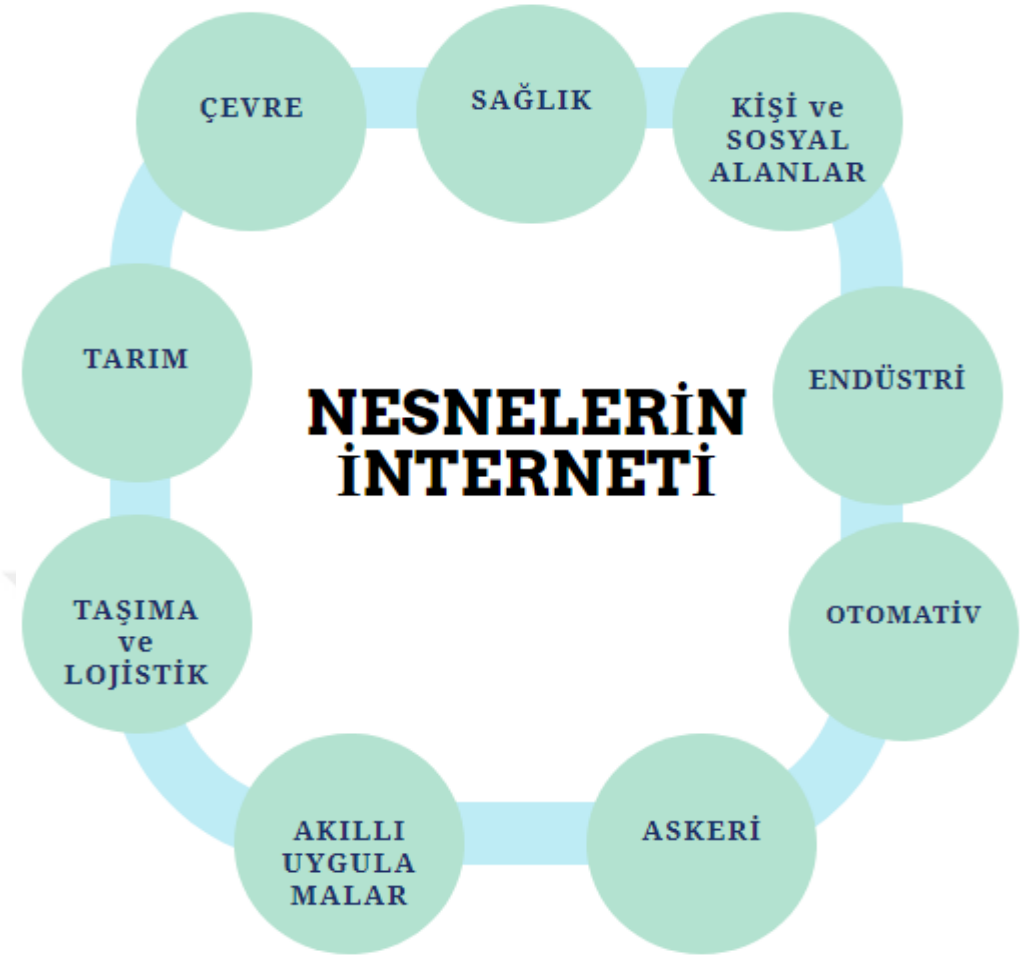
Bulut ve sis tabanlı mimari

Bulut işleme mimarisi büyük miktardaki verilerin depolanması için kullanılmaktadır. Sis hesaplama ise yeni bir işleme mimarisi olarak sensör ve ağ geçitlerinde bilginin işlenmesi ve analizi için ortaya çıkmıştır. Bulut ve sis tabanlı mimari katmanları nesneler, sis düğümleri, sis veri servisleri ve bulut yapısından oluşmaktadır [1]. Nesne katmanında sensörler ve uç sistemde yer alan cihazlar yer almaktadır. Sis katmanında ise daha küçük çaplı veri işleme ve yönlendirme yapan yerlerdir. Bulut katmanında ise sis katmanında yapılamayan genellikle zaman bağımsız işlemler yapılmaktadır [76].

2.2.2. IoT uygulama alanları

IoT, araştırmacıların ilgisini çekmesiyle birlikte birçok alanda uygulamalar geliştirilmiştir.

Literatürde IoT alanı ile ilgili çalışmaların artması ile birlikte günlük hayatta birçok alanda IoT uygulamaları görülmektedir. Bunlar sağlık, endüstri, otomotiv, askeri, çevre, tarım, taşıma ve lojistik, akıllı şehir, akıllı ev, kişi ve sosyal alanlar olarak gruplandırılmıştır [1].



Şekil 2. 4. Nesnelerin interneti uygulama alanları

Sağlık: Akıllı sensörler sayesinde düzenli olarak yapılan ölçümler ile sağlık verilerinin izlenmesi ve analiz edilmesine imkan sağlanmıştır. Örneğin kalp ritmi, kan basıncı, oksijen satürasyon seviyesi gibi giyilebilir akıllı cihazlarla elde edilebilen verilerin işlenmesi ile birlikte kişilerin sağlık durumları hakkında bilgi alınabilmektedir. Bununla birlikte kritik hasta koşulları, medikal ve ilaç endüstrisi alanında da uygulamalar mevcuttur [77].

Endüstri: Endüstriyel IoT sensörlerden üretilen verinin toplanması, birleştirilmesi ve analiz edilmesi ile üretim ve benzeri safhalarda makinelerin verimliliklerini artırmayı sağlamaktadır. Bu alanda akıllı otomasyon sistemleri, hareket kontrolü, makineden makineye iletişim ve akıllı şebekeler yer almaktadır [1].

Otomotiv: Araç kullanıcıları ve diğer araçlar arası bilgi paylaşımı sayesinde trafik ve kaza oranlarını aza indirmek amacı ile genel olarak verimliliği artırmaya yönelik çalışmalar

bulunmaktadır. Bununla birlikte üretim endüstrisinde yapılacak iyileştirmeler için uygulamalar yer almaktadır [78]. Özellikle son dönemde popüler olan sürücüsüz araçlar ile kaza oranlarının azaltılması hedeflenmektedir. Bu kapsamda yapılan araştırmalar dünya çapında 2040 yılında sürücüsüz araçların yüzde 75 oranında olacağı tahmin edilmektedir [79].

Askeri: Askeri alanda gerçekleştirilen IoT uygulamaları askeriye ile alakalı oluşabilecek riskleri hafifletmeyi ve bunu mevcut IoT teknolojileri ile çözmeyi hedeflemektedir. Özellikle sensör teknolojisinin gelişmesi ile birlikte savaş alanının izlenmesi, askerlerin savaş alanında sağlığının izlenmesi veya özel bölgelerin izlenmesi ile oluşacak verilerin IoT konseptine göre değerlendirilmesi gibi uygulamalar ön plana çıkmaktadır [80].

Çevre: Atık yönetimi, su yönetimi, hayvanların takibi ve korunması, hava durumu izleme sistemleri, hava durumu tahmin, çevre koruma ve ticari amaçlı çiftlik gibi birçok uygulamaları içermektedir. Bununla birlikte mevsimsel değişiklikleri izlemek amacı ile izleme ve tespit etme sistemleri de bulunmaktadır [1].

Tarım: IoT ile birlikte tarıma bağlı sektörlerde bilgili, gelişmiş ve uyarlanabilir bir kırsal topluluk oluşturulmak istenmektedir. Düşük maliyetli cihazlar insan etkileşimini iyileştirerek değerli analizler üretilmesi sağlanmaktadır [81]. Sensörler aracılığı ile çevre ve toprak koşullarının takibi ile faaliyetlerinin etkin yönetimi verimliliğin artmasını sağlayacaktır.

Taşıma ve Lojistik: tren, otobüs ve araba gibi tüm araçların yer aldığı taşıma ağı sensörler ile birlikte daha yönetilebilir olmaktadır. Ulaşım ağında taşınan malların efektif olarak yönlendirilmesi amacı ile trafik sıkışıklığını kontrol altına almak uygulama alanları içerisinde yer almaktadır. Lojistik, mobil etiketleme, çevre izleme ve geliştirilmiş harita seçenekleri uygulamalar arasındadır [82].

Akıllı uygulamalar: Akıllı uygulamalar günlük yaşantımızın her aşamasında karşımıza çıkmaktadır. Yaşadığımız alanları daha kolay ve yönetilebilir hale getirme hedefi taşımaktadır. Akıllı ev ve binalarda gerçekleştirilen uygulamalar sayesinde enerji ve tüketim maliyetleri azaltılarak optimize edilerek yönetim açısından şeffaflık sağlanmaktadır [83].

Kişisel ve sosyal alanlar: Sosyal ağ uygulamaları, kişisel bilgi yönetimi ve sosyal medyayı kapsayan uygulamaları bünyesinde barındırmaktadır [84]. Bununla birlikte IoT cihazlarının sosyal aktivitelere entegre edilmesi ile birlikte akıllı alışveriş, sağlık ve teletıp, akıllı multimedya verileri ve akıllı sosyal etkileşim uygulamaları ile kişisel ve sosyal çevre ile etkileşim yönetimi sağlanarak sosyal hayatın kolaylaştırılması sağlanmaya çalışılmıştır [85].

2.2.3. IoT zorlukları

IoT geniş bir alanı kapsaması sebebi ile bilim adamları ve araştırmacıların ilgisini çekmektedir. Bu kapsamda yapılan çalışmalar geliştirme aşamasında yer almakta olup geliştirilmesi gereken birçok unsur bulunmaktadır. Genel olarak IoT alanında yapılması gereken araştırma zorlukları mimari, uygulama, ağ, gizlilik, güvenlik, standartlar olarak ifade edilmiştir [84]. Bununla birlikte daha spesifik olarak [86] çalışmasında aşağıdaki hususlar ifade edilmiştir.

• İnternetin her yerde ve ücretsiz olarak kullanılabilirliği: İnternetin her yerden erişebilmesi mevcut sisteme ek olarak maliyet açısından yük getirmektedir. Sadece kurulum maliyetinin yanı sıra işletme maliyeti de oluşacaktır. Bununla birlikte gizlilik güvenlik gibi kavramların bu yapıya uygun olarak güncellenmesi gerekecektir [86].
• Güvenlik sorunları: IoT mimarisinin gereksinimi olarak her katmanda farklı güvenlik gereksinimlerine ihtiyaç duyulabilmektedir. Bu kapsamda [87] çalışmasında güvenlik ile alakalı unsurlar aşağıdaki şekilde ifade edilmiştir.

i) IoT cihazlarının fiziksel müdahalelere karşı dayanaklı olması ve sadece yetkili kullanıcılar tarafından güncellenmesi gerekir. Yani nesnelerin veri kaybına ve sızıntısına izin vermeden gerekli güvenlik önlemleri alınabilmedir.
ii) IoT cihazları belleklerinde kullanılan şifreleme mekanizması ve tutulan veriler gizlilik prensiplerine uygun şekilde tasarlanmalıdır. Yani veri gizliliği, güvenliği ve mahremiyet kurallarının her nesne ve katman için tasarlanması gerekmektedir.
iii) IoT cihazları arası iletişim, gizlilik ve bütünlük prensiplerine uygun olarak yapılmalıdır. Nesneler tarafından kablolu veya kablosuz olarak gerçekleştirilen veri iletimi için gerekli güvenlik önlemleri alınmalıdır.

iv) IoT kaynaklarına sadece yetkili kurum ve kişilerin erişebilmesi için tanımlama ve yetkilendirme mekanizmaları oluşturulmalıdır. Bu kapsamda izinsiz erişim mekanizmasını sağlayacak önlemler gerekmektedir.

v) Sistem, kötü niyetli kullanıcılar tarafından cihazlara fiziksel hasar verilse bile normal çalışma içinde hazır olmalıdır. Bu, IoT'nin sağlamlığını garanti eder.

• Düşük maliyetli akıllı algılama sistemi geliştirme: IoT gelişmesi ve genişlemesi ile birlikte çevremizde milyonlarca sensör olacaktır. Sensörler ile birlikte kurulan sistem gereksinimleri artacaktır. Bu kapsamda karşılaşılan en önemli zorluklardan biriside bunları çok ucuza geliştirmek ve imal etmektir [65].

• Enerji: IoT içerisinde yer alan nesnelerin piller ile yıllarca çalışması beklenmektedir. Bu durum her bir nesnenin ömrü boyunca pil değişimine gerek kalmadan düşük güçte enerji açısından verimli çalışan sensör tasarımını gerektirmektedir. Bununlar birlikte enerji verimliliği açısından IoT mimarisindeki katmanlar düşünülerek tasarım ihtiyacı oluşacaktır [66].

• Hesaplama yeteneği: IoT uygulamaları genel olarak gerçek zamanlı olarak çalışmaktadır. Daha spesifik, alana özgü kabiliyetleri olan uygulamalar zaman bağımlı olarak çalışmaktadır. Bu kapsamda belli zaman içerisinde belli işlemlerin yapılması gerekmektedir. Bununla birlikte, tek tek uç düğümlerinin hesaplama kapasitesi sınırlı olması sebebi ile uç hesaplama kapasitesinin ölçeklenebilirliği ayrı bir sorun olarak karşımıza çıkmaktadır [67]. Aynı zamanda heterojen ortamdan sensörlerin ürettikleri verilerin analiz edilmesi için yeni gelişmiş yöntemlere ihtiyaç duyulmaktadır.

• Ölçeklenebilirlik: IoT mimarisinde çok sayıda sensör yer almaktadır. Bu sensörlerin belli zaman aralığında veri iletimini gerçekleştirmesi gerekmektedir. Daha çok sayıda cihaz arasında ağ oluşturma ve iletişim sağlamak IoT 'nin bir başka zorluğudur. Bu zorlukların üstesinden gelmek için ölçeklenebilir yaklaşımlar önerilmektedir. Bu kapsamda bulut ve sunucu mimarisi ile birlikte ağ ve iletişim protokolleri düşünülerek geliştirmelerin yapılması önerilmektedir [68]. Bu sebeple IoT ’ye bağlanan nesne sayısının orantısal olması gerekmektedir.

• Hata Toleransı: IoT ’de sürekli değişen ortam varlığı sebebi ile nesnelerin kolayca uyum sağlaması ve güvenilir bir yedekleme sistemi önemli olmaktadır. Uç sistem sensörlerinin kısıtlı cihazlar olması sebebi ile kötü niyetli sistemler doğrudan veya dolaylı olarak tarafından uç sistem nesnelerini üzerinde kontrol sağlanmaya çalışacaktır. Bu sebeple hata toleransının ölçeklenebilirlik açısından değerlendirilmesi gerekmektedir [69].

• Güç tüketimi: Sensör bataryalarının seçiminde maliyet, büyüklük, hafıza, çevre etkisi ve yaşam ömrü gibi parametrelere dikkat edilmektedir. Bununla birlikte çevre koşulları ile tekrar şarj edilebilen bataryalar kullanılmaktadır [65]. Bununla birlikte sensörlerin işleme ve iletişim işlemleri yoğunluklu olarak gücün tüketiminde kullanılmaktadır. İletişim ve işleme yöntemlerinde etkin yöntemlerin belirlenmesi için yeni yöntemlere ihtiyaç duyulmaktadır.
• Toplum içinde kabul edilebilirlik: Günümüzde IoT uygulamaları mevcutta gerçek yaşamda bulunmayan, ileriye dönük projeler olarak karşımıza çıkmaktadır. Bu sebeple kişilerin bu projeleri benimsemeleri ve günlük yaşam içerisinde kullanmaları önem taşımaktadır [70].

2.3. Tezde Kullanılan Makine Öğrenmesi Teknikleri

Bu alt bölümde tez kapsamında kullanılan yöntem ve algoritmalar anlatılmıştır.

2.3.1. Veri madenciliği yöntemleri

Veri madenciliği büyük veri kümelerindeki ilginç, beklenmedik örüntülerin veya önemli bilgilerin keşfedilmesidir [88]. Bu kapsamda veri madenciliği algoritmalarından kümeleme, regresyon yöntemleri ile tahmin ve kural çıkarım algoritmaları tez bölümlerinde kullanılmıştır. Bu bölümde tez kapsamında kullanılan veri madenciliği yöntemleri sunulmuştur.

K-Means kümeleme

Kümeleme çok boyutlu veri üzerinde benzer nesnelerin bulunması amacı kullanılan veri madenciliği tekniğidir. Her küme veya grup kendi içerisinde benzer karakteristik özellikleri sahiptir. Bu kümelerin belirlenmesi için önerilen K-Means algoritması basit ve kümeleme problemlerine literatürde çoklukla uygulanan bir yöntemdir. Kümeleme algoritmalarının başarımı küme sayısının doğru seçilmesi ile doğru orantılıdır. Bu kapsamda literatürde aşağıdaki yaklaşımlar küme sayısını belirleme işlemi için kullanılmaktadır [89].

- Elbow method
- Information Criterion Approach
- An Information Theoretic Approach

- Choosing k Using the Silhouette
- Cross-validation
- Canopy kümeleme [90]

Küme sayısı algoritması başlangıç küme sayısını belirler ve K-means algoritmasına girdi üretir [91].“k” değerinin belirlenmesinden sonra K-means algoritması ile veri “k” adet kümeye bölünmektedir. Algoritma kümenin merkezi ile başlayarak ilk aşamada her örnek rastgele yerleştirilmektedir. Ardından kümeleme hatasını en aza indirmek için merkezi noktayı değiştirilerek işleme devam edilmektedir [92]. Verilerin hangi kümeye ait olduğunu belirlemek amacı ile Öklid uzaklığı [93], Manathan uzaklığı ve Minkowski uzaklığı tercih edilen yöntemlerdendir [90].

Öklid uzaklığı: $d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + . + (x_{in} - x_{jn})^2}$ (2.1)

Manathan uzaklığı: $d(i,j) = \left|(x_{i1} - x_{j1})\right| + \left|(x_{i2} - x_{j2})\right| + \ldots + \left|(x_{i1} - x_{j1})\right|$ (2.2)

Minkowski uzaklığı: $d(i,j) = (\left|(x_{i1} - x_{j1})\right|p + \left|(x_{i2} - x_{j2})\right|p + \ldots + \left|(x_{i1} - x_{j1})\right|p)\ 1/p$ (2.3)

x, y iki boyutlu veri nesnesini ifade etmektedir.

Random Forest Regresyon

Regresyon yöntemleri sürekli değişkenlerin tahmin problemlerinde yaygın olarak kullanılmaktadır. Random Forest Regresyon (RFR) ilk olarak 2001 yılında Breiman tarafından önerilmiştir [94]. RFR başlangıçta bir karar ağacı oluşturarak verilen değerlerle tahminlerini yapmak için kullanılmaktadır. Oluşturulan karar ağacı tekrarlanan bir şekilde bölümlere ayıran ağaç tabanlı bir yöntemdir [95]. Algoritması aşağıdaki şekilde ifade edilmektedir [96].

1. Orijinal veri üzerinden başlangıç yordam ağacı oluşturulur
2. Her bir başlangıç yordam değeri için regresyon ağacı geliştirilir. Her düğüm için en iyi seçim yerine rastgele örnekleme yapılarak en iyi ayrım seçilir.
3. Ağaç tahminleri kullanılarak yeni veri tahmin edilir. Regresyon için ortalama tercih edilir.

RFR rastgele olarak seçilen yordam örneklemleri ile başlar. Her bir yordam ağacı örneklemi için ağaç tekrar oluşturulur. Her düğüm için en ayrılma rastgele yapılarak ağaç bölme kriteri olarak en düşük gini indeksine sahip olan girdi seçilir. Gini indeksi Eş. 2.4'te verilmiştir [97].

Gini İndeksi: $$I_G(t_{X(x_i)})=1-\sum_{j=1}^{m} f(t_{X(x_i)},j)^2 \quad (2.4)$$

k-Nearest-Neighbors Regresyon

k-nearest negihboors (KNN) algoritması denetimli bir öğrenme yöntemidir. İki çeşidi bulunmaktadır. Bunlar KNN sınıflandırma ve KNN regresyon yöntemleridir. KNN regresyon, sınıflandırma algoritmasına göre daha az tercih edilen bir yöntemdir [98]. KNN regresyon yönteminde örnekler bir sınıf veya kategori yerine ilişkili bir sayısal hedef değerine sahiptir. Hedef değerini tahmin etmek için "k" en yakın komşunun hedef sayısal değerlerinin ortalaması veya medyanı ile belirlenmektedir [99]. KNN regresyon algoritması genellikle öklid uzaklığını kullanmaktadır. Bununlar birlikte KNN regresyon özellikle gürültülü veri olması durumunda olumsuz performans sergilemektedir [98].

Destek Vektörü Regresörü

Destek vektör makineleri (SVM) ikili sınıflandırma problemlerini çözmektedir. Destek vektörü regresörü ise destek vektör makinelerinin genişletilmiş bir halidir [100]. SVR, eğitim verilerine dayalı olarak bir girdi dizisini ele alarak gerçek bir sayıya eşleyen bir tahmin etme yöntemidir. SVM sınıflandırmasına benzer şekilde, doğrusal olmayan haritalama için marjin maksimizasyonu ve çekirdek numarasıyla aynı özellikleri kullanır [101]. SVR deneysel ölçüm hatası ve yapısal riski minimuma indirmeyi hedeflemektedir. SVR daha çok doğrusal olmayan regresyon problemlerinin çözümünde kullanılmaktadır [102].

Kural madenciliği algoritmaları

Kural madenciliği algoritmaları etiketli veriden kurallar çıkarmaktadır. Parçala ve fethet yöntemi kullanmaktadır [103]. Kural madenciliği algoritmaları boş bir kural kümesi ile

başlayarak bu kural kümesine yeni kurallar ekleyerek çalışır. Her aşamada performans başarımını ölçerek ilerlemektedir. Kural madenciliği algoritmaları ile anlamlı ve rahatlıkla yorumlana kurallar elde edilebilmektedir. Kolayca yorumlanabilir olması sebebi ile birçok alanda rahatlıkla kullanılabilir. Bu kapsamda tez çalışmasında kullanılan kural madenciliği algoritmaları aşağıda verilmiştir [58].

Projective adaptive resonance theory (PART): Algoritma C4.5 karar ağacı yapısını kullanır. Tekrarlanan artırımlı budama tekniği ile hataları azaltarak kuralları oluşturur. Yerel optimizasyon kullanır. İlk kural oluşturulduktan sonra, bu kuralın örnekleri kaldırılır. Diğer kurallar, tüm örnekler kaldırılana kadar aynı şekilde oluşturulur [12].

Karar Tabloları: Kural tabloları oluşturarak her sütun kural olarak tanımlanır [104]. Karar kalitesini değerlendirerek mantıksal kurallar üretir. Özellik alt kümelerini değerlendirmek için en iyi ilk aramayı ve çapraz doğrulamayı kullanır [105,106].

JRIP: Hata azaltma için artırımlı budama tekniği kullanan bir algoritmadır. Büyüme, budama, optimizasyon ve seçme aşamalarında oluşmaktadır [107]. Pozitif örnekleri dikkate alarak kural seti oluşturur. Başlangıçta kural seti boş olarak başlar. Artırımlı olarak kural setine ekleme yapılarak negatif örneklerden arındırma işlemi yapılır [104].

ONE-R: Tek seviyeli bir karar ağacı oluşturarak kuralları düşük hata oranına göre oluşturur [108]. Sadece veri seti üzerinden tek nitelik kullanarak kuralları oluşturmayı hedefler [106,109].

RIDOR: Algoritma direk sınıflandırma yapar. İlk olarak başlangıç kurallarını üretir. Her kural için istisnalar üreterek işleme devam eder. Sonrasında artırımlı budama tekniği kullanarak hataları azaltmayı hedefler. İstisna olarak nitelendirilen kuralları en iyi istisnaları oluşturasıya kadar temizleyerek işleme devam eder [12,107].

NNGE: Aşırı genelleme probleminin çözümü için önerilmiştir. Büyük dikdörtgenler benzeri genelleştirilmiş örnekleri ele alarak işleme devam eder. Verilere her seferinde yeni bir örnek eklenir, aynı sınıfın en yakın komşusu seçilerek bir genelleme yapılır[12,105].

FURIA: RIPPER algoritmasının genişletilmiş hali olarak ifade edilmektedir. Geleneksel kurallar yerine bulanık kurallar üretir ve kural listesi yerine sırasız kural kümeleri kullanır [110].

2.3.2. Derin öğrenme yöntemleri

Derin öğrenme yöntemli birçok alanda üstün performansı sebebi ile tercih edilen yöntemlerdir [111,112].Yüksek boyutlu veriler üzerinde güçlü hesaplama yetenekleri sebebi ile birçok alanda kolaylıkla uygulanabilmektedir. Yapısında kullanılan derin sinir ağları ile görüntü, metin, zaman serisi, ses ve benzeri olarak nitelendirilen verilerden veri özelliklerini otomatik olarak çıkartabilmektedir [113]. Bilgisayarla görme, tahmin, semantik analiz, doğal dil işleme, bilgi çıkarımı, müşteri ilişkileri yönetimi gibi birçok alanda uygulamaları bulunmaktadır. Uygulama alanları içerisinde RBM, AE, LSTM, GRU, CNN, RNN, DNN ve pekiştirmeli öğrenme gibi birçok teknik kullanılmaktadır. Tez çalışması kapsamında adreslenen problemlere aşağıda belirtilen derin öğrenme yöntemleri zaman serilerinin tahmini için kullanılmıştır.

Otomatik kodlayıcılar (AE): AE derin öğrenme yöntemlerinin temelini oluşturmaktadır. Genel olarak denetimsiz öğrenme de kullanılmakla birlikte transfer öğrenme gibi birçok alanda uygulaması bulunmaktadır [114]. Yeniden çatılma işlemi için aşamasında kodlayıcı ve kod çözücü kullanılmaktadır. Girdi olarak ifade edilen x değerini daha düşük boyutlu z çıktısına dönüştürür. Kod çözme işleminde, kodlanan z değeri x girdisinin yaklaşık temsiline dönüştürülür. Başka bir ifade ile orijinal boyutuna yeniden yapılandırır. Bu bağlamda girdi ve çıktı arasında yeniden çatılması hatası (RE) hesaplayarak hata oranını aza indirir. Kodlama işlemi, çıktı hesaplanması ve yeniden çatılma hatası Eş. 2.5, Eş. 2.6, Eş. 2.7'de verilmiştir [115,116].

Kodlama işlemi: $$z = \sigma(W\,x + b) \tag{2.5}$$

Çıktı: $$\bar{x} = \sigma(W'\,z + b') \tag{2.6}$$

Yeniden çatılma hatası $$RE = (\bar{x} - x) \tag{2.7}$$

x girdi, W, W′ kodlama ve kod çözme ağırlıkları, b, b′ sapma oranları, σ sigmoid fonksiyonu olarak tanımlanmıştır.

Yinelemeli sinir ağı (RNN): RNN yapay sinir ağlarının bir alt dalı olup zaman serilerinde dinamik değişim modelleme problemini çözmek amacı ile kullanılır. Basit yapısı sebebi ile doğal dil işleme, ses tanıma ve el yazısı tanıma gibi birçok problemde kullanılmaktadır [117]. RNN yapısı girdi, gizli ve çıktı katmanlarından oluşmaktadır. En önemli aşaması gizli katmanda gerçekleşmektedir. Bu katmanda önceki girdi ve çıktılar hatırlanarak işlem yapılmaktadır. Sonuç olarak aynı girdi dizisi farklı sonuçlar üretebilir. RNN yapısı gereği girdi den çıktı üretilmesine kadar birbiri ile bağımlıdır [118]. Aşamalar Eş. 2.8 ve Eş. 2.9'da ifade edilmiştir.

Gizli katman: $$h(t) = f_H\ (W_{IH}\ \mathrm{x(t)} + W_{HH}\ \mathrm{h(t\text{-}1)}) \quad (2.8)$$

Çıktı: $$y(t) = f_O\ (W_{HO}\mathrm{h(t)}) \quad (2.9)$$

Uzun kısa süreli bellek (LSTM):LSTM Hochreiter ve Schmidhuber[119] tarafından önerilmiştir. Zaman serisi tahmini sorununu ele alan güçlü bir tekniktir [120]. LSTM, uzun vadeli bağımlılıkları öğrenmek için yenilikçi bir bellek hücresine sahip olan RNN yapısının özel bir versiyonudur [121]. Geleneksel LSTM, geçmiş giriş bilgilerini korurken, çift yönlü LSTM, geçmiş ve gelecekteki giriş bilgilerini korumaktadır. Ayrıca, bloğun ve çıktının [122] durumunu idare etmek için giriş geçidi, unutma geçidi ve çıkış geçidi dahil olmak üzere üç tip geçit içeren bellek bloklarına sahiptir. Bu kapılar aşağıdaki gibi formüle edilebilir [123].

Blok girdi: $$z^t = \mathrm{g}(\mathrm{W_z x^t} + \mathrm{R_z y^{t-1}} + b_z) \quad (2.10)$$

Giriş Kapısı: $$i^t = \sigma(\mathrm{W_i x^t} + \mathrm{R_i y^{t-1}} + p_i \odot c^{t-1} + b_i) \quad (2.11)$$

Unutma Kapısı: $$f^t = \sigma\left(\mathrm{W_f x^t} + \mathrm{R_f y^{t-1}} + p_f \odot c^{t-1} + b_f\right) \quad (2.12)$$

Hücre Durumu: $$c^t = i^t \odot z^t + f^t \odot c^{t-1} \quad (2.13)$$

Çıkış kapısı: $$o^t = \sigma(\mathrm{W_o x^t} + \mathrm{R_o y^{t-1}} + p_o \odot c^t + b_o) \quad (2.14)$$

Blok çıktı: $$y^t = o^t \odot h(c^t) \quad (2.15)$$

$\mathrm{x^t}$ girdi vektörü, $\mathrm{W_z, W_i, W_f, W_o}$kapılarda ve girdilerde kullanılan ağırlık matrislerini, $\mathrm{R_z, R_i, R_f, R_0}$ tekrarlanan ağırlık matrislerini, b_z, b_i, b_f, b_o yanılma (bias) vektörlerini, $\odot$ ise Hadamard çarpımını, $\sigma(\cdot)$ ise lojistik sigmoid fonksiyonunu ve $\mathrm{h}(\cdot)$ hiperbolik tanjant fonksiyonunu, g ise blok girdi ve çıkış aktivasyon fonksiyonlarında kullanılan fonksiyonu temsil etmektedir.

Kapılı yineleyen birim (GRU): GRU, ilk olarak Cho ve arkadaşları [124] tarafından, LSTM çalışma zamanı işleminden kaçınmak için LSTM'in genişletilmiş ve değiştirilmiş bir versiyonu olarak 2004 yılında önerilmiştir. GRU, LSTM'in daha basit bir şeklidir ve uzun sıralı uygulamalarda daha iyi sonuçlar verir. Kapıyı sıfırlayan ve geçidi güncelleyen iki geçit kullanır. LSTM'den farklı olarak, giriş ve unutma kapıları GRU yapısındaki güncelleme geçidinde birleştirilir. Birleştirme hücre durumları ve gizli durumlar açısından LSTM'den farklıdır. GRU modellerinin genel yapısı aşağıdaki şekilde formüle edilmektedir [125].

$$\mathrm{h^t} = (1-\mathrm{z^t}) \odot \mathrm{h^t}-1+\mathrm{z^t} \odot \hat{\mathrm{h}}^{\mathrm{t}} \tag{2.16}$$

$$\hat{\mathrm{h}}^{\mathrm{t}}=g(\mathrm{W_h x^t} + \mathrm{U_h}(r_t \odot h^{t-1} + b_z) \tag{2.17}$$

GRU'nun güncelleme ve reset kapıları aşağıdaki şekilde formüle edilmektedir.

Güncelleme kapısı: $$z^t = \sigma(\mathrm{W_z x^t} + \mathrm{U_z h^{t-1}} + b_z) \tag{2.18}$$

Reset kapısı: $$r^t = \sigma(\mathrm{W_r x^t} + \mathrm{U_r h^{t-1}} + b_r) \tag{2.19}$$

$\mathrm{x^t}$ giriş vektörü, z^t güncelleme kapısı vektörü, r^t reset kapısı vektörü, W, U ve b sırası ile ağırlık ve yanılma vektörlerini temsil etmektedir.

Evrişimli sinir ağı (CNN): Zaman serileri ve resim verileri üzerinden gizli nitelikler çıkartılarak nesne algılamada kullanılan bir tekniktir. Konvolüsyon ve havuzlama katmanları ile derin öğrenme sayesinde gizli niteliklerin ham veriden çıkarılması sağlanır [126]. Konvolüsyon katmanında filtreleme yapılarak iki boyutlu matrisler ile veri üzerinde kaydırma işlemi ile anlamlı bilgiler çıkarılır [127]. Giriş verileri bu katmanda daha küçük boyutta temsil edilen öznitelik haritalarına dönüştürülmektedir [128]. Havuzlama katmanında ise özellik seçimi yapabilmek için girdi boyutu azaltılır [129]. Bu sayede hesaplama maliyeti azaltılmaktadır. Havuzlama katmanında ortalama, maksimum ve stokastik yöntemler genellikle tercih edilen yöntemlerdir. Evrişimli sinir ağı yapısı aşağıdaki şekilde formüle edilmektedir [117].

$$w_{out} = \frac{w_{in-F}}{\mathrm{s}} + 1 \tag{2.20}$$

$$w_{out} = \frac{w_{in+2*p-F}}{\mathrm{s}} + 1 \tag{2.21}$$

Sigmoid aktivasyon fonksiyonu: $$f\,(z) = \frac{1}{1+ e^{-z}} \quad (2.22)$$

w_{in} giriş nitelik haritası, w_{out} çıkış nitelik haritası, s konvolüsyon aşaması boyutu, F çekirdek boyut, p piksel sayısı olarak temsil edilmektedir.

3. KARMAŞIK OLAY İŞLEME OLAY SORGULARININ KÜMELEME YÖNTEMİ İLE BELİRLENMESİ

Sensörler ve akıllı cihazların birbirine bağlanarak büyük hacimli, hızda ve çeşitli veriler üretilmektedir. Çoğu alanda yaşanan dijitalleşme ile birlikte veri akışlarının üretimi artacaktır. Akış verilerinde meydana gelen ani değişiklikleri gerçek zamanlı izlemek ve hızlı aksiyonlar almak oldukça önemli olacaktır. Akış verilerinin gerçek zamanlı olarak izlenmesi ve çok boyutlu veri üzerinde belli durumların tespiti CEP sistemleri ile mümkün olmaktadır. CEP sosyal ağlar, trafik izleme, kriz yönetimi gibi farklı alanlarda kullanılmaktadır [130,131]. Temel olarak bir CEP sistemi, atomik olayları karmaşık olaylarla eşleştiren bir hizmettir [46]. CEP sistemine gelen akış verileri, sisteme daha önceden tanımlanan kurallara bağlı olarak analiz edilir. Kurallara uymayan herhangi bir veri göz ardı edilir. Sonuç olarak, CEP kuralları sistemin ana parçasını oluşturmaktadır.

CEP kuralları alan uzmanları tarafından çoğunlukla manuel olarak belirlenmektedir. Verinin üretim aşamasındaki durumu, karmaşıklığı ve verinin kullanılabilirliği kuralların nasıl tanımlanacağını belirlemektedir [8]. Bu kapsamda nesnelerin interneti ile birlikte üretilen verilerin çeşitli olması ve karmaşıklığın artması düşünüldüğünde kuralların otomatik olarak tanımlanması önemli olacaktır. Bir başka deyişle, verinin karmaşıklığının artması ile alanında uzman kişilerin kural tanımlama işlemini yerine getirmesi zorlaşacaktır. Özellikle nesnelerin interneti ile birlikte verinin sıklığı, miktarı ve hetorojenliğinin artacağı düşünüldüğünde olay sorgularının otomatik olarak tanımlanması gerektiği ortaya çıkmaktadır.

Bu kapsamda, IoT kapsamında üretilen akış verilerinin etiketsiz ver olması sebebi ile IoT verilerinden CEP kurallarının otomatik olarak çıkarılmasını sağlamak amacı ile bir model önerilmiştir. Önerilen model günümüzde akıllı şehir uygulamalarında yer alan hava kalitesi gazlarından kural çıkarılması probleminin çözümünde kullanılmıştır. Önerilen model kümeleme ve kural madenciliği aşamalarını içermektedir. Bu doğrultuda önerilen modelin değerlendirilmesi amacı ile hava kirliliğine etki eden gaz verileri kullanılmıştır. Bu kapsamda 3.1 bölümünde kullanılan veri seti ve özellikleri, 3.2 bölümünde literatür taraması, 3.3. bölümünde model mimarisi ve son olarak model başarımı ve değerlendirmeler sunulmuştur.

3.1. Kullanılan Veri Seti ve Nitelikleri

Önerilen modelin eğitim ve test süreçlerinde IoT verisi olarak CityPulse EU FP7 Projesi [132] kapsamında toplanan kirlilik verileri kullanılmıştır. Proje kapsamında zamansal olarak aşağıda belirtilen nitelikte veriler toplandığı ifade edilmiştir [133].

Çizelge 3. 1. Veri zaman aralıkları ve lokasyon bilgileri

Veri Tipi	Zaman	Lokasyon
Yol trafik verileri	2/2014 - 6/2014 8/2014 - 9/2014 10/2014 - 11/2014 07/2015 - 10/2015	Aarhus, Danimarka
Kirlilik verileri	8/2014 - 10/2014 2/2014 - 6/2014 8/2014 - 9/2014	Aarhus, Danimarka (Open Data Aarhus) Brasov, Romanya Brasov, Romanya
Hava verileri	2/2014 - 6/2014 8/2014 - 9/2014 2/2014 - 6/2014 8/2014 - 9/2014	Aarhus, Danimarka Aarhus, Danimarka Brasov, Romanya Brasov, Romanya
Kültürel olay verileri	5/2014 - 1/2015	Aarhus, Danimarka (Open Data Aarhus)
Sosyal olay verileri	6/2012 - 6/2014	Surrey, Birleşik Krallık
Kütüphane olay verileri	10/2013 - 6/2015	Aarhus, Danimarka
Park verileri	5/2014 - 11/2014	Danimarka

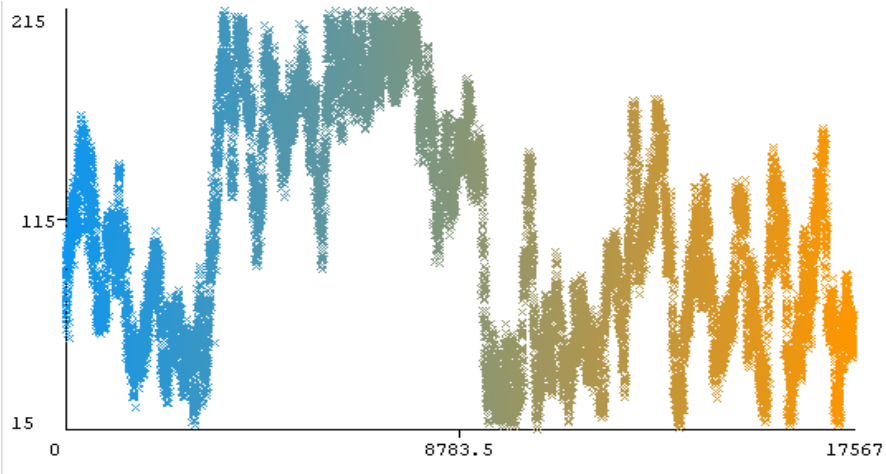
Bu çalışma kapsamında kirlilik verileri kapsamında toplanan ozon, partikül madde, karbon monoksit, kükürt dioksit ve nitrojen dioksit verileri kullanılmıştır.

3.1.1. Veri nitelikleri

Önerilen model kapsamında hava kirlilik verilerinden kural çıkarımı yapılmıştır. Hava kirlilik verileri hava kalitesi indeksi (Air Quality Index-AQI) olarak adlandırılan havanın sağlık şartları dikkate alınılarak hava kalitesini tanımlayan bir indekstir. Bu kapsamda aşağıdaki veriler önerilen modelin test ve eğitim süreçlerinde kullanılmıştır. Gazların nasıl oluştuğu ve insan sağlığına etkileri aşağıdaki verilmiştir [134].

- Ozon: Troposfer tabakasında rafineri, enerji santralleri veya kimyasal üretim yapan fabrikalar tarafından atmosfere yayılan gazlardan üretilmektedir. Akciğer rahatsızlıklarına neden olduğu belirtilmiştir.
- Partikül madde: Atmosferde asılı olarak bulunan katı partiküllerin ve sıvı damlacıkların karışımıdır. Bu asitler, organik kimyasallar, metaller vb. maddelerden oluşmaktadır. Özellikle kalp ve akciğer rahatsızlığı olan kişilerde rahatsızlıkları artırmakta ve solunum yolları problemlerini artırmaktadır.
- Karbon monoksit: Taşıtlar, endüstri tesisleri, odun yakılması ve orman yangınları sonucu ortaya çıkmaktadır. Dokulara oksijen taşınmasını engelleyerek ciddi solunum problemlerine sebep olmaktadır.
- Kükürt dioksit: Kömürün yakıt olarak kullanıldığı termik santraller, petrol rafineleri vb. tesislerde kullanılan katı ve sıvı yakıtlar sebebi ile veya orman yangınları, volkanik faaliyetler gibi doğal afetler sonucu oluşmaktadır. İnsanlarda burun ve solunum yolu enfeksiyonlarına sebep olmakla birlikte neden olduğu asit yağmuru sebebi ile çevreye zarar vermektedir.
- Nitrojen dioksit: Katı ve sıvı yakıtların yüksek sıcaklıkta yanması sonucu atmosfere yayılır. Kanın oksijen taşıma kapasitesini etkileyerek solunum yolu rahatsızlıklarına sebep olur.

Hava kirliliği gazları her 5 dakika bir sensörler tarafından üretilen okuma değeri ile birlikte lokasyon bilgisinden oluşturmaktadır. Aşağıdaki grafikte örnek olarak nitrojen dioksit verisinin zamansal olarak değişim grafiği verilmiştir.



Şekil 3. 1. Nitrojen dioksit verisi değişim grafiği

Önerilen model kapsamında kullanılan hava kirlilik verisi beş adet gazın tanımlayıcı istatistikleri aşağıdaki tabloda verilmiştir.

Çizelge 3. 2. Veri tanımlayıcı istatistikleri

	Ozon	Partikül Madde	Karbon monoksit	Sülfür dioksit	Nitrojen Dioksit
Min	15	15	15	15	15
Max	215	215	215	215	215
Ort	111.04	124.90	98.13	116.59	107.100
Standart Sapma	55.04	54.04	49.70	54.61	54.09

3.2. Literatür Taraması

CEP sistemlerinde genel eğilim kuralların manuel olarak tanımlanmasıdır [24,38,135]. Ancak IoT alanı veri boyutu ve niteliklerinin artması ile birlikte CEP kurallarının manuel olarak tanımlanması giderek zorlaşmaktadır [136]. Bu sebeple araştırmacıların ilgisi CEP kurallarının otomatik çıkarılmasına kaymıştır. Buna rağmen CEP kurallarının otomatik olarak çıkarılması erken bir aşamada olup literatürde fazla çalışma yer almamaktadır. Literatürde veri madenciliği ve optimizasyon teknikleri bu problemin çözümünde kullanılan tekniklerdir. Literatürde yer alan mevcut çalışmalar aşağıda özetlenerek sunulmuştur.

Veri madenciliği yöntemleri genellikle otomatik kural çıkarma problemlerinde kullanıldığı gözlemlenmiştir. Bunun sebebi akış verileri içerisinde yer alan örüntülerin veri madenciliği teknikleri ile çıkarılabilmesidir. Bu kapsamda Margara ve arkadaşları ICEP [131] ismini verdikleri çalışmalarında modüler bir otomatik kural çıkartma yaklaşımı önermişlerdir. Modüler yapı sayesinde kuralların farklı özellikleri incelenerek kullanıcılara eğitim sürecinde esneklik sağlamışlardır. Önerdikleri yaklaşım ile CEP kural çıkarımı problemini alt problemlere ayrıştırarak çözüm bulmaya çalışmışlardır. Olay kurallarının okunabilirliğini artırmak amacı ile Adhoc öğrenme algoritmasını kullanmışlardır. Yazarlar, önerdikleri modelin doğruluğunu hem sentetik hem de gerçek dünya veri kümelerinde kesinlik ve hassasiyet metrikleri ile değerlendirmişlerdir. Ancak çıkarılan kuralların statik olması ve alan uzmanları tarafından düzenlemesinin zor olduğu gözükmektedir.

Hasan ve arkadaşları [137] sık ve seyrek dizileri keşfetmek amacı ile CEP kurallarını otomatik olarak çıkaran bir model önermişlerdir. Model CEP kurallarının tespitinde örüntü madenciliği yaklaşımını kullanmaktadır. Olasılık tabanlı yöntem kullanarak olay örüntülerini genişletmişlerdir. Modeli tespit süresi, uzatma süresi ve mutlak hata açısından değerlendirmişlerdir. Bununla birlikte sık ve seyrek örüntülerin tespitinde hassasiyet ve kesinlik metrikleri ile değerlendirmişlerdir. Ancak önerilen modelin hedef örüntü sayısı arttığında kesinlik ve hassasiyet değerlerinin azaldığı gözükmektedir.

George [138] çalışmasında sıralı örüntü madenciliği kullanmıştır. Önerdiği model içerisinde geçmiş dönem verilerinden faydalanarak verilerin frekans ve meydana gelme sıklığı analizi yaparak örüntü çıkarımı yapmıştır.

Petersen ve arkadaşları [136] çalışmalarında K-means kümeleme yöntemi ve SVM sınıflandırıcı ile hibrit bir model önermişlerdir. Etiketsiz veriler kümeleme aracılığı ile etiketlenerek sınıf bilgisi oluşturulmuştur. Kurallar ise SVM sınıflandırıcı ile kendi geliştirdikleri algoritma ile oluşturulmuştur. Önerdikleri modelin geçerlemesini gerçek ortamda oluşan veri kullanarak hassasiyet ve kesinlik metrikleri ile değerlendirmişlerdir. Model, hem performans açısından hem de eksik veri olması durumunda başarılı sonuçlar verdiği ifade edilmiştir.

Örüntü madenciliği teknikleri kullanılarak yapılan çalışmalarda [136-138] sık geçen veri desenleri çıkarılmıştır. Ancak seyrek olan verilere ilişkin kuralların anomali olarak nitelendirilebileceği göz ardı edilmiştir.

Lee ve arkadaşları [139] çalışmalarında SCARG adını verdikleri hibrit bir model önermişlerdir. Çalışmalarında veri kümesi ilk olarak k- en yakın komşuluk yöntemi ile kümelememişlerdir. Benzerlik tabanlı kümeleme ile olay örüntülerine sıra ve benzerlik analizi yapmışlardır. CEP kurallarını çıkarmak amacı ile Markov modeli kullanmışlardır. Model etkinliğini değerlendirmek amacı ile borsa verileri kullanmışlardır. Yazarlar uyarlamalı CEP ve dinamik CEP modelleri açısından başarım sonuçlarını değerlendirmişlerdir. Yazarların önerdikleri modelde alan uzmanlarının müdahalesi gerekmektedir. Bununla birlikte, uzmanlar genellikle sensörler tarafından oluşturulan IoT verilerine aşina olmadığı için ekstra zorluklara neden olabileceği düşünülmektedir.

Mutschler ve arkadaşları [140] alan uzmanları tarafından belli bir olaya ilişkin olarak işaretlenen olaylara ilişkin kuralların otomatik olarak çıkarılmasına ilişkin yöntem önermişlerdir. Önerdikleri yöntem, uzmanlar tarafından belirlenen hedef olaylardan kurallar çıkartmaktadır. Hem sentetik hem de gerçek dünya veri kümelerinde düşük seviyeli olay verilerini eğitmek Noise Hidden Markov modeli kullanılmıştır. Önerdikleri yöntemi güven açısından değerlendirmişlerdir. Önerilen yöntem, verilerin sık örüntüleri barındırması durumunda veya önemsiz olayların ve belirsizliğin varlığında etkinliğini gösterdiğini gözükmektedir. Ancak önerilen yöntem için hedef olayları belirlemek için alan uzmanlarının müdahalesi gerekmektedir.

Mehdiyev ve arkadaşları [12] etiketli veri kullanarak kural madenciliği algoritmaları ile kural çıkarımı yapmışlardır. Önerdikleri model ile manuel olarak tanımlanması gereken kuralların otomatik olarak oluşturulmasını hedeflenmişlerdir. Önerilen modelde kural tabanlı sınıflandırıcıların başarımı kappa katsayısı metriği ile değerlendirilmiştir. Önerilen modelde kural madenciliği algoritmalarından PART algoritması diğer ONE-R, RIPPER, DTNB, RIDOR, NNGE algoritmalarla performans metrikleri ile karşılattırıldığında daha başarılı sonuçlar üretmektedir. Ancak IoT alanında üretilen verilerin geneli etiketsiz veri olması sebebi ile kullanım alanı sadece etiketli veriler üzerinde olmaktadır.

Mousheimish ve arkadaşları [35] CEP kural çıkarımı için veri madenciliği temelli shapelet öğrenme algoritması kullanarak bir model önermişlerdir. Shapelet öğrenme algoritmaları, gerçek dünya veri kümelerinde çok değişkenli zaman serilerinden örüntüleri çıkarmak için bir öğrenme algoritması olarak kullanılmaktadır. Zaman serilerinden çıkarılan shapelet serileri bir algoritma ile CEP kurallarına dönüştürülerek kurallar oluşturulmuştur. Yazarlar önerdikleri modeli f- ölçümü ve kendi belirledikleri earliness paremetresi ile değerlendirmişlerdir.

Frommgen ve arkadaşları [141] dağıtık CEP sistemlerinde FOSSA isimli bir model önermişlerdir. Model içerisinde genetik algoritma ve istatiksel yöntemlerden faydalanarak kuralları oluşturmayı hedeflemektedirler.

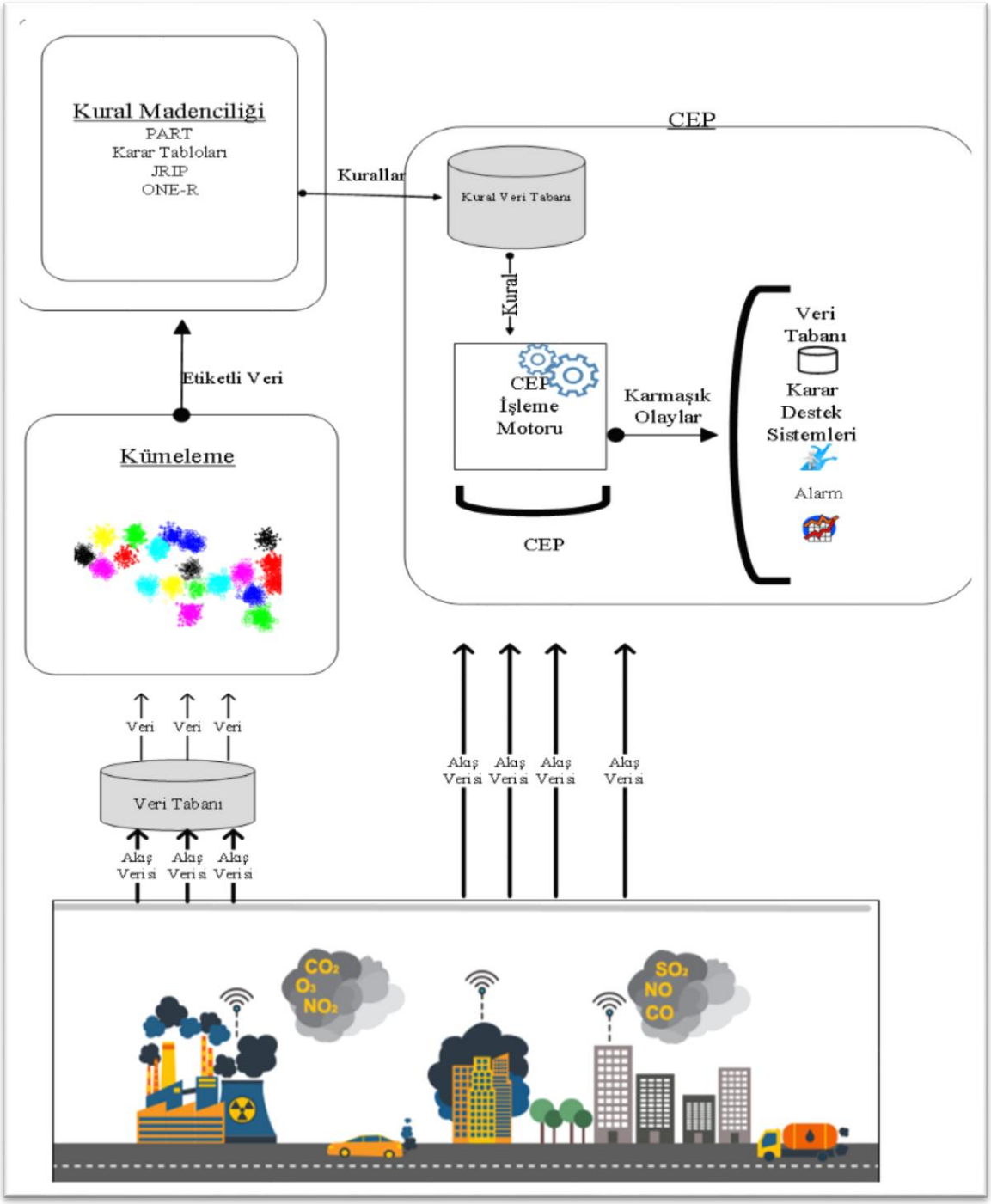
Bruns ve arkadaşları [49] GP4CEP isimli genetik algoritma kullanarak bir model önermişlerdir. Genetik algoritma ile alan uzmanları tarafından açıkça bilinmeyen veriler arasındaki ilgili karşılıklı bağımlılıkları ve ilişkileri dikkate alarak kural çıkarımını

hedeflemişlerdir. Genetik programlama, pencere boyutu, koşul durumları (Öznitelik Koşul Ağacı (ACT), Olay Koşul Ağacı (ECT)) ve bir eylem bölümü olan gelişmiş bir ağaç sözdizimi olarak formüle edilen CEP kurallarını çıkarmak için kullanılmıştır. ACT ve ECT ağaçları hem sentetik hem de gerçek veriler kullanılarak oluşturulmuştur. Yazarlar model başarımını hassasiyet ve kesinlik metrikleri ile değerlendirilmişlerdir. Başarım performansı açısından kuralların genetik algoritma modeli ile başarılı şekilde oluşturulduğu ifade edilmiştir.

3.3. IoT Verilerinden Genel Kuralların Çıkarılmasına Yönelik Önerilen Model Mimarisi

Bu bölümde IoT verilerinden CEP sistemlerinde kullanılan genel kuralların çıkarılmasını sağlamak amacı ile bir model önerilmiştir. Önerilen modelin genel görünümü Şekil 3.2' de sunulmuştur. Model, hava kalite indeksi belirlenmesinde kullanılan kirletici gazların kümeleme yöntemi ile etiketleme yaparak sonrasında kural madenciliği algoritmaları ile genel kurallarının çıkarılmasını sağlamaktadır.

Model içerisinde kullanılan veriler akıllı şehir, sensör, trafik veya hava durumu izleme sistemlerinde üretilen akış verilerini temsil etmektedir. Akış verileri hem hacmi yüksek hem de etiketsiz veri olması sebebi ile ilk olarak kümeleme işlemi gerçekleştirilmektedir. Kümeleme işlemi ile etiketlenen veriler sonrasında kural madenciliği algoritmaları ile eğitilerek otomatik kurallar çıkarılmaktadır. Kurallar CEP sistemi içerisinde yer alan kuralların sisteme entegre edildiği kural veritabanında saklanmaktadır. Önerilen sistem içerisinde kümeleme ve kural çıkarımı olmak üzere iki aşama yer almaktadır.



Şekil 3. 2. Önerilen model modeli

Akış verilerinin hepsi analiz için veri tabanında saklanmaktadır. Saklanan veri etiketlenmemiş bir veri olduğundan dolayı kural çıkarımı yapabilmek için verilerin birbirleri ile olan ilişkilerini görmek önem arz etmektedir. Verilerin oluşturdukları grupları görmek amacı ile ilk aşamada kümeleme algoritması kullanılarak veri etiketleme işlemi gerçekleştirilmiştir. Kümeleme işleminde başarımı etkileyen en önemli faktör küme sayısının belirlenmesidir. Bu kapsamda bölüm 2.3.1 de anlatılan Canopy kümeleme tekniği

kullanılmıştır. Küme sayısının belirlenmesinden sonra K-means kümeleme algoritması olarak kullanılarak veri k adet kümeye ayrılmıştır. İkinci aşamada ise kümelenen veri içerisinden kural çıkarımı yapılmıştır. Etiketlenmiş veri üzerinde PART, JRIP, Karar tabloları ve ONE-R algoritmaları uygulanarak kural çıkarımı yapılmıştır. Çıkarılan kurallar kural veri tabanına kayıt edilerek karmaşık olay işleme motorunda kullanılmak üzere hazır hale getirilmiştir.

Önerilen modelin işleyişi Çizelge 3.3'te verilmiştir.

Çizelge 3. 3. Önerilen model adımları

Önerilen CEP Genel Kuralların Akış İşlemleri
Girdi: Kirlilik Verileri (O_3, CO, PM, SO_2, NO_2) Çıktı: CEP Kuralları
Adım 1: Kapsama alanında yer alan sensörlerden kirlilik verilerini al ve veritabanına at Adım 2: Canopy kümeleme ile küme sayısı belirle Adım 3: K-means algoritması ile kümeleme yap ve verileri etiketle Adım 3: Etiketli verilerden kural madenciliği algoritmaları ile kural çıkarımı yap Adım 4: Kuralları CEP sistemi kural veritabanına yaz

3.4. Model Başarım Sonuçları ve Değerlendirilmesi

Bu bölümde önerilen modele ilişkin başarım sonuçları ve değerlendirmeler yer almaktadır.

3.4.1. Kümeleme sonuçları ve değerlendirme

Kümeleme işleminde küme sayısının belirlenmesi en önemli problemlerden birisidir. Küme sayısı model başarımını etkilemesi sebebi ile küme sayısı belirleme problemi önemli bir konudur. Önerilen modelde kümeleme işleminin gerçekleştirilmesi amacı ile küme sayısının belirlenmesi amacı ile Canopy kümeleme tekniği kullanılmıştır. Canopy kümeleme denetimsiz öğrenme tekniği ile çalışan bir ön kümeleme algoritmasıdır. Literatürde ifade edildiği üzere K-means algoritması ile birlikte kullanıldığında özellikle büyük veri setleri üzerinde kümeleme işlemini hızlandırdığı ifade edilmiştir [90].

Canopy kümele işleminde küme merkezi sayısı belirlenmektedir. Bu kapsamda belirlenen küme merkezine göre T2 yarıçapı uzaklığında bulunan noktalar bu kümenin elemanı sayılır. Bu oluşan küme canopy olarak adlandırılır. T1 uzaklığında yer alan noktalar ise yeni canopy merkezleri olarak ifade edilmektedir [142].

Önerilen modelin ilk aşamasında küme sayısını belirlemek amacı ile Canopy kümeleme tekniği kullanılmıştır. "k" değerini elde etmek için kullandığımız Canopy kümeleme algoritması sonuçları aşağıda verilmiştir.

Çizelge 3. 4. Canopy kümeleme algoritması sonuçları

Algoritma	Açıklama
Canopy Kümeleme	Canopy Sayısı (küme merkezi) bulunan: 8 T2 yarıçap: 0,818 T1 yarıçap: 1,023

"k" değeri 8 olarak elde edilmiştir. K-means algoritmasında girdi olarak küme sayısı bu değer olarak belirlenmiştir. Başka bir ifade ile veri seti içerisinde 8 adet küme bulunduğu ve buna göre etiketleme işleminin yapılacağı belirtilmiştir.

Önerilen modelin kümeleme aşamasında kullanılan K-means algoritması parametreleri aşağıdaki gibidir.

Çizelge 3. 5. K-Means algoritması parametreleri

Parametreler	Değerler
Hafızada tutulan maksimum canopy sayısı	100
Minimum canopy sayısı	2
Uzaklık fonksiyonu	Öklit uzaklığı
Maksimum iterasyon	500
Küme sayısı	8

Bu bölümde geliştirilen kümeleme algoritmasının performansı Hata Kare Toplamı (Sum of Squared Error - SSE) metriği ile değerlendirilmiştir. Bu metrik literatürde denetimsiz öğrenme yöntemlerinin değerlendirilmesinde yaygın olarak kullanıldığı için tercih edilmiştir. RMSE'nin matematiksel formülü Eş. 3.1 de sunulmuştur.

$$SSE= \sum_{i=1}^{n} |a_i - \grave{a_i}| \quad (3.1)$$

Önerilen modelin eğitim seti olarak her bir kirlilik gazına ait 17550 adet sensör okuma değeri kümeleme işlemine tabi tutulmuştur. Veri seti eğitim ve test verisi olarak %66 ve %34 olarak ayrılmıştır. Önerilen modelin kümeleme aşamasında kullanılan K-means algoritması hata metrik değeri Çizelge 3.6. 'da verilmiştir.

Çizelge 3. 6. K-means algoritması sonuçları

Algoritma	k	Hata kareleri toplamı
K-Means	8	2292.75

Etiketsiz IoT verisi K-means algoritması kullanılarak 8 adet kümeye ayrılmıştır. Bu kapsamda her bir kirlilik gazı verisi etiketlenerek modelin ikinci aşamasında ifade edilen kural çıkarım aşamasında kullanılmıştır. Etiketlenen veriden kuralların çıkarılması amacı ile kural madenciliği algoritmaları kullanılmıştır.

3.4.2. Kural madenciliği algoritmaları sonuçları ve değerlendirme

Veri madenciliği karar mekanizmaları için bilgi ve örüntü çıkarımı olarak kullanılmaktadır [103]. Kural tabanlı sınıflandırma algoritmaları denetimli öğrenmenin bir alt dalıdır. Kural tabanlı sınıflandırma kural örüntülerini çıkarmak amacı ile kullanılmaktadır [12]. PART, JRIP, Karar tabloları ve ONE-R algoritmaları bu çalışma kapsamında kural tabanlı sınıflandırma algoritmaları olarak kullanılmıştır. Kural madenciliği algoritmaları denetimli veri seti üzerinde literatürde yaygın olarak kural çıkarımı için kullanılmaktadır. Eğer – ise [143] şeklinde kurallar üretilmesi sebebi ile rahatlıkla yorumlanabilir ve CEP sistemlerine entegre edilebilir.

Kural madenciliği aşamasında veriler 10 katlı çapraz geçerleme kullanılarak yapılmıştır [144]. Kural çıkarma algoritmalarının performansını değerlendirmek için kesinlik,

hassasiyet ve f1 ölçümü değerleri kullanılmıştır. Bu metriklerin hesaplanmasında aşağıdaki parametreler dikkate alınmaktadır.

- Doğru Pozitif (True Positive / TP)
- Yanlış Pozitif (False Positive / FP)
- Doğru Negatif (True Negatif / TN)
- Yanlış Negatif (False Negative / FN)

Yukarıda bahsedilen parametreler dikkate alınarak kesinlik, hassasiyet ve f1- ölçümü hesaplamaları Eş. 3.2, Eş. 3.3 ve Eş. 3.4'daki gibi hesaplanmaktadır.

Kesinlik: $\frac{TP}{TP+FP}$ (3.2)

Hassasiyet: $\frac{TP}{TP+FN}$ (3.3)

F1- Ölçümü: $\frac{2*Kesinlik*Hassasiyet}{Kesinlik+Hassasiyet}$ (3.4)

Önerilen modelin kural çıkarımı aşamasında oluşan kesinlik, hassasiyet ve f-ölçümü değerleri çizelge 3.7'de verilmiştir.

Çizelge 3. 7.Kural çıkarımı başarım sonuçları

Algoritma	Kesinlik	Hassasiyet	F- Ölçümü
PART	0.978	0.978	0.978
JRIP	0.978	0.978	0.978
Karar Tabloları	0.832	0.816	0.820
ONE-R	0.343	0.402	0.35

Çizelge 3.7'de model içerisinde kullanılan PART, JRIP, Karar tabloları ve ONE-R algoritmaları sınıflandırma metrikleri sonuçları verilmiştir. Kesinlik, hassasiyet ve f- ölçümü açısından PART ve JRIP algoritması 0.978 olarak ölçülmüştür. Bu kapsamda bu iki algoritma küme sınıflarını temsil eden kuralları en iyi şekilde ifade ettiğini söyleyebiliriz. Buna karşın karar tabloları kesinlik, hassasiyet ve f- ölçümü değerleri sırası ile 0.832, 0.816 ve 0.820 olarak ölçülmüştür. ONE-R algoritması ise 0.343, 0.402 ve 0.35 ölçüm değerleri ile performansı diğerlerine göre aşağıda kalmıştır.

Performans metriklerine göre PART ve JRIP algoritmaları belirlenen kümelerden en iyi kuralları çıkardığı gözükmektedir. Karar tabloları bu iki algoritmaya göre nispeten daha az kötü sonuç verirken ONE-R algoritmasının kötü performans sergilediği gözükmektedir. ONE-R algoritmasının başarım oranının bu kadar düşük olmasının nedeni tüm niteliklere ait sınıf bilgisini tek bir nitelik ile tahmin etmeye çalışmasıdır.

Önerilen model içerisinde yer alan kural madenciliği algoritmalarının uygulanması sonucu çıkarılan kuralların sayısı Çizelge 3.8'de verilmiştir.

Çizelge 3. 8. Çıkarılan kural sayısı

Algoritma	Kural Sayısı
PART	102
JRip	76
Karar Tabloları	2771

Çıkarılan kurallar küme sınırlarını ifade ederek küme içerisinde kalan verileri temsil etmektedir. Kurallar kolaylıkla alanındaki uzmanlar tarafından değerlendirilebilmekte ve rahatlıkla karmaşık olay işleme sorgularına çevrilebilmektedir. Örnek olarak aşağıda çıkarılan kurallara örnek verilmiştir.

Çizelge 3. 9. Kural örnekleri

Kural Örnekleri
Sülfür_dioksit <= 56 AND Partikül_madde <= 133 AND Nitrojen_dioksit <= 142 AND Nitrojen_dioksit > 70 AND Karbon_monoksit <= 108: Sınıf 1
Ozon > 130 AND Nitrojen_dioksit > 167 AND Nitrojen_dioksit <= 196 AND Karbon_monoksit > 54 AND Karbon_monoksit <= 130 AND Sülfür_dioksit <= 127 : Sınıf 2

3.5. Bölüm Değerlendirmesi

IoT alanındaki gelişmeler ile birlikte günlük hayatımız birçok alanında sensörler veri üretmektedir. Üretilen veri heterojen ve gerçek zamanlı analiz edilmesi gereken niteliktedir. Bu kapsamda gerçek zamanlı olarak belirli olay ve örüntüleri yakalamak amacı ile CEP sistemleri kullanılmaktadır. CEP sistemlerinin ise temelinde kurallar yer almaktadır. Tanımlanan kurallara göre karmaşık olaylar yakalanmaktadır. Ancak verinin heterojen olması sebebi ile CEP alanında kuralları yazmak zorlu bir görevdir. Bu sebeple IoT verilerinden kuralların otomatik olarak çıkarılması oldukça önemlidir. Bu bağlamda önerilen model ile IoT verisinden genel kuralların çıkarılması amaçlanmıştır.

Önerilen model temel olarak iki aşama içermektedir. Bunlardan ilki kümeleme aşaması olup Canopy kümeleme ve K-means algoritması kümeleme için tercih edilmiştir. Canopy algoritması küme sayısını belirlemek amacı ile K-means algoritması ile birlikte kullanıldığında iyi sonuç vermesi sebebi ile tercih edilmiştir. K-means algoritması ise literatürde yaygın olarak kullanılmakla birlikte kümeleme alanında başarılı sonuçları olması sebebi ile tercih edilmiştir. K-means algoritması ile veri etiketlenerek ikinci aşamada kullanılmıştır. İkinci aşamada ise kuralların rahatlıkla hem uzmanlar tarafından yorumlanması hem de CEP sistemine kolaylıkla entegre edilebilmeleri amacı ile kural madenciliği algoritmaları kullanılmıştır. Bu sebeple denetimli öğrenmede kural çıkarımını en iyi şekilde yapan kural madenciliği algoritmaları tercih edilmiştir. Kural madenciliği algoritma sonuçları ise yaygın olarak kullanılan metrikler ile değerlendirilmiş ve sonuçların PART ve JRIP algoritması için %90 üzerinde başarım gösterdiği deney sonuçlarından anlaşılmaktadır. Önerilen model IoT zaman serisi içeren veri setindeki kümelere ilişkin genel kuralları çıkarmada başarılı sonuçlar ortaya koyacağını ortaya koymuştur.

Çalışmanın genel kuraları çıkarmakta başarımını kanıtladığı gözükmektedir. Ancak kümeleme aşamalarında belirlenen sınıf bilgilerinden uzman değerlendirmesi ile geçerlenmesi gerekebilir. Bu durum çalışmanın sınırlılığı olarak kabul edilerek kuralların geçerlenmesi için ayrı bir model ile genişletilebilir.

4. KARMAŞIK OLAY İŞLEME OLAY SORGULARININ DERİN ÖĞRENME YÖNTEMLERİ İLE BELİRLENMESİ

IoT sensör, mobil cihazlar ve ev ekipmanları gibi birbirine bağlı cihazların büyük miktarda ham veri ürettiği bir paradigmadır. Üretilen veri zaman nitelikli veri olup işlendiğinde üretici, tüketici ve hizmet sağlayıcılara değerli bilgiler sağlamaktadır [3,4]. Bu kapsamda, IoT verileri zaman duyarlı ve sürekli olması sebebi ile gerçek zamanlı analiz edilmesi önemlidir [145,146]. Gerçek zamanlı analiz işlemlerinde birbirleri ile ilişkili IoT verileri birlikte işlendiğinde daha değerli bilgiler elde edilecektir. Bu kapsamda IoT alanında gerçek zamanlı analiz imkanı sunan CEP sistemleri kullanılabilir.

CEP, verileri depolanmadan anında sorgulayan akış işleme teknolojisi sunarak IoT gerçek zamanlı veri analizi ile kaynağından değerli bilgileri çıkarmak için iyi bir adaydır. CEP 'in sosyal ağlar, trafik izleme sistemleri, ulaşım, sağlık yönetimi [147-149] vb. olmak üzere farklı uygulama alanlarına uygulanabilir olması rahatlıkla IoT alanında kullanılabileceğinin göstergesidir. CEP sistemleri veri akışlarını işlemek ve analiz etmek için filtreleme ve toplama yöntemlerini kullanır. CEP sisteminin temelinde atomik olayların karmaşık olaylara eşleştirilmesi yapılmaktadır [3,46]. Akış verileri CEP kurallarına (olay sorguları) göre analiz edilerek ilişkili karmaşık olaylar belirlenmektedir. CEP sistemi önceden belirlenen olay kurallarına göre veriyi işleyerek değerli bilgiler üretir. Bu kuralların tanımlanması genellikle alan uzmanları tarafından yani insan eli ile yapılmaktadır. Kuralların tanımlanması için veriler arası ilişkilerin bilinmesi önemli olacaktır [49]. Literatürde kural tanımlama yöntemleri alan uzmanları ve otomatik olarak yapıldığı ifade edilmiştir. Otomatik kural çıkarımı ise optimizasyon ve veri madenciliği teknikleri olarak ayrılmaktadır [8]. Özellikle alan uzmanların manuel olarak tanımladığı kurallar sebebi ile IoT alanında yer alan CEP uygulamalarının genişleme alanının daraldığı gözükmektedir. IoT alanında üretilen yüksek hacimli ve çok nitelikli verilere ait CEP kurallarının alan uzmanları tarafından oluşturulması neredeyse imkânsızdır. Örnek vermemiz gerekirse otuz nitelikli bir sensör verisine ait bir anomali kuralını yazmak alan uzmanları tarafından oldukça zor olacaktır. Bu sebeple IoT verilerinden otomatik olarak CEP kuralların çıkarılması problemi ortaya çıkmaktadır. Bununla birlikte IoT verisinin doğası ve hızla değişen koşulları sebebi ile kuralların belli aralıklarla güncellenme gerekliliği ortaya çıkmaktadır.

Tez çalışmasının bu bölümünde IoT verileri kullanılarak otomatik kural çıkarımını sağlayacak ARECEP isimli bir çerçeve model önerilmiştir. Önerilen çerçeve genelleştirilmiş bir model olup IoT verilerinden anomali olarak nitelendirilen kuralların çıkarılmasını sağlamaktadır.

4.1. İlişkili Çalışmalar

Bu bölüm kapsamında derin öğrenme yöntemleri ile anomali belirleme ve CEP sistemi kural çıkarma yaklaşımlarına ait literatür çalışmaları verilmiştir.

4.4.1. Derin öğrenme yöntemleri ile zaman serilerinde anomali belirleme

Anomali belirleme farklı uygulama alanlarında araştırılan önemli problemlerden birisidir. Yıllar boyunca birçok farklı teknik ve yaklaşımlar kullanılan geniş bir alandır [150]. Literatürde geniş bir çalışma alanına sahiptir. Çok geniş alanda çalışılmasına rağmen anomali belirleme algoritmaları ilgili alana özel ve uzmanların bilgisi dahilinde uygulama alanı bulur [21]. Makine öğrenmesi, veri madenciliği istatiksel yöntemler ve kural tabanlı programlama anomali belirleme işlemi için kullanılan popüler tekniklerdir [150,151]. Geleneksel makine öğrenme teknikleri derin öğrenme tekniklerine göre ön işleme gerektiren ve zaman açısından efektif olmayan tekniklerdir. Derin öğrenme teknikleri ise geleneksel yöntemlere göre daha efektif ve zaman açısından daha iyi performans göstermektedir [152-154]. Özellikle veri boyutu arttığında derin öğrenme teknikleri daha iyi performans sağlayarak daha fazla katman kullanarak gizli niteliklerin öğrenilmesini sağlamaktadır [153]. Literatürde derin öğrenme tekniklerinin kullanılmasının yaygınlaşması ile birlikte derin öğrenme teknikleri anomali belirleme yöntemleri için kullanılmaya başlanmıştır [151]. Derin öğrenme taksonomisinde anomali belirleme işlemi için yarı denetimli, denetimsiz, hibrit ve tek sınıflı yapay sinir ağları kullanılmaktadır [153]. Derin öğrenme ile anomali belirleme işlemi geleneksel makine öğrenmesi ile sınıflandırma yöntemleri ile karşılaştırıldığında daha iyi sonuç verdiği söylenmektedir [153,154]. Farklı derin öğrenme modelleri başarılı şekilde anomali belirleme işlemlerinde kullanılmaktadır. CNN, RNN ve LSTM yöntemleri sıralı veriler için CNN ve autoencoder yöntemleri ise sıralı olmayan verilerden anomali belirleme işlemi için kullanılmaktadır [152].

LSTM yaygın olarak zaman serilerinin tahmininde kullanılmaktadır. Bununla birlikte Malhotra ve arkadaşları [155] çalışmalarında zaman serilerinden anomali belirleme amacı

ile LSTM kullanmışlardır. LSTM ile zaman serilerinin tahmini yapılmış belirlenen yeniden çatılma hatası oranı ile birlikte performans sonuçları değerlendirilerek anomali belirlenmiştir. Yeniden çatılma hatası vektörü zaman serisinin gerçek ve tahmin edilen değerleri arası farktan oluşmaktadır.

LSTM saklama ünitesi sayesinde uzun zaman serilerinden örüntüleri öğrenme konusunda yeteneklidir. Bu kapsamda Malhotra ve arkadaşları bir diğer çalışmalarında [156] stacked LSTM kullanılmış olup anomali belirleme işlemi için gauss dağılımı kullanılarak zaman serilerinin anomali davranışları belirlenmiştir. Önerdikleri modelin değerlendirmesini ECG, Elektro kardiyo verileri, uzay mekiği, güç talebi ve çok sensörlü motor veri kümesi üzerinde test etmişlerdir.

LSTM doğrusal olmayan zaman serilerine ilişkin problemlerin çözümünde de kullanılmaktadır. Chauhan ve arkadaşları [157] çalışmasında LSTM modelini kullanarak sağlıklı bir tahmin modeli önermişlerdir. Önerdikleri sistem ile ECG sinyallerinden anomali belirleme işlemi yapmışlardır. Tahmin hatalarının olasılıksal dağılımı kullanılarak zaman serileri normal ve anomali olarak sınıflandırılmıştır.

Konvolüsyonel sinir ağları yaygın olarak video verilerinde kullanılmakla beraber kullandığı havuzlama ve filtreleme sistemi ile veriye ait özelliklerin çıkarılmasında kullanılmaktadır. Zhang ve arkadaşları [158] çalışmasında ise konvolüsyonel sinir ağları ve LSTM yapısı birlikte kullanılarak hibrit bir sistem ile anomali belirleme işlemi yapmışlardır. Çok değişkenli zaman serilerinin modellenmesinde kullanılan bir yöntem önermişlerdir.

Munir ve arkadaşları [159] çalışmasında ise derin öğrenme modellerinden CNN yöntemini kullanmışlardır. Önerdikleri modelde girdi olarak kullanılan zaman serisinden bir sonraki değer tahminini yapmışlardır. Tahmin edilen değer ile gerçek değer arasındaki öklit uzaklığı baz alınarak anomali belirleme işlemi yapmışlardır.

4.4.2. Derin öğrenme ile karmaşık olay işleme sistemlerinde kural çıkarma

CEP kural çıkarımı çalışmalarında derin öğrenme yöntemleri ile yapılan çalışmalara literatürde karşılaşılmamıştır. Bu kapsamda geleneksel veri madenciliği ve optimizasyon teknikleri ile yapılan çalışmalar bölüm 3.2'de sunulmuştur. DL yöntemleri CEP sistemleri

içerisinde atomik olayların belirlenmesi [160] aşamasında kamera verilerinin semantik anlam kazanmasını sağlayıp CEP mimarisi içerisinde kullanılmıştır. Bu kapsamda önerilen çerçeve CEP kural çıkarımı problemi için derin öğrenme yöntemleri kullanılarak yapılan ilk çalışma olacaktır.

4.2. Problem Tanımlanması

Bu bölüm içerisinde IoT 'de CEP kurallarının otomatik olarak çıkarılması probleminin çözümüne odaklanılmıştır.

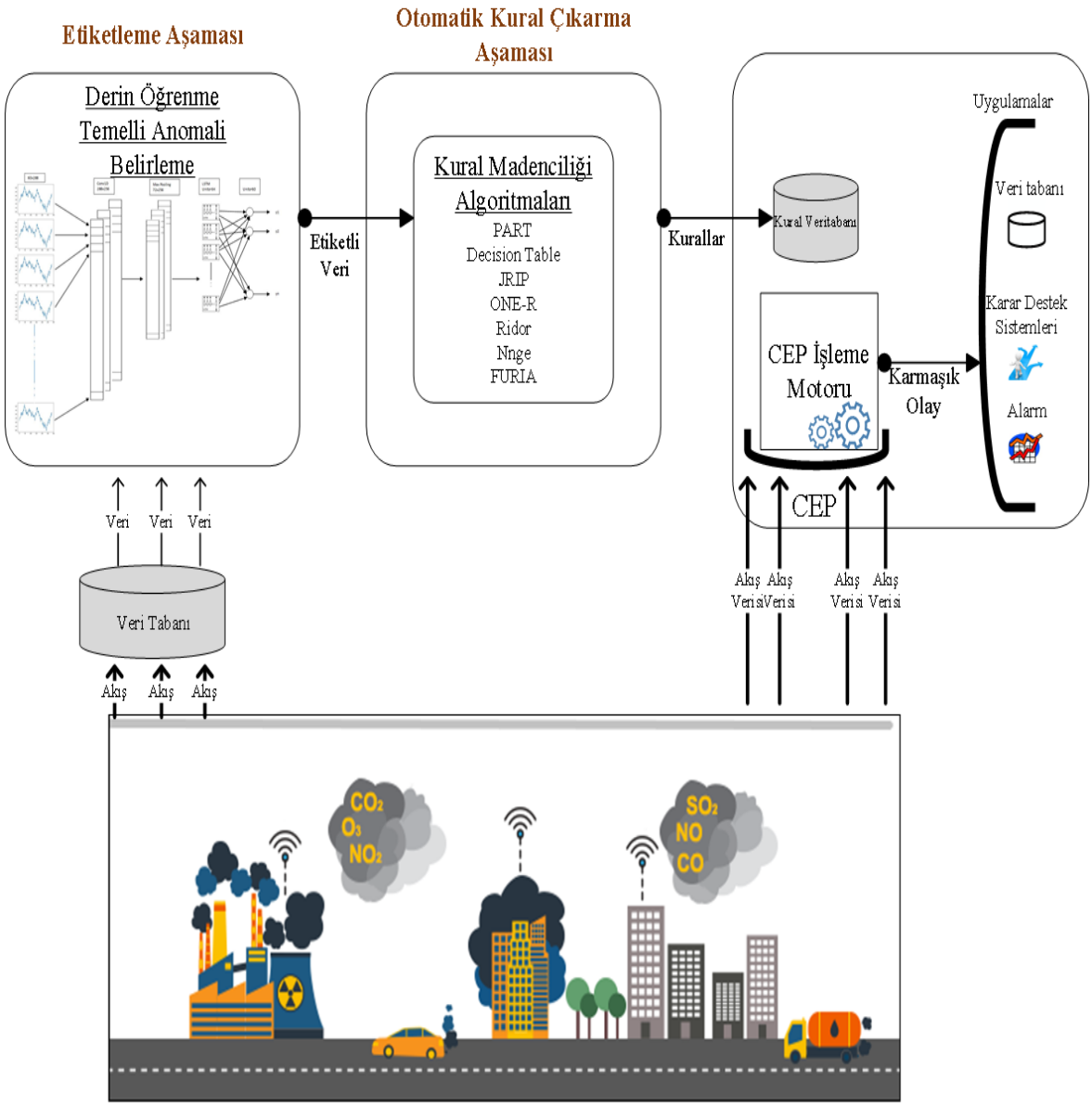
Literatürde IoT verilerinden CEP kuralları çıkaran birkaç çalışma bulunmaktadır. Mevcut çalışmalar genellikle etiketli verileri ele almaktadır. Ancak IoT doğası gereği genel olarak sensör verileri etiketsiz verilerden oluşmaktadır. Etiketsiz veri kullanan çalışmalar ise alana özel olup anomali durumları içermemektedir. Bununla birlikte mevcut çalışmalardan bir kısmında yeni kuralları oluşturmak için model başlangıç kural kümesi kullanmakta veya alan uzmanları tarafından hedef kurallar belirlenmektedir. Bu sebeple IoT alanında CEP kurallarının otomatik olarak tespitine yönelik mekanizmaların geliştirilmesi gerekmektedir.

Yukarıda belirtilen problem için ARECEP isimli IoT verilerinden CEP kurallarının çıkarılması amacı ile bir çerçeve model önerilmiştir. Önerilen çerçeve model, IoT ortamında CEP kurallarını çıkarmak için genelleştirilmiş bir çerçeve sunmaktadır. Çalışmamız, etiketlenmemiş IoT verilerinin DL yöntemleriyle işlenmesi ve ardından oluşturulan etiketli verilerden kurallar çıkarılması nedeniyle önceki çalışmalardan farklıdır. CEP kural çıkarma alanında DL algoritmalarının kullanılması ile birlikte bu alanda genişleme sağlayacağı düşünülmektedir.

4.3. Derin Öğrenme Temelli CEP Kural Çıkarım Çerçevesi

Bu alt bölümde tez kapsamında otomatik CEP kural çıkarımına ilişkin ARECEP isimli yeni bir çerçeve model önerilmiştir. Önerilen çerçeve model, gelen IoT akış verisini hem CEP sistemine hem de veritabanına kayıt olarak iletmektedir. Veritabanına kayıt edilen veriler analiz edilmek üzere saklanmaktadır. CEP sistemine giden akış verileri ise gerçek zamanlı olarak CEP kurallarına göre analiz edilmektedir. Önerilen çerçevenin etkinliği akıllı şehir senaryosu dikkate alınarak IoT alanında oluşturulan hava kirlilik verileri ile test edilmiştir.

Bu senaryoda, bir şehirde bulunan çeşitli IoT sensörleri zaman serisi olarak veri akışı üretir. Bu veriler, bize sensör verilerinin seviyesi hakkında ipuçları verdikleri için önemli olmaktadır. Özellikle sensör verilerinin belli eşik değerlerini aştığından acil önlemlerin alınması gerekir. CEP sistemi gerçek zamanlı olarak akış verilerini hızlı bir şekilde analiz ederek zaman serilerinden istenilen verileri çıkarmaktadır. Bu kapsamda IoT sensörlerinin miktarı arttıkça farklı sensör türleri artacak ve daha karmaşık veriler üretilecektir. Bu verilere ilişkin kuralların manuel olarak tanımlanmasını imkânsız hale getirecektir. Özellikle anlık değişiklikleri ve anomalilerin sıklıkla meydana geldiği akıllı şehir uygulamalarında kuralların manuel tanımlanması pratik olmayacaktır. Bu problemlerin üstesinden gelmek amacı ile alan uzmanlarının müdahalesi olmadan DL ve kural madenciliği algoritmaları kullanılarak ARECEP isimli otomatik CEP kuralları oluşturma çerçevesi önerilmiştir. Önerilen çerçevenin genel yapısı Şekil 4.1’ de verilmiştir.



Şekil 4. 1. Derin öğrenme temelli cep kural çıkarımı çerçevesi

ARECEP çerçevesi iki ana aşamadan oluşmaktadır. Bunlar etiketleme ve otomatik kural çıkarma aşamasıdır.

4.3.1. Etiketleme aşaması

Etiketleme aşamasında, akıllı bir şehirdeki sensörlerden gelen veriler hem CEP motoru hem de veri tabanına gönderilir. CEP motoru, önceden tanımlanmış kurallarla karmaşık olay verilerini tespit eder. CEP kuralları ile tespit edilemeyen veriler dikkate alınmaz ve kayıt edilmez. Tüm veriler, analiz edilmek ve kurallar çıkarılmak üzere tarihsel bir veritabanında

saklanmaktadır. Geçmiş veriler, verileri normal veya anormal olarak etiketlemek amacıyla DL tabanlı ve geleneksel regresyon yöntemleri ile eğitilmektedir. Daha spesifik olarak belirtmemiz gerekirse zaman serisi verilerindeki anormallikleri belirlemek için çift yönlü LSTM, AE, CNN, CNN-LSTM, GRU ve RNN olarak altı DL tabanlı yöntem ile RF regresyon ve KNN Regresyon algoritmaları etiketleme aşamasında kullanılmıştır.

4.3.2. Otomatik kural çıkarımı aşaması

Önerilen çerçeve modelin ikinci aşamasında ise DL yöntemi ile anomali olarak etiketlenen veriler CEP kurallarına dönüştürülmektedir. Bu aşamada anomali olarak ifade edilen zaman serileri

PART, DT, JRIP, ONE-R, RIDOR, NNGE ve FURIA algoritmaları olarak yedi kurala dayalı sınıflandırıcı kullanılarak CEP kuralları çıkarılmıştır. Çıkarılan kurallar anlaşılabilir olması ve SQL dizilimine sahip olması sebebi ile rahatlıkla CEP sistemlerine entegre edilebilmektedir. Çıkarılan kurallar CEP kural veritabanına kayıt edilerek CEP sisteminde kullanılabilir hale getirilmiştir.

4.3.3. Önerilen çerçevenin adımları ve karmaşıklık analizi

Bu bölümde önerilen çerçeve aşamaları ve karmaşıklık analizi verilmiştir. Önerilen çerçevenin akış işlemleri Çizelge 4.1' de verilmiştir.

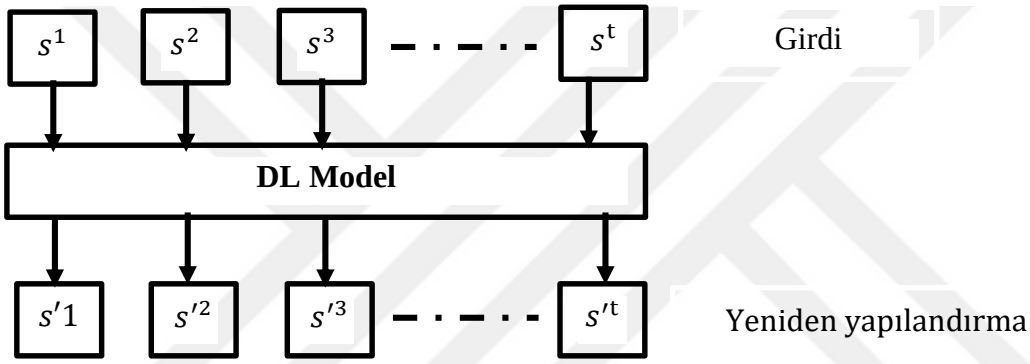
Çizelge 4. 1. ARECEP çerçevesi adımları

Önerilen CEP Genel Kuralların Akış İşlemleri
Girdi: Kirlilik Verileri ($O_3, CO, PM, SO_2, NO_2, time$) Çıktı: CEP Kuralları
Adım 1: Kapsama alanında yer alan sensörlerden kirlilik verilerini al ve veritabanına at Adım 2: Veri anomalilerini DL yöntemi ile belirle Adım 3: etiketli veriden kural madenciliği algoritmaları ile kuralları çıkart Adım 3: Etiketli verilerden kural madenciliği algoritmaları ile kural çıkarımı yap Adım 4: Kuralları CEP sistemi kural veri tabanına yaz

Model içerisinde kullanılan zaman serileri Eş. 4.1 deki ifade edildiği gibi formüle edilebilir. Eşitlikteki s değeri zaman serisini, t değeri ise bulunduğu zamanı ifade etmektedir.

S= $s^1, s^2, s^3, \ldots, s^t,$ (4.1)

s matrisi her bir sensör değerinin t zamanındaki okuma değerini içermektedir. s matrisi geçmiş veriler ile birlikte eğitilerek bir sonraki zaman dilimindeki tahmini yapılmaktadır. s'^t ise t+1 zamanındaki tahmin değerini içermektedir. Girdi olarak 1 den t zamanına kadar s matrisi s' tahmin matrisini oluşturmaktadır. Verinin Dl yöntemi ile yeniden tahminine ilişkin süreç Şekil 4.2' de verilmiştir.



Şekil 4. 2. DL yöntemleri ile verinin yeniden yapılandırılması

Yeniden çatılma hatası ise verinin gerçek değeri ve tahmin edilen değeri arasındaki fark olarak tanımlanmaktadır. Model içerisinde kullanılan yeniden çatılma hatası hesaplaması ve oluşturulan matris Eş. 4.1 ve Eş. 4.2 ifade edildiği gibi formüle edilmiştir. Eşitlikteki r değeri yeniden çatılma hatasını, R matrisi ise yeniden çatılma hata matrisini ifade etmektedir.

Yeniden Çatılma Hatası $r^t = |s^t - s'^t|$ (4.2)

R= $r^1, r^2, r^3, \ldots, r^t,$ (4.3)

Ortalama yeniden çatılma hatası sıralı zaman serisinin normal veya anomali olup olmadığını belirlemek için kullanılmıştır. Zaman serilerinin anomali olarak belirlenmesi amacı ile bir eşik değeri seçilmiştir. Denetimsiz öğrenmede anomali belirleme işleminde eşik değeri belirleme literatürde bir problem olup genel olarak benimsenen yaklaşım standart sapmanın üç katı olarak ele alınmasıdır. Eşik değeri Eş. 4.4'de verilmiştir. Eşitlikteki a değeri standart sapmayı ifade etmektedir.

σ= 3a, (4.4)

Önerilen çerçevenin etiketleme aşamasında kullandığı algoritma Şekil 4.2' te verilmiştir.

Algorithm 1: DL Yöntemi ile Anomali Belirleme

Input: Model(Bidirectional LSTM, AE, CNN-LSTM, CNN one dimentional, RNN, GRU) SensorVerisi (Ozon, Partikülmadde, Carbonmonoksit, Kükürtdioksit,Nitrojendioksit, Zamanbilgisi)
Output: Etiketli Sensor Verisi (S_{nl})

$s_{test} = test\ data\ of\ S_n$
$s_{train} = train\ data\ of\ S_n$
repeat
  $Model_x.fit()$
  $avgRE_x \leftarrow OrtalamaHata\ Hesapla(Model_x.predict(s_{train}))$
until *Tüm algoritmaları eğit*;
$model_s = En\ iyi\ hata\ oranına\ sahip\ algoritmayı\ seç\ (min\ avgRE_x)$
$p^t = model_s.tahmin(s_{test})$
$r^t = |p^t - s_n|$
$RE[t] \leftarrow r^t$
for $k \leftarrow 1\ to\ N$ **do**
  if *RE[t]>eşik$_d$eğeri* **then**
    etiketle $S_{nl} <-$ *anomali*;
  else
    etiketle $S_{nl} <-$ *normal*;
  end
end

Şekil 4. 3. DL yöntemleri ile anomali belirleme algoritması

Önerilen ARECEP çerçevesinin karmaşıklığı Çizelge 4.1'de ifade edilen adımlara göre aşağıdaki şekilde hesaplanmıştır.

Çizelge 4.1'in ilk adımında sensör değerlerinin veri kayaklarından okunması işlemi için yürütme karmaşıklığı O(1) olacaktır.

İkinci adımda ise etiketleme işlemi derin öğrenme yöntemleri ile Şekil 4.3 'de ifade edilen algoritmaya göre yapılmıştır. Bu algoritma içerisinde birbirinden farklı derin öğrenme yöntemleri karşılaştırmalı olarak değerlendirilmiştir. Bir sonraki aşama için ise hata oranı en düşük olan algoritma seçilerek devam edilmiştir. Bu aşamada kullanılan derin öğrenme de

regresyon algoritmalarının yürütme karmaşıklığı kullanılan parametreler göre değişmektedir. Bu aşamada en kötü yürütme karmaşıklığına sahip olan derin öğrenme algoritması yürütme karmaşıklığı O(n^2+m*n) olarak ifade edilmiştir [123]. n girdi örnek sayısı, m girdi neron sayısı veya gizli katman sayısı olarak ifade edilmektedir.

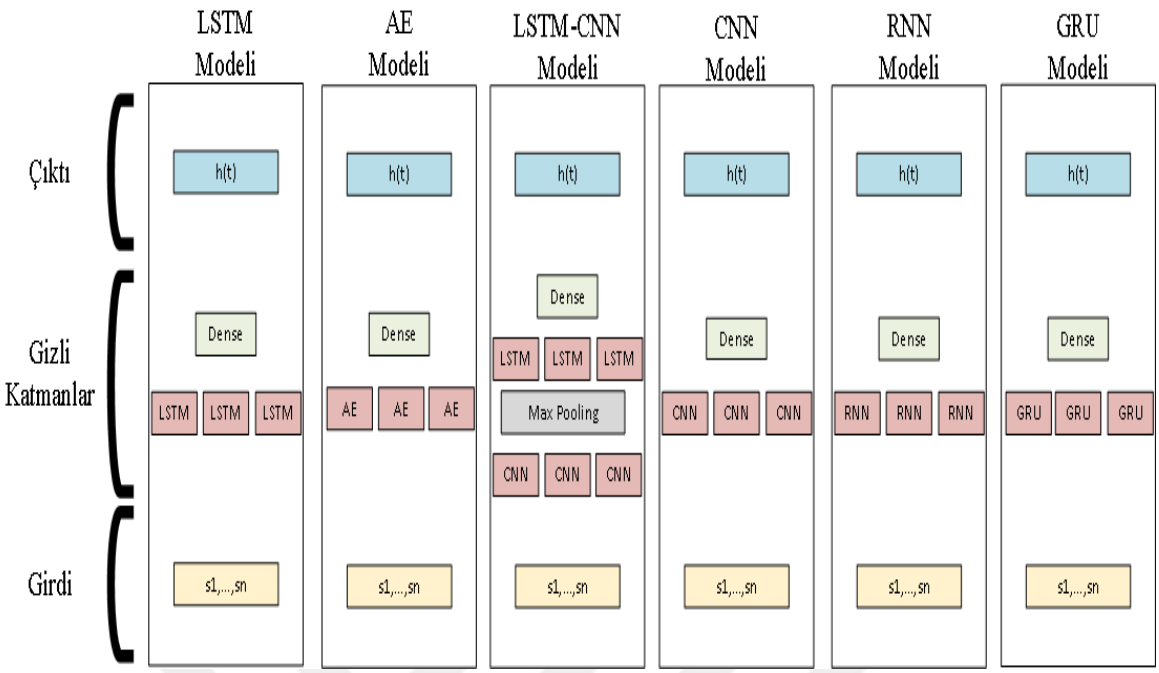
Üçüncü adımda ise farklı kural madenciliği algoritmaları kullanılmıştır. Her algoritmanın karmaşıklığı farklı olup bu aşamada en kötü yürütme performansına sahip kural madenciliği algoritması O(n^2) olarak ifade edilmiştir [161].

Son aşamada ise çıkarılan kuralların CEP kural veritabanına kayıt edilmesi işlemi gerçekleştirilmiştir. Bu aşamada çıkarılan kural sayısı kadar yürütme işlem karmaşıklığı hesaplanmıştır. Çıkarılan karmaşıklık O(r) olarak ifade edilebilir. r, kural sayısı olarak ifade edilmiştir.

Önerilen çerçeve modelin toplam karmaşıklığı ise dört adımda ifade edilen zaman karmaşıklığının toplamı olarak ifade edilebilir.

4.4. Önerilen Çerçevede Kullanılan Derin Öğrenme Yöntemleri ve Regresyon Yöntemleri

Bu bölüm çalışmalarında hava kirlilik veri setinde bulunan ozon, partikül madde, karbon monoksit, kükürt dioksit ve nitrojen dioksit değerlerine ilişkin altı DL modeli ve iki regresyon modeli geliştirilmiştir. Önerilen çerçeve kapsamında karşılaştırmalı olarak LSTM, AE, CNN-LSTM, CNN, RNN, GRU, RF regresyon ve Knn regresyon algoritmaları kullanılmıştır. ARECEP, sensör verilerini girdi alarak çoktan-çoğa (many-to-many) ağ yapısı kullanılarak tasarlanmıştır. Her bir model girdi olarak beş sensör verisini ele alarak her bir sensör değerinin bir sonraki zamana ait tahmini yapılmaktadır. Önerilen derin öğrenme modelleri katmanları aşağıdaki şekil 4.3'te verilmiştir.



Şekil 4. 4. Derin öğrenme modelleri yapısı

Modelin başarım performansını değerlendirmek amacı ile geleneksel yöntemlerde RF regresyon ve KNN regresyon algoritmaları kullanılmıştır.

Hava kirlilik veri seti

Bu bölüm kapsamında çerçevenin eğitim ve test aşamalarında CityPulse EU FP7 Projesi [132] kapsamında toplanan kirlilik verileri kullanılmıştır. Veri nitelikleri ve istatiksel özellikleri bölüm 3.1 kapsamında sunulmuştur. Sensör düğümlerinden her beş dakikada bir zaman bilgisi ve gaz seviyesi bilgisi elde edilmiştir. Her bir gaz değeri için 17550 adet veri kullanılmıştır.

4.4.1. Derin Öğrenme Modelleri Eğitimi ve Parametreleri

Bu bölüm kapsamında geliştirilen DL yöntemlerinde kullanılan parametreler verilmiştir. DL yöntemlerinin başarısı modelin eğitim aşamasından önce belirlenen parametreler ile ilişkilidir. Bu kapsamda kullanılan parametreler aşağıdaki gibi ifade edilebilir.

- Gizli katman sayısı

- Nöron/ünite sayısı
- Optimizasyon yöntemi
- Öğrenme katsayısı
- Düzenleme yöntemi
- Batch boyutu
- Epoch sayısı

Yukarıda ifade edilen parametreler haricinde kullanılan modelin özel yapısına bağlı olarak çeşitli parametreler kullanılmaktadır. CNN yapısında kullanılan havuzlama yöntemi ve konvolüsyon işleminde kullanılan matris yöntemi buna örnek verilebilir [162]. Bu çalışma kapsamında optimum parametreleri bulmak amacı ile aşağıdaki Çizelge 4.2'de ifade edilen oranlarda test edilmiştir.

Çizelge 4. 2. Önerilen DL modelleri test parametreleri

Parametre	Test Edilen Değerler
Katman Sayısı	1, 2, 3, 4, 5
Epoch	20, 50, 100, 200
Batch	4, 8, 16, 32, 64
CNN filtre sayısı	64, 128, 256
Havuzlama yöntemi	Max, min, ortalama
DL ünite sayısı	16, 32, 64, 128, 256

Yukarıda ifade edilen parametreler haricinde optimizasyon yöntemi olarak literatürde SGD, AdaGrad, AdaDelta, RMSProp, Adam, Adamax, Nadam ve AMSGrad yöntemleri kullanılmaktadır [163]. Önerilen çerçeve içerisinde kullanılan DL yöntemleri için adam yöntemi tercih edilmiştir. Adam yöntemi derin öğrenme teknikleri ile zaman serilerinin tahmininde yaygın olarak kullanılan bir tekniktir [164]. Bununla birlikte hesapla açısından verimli olması, daha az bellek gereksinimine ihtiyaç duyması ve sabit olmayan hedefler için uygun olması [163] sebebi ile tercih edilmiştir.

Gerçekleştirilen optimizasyon ile elde edilen optimum parametreler Çizelge 4.3'te verilmiştir.

Çizelge 4. 3. DL yöntemleri optimum parametreleri

Metot	Parametre
LSTM	Epoch: 100, Batch Sayısı: 8, Ünite sayısı:64
AE	Epoch:100, Batch Sayısı: 32, Test Sayısı: 0.2
CNN-LSTM	Epoch: 100, Batch Sayısı: 8, Ünite Sayısı: 128, Max havuzlama Filtre sayısı: 256
CNN	Epoch: 100, Batch Sayısı: 8, Max havuzlama, Filtre sayısı: 256
RNN	Epoch: 100, Batch Sayısı: 8, Ünite sayısı:128
GRU	Epoch: 100, Batch Sayısı: 8, Ünite sayısı:128

Bu çalışma kapsamında modellerin değerlendirilmesi amacı ile Intel i7-7700HQ işlemcili, 16 GB RAM ve 6 GB GTX 1060 GPU ekran kartına sahip laptop kullanılmıştır. DL ve regresyon modelleri için Tensorflow [165], Keras [166] ve Scikit-learn [167] kütüphaneleri kullanılmıştır.

4.5. Önerilen Çerçeve Başarım Sonuçları ve Değerlendirilmesi

Bu bölümde önerilen çerçeve modele ilişkin başarım sonuçları ve değerlendirmeler yer almaktadır.

4.5.1. Derin öğrenme modelleri sonuçları ve değerlendirmesi

Deneysel sonuçlarda, veri seti eğitim, doğrulama ve test setleri olarak sırasıyla % 70,% 10 ve% 20 olarak bölünmüştür. Model, eğitim ve doğrulama setleri ile eğitildikten sonra, kural çıkarma aşamasında kullanılan test seti üzerinden değerlendirilir. Etiketleme aşamasının performansı, yeniden yapılandırma hata kriterleri ile değerlendirilmiştir. Önerilen DL modellerinin tahmin performansı yeniden çatılma hatası [168] metriğine göre değerlendirilmiştir. Yeniden çatılma hatası matematiksel formülasyonu Eş. 4.5 ‘te verilmiştir.

$$RE = |a^t - a'^t| \tag{4.5}$$

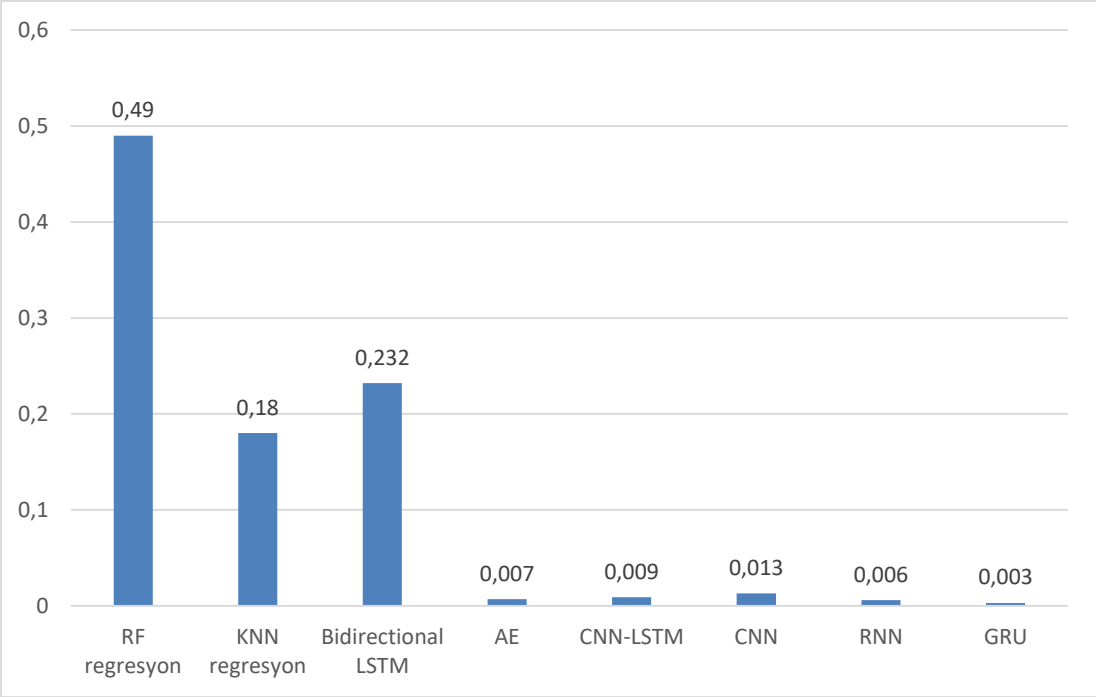
Eş. 4.5 ‘te RE yeniden çatılma hatasını, a^t gerçek değeri ve a'^t tahmin edilen değeri ifade etmektedir.

Model parametreleri optimizasyonu için çok sayıda deney yapılarak optimum parametreler elde edilmiştir. Optimum parametreler Çizelge 4.3. de sunulmuştur. Önerilen model içerisinde anomali belirleme işlemi için altı DL yöntemi ve iki regresyon yöntemi başarım sonuçları karşılaştırılmıştır. KNN regresyon ve RF regresyon yöntemleri geleneksel makine öğrenmesi algoritmaları olarak seçilmiş olup DL yöntemleri ile karşılaştırma amacı ile seçilmiştir. Değerlendirme metriği olarak seçilen RE değerine göre nicel karşılaştırmalar Çizelge 4.4’te verilmiştir.

Çerçeve içerisinde test edilen yöntemlerin başarım oranlarına ilişkin karşılaştırma Çizelge 4.4 ve Şekil 4.4’ te verilmiştir. Tablodan da anlaşılacağı üzere GRU yöntemi diğer DL yöntemleri ve regresyon yöntemleri ile karşılaştırıldığında daha iyi sonuç verdiği gözükmektedir. Çerçevenin etiketleme aşamasından sonra kural çıkarım aşaması için DL modellerinden en iyi model ve regresyon modellerinden en iyi model RE metrik sonuçlarına göre değerlendirilerek etiketleme yapılmıştır. Başka bir ifade ile kural çıkarım aşaması için karşılaştırmalı sonuçları görmek amacı ile GRU ve KNN regresyon algoritması seçilmiştir. Kural çıkarım aşaması için veriler etiketlenerek kural madenciliği algoritmaları kullanılmıştır.

Çizelge 4. 4. Tahmin sonuçları karşılaştırılması

Metot	Yeniden Çatılma Hatası
RF regresyon	0.490
KNN regresyon	0.180
Bidirectional LSTM	0.232
AE	0.007
CNN-LSTM	0.009
CNN	0.013
RNN	0.006
GRU	0.003



Şekil 4. 5. Tahmin sonuçları karşılaştırma grafiği

4.5.2. Kural madenciliği sonuçları ve değerlendirmesi

Önerilen modelin kural çıkarım aşaması için PART, DT, JRIP, ONE-R, RIDOR, NNGE ve FURIA algoritmaları kullanılmıştır. Her algoritma, bölüm 4.4. de etiketlenen hava kirlilik verisi üzerinde 10 kat çapraz geçerleme ile uygulanır. Kural çıkarma algoritmalarının performansını değerlendirmek için kesinlik, hassasiyet ve f1 ölçümü değerleri kullanılmıştır. Bu metriklerin hesaplanmasında aşağıdaki parametreler dikkate alınmaktadır.

Yukarıda bahsedilen parametreler dikkate alınarak kesinlik, hassasiyet ve f1- ölçümü hesaplamaları Eş. 3.2, Eş. 3.3 ve Eş. 3.4 ‘te verilmiştir.

Önerilen çerçevenin kural çıkartma aşamasında en iyi DL ve regresyon yöntemlerine ilişkin kural madenciliğine ilişkin sonuçlar Çizelge 4.5 ‘te ve Çizelge 4.6 ’da verilmiştir. Sensör sayısının artması değerlendirilerek başarım sonuçları değerlendirilmiştir. Her bir sensör değeri hava kirlilik gazlarını temsil etmektedir. Seçilen sensörlerin birbirleri ile korelasyonu dikkate alınarak iki, üç, dört ve beş hava kirlilik gazının başarımı değerlendirilmiştir.

Çizelge 4.5'te KNN regresyon modeli kullanılarak yapılan anomali etiketleme sonuçlarına göre yapılan kural madenciliği yöntemleri başarım sonuçları verilmiştir. Deneysel sonuçlar incelendiğinde, kural çıkarma için regresyon tabanlı yöntemlerin, CEP kural çıkarımında DL tabanlı yöntemlerle karşılaştırıldığında düşük performans gösterdiğini ortaya koymaktadır. Sonuçlara göre regresyon temelli yöntem ile alarm kurallarını tespit etmede %50 civarında performans sergilediği gözükmektedir. BU durum regresyon temelli olarak yapılan anomali etiketleme sonucu çıkarılan kuralların alarm durumlarını belirlemede yetersiz olduğunu göstermektedir. Bununla birlikte DT algoritması için herhangi bir kural çıkarılamadığı görülmüştür.

Çizelge 4.6'da GRU modeli kullanılarak yapılan anomali etiketleme sonuçlarına göre yapılan kural madenciliği yöntemleri başarım sonuçları verilmiştir. Deneysel sonuçlar incelendiğinde sensör sayısı arttıkça başarım oranının arttığı görülmektedir. Hava kirlilik verilerinden beş gazın birlikte değerlendirilmesi ile elde edilen sonuçlar incelendiğinde FURIA, JRIP ve NNGE algoritmalarının %95 üzerinde başarımı ölçülmüştür. PART ve RIDOR algoritmaları başarım performansı sırası ile %93 ve %94 olarak ölçülmüştür. DT algoritması başarım performansı yaklaşık olarak %90 ölçülmüştür. ONE-R algoritması ise diğer algoritmalara göre daha düşük başarım oranı performansı ölçülmüştür. ONE-R algoritmasının veri nitelikleri arasından tek bir nitelik ile tahmin sonuçlarını gerçekleştirmesi sebebi ile başarım oranı diğer algoritmalara göre daha düşüktür.

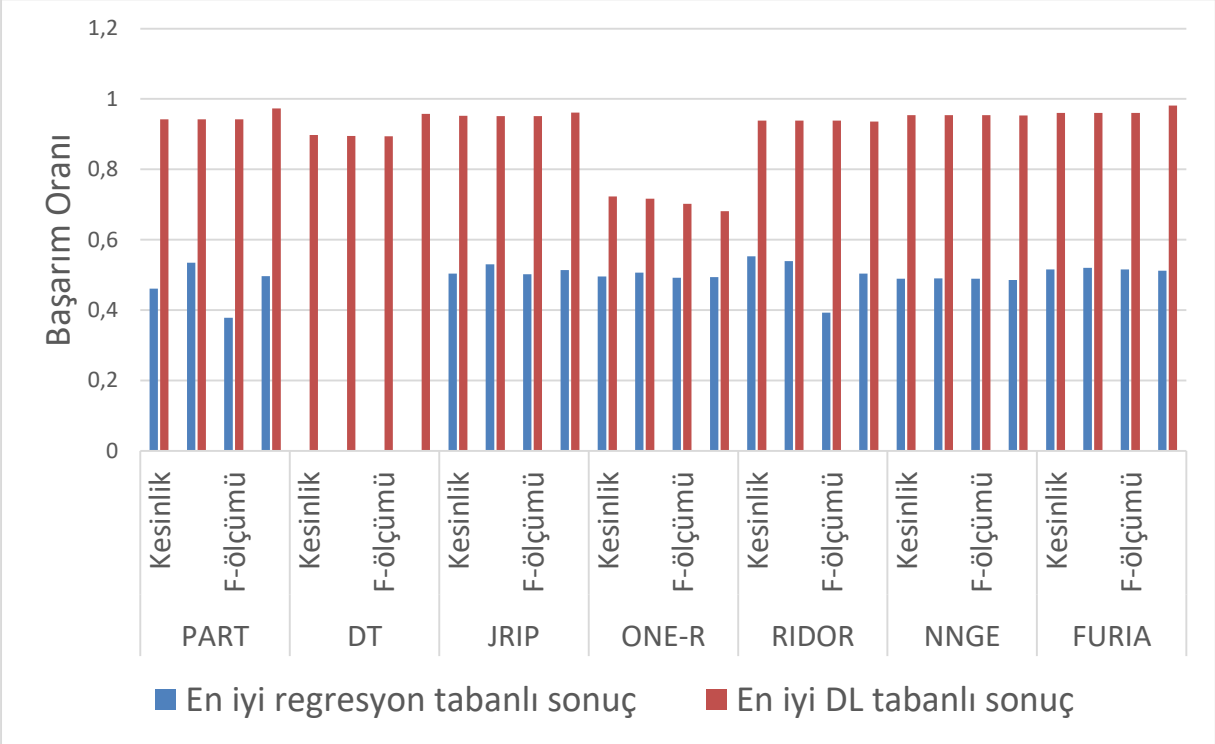
Çizelge 4.5. En iyi regresyon modeli kural çıkartma algoritmaları başarım sonuçları

Sensör Sayısı	2 sensör				3 sensör				4 sensör				5 sensör			
	Kesinlik	Hassasiyet	F-Ölçümü	ROC alanı	Kesinlik	Hassasiyet	F-Ölçümü	ROC alanı	Kesinlik	Hassasiyet	F-Ölçümü	ROC alanı	Kesinlik	Hassasiyet	F-Ölçümü	ROC alanı
PART	-	-	-	-	0,461	0,535	0,378	0,497	0,461	0,535	0,378	0,497	0,461	0,535	0,378	0,497
DT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
JRIP	0,518	0,532	0,487	0,514	0,515	0,527	0,498	0,515	0,517	0,528	0,504	0,515	0,504	0,53	0,502	0,514
ONE-R	0,506	0,517	0,499	0,503	0,498	0,51	0,492	0,496	0,496	0,507	0,492	0,494	0,496	0,507	0,492	0,494
RIDOR	0,541	0,538	0,396	0,504	0,491	0,533	0,391	0,499	0,519	0,536	0,409	0,503	0,553	0,539	0,393	0,504
NNGE	0,505	0,505	0,505	0,503	0,498	0,499	0,499	0,496	0,505	0,508	0,506	0,503	0,489	0,49	0,489	0,486
FURIA	0,499	0,491	0,488	0,496	0,5	0,498	0,499	0,496	0,519	0,522	0,52	0,516	0,516	0,52	0,516	0,512

Çizelge 4.6. En iyi DL modeli kural çıkartma algoritmaları başarım sonuçları

Sensör Sayısı	2 sensör				3 sensör				4 sensör				5 sensör			
	Kesinlik	Hassasiyet	F-Ölçümü	ROC Alanı	Kesinlik	Hassasiyet	F-Ölçümü	ROC Alanı	Kesinlik	Hassasiyet	F-Ölçümü	ROC Alanı	Kesinlik	Hassasiyet	F-Ölçümü	ROC Alanı
PART	0,811	0,806	0,800	0,880	0,910	0,910	0,910	0,968	0,915	0,915	0,915	0,973	0,942	0,942	0,942	0,973
DT	0,779	0,780	0,777	0,855	0,875	0,874	0,873	0,947	0,898	0,895	0,893	0,959	0,898	0,895	0,894	0,958
JRIP	0,817	0,817	0,815	0,825	0,916	0,916	0,916	0,932	0,944	0,944	0,944	0,948	0,952	0,951	0,951	0,961
ONE-R	0,698	0,701	0,698	0,684	0,723	0,717	0,702	0,681	0,723	0,717	0,702	0,681	0,723	0,717	0,702	0,716
RIDOR	0,798	0,798	0,795	0,782	0,907	0,904	0,904	0,907	0,930	0,930	0,930	0,930	0,939	0,939	0,939	0,936
NNGE	0,793	0,793	0,793	0,786	0,934	0,934	0,934	0,932	0,949	0,949	0,949	0,946	0,954	0,954	0,954	0,953
FURIA	0,808	0,805	0,801	0,817	0,924	0,924	0,924	0,948	0,949	0,949	0,949	0,973	0,960	0,960	0,960	0,981

Sonuç olarak, Çizelge 4.5 ve 4.6'da görüldüğü üzere sensör sayısındaki değişiklik dikkate alınarak analiz edildiğinde, sensör sayısı arttıkça DL tabanlı önerilen ARECEP çerçevemizin regresyon tabanlı yöntemlere göre daha doğru sonuçlar verdiğini ortaya koymaktadır. Şekil 4.5'te DL tabanlı ve regresyon tabanlı kural çıkarım modellerinin karşılaştırılması verilmiştir.



Şekil 4.6. DL ve regresyon tabanlı yöntemler kural madenciliği başarım oranı değişimi karşılaştırması

ONE-R algoritması dışında diğer kullanılan tüm kural madenciliği algoritmaları tüm veri niteliklerini içeren kurallar oluşturmuştur. ONE-R ve RIDOR algoritması anomali durumlara ilişkin sadece alarm kuralları oluştururken, PART, DT, JRIP, NNGE ve FURIA algoritmaları hem alarm hem de normal kuralları oluşturmaktadır. Özellikle parametre sayısı arttığında alan uzmanlarının manuel olarak kural tanımlaması zorlaşmaktadır. Bu sebeple nitelikleri fazla kuralların otomatik olarak yapılması önem arz etmektedir. Önerilen ARECEP çerçevesi sensör sayısı arttıkça başarımı artması açısından iyi performans sergilemiştir. Oluşturulan kural boyutları ve kuralların bir örneği Çizelge 4.7 'de verilmiştir.

Çizelge 4.7. Tüm sensör değerleri için oluşturulan kural sayısı ve örneği

Algoritma	Parametreler	Kural Sayısı	Örnek Kural	Kural Tipi
PART	Reduced error pruning: true Reduced error pruning: false	R1:31 R2: 23	Partikül_Madde <= 207 ve Partikül_Madde >= 111 ve Sülfür_Dioksit > 148 ve Ozon > 195 ve carbon_monoxide > 120: 1 (58.0)	Normal- Alarm
DT	Search:Best first Greedy Stepwise	R1:1517 R2:843	-	Normal- Alarm
JRIP	Use pruning: true Use pruning: false	R1: 38 R2: 55	(Nitrojen_Dioksit >= 165) ve (Nitrojen_Dioksit <= 190) ve (Partikül_Madde <= 148) and (Sülfür_Dioksit >= 63) ve (Ozon >= 207) and (Karbon_Monoksit >= 133)	Normal- Alarm
ONE-R	-	R1: 1	Ozon >200	Alarm
RIDOR	-	R1: 23	Except (Karbon_Monoksit > 113.5) ve (ozone >189.5) ve (Sülfür_Dioksit > 157.5) ve (Ozon < 202.5) ve (Partikül_Madde <= 136.5) ve (Nitrojen_Dioksit > 144.5)	Alarm
NNGE	-	R1: 959	IF : 69.0<=ozone<=162.0 ^142.0<= Partikül_Madde <=211.0 ^177.0<= Karbon_Monoksit <= 212.0 ^176.0<= Sülfür_Dioksit <=207.0 ^55.0<= Nitrojen_Dioksit <=85.0	Normal- Alarm
FURIA	-	R1: 81	-	Normal- Alarm

Çizelge 4.8'de çıkarılan kurallardan birisinin örneği verilmiştir.

Çizelge 4.8. Kural madenciliği yöntemleri ile çıkarılan kural örneği

Partikül_madde <= 207 AND Partikül_madde >= 111 AND Sülfür_dioksit > 148 AND Ozon> 195 AND Karbon_Monoksit > 120"

Hava kalite indeksi için Çevre Koruma Ajansı tarafından tanımlanan AQI verileri kullanılmaktadır [169]. Bu kapsamda Çizelge 4.9' da belirtilen seviyeler hava kirliliği hakkında bilgi vermektedir. Bu kapsamda Kural örneğini detaylı olarak incelersek, AQI indeks verileri ile benzerlik göstermektedir. Örneğin, çıkarılan kuraldaki partikül madde değeri 111 ile 207 arasında olup bu kategori AQI verilerine göre sağlıksız olarak ifade edilmiştir. Sonuç olarak, çıkarılan kurallar hava kirliliği gazlarının anomalileri hakkında ipucu vermektedir. Bu tür kurallar, IoT alandaki anomalileri tespit etmek için CEP motorunda kolayca kullanılabilmektedir.

Çizelge 4. 9. Hava kalite indeks verileri

AQI indeks seviyesi	Sağlık seviyesi	Renklendirme
0-50	İyi	Yeşil
51-100	Orta	Sarı
101-150	Hassas guruplar için riskli	Turuncu
151-500	Sağlıksız- tehlikeli	Kırmızı

4.6. Bölüm Değerlendirmesi

CEP sistemlerinde uzmanlar genellikle CEP kurallarını manuel olarak tanımlamaktadırlar. Bununla birlikte, kuralları manuel olarak tanımlamak, bilgi uzmanlığını gerektirmekte ve özellikle anlık değişikliklerin veya anormalliklerin sıklıkla meydana geldiği IoT alanında pratik olmayacaktır. Bu bağlamda, bu bölüm kapsamında IoT verilerinden CEP sistemleri için derin öğrenme yöntemi kullanarak ARECEP isimli genelleştirilmiş bir otomatik kural çıkarma çerçeve modeli önerilmiştir. Bu bağlamda çerçevenin uygulanabilirliğini göstermek amacı ile akıllı şehir uygulamalarında yer alan hava kirliliği veri seti kullanılmıştır.

Önerilen çerçevenin başarım performansını test etmek amacı ile altı DL yöntemi ve iki regresyon modeli seçilerek karşılaştırmalı sonuçlar verilmiştir. Zaman serilerinden anomali

belirleme işleminde DL yöntemlerinin geleneksel yöntemlere göre daha iyi sonuç verdiği ölçülmüştür. DL ve regresyon modelleri arasından en iyi sonuçlar seçilerek veri anomali olarak etiketlenmiştir. Yeniden çatılma hatası değerlendirilerek GRU ve KNN regresyon algoritmaları seçilerek çerçevenin ikinci aşamasında kural çıkarımı için kullanılmıştır. Çerçevenin ikinci aşamasında etiketlenen verilerden kural madenciliği algoritmalarından kurallar çıkarılmıştır. Bu bağlamda sensör sayısı dikkate alınarak yapılan değerlendirmelerde, DL tabanlı kural çıkartma modelinin sensör sayısı arttıkça kural çıkarma başarım oranının arttığı ölçülmüştür. Regresyon tabanlı kural çıkartma modeli ise sensör sayısının artmasının başarım performansını çok etkilemediği görülmüştür. DL tabanlı kural çıkartma modelinin başarım performansının regresyon tabanlı modele göre oldukça yüksek olduğu ölçülmüştür. Bu durumun sebebi zaman serilerinde tahmin performansında DL yöntemlerinin geleneksel yöntemlere göre daha iyi sonuç verdiği literatürde de ifade edilmiştir. Bunula birlikte DL yöntemlerinin tahmin performansında, geçmiş bilgileri çıkartarak tekrarlayan sinir ağları yapısı ile diğer hücrelere iletmesi sebebi ile regresyon yöntemlerine göre daha başarılı sonuçlar vermektedir. RF ve KNN regresyon yöntemlerinde ise genel olarak tahmin edilmek istenen veri kümesine yakın komşular seçilmektedir. Bu sebeple de anomali belirleme performansı bu kapsamda DL yöntemlerine göre daha düşüktür. Bu bağlamda deneysel sonuçlar dikkate alındığında ARECEP çerçevesinin IoT verilerinden %90 üzerinde başarım oranı ile kuralları çıkardığı ölçülmüştür.

ARECEP çerçevesinin IoT verileri için genelleştirilmiş bir çerçeve model olması sebebi ile literatüre katkı sağlayacaktır. Bununla birlikte derin öğrenme yöntemleri ve kural madenciliği algoritmaları kullanılarak kural çıkarma işlemini yapan bir model olması sebebi ile alandaki hibrit çözümlere katkı sağlayacağı düşünülmektedir. Önerilen çerçevenin gerçek zamanlı veri analizinde kullanılan CEP sisteminin IoT alanında da yaygınlaşmasına katkı sağlayacağı düşünülmektedir.

5. DEEPFOGCEP: SİS KATMANINDA DERİN ÖĞRENME TEMELLİ CEP TAHMİN SİSTEMİ

Günümüzde IoT uygulamalarının yaygınlaşması ile birlikte dinamik ve geleceğe dönük kararlar alabilmek amacı gerçek zamanlı olarak verilerin analizine ihtiyaç duyulmaktadır. Gerçek zamanlı olarak IoT verilerinin işlenmesi ile erken uyarı sistemleri gerçekleştirilebilmekledir [170]. Özellikle gelecekte öngörülemeyen olaylara karşı önlem almak veya geleceğe dönük kararlarda kullanılmak üzere erken alarm sistemleri kullanılmaktadır. Bu modellerin gerçekleştirilmesi için veri tahmini ve verinin gerçek zamanlı olarak işlenmesine imkân tanıyan sistemlere ihtiyaç duyulmaktadır. Bu kapsamda bu problemlerin çözümüne yönelik hibrit ve gelişmiş modellere ihtiyaç duyulmaktadır.

IoT alanında oluşturulan akış verilerinin analiz edilmesi kümeleme, sınıflandırma, anomali belirleme ve örüntü madenciliği teknikleri kullanılmaktadır [171]. Bu belirtilen işlemlerin hepsi IoT ağı içerisinde gerçekleştirildiği katmana göre enerji ve zaman verimliliği yönünden incelenmesi gerekir. Özellikle nesne ve servis sayısı arttığında ağ yükü önemli ölçüde artacaktır [172]. Bu bağlamda sis hesaplama yaklaşımları düşük gecikme, düşük işlem maliyeti, ölçeklenebilirlik ve esneklik [173] gibi sağladığı faydalar sebebi ile literatürde sıklıkla tercih edilmektedir.

CEP sistemi, belirli bir alan için ilgili veya kritik durumları tespit etmek amacıyla çok sayıda heterojen veri akışını işlememize ve analiz etmemize olanak tanır. Bu sayede olayları gerçek zamanlı olarak çıkarmak ve olayların sonuçlarını değerlendirmek mümkün hale gelir. Öte yandan, CEP sistemleri makine öğrenimi ve istatistiksel veri analizi yöntemlerinin tahmin gücünden yoksundur [174]. Bu nedenle, CEP sistemi tek başına geleceğe dönük olarak ve erken uyarı sistemi geliştirmede yetersizdir [175]. Bu sebeple verinin tahmin edilmesine yönelik ve gerçek zamanlı olarak izlenmesine yönelik hibrit protokollere ihtiyaç duyulmaktadır.

Bu bağlamda, bu bölüm kapsamında IoT verileri için DL ve geleneksel makine öğrenmesi yöntemlerini birleştirerek sis konseptine uygun olarak DeepFogCEP isimli bir CEP sistem protokolü önerilmiştir. Model ile erken uyarı amaçlı olarak DL ve geleneksel yöntemlerin tahmin yeteneği ile CEP sisteminin etkin veri ve olay işleme gücünü sentezlenmiştir. Bu

kapsamda DeepFogCEP sistemi ile sis hesaplama konseptine uygun olarak test edilmiştir. Sistemin test edilmesi ve örnek senaryo uygulanması amacı ile hava kirlilik verileri kullanılmıştır.

Hava kirlilik gazlarının tahmin sürecinin karmaşık olması hem de gerçek zamanlı olarak veri akışlarının işlenerek değerlendirilmesi karmaşık bir süreçtir. Hava kirliliği, insan sağlığı ve sürdürülebilir kalkınma üzerindeki olumsuz etkileri nedeniyle dünya çapında dikkat çeken ciddi bir çevre sorunudur [176]. Sanayileşmenin, kentleşmenin ve ekonominin hızla gelişmesiyle birlikte, birçok gelişmekte olan ülke ağır hava kirliliği problemi ile uğraşmaktadır [177]. Hava kirliliği gazlarının seviyelerinin artması durumunda hava kalitesi düşerken hava kirliliği dolayısı ile sağlık problemleri artmaktadır. Bununla birlikte uzun vadede ekonomi, sağlık, ekosistem ve iklim için tehditler oluşturmaktadır [178]. Bu sebeple hava kirlilik gazlarının tahmin edilmesi ve erken uyarı sistemleri ile izlenerek AQI seviyelerinde tutulması önem arz etmektedir. Bu sebeple DeepFogCEP sistemi bu problemin çözümünde kullanılmak üzere önerilmiştir.

5.1. Literatür Taraması

Son yıllarda DL araştırmacıların ilgili artmış ve birçok alana özgü uygulama geliştirilmiştir. Literatürde hava kalitesini DL ve istatistiksel yöntemlere dayalı olarak tahmin eden birçok çalışma olmasına rağmen, CEP tahmin modeli ile birlikte DL uygulamaları literatürde rastlanmamıştır. Bu nedenle, bu bölümde, geleneksel makine öğrenimi ve CEP sistemini birlikte inceleyen güncel makaleler özetlenmiştir.

Fülöp ve arkadaşları [179] tahmine dayalı analiz ile CEP sisteminin birleştirilmesinin etkinliğini göstermek için kavramsal bir model önermişlerdir. Model, tahmine dayalı analizler için BFtree, karar ağacı ve J48 algoritmalarını kullanmaktadır. Önerdikleri modelin performansını hassasiyet, kesinlik ve f-ölçüm ölçümleri açısından değerlendirmişlerdir.

Schwegmann ve arkadaşları [57] çalışmalarında tahmin modelleri ve CEP sistemini birleştiren bir çerçeve sunmuşlardır. Bu çerçeve içerisinde yer alan tahmin aşamasında regresyon ve sınıflandırma yöntemlerini kullanmışlardır. Model içerisinde yer alan tahmin aşamasını olay üreticisi olarak tanımlayıp belli metriklere göre tahmin yapılmaktadır.

Önerdikleri çerçevenin başarım performansını MSE ve sınıflandırma metriklerine göre değerlendirmişlerdir.

Cabanillas ve arkadaşları [56] önerdikleri çerçeve ile tahmin yöntemleri ile iş süreci yönetimi alanında kullanılan CEP modelini birleştirmişlerdir. Tahmin algoritması olarak SVM kullanmışlardır. CEP sisteminde Esper Kütüphanesi kullanarak hava uçuş verileri üzerinde test etmişlerdir.

Wang ve arkadaşları [180] Bayes ağları kullanarak geleceğe yönelik olayların tahminine yönelik bir metot önermişlerdir. Önerdikleri metot içerisinde olasılıksal olarak CEP bileşeni kullanmışlardır.

Wang ve arkadaşları [181] bir diğer çalışmaların CEP tahmin modülü kullanarak tahmin edilen değerleri karar destek sistemine gönderen bir yöntem geliştirmişlerdir. Gelecek olayların tahmini için bayes modeli kullanmışlardır. Önerilen yöntemin testi için karayolu trafik verileri kullanılmış olup pencere boyutunun arttıkça doğruluk değerlerinin arttığı görülmüştür.

Nechifor ve arkadaşları [182] tedarik zincirinde yer alan sıcaklık tahminine yönelik bir mimari önermişlerdir. Önerdikleri mimaride zaman serileri kullanarak CEP sisteminde çok katmanlı perceptron modeli geliştirmişlerdir. Ancak tahmin aşamasının değerlendirilmesi detaylı olarak ele almamışlardır.

Christ ve arkadaşları [183] geliştirdikleri mimari ile CEP sistemine tahmin analiz bileşeni eklemişlerdir. Bu çalışmada tahmin bileşeni için durumsal yoğunluk tahmini (Conditional Density Estimation) algoritması kullanmışlardır. Ancak çalışmada tahmin yönteminin başarısı detaylı olarak değerlendirilmemiştir.

Akbar ve arkadaşları [9] önerdikleri mimaride makine öğrenmesi teknikleri ile CEP sisteminin güçlü yanlarını birleştirmişlerdir. Önerdikleri adaptive moving window regresyon algoritması ile tahmin işlemini gerçekleştirmişlerdir. Önerdikleri mimari için trafik veri seti kullanmışlardır. Sonuçlar değerlendirildiğinde trafik veri seti üzerinde yüksek doğrulukta erken uyarı sisteminin başarı ile gerçekleştirildiği görülmüştür.

Emerson ve arkadaşları [184] çalışmalarında logistic regresyon ve naive bayes algoritmaları kullanarak CEP tahmin sistemleri için model geliştirmişlerdir. Modellerini doğruluk, kesinlik, hassasiyet ve f-ölçümü metrikleri ile değerlendirmişlerdir.

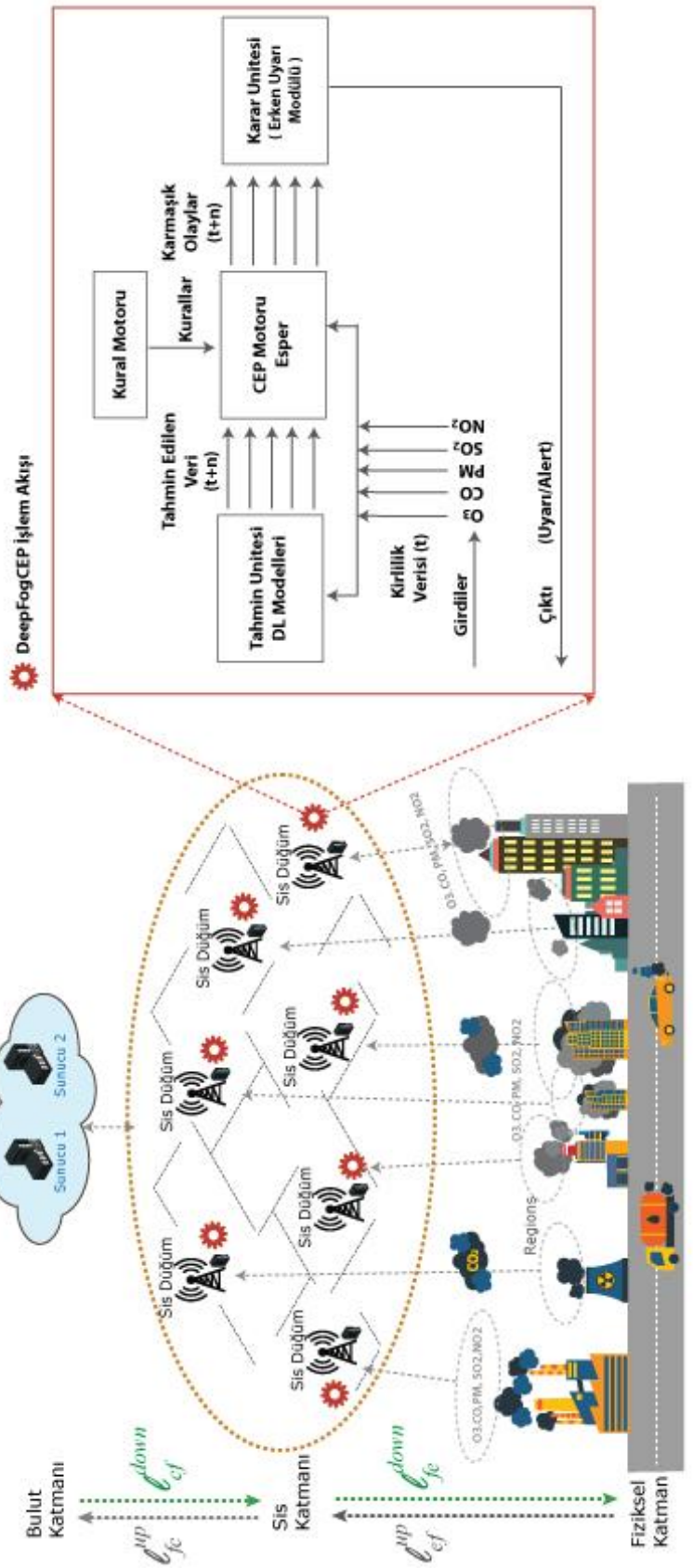
Xing ve arkadaşları [160] ise önerdikleri mimaride video verilerinde kullanmak üzere derin öğrenme modelleri kullanarak atomik olayları etiketlemişlerdir. Etiketlenen videoların semantik anlam kazanmasını sağlayarak CEP sistemi içerisinde kullanmışlardır.

Yadav ve arkadaşları [185] video kamera verileri üzerinde çalışan derin öğrenme teknikleri ile CEP sistemini birleştiren bir mimari önermişlerdir. Önerile mimaride trafik tahmini yapılmış olup gerçek zamanlı gecikmeler ve f-ölçümü dikkate alınarak mimari performansı değerlendirilmiştir. Önerdikleri model sayesinde DL yöntemleri ile CEP mimarisi entegrasyonunun çok boyutlu analizlerin yapılmasına imkan verildiği görülmüştür.

5.2. Akış Serilerinin Tahmini ile CEP Sistemlerinde Erken Uyarı Sistemi

Bu alt bölümde akıllı şehir uygulamaları içerisinde kullanılan CEP tahmin sistemini içeren bir protokol önerilmiştir. Önerilen sistem konsepti üç katmanlı IoT mimarisi ele alınarak sis hesaplama yaklaşımına göre tasarlanmıştır. Bu katmanlar fiziksel, sis ve bulut katmanı olarak Şekil 5.1 ‘de verilmiştir.

Fiziksel katmanda, hava kirlilik gazları olarak ifade edilen ozon, partikül madde, karbon monoksit, sülfür dioksit ve nitrojen dioksit gazlarına ait kapsadıkları alana ait konsantrasyon verilerin yakın sis sunucusuna gönderilmesi işlemleri yapılmaktadır. Sis katmanında ise DeepFogCEP sistemi sunucular üzerinde çalışmaktadır. Sis sunucuları gelen veriyi işleyerek DeepFogCEP sistemine gönderir. DeepFogCEP sistemi her bir hava kirlilik gazlarına ilişkin geleceğe dönük tahmin işlemini gerçekleştirir. Tahmin edilen değerler ve gerçek zamanlı olarak gelen hava kirlilik gazları verileri CEP motoruna gönderilerek karmaşık olaylar çıkarılır. Çıkan karmaşık olaylar değerlendirilerek alarm seviyelerine göre çıkarılan bilgi bulut sunucusuna iletilmektedir.



Şekil 5.1. DeepFogCEP sistemi

5.2.1. Sistem bileşenleri

Önerilen sistem mimarisi şekil 5.1'de verilmiştir. DeepFogCEP dört temel bileşenden oluşmaktadır. Bu bileşenler tahmin ünitesi, CEP motoru, kural motoru ve karar ünitesi olarak adlandırılmıştır. Aşağıda bu bileşenlerin açıklaması verilmiştir.

Tahmin ünitesi

Bu ünite gaz konsantrasyonlarının gelecek değerlerini derin öğrenme metotları ile tahmin etmektedir. Tahmin edilen gaz konsantrasyonları CEP motoruna gönderilir. Bu ünite içerisinde derin öğrenme metotlarından RNN, LSTM, CNN ve geleneksel veri madenciliği yöntemlerinden SVR test edilmiştir.

Kural motoru

Kural motoru, olayların tespitinde kullanılan kuralların alan uzmanlarının bilgilerine veya referans alınan standartlara göre belirlenmektedir. Bu kurallar CEP motoruna gönderilerek karmaşık olaylar yakalanmaktadır. Kural moturu içerisinde basit ve karmaşık kurallar oluşturulabilmektedir. Önerilen sistem kapsamında ele alınan hava kirlilik verileri sağlık seviyeleri için Hava Koruma Ajansı tarafından Çizelge 5.1'de belirtilen değerler dikkate alınmıştır [169]. Bu seviyeler hava kirlilik gazlarının kritik seviyeleri hakkında bilgi vermektedir.

Çizelge 5.1.Hava kalite indeks verileri

AQI indeks seviyesi	Sağlık seviyesi
0-50	İyi
51-100	Orta
101-150	Hassas guruplar için riskli
151-200	Sağlıksız
201-300	Çok sağlıksız
301-500	Tehlikeli

Hava kalite indeks verilerinde ifade etilen sağlık seviyelerine göre üç ana seviye belirlenmiştir. Bu seviyeler sırası ile 0-50, 51-100 ve 101-500 arası konsantrasyon verilerini içeren iyi, orta ve sağlıksız olarak Çizelge 5.2'de CEP kurallarına dönüştürülerek verilmiştir.

Çizelge 5.2. DeepFogCEP sisteminin kullandığı CEP kuralları

AQI indeks seviyesi	Sağlık seviyesi	Alarm Seviyesi	CEP kuralı
0-50	İyi	Normal	select * from AirQualityEvent match recognize (measures A as temp1"pattern (A) define A as A.pollutantGas _ 50
51-100	Orta	Uyarı	select * from AirQualityEvent match recognize (measures A as temp1"pattern (A) define A as A.pollutantGas>50 and A.pollutantGas _ 100)
101-500	Sağlıksız-tehlikeli	Kritik	select * from AirQualityEvent match recognize (measures A as temp1"pattern (A) define A as A.pollutantGas>100)

Oluşturulan CEP kuralları CEP motoru tarafından kullanılarak hava kalitesine ilişkin nomal, uyarı ve kritik durumlara ilişkin veriler elde edilmektedir.

CEP motoru

Basit, karmaşık ve bileşik olayları tespit etmek amacıyla kural motoru tarafından oluşturulan kuralları kullanarak büyük miktarda kirletici gaz verilerini işleyen ve analiz eden DeepFogCEP bileşenidir. Sonrasında kurallara göre yakalanan karmaşık olayları karar ünitesine gönderir. CEP motoru tasarımında ESPER [186] kütüphanesi kullanmıştır.

Karar ünitesi

Tüm karmaşık olayların değerlendirildiği ve erken uyarı sistemi olarak işlev gören DeepFogCEP bileşenidir. Alarm seviyelerine göre gelen karmaşık olayların anlaşılabilir formatta üçüncül kişilerle paylaşılma işlemi yapılır.

5.2.2. Önerilen sistem akış algoritması

Önerilen DeepFogCEP sistemi akış adımları aşağıdaki algoritmada verilmiştir.

Çizelge 5.3. DeepFogCEP sistemi akış işlemleri

Önerilen DeepFogCEP Sistemi Akış İşlemleri
Girdi: Kirlilik Verileri (O_3, CO, PM, SO_2, NO_2), Modeller(RNN, LSTM, CNN, SVR), Kurallar, AQI Seviyeleri Çıktı: Alarmlar
Adım 1: Sensörler aracılığı ile belli bir alana ait hava kirlilik verilerine ait gaz konsantrasyon bilgilerini topla Adım 2: Gazların gelecek konsantrasyonlarını tahmin et ($p_t = Model_x.predict\ (O_3, CO, PM, SO_2, NO_2)$) ve CEP motoruna gönder Adım 3: Kural motorundan kuralları al Adım 3: Kurallara göre basit ve karmaşık olayları tespit et Adım 4: Geleceğe ilişkin alarm sevilerini belirle ve üçüncü parti uygulamalara bildir ($Alarm_{seviyesi} = KararUnitesi(KarmaşıkOlay, AQI)$)

5.2.3. Ağ modeli ve formülasyonu

DeepFogCEP sistemi (Şekil 5.1) hava kirlilik gazlarının konsantrasyonunu ölçen fiziksel cihazlar, sis sunucuları ve bulut sunucusu yapısından oluşan sis tabanlı IoT ağını benimsemektedir. Benimsenen ağ modelinde sensörlerin farklı coğrafyada belli bir bölgeye ait hava kirlilik verilerinin ölçümünün yapıldığı varsayılmıştır. Çizelge 5.4'te network modelinde kullanılan parametreler verilmiştir.

Çizelge 5.4. Ağ modelinde kullanılan semboller

Sembol	Açıklama
P	Sensör seti
R	Bölge seti
ℓ_{ef}^{w}	Sensörler ile sis cihazı arasındaki iletişim linki olarak tanımlanmıştır. *w* iletişimin yönünü belirten parametredir. Up yukarı yönü, down ise aşağı yönü ifade eder.
ℓ_{fc}^{w}	Sis cihazı ile bulut cihazları arasındaki iletişim linki olarak tanımlanmıştır. w iletişimin yönünü belirten parametredir. Up yukarı yönü, down ise aşağı yönü ifade eder.
e	Fiziksel cihaz
f	Sis cihazı
D_r^{down}	Sis ve bulut cihazlarından fiziksel cihazlara iletilen veri
D_r^{up}	Sensörler tarafından üretilen çıkış veri seti
B^{up}	Toplam uplink/çıkış hattı bant genişliği
B^{down}	Toplam downlink/iniş hattı bant genişliği
f^{cc}	Sis sunucusu işleme kapasitesi
$T_{DeepFogCEP}^{proc}$	Sistem işleme gecikmesi
T_{pu}^{proc}	Tahmin işleme gecikmesi
T_{cu}^{proc}	CEP motoru işleme gecikmesi
T_{du}^{proc}	Karar modülü işleme gecikmesi
$T_{total}^{network}$	n paket için toplam ağ gecikmesi

Yukarıdaki notasyonlar baz alınarak, DeepFogCEP 'in ağ modelindeki performansını değerlendirmek için hesaplama gecikmesini ve iletim gecikmelerini aşağıdaki gibi formülize edilmiştir.

$$P = \{ p_1,\ p_2,\ p_3,\ p_4, \ldots\ldots\ldots\ldots, p_n\} \quad (5.1)$$

$$R = \{ r_1,\ r_2,\ r_3,\ r_4, \ldots\ldots\ldots\ldots, r_n\} \quad (5.2)$$

Katmanlar arası iletişim gecikmesi Eş. 5.3 ve 5.4 gibi ifade edilebilir.

$$T_{cf/fc}^{up} = \frac{D_r^{up}}{B^{up}} \quad (5.3)$$

$$T_{cf/fc}^{down} = \frac{D_r^{down}}{B^{down}} \quad (5.4)$$

Her bir paket için aşağı ve yukarı yönlü uçtan uca (end to end) iletişim gecikmesi ise Eş 5.5'deki gibi ifade edilebilir.

$$T_{e2e}^{trans} = T_{c,f}^{up} + T_{f,c}^{up} + T_{c,f}^{down} + T_{f,c}^{down} \quad (5.5)$$

DeepFogCEP sistemi tahmin ünitesi gecikmesi Eş. 5.6'daki gibi ifade edilebilir.

$$T_{pu}^{pred} = \frac{D_r^{up}}{f^{cc}} \quad (5.6)$$

DeepFogCEP sistemi toplam gecikmesi her bir ünitenin işlem gecikmesi toplam hesabından oluşmaktadır. Bu hesaplama Eş. 5.7'de verilmiştir.

$$T_{DeepFogCEP}^{proc} = T_{pu}^{proc} + T_{cu}^{proc} + T_{du}^{proc} \quad (5.7)$$

n adet paket için her bir bölge için toplam ağ gecikmesi Eş. 5.8'de verilmiştir.

$$T_{total}^{network} = \sum_{n=1}^{N} (T_{DeepFogCEP}^{proc} + T_{e2e}^{trans}) \quad (5.8)$$

5.2.4. Önerilen tahmin modelleri ve parametreleri

Bu bölüm kapsamında tahmin ünitesi için RNN, LSTM, CNN ve SVR algoritmaları kullanılmıştır. Tüm modellerin ağ başarım oranını artırmak amacı ile birçok deney yapılmıştır. Her bir gaza ait sensör verisi ayrı olarak ele alınıp bir sonraki değeri tahmin edilmek üzere model tasarlanmıştır. Bu kapsamda DL modelleri hava kirlilik gazlarını girdi olarak alarak çoktan – teke (many to one) ağ yapısı kullanılarak tasarlanmıştır. Model sekiz sensör değerini girdi olarak ele alıp bir sonraki sensör değerini tahmin etmektedir. Bu

kapsamda optimum parametreleri bulmak amacı ile RNN ve LSTM modelleri 1,2 ve 3 katmanlı olarak, ünite sayısı olarak 128, 256 ve 512 değerleri test edilmiştir. CNN modeli için 128, 256 ve 512 filtre sayısı test edilmiştir. DL modellerinin tahmin doğruluğunu iyileştirme amacı ile Bırakma (Dropout) düzenleme tekniğini kullanılmıştır. DL modelleri epoch sayısı 50, 100, 200, 300, 400 ve 500 olarak test edilmiştir. Modellerin optimum parametrelerini bulmak amacı ile grid arama tekniği kullanılmıştır. Bu kapsamda elde edilen optimum parametreler Çizelge 5.5'te verilmiştir.

Çizelge 5.5. DL yöntemleri optimum parametreleri

Metot	Parametre
CNN	Epoch: 100, Max havuzlama, Filtre sayısı: 256
LSTM	Epoch: 100, Batch Sayısı: 8, Gizli katman sayısı: 1, Ünite sayısı:128
RNN	Epoch: 100, Batch Sayısı: 6, Gizli katman sayısı: 1, Ünite sayısı:128
SVR	C=1.0, Epsilon=0.2

DeepFogCEP modelinin tahmin ünitesinde yer alan CNN, LSTM, RNN ve SVR modellerinin gerçekleştirilmesi amacı ile gerçekleştirilmesi amacı ile Tensorflow [165], Keras [166] ve Scikit-learn [167] kütüphaneleri kullanılmıştır. CEP sisteminin gerçekleştirilmesi için Eclipse platformu ile ESPER [186] kütüphanesi kullanılarak java dilinde yazılmıştır. Modellerin değerlendirilmesi amacı ile Intel i7-7700HQ işlemcili, 16 GB RAM ve 6 GB GTX 1060 GPU ekran kartına sahip dizüstü bilgisayarı kullanılmıştır.

Hava kirlilik veri seti

DeepFogCEP sistemi eğitim ve test aşamalarında CityPulse EU FP7 Projesi [132] kapsamında toplanan kirlilik verileri kullanılmıştır. Veri nitelikleri ve istatiksel özellikleri bölüm 3.1 kapsamında sunulmuştur.

5.3. Başarım Sonuçları ve Değerlendirilmesi

Bu bölüm kapsamında önerilen sistem çıktıları değerlendirilmiştir.

5.3.1. DL modelleri sonuçları ve değerlendirme

Bu bölüm altında önerilen DL ve geleneksel yöntemlerin performansı RMSE ve MAE metrikleri ile değerlendirilmiştir. RMSE ve MAE metrikleri genel olarak tahmin doğruluğu değerlendirmek için kullanılmaktadır [187]. Bu metriklere ait matematiksel formülasyonları Eş. 5.9 ve Eş. 5.10 da verilmiştir.

$$\text{RMSE (y,x)} = \sqrt{\frac{1}{n}\sum_{n=1}^{N}(y-x)^2} \tag{5.9}$$

$$\text{MAE (y,x)} = \frac{1}{n}\sum_{n=1}^{N}|(y-x)| \tag{5.10}$$

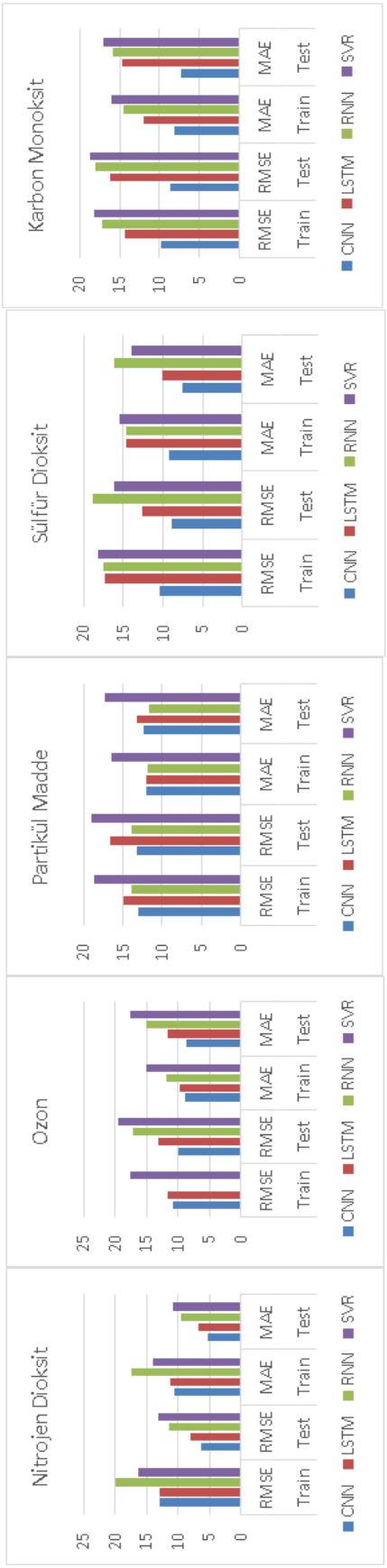
Eşitliklerde x girdi zaman serisi, y tahmin edilen zaman serisi ve N örneklem sayısını belirtmektedir.

Tahmin ünitesinde CNN, LSTM, RNN DL algoritmaları geleneksel yöntemlerden SVR ile karşılaştırılmıştır. Tüm eğitim modellerinde 10 katlı çapraz doğrulama kullanılmıştır. Kullanılan her bir modele ilişkin değerlendirme metrikleri sonuçları Çizelge 5.6'da ve Şekil 5.3'te verilmiştir.

Çizelge 5.6 incelendiğinde CNN modelinin zaman serilerini tahminde hem eğitim hem de test veri setlerinde karşılaştırılan diğer modellerden daha az hata RMSE ve MAE değeri ürettiği görülmektedir. Tüm eğitim modellerinde SVR tahmini hata değeri DL modelleri ile karşılaştırıldığında daha yüksek RMSE ve MAE değeri ürettiği ölçülmüştür.

Çizelge 5.6. Tahmin modelleri performans karşılaştırması

	Nitrojen Dioksit			Ozon				Partikül Madde				Sülfür Dioksit				Karbon Monoksit			
	Train RMSE	Test RMSE	Train MAE	Test MAE	Train RMSE	Test RMSE	Train MAE	Test MAE	Train RMSE	Test RMSE	Train MAE	Test MAE	Train RMSE	Test RMSE	Train MAE	Test MAE	Train RMSE	Test RMSE	Train MAE	Test MAE
CNN	12,86	6,40	10,50	5,24	10,67	9,86	8,83	8,73	13,08	13,26	11,94	12,29	10,48	8,89	9,20	7,46	9,88	8,69	8,13	7,27
LSTM	12,84	8,09	11,17	6,74	11,56	13,17	9,64	11,59	14,93	16,59	12,01	13,18	17,33	12,52	14,67	10,15	14,30	16,21	11,97	14,62
RNN	20,03	11,43	17,38	9,58	14.29	17,18	11,88	15,11	13,90	13,92	11,90	11,73	17,42	18,89	14,67	16,07	17,23	17,94	14,544	15,88
SVR	16,21	13,14	13,99	10,83	17,51	19,60	15,00	17,52	18,68	19,02	16,53	17,24	18,09	16,19	15,50	13,91	18,17	18,71	16,09	17,02



Şekil 5.2. Tahmin sonuçları karşılaştırma grafiği

5.3.2. CEP sonuçları ve değerlendirme

Bu kapsamda CEP motoru çıktıları performansını değerlendirmek için kesinlik, hassasiyet ve f1 ölçümü değerleri kullanılmıştır. Bu metriklerin hesaplanmasında aşağıdaki parametreler dikkate alınarak kesinlik, hassasiyet ve f1- ölçümü hesaplamaları Eş. 3.2, Eş. 3.3 ve Eş. 3.4 belirtilen formülasyonlara göre hesaplanmaktadır.

Karmaşık olaylar, kirletici gaz konsantrasyonlarının gözlemlenen değerleri ve tahmin edilen değerleri değerlendirilerek CEP kuralları tarafından belirlenir. Önerilen sistemde DL ve geleneksel yöntem ile tahmin edilen veriler ve gözlemlenen veriler CEP motoruna gönderilerek işlenmektedir. CEP motorunun performansını birden fazla perspektiften değerlendirmek amacı ile her bir gaza ait tahmin verilerine ilişkin CEP motor çıktısı ile gerçek değerlerine ilişkin CEP motor çıktısı değerlendirilmiştir. CEP motorunda, tespit edilen karmaşık olaylar, tahmin edilen veriler ve gözlemlenen veriler analiz edilerek önceden tanımlanmış kurallar aracılığıyla Çizelge 5.2'de belirlenen alarm seviyelerine göre belirlenmektedir. Tüm gazlar için karşılaştırmalı tahmin sonuçları Çizelge 5.7, 5.8, 5.9, 5.10 ve 5.11 de sunulmuştur.

Çizelge 5.7 nitrojen dioksit verilerinin tahmin verilerinin CEP motorunda işlenmesi sonucu oluşan alarm seviyelerine göre başarım sonuçları değerlendirmesi verilmiştir. CNN'in diğer modellere göre normal, uyarı ve kritik alarm türlerinin hepsinde diğer modellere göre daha iyi sonuç verdiği ölçülmüştür. Başarım sonuçları sırası ile LSTM, RNN ve SVR olarak sıralanmıştır.

Çizelge 5.8 ozon verilerinin CEP motoru çıktıları değerlendirilmiştir. Bu kapsamda CNN'in performansı uyarı ve kritik alarm durumları için %90 üzerinde ölçülmüştür. SVR modelinin ise diğer modeller ile karşılaştırıldığından başarım oranını düşük olduğu gözlemlenmiştir.

Çizelge 5.9 partikül madde verilerinin CEP motoru çıktıları değerlendirildiğinde tüm modellerin kritik alarm seviyesindeki tahmin başarım oranları %90 üzerinde ölçülmüştür. CNN modelinin diğer alarm durumlarından diğer modellere göre daha iyi sonuç verdiği ölçülmüştür. CEP motorunda, uyarı alarm durumuna ait başarım sonuçlarının CNN modelinin diğer modellere göre kısmen iyi sonuç verdiği gözlemlenmiştir.

Çizelge 5.7. Nitrojen dioksit CEP motoru sonuçları

	LSTM			RNN			SVR			CNN		
	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü
Normal	1.000	0.502	0.669	1.000	0.103	0.186	0	0	0	1.000	0.741	0.852
Uyarı	0.835	0.983	0.903	0.747	0.955	0.838	0.714	0.879	0.788	0.891	0.988	0.937
Kritik	0.977	0.973	0.975	0.942	0.987	0.964	0.859	0.996	0.923	0.984	0.959	0.972
Ortalama	0.913	0.899	0.889	0.747	0.955	0.838	0.646	0.774	0.704	0.942	0.937	0.935

Çizelge 5.8. Ozon CEP motoru sonuçları

	LSTM			RNN			SVR			CNN		
	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü
Normal	1.000	0.503	0.670	1.000	0.184	0.311	0	0	0	1.000	0.774	0.852
Uyarı	0.739	0.949	0.831	0.627	0.928	0.749	0.562	0.866	0.681	0.891	0.988	0.937
Kritik	0.920	1.000	0.958	0.890	0.999	0.941	0.812	1.000	0.896	0.984	0.959	0.972
Ortalama	0.864	0.829	0.815	0.806	0.723	0.667	0.457	0.641	0.532	0.942	0.937	0.935

Çizelge 5.9. Partikül madde CEP motoru sonuçları

	LSTM			RNN			SVR			CNN		
	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü
Normal	1.000	0.011	0.021	1.000	0.234	0.379	0	0	0	1.000	0.551	0.711
Uyarı	0.422	0.706	0.529	0.506	0.767	0.610	0.446	0.786	0.569	0.643	0.790	0.709
Kritik	0.921	1.000	0.959	0.937	1.000	0.967	0.941	1.000	0.970	0.942	1.000	0.970
Ortalama	0.843	0.767	0.710	0.869	0.819	0.795	0.680	0.780	0.721	0.898	0.880	0.875

Çizelge 5.10. Sülfür dioksit CEP motoru sonuçları

	LSTM			RNN			SVR			CNN		
	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü
Normal	0	0	0	1.000	0.533	0.695	0	0	0	1.000	0.526	0.690
Uyarı	0.831	0.733	0.779	0.912	0.690	0.786	0.831	0.733	0.779	0.923	0.812	0.864
Kritik	0.934	0.999	0.965	0.925	1.000	0.961	0.934	0.999	0.965	0.953	1.000	0.976
Ortalama	0.887	0.916	0.900	0.924	0.924	0.918	0.887	0.916	0.900	0.948	0.948	0.945

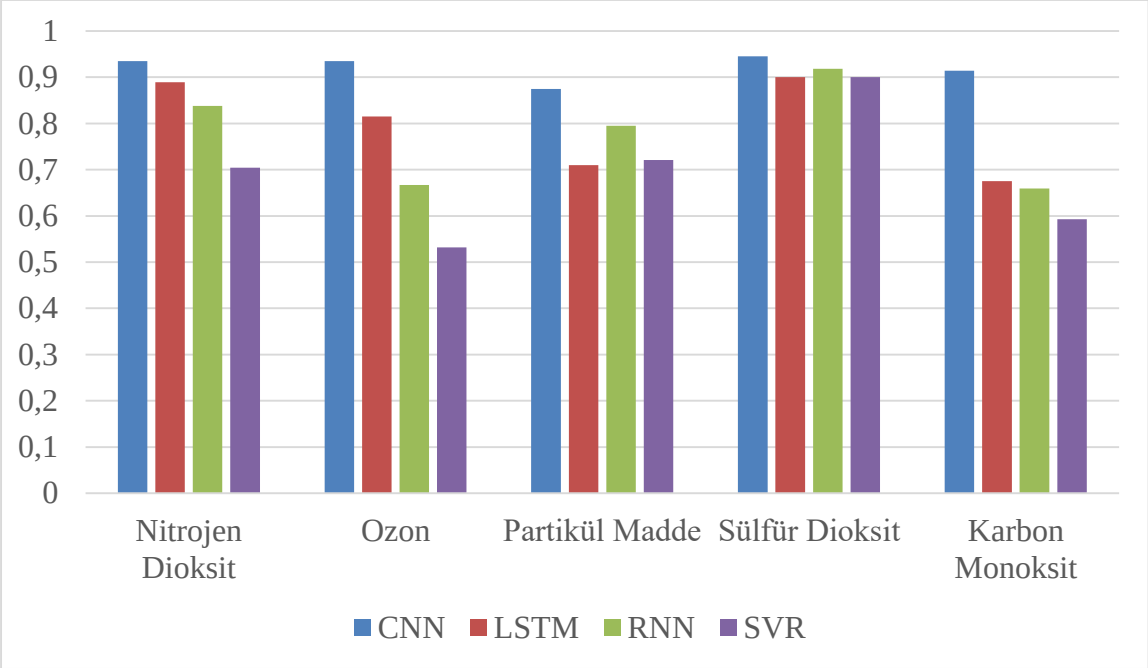
Çizelge 5.11. Karbon monoksit CEP motoru sonuçları

	LSTM			RNN			SVR			CNN		
	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü	Kesinlik	Hassasiyet	f-ölçümü
Normal	1.000	0.105	0.191	1.000	0.035	0.068	0	0	0	1.000	0.687	0.814
Uyarı	0.684	0.905	0.779	0.676	0.944	0.788	0.644	0.843	0.730	0.867	0.990	0.924
Kritik	0.842	1.000	0.914	0.900	0.997	0.946	0.765	1.000	0.867	0.981	0.987	0.984
Ortalama	0.800	0.741	0.675	0.810	0.743	0.659	0.523	0.685	0.593	0.928	0.918	0.914

Çizelge 5.10 sülfür dioksit verilerinin CEP motoru çıktıları değerlendirilmiştir. Bu kapsamda kritik alarm durumları için tüm modeller içim başarım oranı %90 üzerinde ölçülmüştür. Buna karşın normal ve uyarı seviyeleri için CNN modeli için diğer modellere göre daha iyi başarım performansı ölçülmüştür.

Çizelge 5.11 karbon monoksit verileri başarım sonuçları değerlendirilmiştir. Bu kapsamda CNN'in performansı uyarı ve kritik alarm durumları için %90 üzerinde ölçülmüştür. Deney sonuçlarına göre değerlendirme metrikleri açısından CNN'in diğer modellere göre daha iyi performans sergilediği gözlemlenmiştir.

Tüm sonuçlar değerlendirildiğinde önerilen CNN modelinin diğer DL ve geleneksel tahmin modelleri ile karşılaştırıldığından değerlendirme metrikleri açısından daha iyi sonuç verdiği gözlemlenmiştir. CNN modelinin hava kirletici gazlarının tümünde daha iyi sonuç verdiği gözlemlenmiştir. Şekil 5.3 'te tüm modellerin ortalama f- ölçümü sonuçları verilmiştir.



Şekil 5.3. Tüm modellerin ortalama f-ölçümü karşılaştırması

5.3.3. DeepFogCEP sistemi karmaşıklık analizi

Önerilen DeepFogCEP modelin karmaşıklığı Çizelge 5.5'te ifade edilen adımlara göre aşağıdaki şekilde hesaplanmıştır.

DeepFogCEP modelinin adımları incelendiğinde karmaşık analizini etkileyen tahmin ve CEP ünitesi işlemleridir. Diğer adımlar O(1) ve alarm sayısı kadar işlem alacaktır. Tahmin ünitesinde farklı DL modelleri ve geleneksel yöntemlerden SVR kullanılmıştır. Modellerin karmaşıklığı kullanılan parametrelere göre değişmektedir. Bu kapsamda en kötü yürütme karmaşıklığına sahip olan derin öğrenme algoritması baz alınmıştır. Yürütme karmaşıklığı O(n^2+m*n) olarak ifade edilmiştir [123] [188]. n girdi örnek sayısı, m girdi neron sayısı veya gizli katman sayısı olarak ifade edilmektedir. CEP motoru aşamasında yapılan işlem karmaşıklığı ise Çizelge 5.5'te belirtildiği üzere girdi sayısı kadar olacaktır. Bu bağlamda bu ünitenin karmaşıklığı O(n) olarak ifade edilebilir. DeepFogCEP sistemi toplam karmaşıklığı ise O(n^2+m*n+n) olarak ifade edilebilir.

5.3.4. Uçtan uca ağ gecikmesi sonuçları ve değerlendirilmesi

Bu bölüm kapsamında DeepFogCEP sisteminin gecikmeleri analiz edilmiştir.

DeepFogCEP 'in performansını değerlendirmek için, literatürde yaygın olarak kullanılan üç ağ ölçüsü kullanılmıştır. Bunlar, i) İletim gecikmesi, ii) DeepFogCEP işleme gecikmesi, iii) Uçtan uca toplam ağ gecikmesi olarak ifade edilebilir. Bu kapsamda iletişim gecikmeleri için sis mimarisi dikkate alınmıştır [123]. Sis mimarisinde kenar cihazları sensör değerlerini okuyarak işlem yapmadan sis sunucusuna iletir. Sis sunucusunda DeepFogCEP sistemi tahmin ve CEP motoru işlemleri gerçekleştirilmektedir. Önerilen DeepFogCEP sistemi ağ performansı Çizelge 5.12'de verilmiştir. Eğitim ve test aşamalarında elde edilen işleme sonuçları ve gecikmeler için %95 güven aralığı dikkate alınarak hesaplanmıştır.

Çizelge 5.12. Uçtan uca ağ gecikmesi karşılaştırma tablosu

	$T^{up}_{\ell_{ef}}$	$T^{up}_{\ell_{fc}}$	$T^{down}_{\ell_{cf}}$	$T^{down}_{\ell_{fe}}$	T^{trans}_{e2e}	$T^{proc}_{DeepFogCEP}$	$T^{network}_{total}$
CNN	77,611	119,693	56,100	15,620	269,024 ± 2,22	2,840 ± 0,05	271.864 ± 14.46
LSTM	75,062	119,541	56,249	15,748	266,600 ± 1,20	4,365 ± 0,02	271.864 ± 15.39
RNN	73,112	119,641	55,988	16,028	264,769 ± 2,03	2,918 ± 0,02	271.864 ± 18.66
SVR	76,196	121,949	55,377	16,035	269,557 ± 2,75	1,715 ± 0.01	271.864 ± 15.08

Çizelge incelendiğinde DeepFogCEP sistemi kapsamında önerilen CNN, LSTM, RNN ve SVR modellerin işleme süresi sırası ile 2.8, 4.3, 2.9 ve 1.7 olarak ölçülmüştür. Bu bağlamda SVR modelinin önceki veri bağımlılıklarına bağlı kalmadan tahmin işlemini gerçekleştirmesi sebebi ile en az işlem gecikmesi ölçülmüştür. İletişim gecikmeleri dikkate alındığında için kenar-sis cihazı arası 73 ve 77 ms arasında, sis-bulut arasında 119 ve 122 ms olarak, bulut-sis arası 55 ve 56 ms ve sis-kenar arasında 15 ve 16 ms olarak ifade edilebilir [123]. İletişim gecikmesi ile birlikte değerlendirdiğimizde LSTM ve RNN modellerindeki toplam gecikmenin diğer modellere göre nispeten daha az ağ gecikmesine neden olduğu ölçülmüştür. CNN ve SVR modelinin, LSTM ve RNN modellerine kıyasla daha yüksek toplam ağ gecikmesi ortaya çıkardığı görülmüştür. Toplam ağ gecikmesi ise LSTM, RNN, CNN ve SVR modelleri için sırası ile 270, 267, 271 ve 271 ms olarak ölçülmüştür. Tüm sonuçlar değerlendirildiğinde önerilen sistemin ağ gecikmesini % 1,1 artırdığı görülmüştür. Önerilen DeepFogCEP sisteminin ağdaki gecikmesi tolere edilebilir olarak kabul edilebilir.

5.4. Bölüm Değerlendirmesi

Bu bölümde IoT uygulamaları için erken uyarı sistemleri gerçekleştirebilmek amacı hem DL modelleri ile tahmin yöntemlerini kullanan hem de CEP ile gerçek zamanlı veri işleyen bir sistem tasarımına odaklanılmıştır. Bu kapsamda DL modelleri ve geleneksel modellerin tahmin yetenekleri ve CEP sistemi gerçek zamanlı işleme yetenekleri birleştirilerek DeepFogCEP isimli bir sistem sis hesaplama mimarisi dikkate alınarak önerilmiştir. Bu bağlamda CNN, LSTM, RNN ve SVR modelleri bulundukları ortamdaki hava kalite gazlarına ilişkin sensör değerlerinden bir sonraki değerleri tahmin etmektedir. Tahmin performanslarını RMSE ve MAE metrikleri ile değerlendirdiğimizde önerilen CNN modelinin tüm hava kirletici gaz konsantrasyonları için tahmin başarımının daha iyi olduğu ölçülmüştür.

Sonrasında, tahmin edilen değerler ile gerçek değerler CEP motoruna gönderilerek belirlenen alarm kurallarına göre işlenmektedir. Bu kapsamda CEP motoru çıktısı belirlenen AQI kurallarına göre yakalanan karmaşık olaylara göre değerlendirilmiştir. Bu kapsamda belirlenen normal, uyarı ve kritik alarm seviyeleri çıktılarına göre CNN modelinin diğer modellere göre daha başarılı sonuçlar verdiği gözlemlenmiştir.

Önerilen sistemin tasarımında sis- bulut tabanlı mimari dikkate alınmıştır. Bu kapsamda iletişim gecikmeleri, tahmin ve CEP ünitesi işleme gecikmeleri hesaplanarak değerlendirilmiştir. Bu kapsamda DeepFogCEP mimarisinin kenar, sis ve bulut mimarisi açısından uygulanabilirliği gösterilerek ağ performansı değerlendirilmiştir. Ağ performansı değerlendirildiğinde önerilen sistemin ağ gecikmesini % 1,1 artırdığı görülmüştür.

Sonuç olarak önerilen DeepFogCEP sistemi tahmin sistemi yetenekleri ile CEP işleme yeteneklerini birleştirerek geleceğe dönük alarmları erken olarak tespit etmektedir.

6. SONUÇ VE ÖNERİLER

CEP sistemleri uzun zamandır gerçek zamanlı veri işleme için kullanılan bir teknolojidir. atomik verileri içeren akış verilerinin analizi ile karmaşık olaylar çıkarılmaktadır. CEP uygulamaları genellikle alana özgü olup daha önceden belirlenen CEP kurallarına göre işlemektedir. Ancak günümüzde sensör sayısının artması ile birlikte birçok alanda sensörler tarafından akış verisi üretilmektedir. Üretilen akış verileri hacimsel olarak büyük ve nitelik sayısı olarak oldukça fazladır. Bu kapsamda CEP sistemlerinin yeni model ve teknolojiler ile bu gelişime uyması gerekmektedir. Bu kapsamda bu tez çalışması kapsamında CEP'in IoT alanında genişlemesine imkan sağlayabilecek otomatik kural çıkarımı ve CEP tahmin problemlerine ilişkin modeller önerilmiştir. Bu kapsamda tez çalışması kapsamında yapılan çalışmalar ve sonuçlar aşağıda özetlenmiştir.

Otomatik olarak CEP kurallarının çıkarılmasına ilişkin yapılan literatür taramasında önerilen modellerin genel olarak alana özgü ve IoT verisini dikkate almadığı tespit edilmiştir. Bu bağlamda tezin ilk aşamasında CEP genel kurallarının makine öğrenmesi teknikleri ile çıkarılmasına odaklanılmıştır. Literatürde yapılan çalışmalarda optimizasyon teknikleri ve makine öğrenmesi teknikleri etiketli veriler için önerildiği görülmüştür. Etiketsiz IoT verilerinden kural çıkarımı için tek bir yöntem kullanmanın yetersiz kalacağı görülmüştür. Bu sebeple önerilen model kapsamında hibrit yöntem kullanılarak model gerçekleştirilmiştir. Bu kapsamda önerilen model ile genel CEP kuralları çıkarılmıştır. Modelde hava kirletici gazları kullanılarak kümeleme işlemi gerçekleştirilmiş ve sonrasında kural madenciliği algoritmaları ile genel kurallar çıkarılmıştır. Bu tez çalışması kapsamında önerilen model IoT alan verilerini ve akıllı şehir uygulamalarını dikkate alması sebebi ile literatüre katkı sağlayacaktır. Bu kapsamda geliştirilen modelde kural madenciliği algoritma sonuçları karşılaştırmalı olarak değerlendirilmiş ve sonuçların PART ve JRIP algoritması için belirlenen başarım metriklerine göre %90 üzerinde başarım gösterdiği deney sonuçlarından anlaşılmaktadır. Bu bağlamda önerilen modelin zaman serisi içeren IoT veri seti dikkate alınarak yapılan testlerde genel kuralları çıkarmada başarılı sonuçlar ortaya koyacağını ortaya koymuştur.

Bu tez çalışmasında ikinci olarak adreslenen problem ise CEP kurallarının DL yöntemleri ile otomatik olarak çıkarılmasıdır. Bu bağlamda önerilen mimari ile DL yöntemleri ile CEP

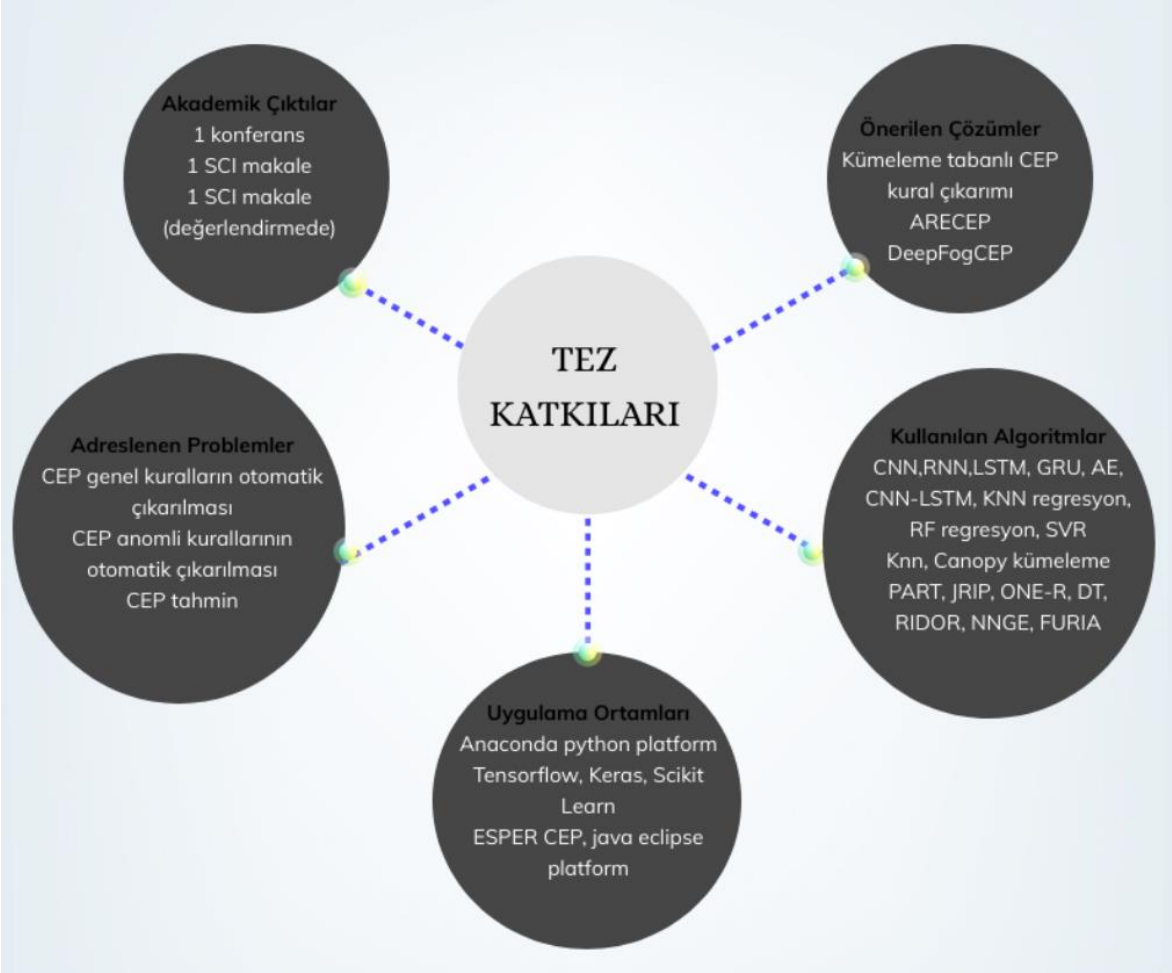
kural çıkarımına odaklanılmıştır. Önerilen çerçeve model literatürde DL yöntemleri kullanılarak CEP kural çıkarımını problemini ele alan ilk çalışmadır. Önerilen çerçeve kapsamında DL ve geleneksel yöntemlerinin zaman serilerinde anomali belirlemedeki başarım sonuçları karşılaştırılmıştır. Bu bağlamda en düşük yeniden çatılma hatasına sahip model GRU olarak ölçülmüştür. Modellerin kural çıkarım performansını karşılaştırmak amacı ile mimarinin ikinci aşamasında kural madenciliği algoritmaları kullanılmıştır. Bu kapsamda DL modellerinden GRU algoritması ve geleneksel veri madenciliği yöntemlerinden KNN regresyon yöntemi tercih edilerek kural çıkarım sonuçları nitelik sayısı değişimi dikkate alınarak yapılmıştır. GRU ile yapılan anomali etiketleme işlemi sonrasında yapılan kural madenciliği başarım sonuçları geleneksel yöntem ile karşılaştırıldığında değerlendirme metriklerine göre daha iyi sonuçlar verdiği gözlemlenmiştir. Özellikle sensör sayısı artması ile birlikte IoT akıllı şehir uygulamalarında analiz edilmesi gereken nitelik sayısı artacaktır. Bu kapsamda önerilen çerçeve ile bu durumun deney sonuçlarına etkisi değerlendirilmiştir. Veri nitelik sayısı arttığında kural çıkartma başarım performansının seçilen DL yöntemi için arttığı görülmektedir. Buna karşın veri nitelik sayısı arttığında geleneksel yöntem tercih edilerek yapılan etiketleme sonrasında kural çıkartma performansının artmadığı gözlemlenmiştir.

Bu tez çalışmasında son olarak CEP tahmin problemine odaklanılmıştır. Bu kapsamda zaman serileri tahmin yetenekleri güçlü olan DL yöntemleri ve SVR regresyon modeli test edilerek karşılaştırmalı sonuçlar sunulmuştur. Bu bağlamda önerilen sistem ile IoT sis konseptinde DL sonuçları ve CEP işleme motoru sonuçları ele alınmıştır. Önerilen sistem literatürde DL yöntemleri kullanılarak CEP tahmin problemini IoT alanında ele alan ilk çalışmadır. Tahmin performanslar RMSE ve MAE metrikleri açısından değerlendirilmiş ve CNN modelinin diğer modellere göre tahmin sonuçlarında daha iyi performans gösterdiği ölçülmüştür. CEP işleme motorunda AQI seviyelerine göre oluşturulan kurallar ile tahmin verileri analiz edilmiştir. Bu kapsamda DL metotlarından CNN modelinin CEP işleme motoru çıktıları diğer modellere göre daha başarılı sonuçlar vermiştir. Önerilen sistemde sis işleme yapısı dikkate alınarak iletişim gecikmeleri, tahmin ve CEP ünitesi işleme gecikmeleri hesaplanarak değerlendirilmiştir. Bu kapsamda sistemin ağ gecikmesini % 1,1 artırdığı görülmüştür.

IoT'nin hayatımızın her alanında yer alması ile birlikte birçok problemin çözümünde dikkate alınması gereken bir unsur halini almıştır. Bu kapsamda CEP alanında yer alan tahmin ve

kural çıkarımı problemlerinin çözümünde IoT mimarisi dikkate alınması gerekmektedir. Bununla birlikte mevcut CEP çalışmaları incelendiğinde kural çıkarımı sonrası kuralların geçerlenmesi gelecekte çözüm bekleyen problemler arasındadır. Bu bağlamda dağıtık mimaride kuralların çıkarılması problemi gelecek çalışma hedefleri arasında yer almaktadır.

Tez kapsamında yapılan katkılar Şekil 6.1'de sunulmuştur.



Şekil 6. 1. Tez Katkıları

KAYNAKLAR

1. Chanal, P.M., and Kakkasageri, M.S. (2020). Security and Privacy in IoT: A Survey. *Wireless Personal Communications*, 115(2), 1667-1693.

2. Mahdavinejad, M.S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P. and Sheth, A.P. (2018). Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks*, 4(3), 161-175.

3. Starks, F., Goebel, V., Kristiansen, S. and Plagemann, T. (2018). Mobile distributed complex event processing—Ubi Sumus? Quo vadimus? *Mobile Big Data*. Springer, 147-180.

4. Monnier, O. (2013). A smarter grid with the Internet of Things. *Texas Instruments*, 1-11.

5. Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I.A.T., Siddiqa, A. and Yaqoob, I. (2017). Big IoT data analytics: architecture, opportunities, and open research challenges. *ieee access*, 5, 5247-5261.

6. Mohammadi, M., Al-Fuqaha, A., Sorour, S. and Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, 20(4), 2923-2960.

7. Flouris, I., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Kamp, M. and Mock, M. (2017). Issues in complex event processing: Status and prospects in the big data era. *Journal of Systems and Software*, 127, 217-236.

8. Pielmeier, J., Braunreuther, S. and Reinhart, G. (2018). Approach for Defining Rules in the Context of Complex Event Processing. *Procedia CIRP*, 67, 8-12.

9. Akbar, A., Khan, A., Carrez, F. and Moessner, K. (2017). Predictive analytics for complex IoT data streams. *IEEE Internet of Things Journal*, 4(5), 1571-1582.

10. Akbar, A., Chaudhry, S.S., Khan, A., Ali, A. and Rafiq, W. (2019, December). *On Complex Event Processing for Internet of Things*. Paper presented at the 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Kiev, Ukraine.

11. Tümer, E.D. (2018). *Nesnelerin İnterneti Ortamındaki Veriler Kullanılarak Makine Öğrenmesi Algoritmaları İle Tahminleme ve Gerçek Zamanlı Veriler Üzerinde Karmaşık Olay İşleme*, Doktora Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.

12. Mehdiyev, N., Krumeich, J., Enke, D., Werth, D. and Loos, P. (2015). Determination of Rule Patterns in Complex Event Processing Using Machine Learning Techniques. *Procedia Computer Science*, 61, 395-401.

13. Buchmann, A. and Koldehofe, B. (2009). Complex event processing. *it-Information Technology*, 51(5), 241.

14. Teymourian, K. (2014). *A Framework for Knowledge-Based Complex Event Processing*, Doctoral Thesis, Freie Universitat Berlin, Berlin.

15. Anicic, D., Fodor, P., Rudolph, S., Stühmer, R., Stojanovic, N. and Studer, R. (2010, Semptember). *A rule-based language for complex event processing and reasoning*. Paper presented at the International Conference on Web Reasoning and Rule Systems, Bressanone, Italy.

16. Cugola, G. and Margara, A. (2012). Low latency complex event processing on parallel hardware. *Journal of Parallel and Distributed Computing,* 72(2), 205-218.

17. İnternet: Luckham D., S.W.R. (2011). Event Processing Technical Society. Event Processing Glossary – Version 2.0. URL: https://complexevents.com/wp-content/uploads/2011/08/EPTS_Event_Processing_Glossary_v2.pdf. Son Erişim Tarihi: 17.05.2021.

18. Luckham, D. (2008, October). *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Rule Representation. Paper presented at the Interchange and Reasoning on the Web, Orlando, FL, USA.

19. George, L., Cadonna, B. and Weidlich, M. (2016). IL-miner: instance-level discovery of complex event patterns. *Proceedings of the VLDB Endowment*, 10(1), 25-36.

20. Paschke, A. (2008). Design patterns for complex event processing. *arXiv preprint arXiv:0806.1100*.

21. Flouris, I., Giatrakos, N., Garofalakis, M. and Deligiannakis, A. (2015, August). *Issues in complex event processing systems*. Paper presented at the IEEE Trustcom/BigDataSE/ISPA, Washington, USA.

22. Michelson, B.M. (2006). Event-driven architecture overview. *Patricia Seybold Group,* 2(12), 10-1571.

23. Al-Haboobi, A.S., Alharan, A.F. and Alsagheer, R.H. (2016). Experimenting with Storm/Esper Integration and Programmatic Generation of Storm Topologies. *International Journal of Computer Science and Information Security,* 14(8), 169.

24. Cugola, G., Margara, A., Matteucci, M. and Tamburrelli, G. (2015). Introducing uncertainty in complex event processing: model, implementation, and validation. *Computing,* 97(2), 103-144.

25. Garcia Izaguirre Montemayor, J.A. (2012). *A Complex Event Processing System for Monitoring of Manufacturing Systems*, Master Thesis, Information Technology, Tampere.

26. Saleh, O. (2013). Complex Event Processing in Wireless Sensor Networks. *Grundlagen von Datenbanken*, 69-74.

27. Schaaf, M., Grivas, S.G., Ackermann, D., Diekmann, A., Koschel, A. and Astrova, I. (2012). Semantic complex event processing. *Recent Researches in Applied Information Science*, 38-43.

28. Paschke, A. and Boley, H. (2009). Rules capturing events and reactivity. *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches*, 215-252.

29. Gehani, N.H., Jagadish, H.V. and Shmueli, O. (1992, August). *Composite event specification in active databases: Model & implementation*. Paper Presented at the Proceedings of the 18th VLDB Conference, Vancover, Canada.

30. Agrawal, J., Diao, Y., Gyllstrom, D. and Immerman, N. (2008, June). *Efficient pattern matching over event streams*. Paper presented at the Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Newyork, USA.

31. Mei, Y. and Madden, S. (2009, June-July). *Zstream: a cost-based query processor for adaptively detecting composite events*. Paper presented at the Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, Rhode Island, USA.

32. Chakravarthy, S. and Jiang, Q. (2009). *Stream data processing: a quality of service perspective: modeling, scheduling, load shedding, and complex event processing* Springer Science & Business Media.

33. Eckert, M. (2008). *Complex event processing with XchangeEQ: language design, formal semantics, and incremental evaluation for querying events*, Doctoral Thesis, Ludwig Maximilians University, Munich.

34. Fülöp, L.J., Tóth, G., Rácz, R., Pánczél, J., Gergely, T., Beszédes, A. and Farkas, L. (2010 September). *Survey on complex event processing and predictive analytics*. Paper presented at the Proceedings of the Fifth Balkan Conference in Informatics, Novi Sad, Serbia.

35. Mousheimish, R., Taher, Y. and Zeitouni, K. (2017, June). *Automatic learning of predictive cep rules: bridging the gap between data mining and complex event processing*. Paper presented at the Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, Barcelona, Spain.

36. Burgueño, L., Boubeta-Puig, J. and Vallecillo, A. (2018). Formalizing complex event processing systems in Maude. *IEEE Access*, 6, 23222-23241.

37. Cugola, G. and Margara, A. (2012). Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44(3), 1-62.

38. Schultz-Møller, N.P., Migliavacca, M. and Pietzuch, P. (2009, July). *Distributed complex event processing with query rewriting*. Paper presented at the Proceedings of the Third ACM International Conference on Distributed Event-Based Systems, Nashville, Tennessee.

39. Akbar, A. (2017). *Extracting Knowledge from Raw IoT Data Streams*. Doctoral Thesis, University of Surrey (United Kingdom), Londra.

40. Zhang, H., Diao, Y. and Immerman, N. (2014, June). *On complexity and optimization of expensive queries in complex event processing*. Paper presented at the Proceedings of the 2014 ACM SIGMOD international conference on Management of data, Utah, USA.

41. Wang, W., Sung, J. and Kim, D. (2008, May). *Complex event processing in epc sensor network middleware for both rfid and wsn*. Paper presented at the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), Orlando, USA.

42. Giatrakos, N., Alevizos, E., Artikis, A., Deligiannakis, A. and Garofalakis, M. (2020). Complex event recognition in the big data era: a survey. *The VLDB Journal*, 29(1), 313-352.

43. Luckham, D.C. and Vera, J. (1995). An event-based architecture definition language. *IEEE transactions on Software Engineering*, 21(9), 717-734.

44. Gyllstrom, D., Wu, E., Chae, H.-J., Diao, Y., Stahlberg, P. and Anderson, G. (2006). SASE: Complex event processing over streams. *arXiv preprint cs/0612128*.

45. Demers, A.J., Gehrke, J., Panda, B., Riedewald, M., Sharma, V. and White, W.M. (2007, January). *Cayuga: A General Purpose Event Monitoring System.* Paper presented at the Conference on Innovative Data Systems Research, Cambridge, USA.

46. Robins, D. (2010, March). *Complex event processing*. Paper presented at the Second International Workshop on Education Technology and Computer Science, Wuhan, China.

47. Almnes, W.W.F.D. (2014). *EsperSens-A Complex Event Processing System for Automated Home Care*. Master Thesis, Institutt for informatikk, Oslo.

48. Liu, M., Rundensteiner, E.A., Dougherty, D., Gupta, C., Wang, S., Ari, I. and Mehta, A. (2010, September). *NEEL: The nested complex event language for real-time event analytics*. Paper presented at the International Workshop on Business Intelligence for the Real-Time Enterprise, Singapore, Singapore.

49. Bruns, R., Dunkel, J. and Offel, N. (2019). Learning of complex event processing rules with genetic programming. *Expert Systems with Applications*, 129, 186-199.

50. Keskisärkkä, R. and Blomqvist, E. (2013, May). *Semantic complex event processing for social media monitoring-a survey*. Paper presented at the Proceedings of Social Media and Linked Data for Emergency Response (SMILE) Co-located with the 10th Extended Semantic Web Conference, Montpellier, France.

51. Dayarathna, M. and Perera, S. (2018). Recent advancements in event processing. *ACM Computing Surveys (CSUR)*, 51(2), 1-36.

52. Nawaz, F., Janjua, N.K. and Hussain, O.K. (2019). PERCEPTUS: Predictive complex event processing and reasoning for IoT-enabled supply chain. *Knowledge-Based Systems*, 180, 133-146.

53. Alakari, A., Li, K.F. and Gebali, F. (2017, August). *Complex event processing enrichment: Motivations and challenges*. Paper presented at the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, Canada.

54. Yang, J., Ma, M., Wang, P. and Liu, L. (2015, August). *From complex event processing to cognitive event processing: approaches, challenges, and opportunities*. Paper presented at the IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China.

55. Khan, M.U.S., Khan, S.U. and Zomaya, A.Y. (2019). *Big Data-enabled Internet of Thing*s, The institute of Engineering and Technology, 2019.

56. Cabanillas, C., Di Ciccio, C., Mendling, J. and Baumgrass, A. (2014, September). *Predictive task monitoring for business processes*. Paper presented at the International Conference on Business Process Management, Eindhoven, Netherlands.

57. Schwegmann, B., Matzner, M. and Janiesch, C. (2013, February). *A Method and Tool for Predictive Event-Driven Process Analytics*. Paper presented at the Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI2013), Leipzig, Germany.

58. Simsek, M.U., Okay, F.Y. and Ozdemir, S. (2021). A deep learning-based CEP rule extraction framework for IoT data. *The Journal of Supercomputing*, 1-30.

59. Cugola, G. and Margara, A. (2013). Deployment strategies for distributed complex event processing. *Computing*, 95(2), 129-156.

60. de Almeida, V.P., Bhowmik, S., Endler, M. and Rothermel, K. (2020). DSCEP: an infrastructure for distributed semantic complex event processing. *arXiv preprint arXiv:2002.05869*.

61. Teymourian, K. and Paschke, A. (2010, March). *Enabling knowledge-based complex event processing*. Paper presented at the Proceedings of the 2010 EDBT/ICDT Workshops, Lausanne, Switzerland.

62. Liu, T. and Lu, D. (2012). *The application and development of IoT*. 2012 International Symposium on Information Technologies in Medicine and Education, 2, 991-994.

63. Suresh, P., Daniel, J.V., Parthasarathy, V. and Aswathy, R. (2014, March). *A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment*. Paper presented at the International conference on science engineering and management research (ICSEMR), Chennai, India.

64. Kumar, N.M. and Mallick, P.K. (2018). The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. *Procedia computer science*, 132, 109-117.

65. Cvar, N., Trilar, J., Kos, A., Volk, M. and Stojmenova Duh, E. (2020). The Use of IoT Technology in Smart Cities and Smart Villages: Similarities, Differences, and Future Prospects. *Sensors,* 20(14), 3897.

66. Mrabet, H., Belguith, S., Alhomoud, A. and Jemai, A. (2020). A survey of IoT security based on a layered architecture of sensing and data analysis. *Sensors,* 20(13), 3625.

67. Mahmoud, R., Yousuf, T., Aloul, F. and Zualkernan, I. (2015, December). *Internet of things (IoT) security: Current status, challenges and prospective measures*. Paper presented at the 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, England.

68. Chen, S., Xu, H., Liu, D., Hu, B. and Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal*, 1(4), 349-359.

69. Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J. and Du, H.-Y. (2010, August). *Research on the architecture of Internet of Things*. Paper presented at the 3rd international conference on advanced computer theory and engineering (ICACTE), Chengdu, China.

70. Patel, K.K. and Patel, S.M. (2016). Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, 6(5).

71. Darwish, D. (2015). Improved layered architecture for Internet of Things. *Int. J. Comput. Acad. Res.(IJCAR)*, 4, 214-223.

72. Al-Qaseemi, S.A., Almulhim, H.A., Almulhim, M.F. and Chaudhry, S.R. (2016, December). *IoT architecture challenges and issues: Lack of standardization*. Paper presented at the Future technologies conference (FTC), San Francisco, United States.

73. Kumar, P.R., Wan, A.T. and Suhaili, W.S.H. (2020). Exploring Data Security and Privacy Issues in Internet of Things Based on Five-Layer Architecture. *International Journal of Communication Networks and Information Security*, 12(1), 108-121.

74. Sethi, P. and Sarangi, S.R. (2017). Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017.

75. Balaji, S., Nathani, K. and Santhakumar, R. (2019). IoT technology, applications and challenges: a contemporary survey. *Wireless personal communications*, 108(1), 363-388.

76. Abdelmoneem, R.M., Benslimane, A., Shaaban, E., Abdelhamid, S. and Ghoneim, S. (2019, May). *A cloud-fog based architecture for iot applications dedicated to healthcare*. Paper presented at the ICC 2019 IEEE International Conference on Communications (ICC), Shangai, China.

77. Gardašević, G., Veletić, M., Maletić, N., Vasiljević, D., Radusinović, I., Tomović, S. and Radonjić, M. (2017). The IoT architectural framework, design issues and application domains. *Wireless personal communications*, 92(1), 127-148.

78. Rahim, M.A., Rahman, M.A., Rahman, M., Asyhari, A.T., Bhuiyan, M.Z.A. and Ramasamy, D. (2020). Evolution of IoT-enabled connectivity and applications in automotive industry: A review. *Vehicular Communications*, 100285.

79. Shah, S.H. and Yaqoob, I. (2016, August). *A survey: Internet of Things (IOT) technologies, applications and challenges*. Paper presented at the IEEE Smart Energy Grid Engineering (SEGE), Oshawa, Canada.

80. Iyer, B. and Patil, N. (2018). IoT enabled tracking and monitoring sensor for military applications. *International Journal of System Assurance Engineering and Management*, 9(6), 1294-1301.

81. Talavera, J.M., Tobón, L.E., Gómez, J.A., Culman, M.A., Aranda, J.M., Parra, D.T., . . . Garreta, L.E. (2017). Review of IoT applications in agro-industrial and environmental fields. *Computers and Electronics in Agriculture*, 142, 283-297.

82. Shahid, N. and Aneja, S. (2017, February). *Internet of Things: Vision, application areas and research challenges*. Paper presented at the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, İndia

83. Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660.

84. Rehman, H.U., Asif, M. and Ahmad, M. (2017, October). *Future applications and research challenges of IOT*. Paper presented at the International conference on information and communication technologies (ICICT), Karachi, Pakistan.

85. Afzal, B., Umair, M., Shah, G.A. and Ahmed, E. (2019). Enabling IoT platforms for social IoT applications: Vision, feature mapping, and challenges. *Future Generation Computer Systems*, 92, 718-731.

86. Mukhopadhyay, S.C. and Suryadevara, N.K. (2014). *Internet of things: Challenges and opportunities*. Springer International Publishing, 1-17.

87. Sisinni, E., Saifullah, A., Han, S., Jennehag, U. and Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, 14(11), 4724-4734.

88. Hand, D.J. (2007). Principles of data mining. *Drug safety*, 30(7), 621-622.

89. Kodinariya, T.M. and Makwana, P.R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal*, 1(6), 90-95.

90. Kumar, A., Ingle, Y.S., Pande, A. and Dhule, P. (2014). Canopy clustering: a review on pre-clustering approach to K-means clustering. *Int. J. Innov. Adv. Comput. Sci.(IJIACS),* 3(5), 22-29.

91. Zhang, G., Zhang, C. and Zhang, H. (2018). Improved K-means algorithm based on density Canopy. *Knowledge-based systems,* 145, 289-297.

92. Likas, A., Vlassis, N. and Verbeek, J.J. (2003). The global k-means clustering algorithm. *Pattern recognition,* 36(2), 451-461.

93. Zhang, W., Hu, H., Hu, H. and Fang, J. (2019). Semantic distance between vague concepts in a framework of modeling with words. *Soft Computing,* 23(10), 3347-3364.

94. Breiman, L. (2001). Random forests. *Machine learning,* 45(1), 5-32.

95. Kane, M.J., Price, N., Scotch, M. and Rabinowitz, P. (2014). Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC bioinformatics,* 15(1), 1-9.

96. Liaw, A. and Wiener, M. (2002). Classification and regression by randomForest. *R news,* 2(3), 18-22.

97. Zhou, X., Zhu, X., Dong, Z. and Guo, W. (2016). Estimation of biomass in wheat using random forest regression algorithm and remote sensing data. *The Crop Journal,* 4(3), 212-219.

98. Song, Y., Liang, J., Lu, J. and Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing,* 251, 26-34.

99. Martínez, F., Frías, M.P., Pérez-Godoy, M.D. and Rivera, A.J. (2018). Dealing with seasonality by narrowing the training set in time series forecasting with kNN. *Expert systems with applications,* 103, 38-48.

100. Ho, C.-H. and Lin, C.-J. (2012). Large-scale linear support vector regression. *The Journal of Machine Learning Research,* 13(1), 3323-3348.

101. Yu, H. and Kim, S. (2012). SVM Tutorial-Classification, Regression and Ranking. *Handbook of Natural computing,* 1, 479-506.

102. Karal, Ö. (2018). EKG verilerinin destek vektör regresyon yöntemiyle sıkıştırılması. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi,* 33(2), 743-756.

103. Mahajan, A. and Ganpati, A. (2014). Performance evaluation of rule based classification algorithms. *International Journal of Advanced Research in Computer Engineering & Technology,* 3(10), 3546-3550.

104. Ponce, J. and Karahoca, A. (2009). *Data Mining and Knowledge Discovery in Real Life Applications* (First Edition), i-Tech Education and Publishing, Croatia.

105. Panda, M. and Patra, M.R. (2009, October). *Ensembling rule based classifiers for detecting network intrusions*. Paper presented at the International Conference on Advances in Recent Technologies in Communication and Computing, Kerala, India.

106. MeeraGandhi, G., Appavoo, K. and Srivasta, S. (2010). Effective network intrusion detection using classifiers decision trees and decision rules. *Int. J. Advanced network and application,* 2.

107. Veeralakshmi, V. and Ramyachitra, D. (2015). Ripple down rule learner (ridor) classifier for iris dataset. *Issues,* 1(1), 79-85.

108. Holte, R.C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning,* 11(1), 63-90.

109. Devasena, C.L., Sumathi, T., Gomathi, V. and Hemalatha, M. (2011). Effectiveness evaluation of rule based classifiers for the classification of iris data set. *Bonfring International Journal of Man Machine Interface,* 1, 05-09.

110. Hühn, J. and Hüllermeier, E. (2009). FURIA: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery,* 19(3), 293-319.

111. Sutskever, I., Vinyals, O. and Le, Q.V. (2014). Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

112. Wang, S., Sun, S., Li, Z., Zhang, R. and Xu, J. (2017). Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology,* 13(1), e1005324.

113. Längkvist, M., Karlsson, L. and Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters,* 42, 11-24.

114. Baldi, P. (2012, July). *Autoencoders, unsupervised learning, and deep architectures*. Paper presented at the Proceedings of ICML workshop on unsupervised and transfer learning, Washington, USA.

115. Shickel, B., Tighe, P.J., Bihorac, A. and Rashidi, P. (2017). Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE journal of biomedical and health informatics,* 22(5), 1589-1604.

116. Druzhkov, P. and Kustikova, V. (2016). A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis,* 26(1), 9-15.

117. Li, T., Zhang, Z. and Chen, H. (2019). Predicting the combustion state of rotary kilns using a Convolutional Recurrent Neural Network. *Journal of Process Control,* 84, 207-214.

118. Gupta, L. and McAvoy, M. (2000). Investigating the prediction capabilities of the simple recurrent neural network on real temporal sequences. *Pattern Recognition*, 33(12), 2075-2081.

119. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

120. Bontemps, L., McDermott, J. and Le-Khac, N.-A. (2016, November). *Collective anomaly detection based on long short-term memory recurrent neural networks*. Paper presented at the International Conference on Future Data and Security Engineering, Can Tho City, Vietnam.

121. Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K. and Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*.

122. Abdel-Nasser, M. and Mahmoud, K. (2019). Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Computing and Applications*, 31(7), 2727-2740.

123. Kok, I. and Ozdemir, S. (2021). DeepMDP: A Novel Deep-Learning-Based Missing Data Prediction Protocol for IoT. *IEEE Internet of Things Journal*, 8(1), 232-243.

124. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

125. Dey, R. and Salem, F.M. (2017, Agust). *Gate-variants of gated recurrent unit (GRU) neural networks*. Paper presented at the IEEE 60th international midwest symposium on circuits and systems (MWSCAS), Medford, United States.

126. Zhao, B., Lu, H., Chen, S., Liu, J. and Wu, D. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1), 162-169.

127. Mittal, S. (2020). A survey of FPGA-based accelerators for convolutional neural networks. *Neural computing and applications*, 32(4), 1109-1139.

128. Pan, H., He, X., Tang, S. and Meng, F. (2018). An improved bearing fault diagnosis method using one-dimensional CNN and LSTM. *J. Mech. Eng*, 64(7-8), 443-452.

129. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J. and Fricout, G. (2012, June). *Steel defect classification with max-pooling convolutional neural networks*. Paper presented at the The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia.

130. Mousheimish, R., Taher, Y. and Zeitouni, K. (2017, June). *Automatic Learning of Predictive CEP Rules*. Paper presented at the Proceedings of the 11th ACM international conference on distributed and event-based systems Barcelona, Spain.

131. Margara, A., Cugola, G. and Tamburrelli, G. (2014, May). *Learning from the past: automated rule generation for complex event processing*. Paper presented at the Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, Mumbai, India.

132. İnternet: The CityPulse Consortium. URL: http://iot.ee.surrey.ac.uk:8080/. Erişim Tarihi: 17.05.2021

133. Ali, M.I., Gao, F. and Mileo, A. (2015, October). *Citybench: A configurable benchmark to evaluate rsp engines using smart city datasets*. Paper presented at the International Semantic Web Conference, Penslyvania, USA.

134. Zencirci, S.A. and IŞIKLI, B. (2017). Hava Kirliliği. *ESTÜDAM Halk Sağlığı Dergisi,* 2(2), 24-36.

135. Boubeta-Puig, J., Ortiz, G. and Medina-Bulo, I. (2014). A model-driven approach for facilitating user-friendly design of complex event patterns. *Expert Systems with Applications*, 41(2), 445-456.

136. Petersen, E., To, M.A., Maag, S. and Yamga, T. (2018, November). *An unsupervised rule generation approach for online complex event processing*. Paper presented at the IEEE 17th International Symposium on Network Computing and Applications (NCA), USA.

137. Hasan, A., Teymourian, K. and Paschke, A. (2015, August). *Probabilistic event pattern discovery*. International Symposium on Rules and Rule Markup Languages for the Semantic Web , Berlin, Germany

138. George, L. (2015, October). *Event pattern mining for smart environments*. Paper presented at the International SDL Forum, Berlin, Germany.

139. Lee, O.J. and Jung, J.E. (2017). Sequence Clustering-based Automated Rule Generation for Adaptive Complex Event Processing. *Future Generation Computer Systems*, 66, 100-109.

140. Mutschler, C. and Philippsen, M. (2012, June). *Learning event detection rules with noise hidden markov models*. Paper presented at the NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Nuremberg, Germany.

141. Frommgen, A., Rehner, R., Lehn, M. and Buchmann, A. (2015, July). *Fossa: Learning ECA Rules for Adaptive Distributed Systems*. Paper presented at the IEEE International Conference on Autonomic Computing, Grenoble, France.

142. Yuan, C. and Yang, H. (2019). Research on K-value selection method of K-means clustering algorithm. *J—Multidisciplinary Scientific Journal*, 2(2), 226-235.

143. Aydilek, İ.B. (2018). Yazılım hata tahmininde kullanılan metriklerin karar ağaçlarındaki bilgi kazançlarının incelenmesi ve iyileştirilmesi. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi,* 24(5), 906-914.

144. Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing,* 21(2), 137-146.

145. Gökalp, M.O., Koçyiğit, A. and Eren, P.E. (2019). A visual programming framework for distributed Internet of Things centric complex event processing. *Computers & Electrical Engineering,* 74, 581-604.

146. Kawashima, H., Kitagawa, H. and Li, X. (2010, November). *Complex event processing over uncertain data streams*. Paper presented at the 2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Fukuoka, Japan.

147. Mahmood, N., Pasha, M.K. and Pasha, K.A. (2017). Survey of Applications of Complex Event Processing (CEP) in Health Domain. *Sukkur IBA Journal of Computing and Mathematical Sciences,* 1(2), 88-94.

148. Alias, C., Rawet, V.L., Neto, H.X.R. and Reymão, J.d.E.N. (2016, March). *Investigating into the Prevalence of Complex Event Processing and Predictive Analytics in the Transportation and Logistics Sector: Initial Findings From Scientific Literature*. Paper presented at the Tenth Mediterranean Conference on Information Systems, Paphos, Cyprus.

149. Mijović, V., Tomašević, N., Janev, V., Stanojević, M. and Vraneš, S. (2019). Emergency management in critical infrastructures: a complex-event-processing paradigm. *Journal of Systems Science and Systems Engineering,* 28(1), 37-62.

150. Fernandes, G., Rodrigues, J.J., Carvalho, L.F., Al-Muhtadi, J.F. and Proença, M.L. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems,* 70(3), 447-489.

151. Kwon, D., Kim, H., Kim, J., Suh, S.C., Kim, I. and Kim, K.J. (2019). A survey of deep learning-based network anomaly detection. *Cluster Computing,* 22(1), 949-961.

152. Yin, C., Zhu, Y., Fei, J. and He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access,* 5, 21954-21961.

153. Chalapathy, R. and Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.

154. Canizo, M., Triguero, I., Conde, A. and Onieva, E. (2019). Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. *Neurocomputing,* 363, 246-260.

155. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P. and Shroff, G. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.

156. Malhotra, P., Vig, L., Shroff, G. and Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. *Proceedings,* 89, 89-94

157. Chauhan, S. and Vig, L. (2015, October). *Anomaly detection in ECG time signals via deep long short-term memory networks*. Paper presented at the IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France.

158. Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Chawla, N.V. (2019, February). *A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data*. Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, Hawai.

159. Munir, M., Siddiqui, S.A., Dengel, A. and Ahmed, S. (2018). DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access,* 7, 1991-2005.

160. Xing, T., Vilamala, M.R., Garcia, L., Cerutti, F., Kaplan, L., Preece, A. and Srivastava, M. (2019, June). *Deepcep: Deep complex event processing using distributed multimodal information*. Paper presented at the IEEE International Conference on Smart Computing (SMARTCOMP), Washington, USA.

161. Kondeti, P.K., Ravi, K., Mutheneni, S.R., Kadiri, M.R., Kumaraswamy, S., Vadlamani, R. and Upadhyayula, S.M. (2019). Applications of machine learning techniques to predict filariasis using socio-economic factors. *Epidemiology & Infection,* 147, e260.

162. Özkan, İ. and Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi,* 6(3), 85-104.

163. Soydaner, D. (2020). A comparison of optimization algorithms for deep learning. *arXiv preprint arXiv:2007.14166*.

164. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L. and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery,* 33(4), 917-963.

165. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Isard, M. (2016, August). *Tensorflow: A system for large-scale machine learning*. Paper presented at the 12th {USENIX} symposium on operating systems design and implementation, Savannah, GA.

166. İnternet: Chollet, F. (2015). Keras: Deep learning library for theano and tensorflow. URL: https://keras. io/k, Son Erişim Tarihi:17.05.2021.

167. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research,* 12, 2825-2830.

168. An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE,* 2(1), 1-18.

169. Agency, E.P. (2014). Air Quality Index: A guide to air quality and your health, URL: https://www.airnow.gov/aqi/aqi-basics/, Son Erişim Tarihi: 17/05/2021.

170. Dong, L., Shu, W., Sun, D., Li, X. and Zhang, L. (2017). Pre-alarm system based on real-time monitoring and numerical simulation using internet of things and cloud computing for tailings dam in mines. *IEEE Access,* 5, 21080-21089.

171. Qin, Y., Sheng, Q.Z., Falkner, N.J., Dustdar, S., Wang, H. and Vasilakos, A.V. (2016). When things matter: A survey on data-centric internet of things. *Journal of Network and Computer Applications,* 64, 137-153.

172. Naas, M.I., Parvedy, P.R., Boukhobza, J. and Lemarchand, L. (2017, May). *iFogStor: an IoT data placement strategy for fog infrastructure*. Paper presented at the IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, Spain.

173. Atlam, H.F., Walters, R.J. and Wills, G.B. (2018). Fog computing and the internet of things: A review. *big data and cognitive computing,* 2(2), 10.

174. Akbar, A., Carrez, F., Moessner, K. and Zoha, A. (2015). *Predicting complex events for pro-active IoT applications*. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT).

175. Bruns, R., Dunkel, J., Masbruch, H. and Stipkovic, S. (2015). Intelligent M2M: Complex event processing for machine-to-machine communication. *Expert Systems with Applications,* 42(3), 1235-1246.

176. Wang, J. and Song, G. (2018). A deep spatial-temporal ensemble model for air quality prediction. *Neurocomputing,* 314, 198-206.

177. Li, X., Peng, L., Hu, Y., Shao, J. and Chi, T. (2016). Deep learning architecture for air quality predictions. *Environmental Science and Pollution Research,* 23(22), 22408-22417.

178. Athira, V., Geetha, P., Vinayakumar, R. and Soman, K. (2018). Deepairnet: Applying recurrent networks for air quality prediction. *Procedia computer science,* 132, 1394-1403.

179. Fülöp, L.J., Beszédes, Á., Tóth, G., Demeter, H., Vidács, L. and Farkas, L. (2012, September). *Predictive complex event processing: a conceptual framework for combining complex event processing and predictive analytics*. Paper presented at the Proceedings of the Fifth Balkan Conference in Informatics, Novi Sad, Serbia.

180. Wang, Y. and Cao, K. (2014). A proactive complex event processing method for large-scale transportation internet of things. *International Journal of Distributed Sensor Networks,* 10(3), 159052.

181. Wang, Y., Gao, H. and Chen, G. (2018). Predictive complex event processing based on evolving Bayesian networks. *Pattern Recognition Letters,* 105, 207-216.

182. Nechifor, S., Târnaucă, B., Sasu, L., Puiu, D., Petrescu, A., Teutsch, J., Moldoveanu, F. (2014). *Autonomic monitoring approach based on cep and ml for logistic of sensitive goods*. IEEE 18th International Conference on Intelligent Engineering Systems INES 2014.

183. Christ, M., Krumeich, J. and Kempa-Liehr, A.W. (2016, September). *Integrating predictive analytics into complex event processing by using conditional density estimations*. Paper presented at the IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW), Vienna, Austria.

184. EMERSON, R.J., HOSSEN, J., ERVINA, E., TAWSIF, K. and JESMEEN, M. (2020). Broadband network fault prediction using complex event processing and predictive analytics techniques. *Journal of Engineering Science and Technology*, 15(4), 2289-2300.

185. Yadav, P., Sarkar, D., Salwala, D. and Curry, E. (2020). Traffic Prediction Framework for OpenStreetMap using Deep Learning based Complex Event Processing and Open Traffic Cameras. *arXiv preprint arXiv:2008.00928*.

186. İnternet: Espertech. URL: https://www.espertech.com/esper/. Erişim Tarihi: 21.05.2021.

187. Chai, T. and Draxler, R.R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development*, 7(3), 1247-1250.

188. Sak, H., Senior, A. and Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*.

GAZİ GELECEKTİR...