

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ELEKTRİKLİ OTOBÜSLERDE ÇİZGE TABANLI ÖZNİTELİK SEÇİMİ VE
MAKİNE ÖĞRENMESİ İLE KESTİRİMCİ BAKIM**

YÜKSEK LİSANS TEZİ

Ayşe Irmak ERÇEVİK

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Ahmet Murat Özbayoglu

AĞUSTOS 2024



TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.



Ayşe Irmak ERÇEVİK



ÖZET

Yüksek Lisans

ELEKTRİKLİ OTOBÜSLERDE ÇİZGE TABANLI ÖZNİTELİK SEÇİMİ VE MAKİNE ÖĞRENMESİ İLE KESTİRİMCİ BAKIM

Ayşe Irmak ERÇEVİK

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ahmet Murat ÖZBAYOĞLU

Tarih: Ağustos 2024

Bu tez çalışmasında, elektrikli otobüslere ait CAN (Controller Area Network) Bus verileri kullanılarak çizge tabanlı öznitelik seçim yöntemi ile sistemin özellikleri arasındaki ilişkiler analiz edilmiş ve üretici tarafından hedeflenen alarmların yapay zekâ teknikleri ile kestirim performansı araştırılmıştır.

Araştırmada yaklaşık iki yıl süresince, farklı araçlardan elde edilen CAN Bus verileri kullanılmıştır. Ham veri paketleri üzerinde kapsamlı ön işleme teknikleri uygulanmış ve makine öğrenmesi modellerinin yapısal ihtiyaçlarına uygun veri setleri oluşturulmaya gayret edilmiştir. Bu süreçte Pearson korelasyon analizi, Cramer's V istatistik tablosu ve Anova F-testi gibi istatistiksel filtreleme yöntemleri ve InfoMap, Leiden, Louvain, Fast Greedy gibi optimizasyon tabanlı topluluk tespiti algoritmaları kullanılarak, hibrit çizge tabanlı öznitelik seçim aracı geliştirilmiştir. Geliştirilen araç kullanılarak özellik altkümeleri belirlenmiş, özelliklerin ilişkisel konumu ve hedeflenen alarmlarla ilişkileri değerlendirilerek ayırt edici öznitelikler seçilmiş ve boyut azaltımı yapılmıştır.

Makine öğrenmesi modellerinin seçilmesinde, Destek Vektör Makinesi (SVM), Rastgele Orman (RF), eXtreme Gradient Boosting (XGBoost), Uzun Kısa Süreli Bellek (LSTM) ve Tekrarlayan Sinir Ağları (RNN) modelleri dahil olmak üzere sinir ağı mimarileri ve geleneksel sınıflandırıcılar uygulanarak performansları

değerlendirilmiştir. Sınıf dengesizliğinin modellerin performansındaki etkisini minimize etmek için SMOTEEN (Synthetic Minority Oversampling Technique-Edited Nearest Neighbours) ve ikili arama tabanlı zaman aralığı örnekleme (binary search based time interval down-sampling) yöntemleri uygulanmıştır. Her modelin en iyi ayırt edici sınıflandırıcı uzayını bulmak için Izgara Arama (Grid Search) ve Rastgele Arama (Random Search) tabanlı hiperparametre optimizasyonu (Hyperparameter Tuning) yapılmıştır. Geliştirilen modellerin performansları geçerleme (validation) metrikleri esas alınarak değerlendirilmiştir. Modellerin kestirimleri özniteliklerin ağırlıkları incelenerek ve LIME (Local Interpretable Model-Agnostic Explanations) aracı kullanılarak açıklanmaya çalışılmış, kestirimleri tetikleyen öznitelikler belirlenmiş ve sistemin işleyişi (tasarımı) ile ilgili içgörüler elde edilmiştir.

Bu çalışma kapsamında CAN Bus verileri kullanılarak araçlardaki belli alarmların makine öğrenmesi teknikleri ile kestirim performansı araştırılmış, çizge tabanlı bir öznitelik seçme aracı geliştirilerek düzeltici, önleyici, kestirimci ve proaktif bakım (tasarımın iyileştirilmesi) yaklaşımlarının iyileştirilmesine yardımcı olacak şekilde öznitelikler arasındaki ilişkilerin görsel olarak yorumlanması imkânı sağlanmıştır. Çalışmada izlenen makine öğrenmesi yaşam döngüsü kapsamında uygulanan tekniklerin farklı sektörlerde, Endüstri 4.0 çalışmalarında ve bakım stratejilerinin makine öğrenimi yoluyla iyileştirilmesinde, özellikle veri kalitesi ve uzman görüşüne erişim zorluğu olan projelerde yararlı olabileceği değerlendirilmektedir.

Anahtar Kelimeler: Kestirimci bakım (PdM), Makine öğrenimi (ML), Çizge tabanlı öznitelik seçimi, Açıklanabilir yapay zekâ (XAI).

ABSTRACT

Master of Science

PREDICTIVE MAINTANANCE WITH GRAPH BASED FEATURE SELECTION & MACHINE LEARNING IN ELECTRICAL BUSES

Ayşe Irmak ERÇEVİK

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Ahmet Murat ÖZBAYOĞLU

Date: August 2024

This thesis employs a graph-based feature selection method to analyse the relationships between the characteristics of a CAN (Controller Area Network) Bus data set pertaining to electric buses. Furthermore, it investigates the prediction performance of targeted alarms utilising artificial intelligence techniques.

The study employed data obtained from CAN Bus systems of diverse vehicles over an extended period. The raw data packages underwent extensive pre-processing techniques to create data sets suitable for the structural needs of machine learning models. In this process, a hybrid graph-based feature selection tool was developed using a combination of statistical filtering methods, including Pearson correlation analysis, Cramer's V statistical table and ANOVA F-test, and optimisation-based community detection algorithms, such as InfoMap, Leiden, Louvain and Fast Greedy. The developed tool was employed to identify feature subsets, evaluate the relational position of the features and their relationship with the targeted alarms, select distinctive features and perform dimension reduction.

In the selection of machine learning models, a number of different approaches were considered, including neural network architectures and traditional classifiers. Amongst these, the Support Vector Machine (SVM), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), and Recurrent Neural

Network (RNN) models were applied, and their performance was then evaluated. In order to mitigate the impact of class imbalance on model performance, SMOTEEN (Synthetic Minority Oversampling Technique-Edited Nearest Neighbours) and binary search-based time interval down-sampling methods were employed. In order to identify the optimal distinctive classifier space for each model, a grid search and random search-based hyperparameter optimisation (hyperparameter tuning) was conducted. The performance of the developed models was evaluated based on the metrics employed for model validation. The predictions of the models were explained by examining the weights (importance) of the features and utilising the LIME (Local Interpretable Model-Agnostic Explanations) tool to identify the features that triggered the predictions, thereby gaining insights into the functionality (design) of the system.

The present study has investigated the prediction performance of specific alarms in vehicles utilising CAN Bus data with machine learning techniques. Additionally, a graph-based feature selection tool has been developed, enabling the visual interpretation of relationships between features and facilitating the enhancement of corrective, preventive, predictive and proactive maintenance (design improvement) approaches. It is concluded that the techniques employed throughout the machine learning life cycle, as conducted in the study, can provide valuable insights and contribute to the advancement of maintenance strategies through machine learning, particularly in contexts where data quality and access to expert opinion may present challenges, in alignment with the principles of Industry 4.0.

Keywords: Predictive maintenance (PdM), Machine learning (ML), Graph based feature selection, Explainable artificial intelligence (XAI).

TEŞEKKÜR

Yüksek lisans çalışmalarım süresince bana rehberlik eden ve her zaman destekleyen danışmanım Prof. Dr. Ahmet Murat Özbayoğlu'na en içten teşekkürlerimi sunarım. Ayrıca, başta Prof. Dr. Ali Aydın Selçuk ve Dr. Öğretim Üyesi Buğra Çaşkurlu olmak üzere, TOBB Üniversitesi Bilgisayar Bilimleri Bölümü'ndeki değerli hocalarıma, TUBİTAK 1711- 3227015 numaralı Projesi'nde emeği geçen TUBİTAK ve TEMSA çalışanlarına, değerli çalışma arkadaşlarım Mehmet Deniz Türkmen, Hande Teke, Başak Demirok, Gökalp Coşgun ve Tunahan Oğuz'a minnettarım. Son olarak, eğitim hayatım boyunca bana destek olan sevgili aileme teşekkür ederim.





İÇİNDEKİLER

Sayfa

ÖZET.....	vii
ABSTRACT	ix
TEŞEKKÜR	xi
İÇİNDEKİLER	xiii
ŞEKİL LİSTESİ	xv
ÇİZELGE LİSTESİ	xvii
ALGORİTMA LİSTESİ.....	xix
KISALTMALAR.....	xxi
1. GİRİŞ.....	1
1.1. Tezin Konusu	1
1.2. Motivasyon	1
1.3. Veriye Dayalı Bakımın İmkanları	2
1.4. Sorun.....	2
1.5. Yaklaşım	3
1.6. Tezin Yapısı.....	3
2. LİTERATÜR TARAMASI	7
2.1. Bakım ve Bakım Stratejileri	7
2.2. CAN Bus.....	9
2.3. Öznitelik Seçimi	10
2.4. Bakım Yaklaşımları ve Makine öğrenimi	12
3. ÖN BİLGİ	17
3.1. Bakım.....	17
3.1.1. Düzeltici bakım	17
3.1.2. Önleyici bakım	18
3.1.3. Kestirimci bakım	19
3.1.4. Proaktif bakım	19
3.2. CAN Bus.....	19
3.3. Öznitelik Seçimi	21
3.3.1. Çizge yapısı üzerinde topluluk tespiti	23
3.3.2. Modülerite ölçümü	23
3.3.2.1. InfoMap.....	24
3.3.2.2. Fast greedy	25
3.3.2.3. Louvain algoritması	26
3.3.2.4. Leiden	29
3.4. Kestirimci Bakım ve Makine Öğrenmesi	30
3.4.1. Rastgele orman (RF)	31
3.4.2. Extreme gradient boosting yöntemi (XGBoost).....	34
3.4.3. Destek vektör makinesi (SVM).....	37
4. VERİ SETİ.....	41
4.1. Ham Veri	41
4.1.1. Veri seti özeti	41
4.1.2. Veri toplama	43
4.1.2.1. Gerçek zamanlı toplanan veriler	43
4.1.2.2. Geçmiş dönem verileri	44

4.2. Ham Veri Analizi.....	44
4.2.1. İstatiksel analiz	44
4.2.2. Kategorik değişkenlerin incelenmesi.....	45
4.2.2.1. Chi-Square testi ve Cramer's V istatistik tablosu ile ilişki analizi	47
4.2.3. Sürekli değişkenlerin analizi	50
4.2.3.1. Sürekli değişken dağılım grafikleri	50
4.2.4. Sürekli değişkenler arasındaki ilişkinin pearson korelasyon analizi ...	52
4.3. Veri Ön İşleme ve Anomali Analizi	55
4.3.1. Veri Sadeleştirme.....	55
4.3.2. Anomali Analizi.....	56
4.3.2.1. Hedef alarmların analizi	56
4.3.2.2. Değişkenlerin sapma gösterdiği zamanların belirlenmesi.....	56
4.3.2.3. Hedeflenen alarmlar anomali gösteren değişkenlerin ilişkilendirilmesi.....	57
4.3.2.4. Sonuçlar ve gözlemler	57
4.3.3. Sadeleştirilmiş veri üzerinde yapılan istatiksel analizler.....	64
5. ÇİZGE TABANLI ÖZNİTELİK SEÇİMİ.....	69
5.1. Değişkenler Arası İstatiksel İlişkilerin Çizge Yapısında Görselleştirilmesi ..	69
5.2. Çizge Yapısı Üzerinde Topluluk Tespiti.....	74
5.3. Ayırt Edici Özniteliklerin Seçimi.....	79
5.3.1. SPEC skorunun hesaplanması	79
5.3.2. K-Shell ayırıştırma yöntemi	81
5.3.2.1. Modifiye edilmiş K-Shell skorunun hesaplanması	82
5.3.3. Topluluklardan ayırt edici özniteliklerin belirlenmesi.....	83
6. EĞİTİM VE TEST VERİ SETİ HAZIRLIĞI	85
6.1. Özniteliklerin Düzenlenmesi	85
6.1.1. Veri ön işleme ve standardizasyon.....	85
6.1.2. Kübik spline interpolasyon	86
6.2. Yeniden Etiketleme	87
6.3. Dengeli Veri Kümesi Oluşturma	88
6.3.1. Eğitim ve test veri setlerinin oluşturulması	89
7. MODELLERİN EĞİTİLMESİ VE AÇIKLANABİLİRLİĞİ.....	91
7.1. Model Eğitimi ve Hiperparametre Optimizasyonu.....	91
7.1.1. SMOTEEN	92
7.1.2. Modellerin eğitimi	93
7.1.2.1. Referans otobüse ait veri seti ile eğitim	93
7.1.2.2. Diğer otobüslere ait veri setleri ile eğitim	94
7.1.2.3. Genelleştirilmiş model eğitimi	98
7.2. Modellerin Açıklanabilirliği.....	99
7.2.1. LIME (Local Interpretable Model-Agnostic Explanations)	100
7.3. Modellerin Genel Yorumlanabilirliği.....	102
7.4. Modellerin Kayıt Bazında Yorumlanabilirliği	103
8. DEĞERLENDİRME.....	107
9. GELECEK ÇALIŞMALAR.....	109
KAYNAKLAR.....	111
ÖZGEÇMİŞ.....	116

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1:	Araştırmada kullanılan makine öğrenmesi yaşam döngüsü ve kullanılan teknikler.....	6
Şekil 2.1:	Bakım stratejileri ref: EN 13306:2017.....	8
Şekil 2.2:	Bakım yaklaşımlarının dönüşümü.....	9
Şekil 2.3:	Saha ve arkadaşlarının [19] arıza teşhis (tahmini) akış şeması.....	13
Şekil 3.1:	Önleyici bakımda tipik maliyet.....	18
Şekil 3.2:	CAN Bus'ın gelişimi.....	20
Şekil 3.3:	OSI katmanları ile CAN katmanları arasındaki ilişki.....	21
Şekil 3.4:	Infomap çalışma prensibi.....	24
Şekil 3.5:	Louvain algoritması çalışma prensibi.....	27
Şekil 3.6:	Louvain ve Leiden algoritmalarının çalışma prensiplerinin karşılaştırması.....	29
Şekil 3.7:	Endüstri 4.0 Kestirimci bakım taksonomisi ve bu tez çalışmasında uygulanan teknikler.....	31
Şekil 3.8:	Random Forest ve XGBoost yöntemlerinin karşılaştırılması.....	35
Şekil 3.9:	SVM algoritması.....	37
Şekil 4.1:	Batarya paketlerinin minimum sıcaklık gösterme frekansı.....	47
Şekil 4.2:	Batarya paketlerinin maksimum sıcaklık gösterme frekansı.....	47
Şekil 4.3:	Cramér's V kategorik değişkenler arası ilişki.....	49
Şekil 4.4:	Dsh_AccelPedalPosition [%] değişkenin dağılım grafiği.....	51
Şekil 4.5:	Dsh high power (kW) değişkenin dağılım grafiği.....	52
Şekil 4.6:	Sürekli değişkenlerin korelasyon matrisi.....	54
Şekil 4.7:	Amber warning alarımının sistem değişkenleri anomali analizi.....	59
Şekil 4.8:	Amber warning alarımının batarya değişkenleri anomali analizi.....	60
Şekil 4.9:	DSH system warning alarımının sistem değişkenleri anomali analizi.....	61
Şekil 4.10:	DSH system warning alarımının batarya değişkenleri anomali analizi.....	62
Şekil 4.11:	PDU fault alarımının sistem ve batarya değişkenleri anomali analizi.....	63
Şekil 4.12:	BMS SOC alarmı sistem ve batarya değişkenleri anomali analizi.....	64
Şekil 5.1:	Sürekli değişkenler ilişki çizge yapısı.....	70
Şekil 5.2:	Kategorik değişkenler ilişki çizge yapısı.....	71
Şekil 5.3:	Leiden algoritması topluluk tespiti.....	79
Şekil 5.4:	Bir özellik çizge yapısının 3 kabuklu katmanlı yapısı [37].....	82

Şekil 7.1: SMOTEEN yönteminin çalışma mekanizması.....	92
Şekil 7.2: LIME çalışma prensibi.	101



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1: CAN standartları.....	10
Çizelge 3.1: Random Forest ve XGboost farklılıkları.	36
Çizelge 4.1: Çalışma kapsamında toplanan veri setleri.	42
Çizelge 4.2: Çalışma kapsamında belirlenen hedef değişken (Alarm) listesi.	42
Çizelge 4.3: 902 Değişkenin istatistiksel analiz tablosu (ilk 25 değişken).	45
Çizelge 4.4: Kategorik değişkenlerin frekans tablosu.	46
Çizelge 4.5: Standart sapması yüksek olan 48 sürekli değişkenin histogram grafikleri.	50
Çizelge 4.6: Sadeleştirilmiş sürekli değişkenlerin pearson korelasyon matrisi.....	65
Çizelge 4.7: Sadeleştirilmiş kategorik değişkenlerin cramer's V ilişki matrisi.	65
Çizelge 4.8: Sürekli değişkenler Anova F-Skor tablosu (İlk 25 değişken).	68
Çizelge 4.9: Kategorik değişkenler anova F-Skor tablosu (İlk 25 değişken).	68
Çizelge 5.1: Sürekli ve kategorik değişkenlerin ilişki çizge yapıları.	72
Çizelge 5.2: Kategorik değişken ve hedeflenen alarm ilişkilerinin görselleştirilmesi.	73
Çizelge 5.3: Sürekli değişken ve hedeflenen alarm ilişkilerinin görselleştirilmesi..	74
Çizelge 5.4: Topluluk tespiti algoritmaların performansı.....	75
Çizelge 5.5: Sürekli değişkenler çizge yapısı ve topluluk tespiti algoritmaları.	76
Çizelge 5.6: Geliştirilen algoritmanın çalışma akışı kesitleri.....	78
Çizelge 5.7: SPEC skoru ve K-Shell skoruna göre oluşturulan özellik kümeleri. ...	83
Çizelge 5.8: Kararlaştırılan öznitelik listesi.....	84
Çizelge 7.1: Örneklem parametreleri ve hiperparametre uzayı.	93
Çizelge 7.2: Referans veri seti üzerinde model başarımları.	94
Çizelge 7.3: Otobüs verilerinde hedeflenen alarmların sıklığı.	94
Çizelge 7.4: Referans otobüs verileri ile test sonuçları.	95
Çizelge 7.5: Otobüslere ait rafine edilmiş çizge yapıları.....	97
Çizelge 7.6: Diğer otobüslerin referans otobüse benzerliğinin spektral benzerlik tablosu.	98
Çizelge 7.7: Genelleştirilmiş model eğitimi sonuçları	99
Çizelge 7.8: Alarm tahminlerinin LIME ile açıklanması.	104
Çizelge 7.9: Rastgele orman modelinin performansı ve öznitelik önemi.....	105
Çizelge 7.10: Xgboost modelinin performansı ve öznitelik önemi.	106



ALGORİTMA LİSTESİ

	<u>Sayfa</u>
Algoritma 3.1: Fast Greedy algoritma adımları	26
Algoritma 3.2: Louvain algoritma adımları	28
Algoritma 3.3: Leiden algoritma adımları.	30
Algoritma 4.1: Anova F-testinin adımları.....	67
Algoritma 5.1: Geliştirilen dinamik eşik değerleri topluluk tespiti algoritması.	77
Algoritma 5.2: K-Shell ayrıştırma yöntemi adımları.	81
Algoritma 7.1: LIME' algoritma adımları	100





KISALTMALAR

ACC	: Doğruluk (Accuracy)
AI	: Yapay Zeka (Artificial Intelligence)
ANN	: Yapay Sinir Ağı (Artificial Neural Network)
CAN	: Kontrolcü Alan Ağı (Controller Area Network)
CCE	: Korelasyon Katsayısı (Correlation Coefficient)
CNN	: Konvolüsyonel Sinir Ağı (Convolutional Neural Network)
ECU	: Elektronik Kontrol Üniteleri (Electronic Control Units)
F1	: F1 Skoru (F1 Score)
FAM	: Özellik İlişkilendirme Haritası (Feature Association Map)
IoT	: Nesnelerin İnterneti (Internet of Things)
LCC	: Doğrusal Korelasyon Katsayıları (Linear Correlation Coefficient)
LIME	: Local Interpretable Model-Agnostic Explanations
LSTM	: Uzun Kısa Süreli Bellek (Long Short-Term Memory)
MDL	: Minimum Açıklama Uzunluğu (Minimum Description Length)
MI	: Karşılıklı Bilgi (Mutual Information) ve
ML	: Makine Öğrenmesi (Machine Learning)
MLPC	: Çok Katmanlı Algılayıcı Sınıflandırıcı (Multi-Layer Perceptron Classifier)
MTL	: Çoklu Öğrenme (Multi-Task Learning)
MVB	: Çok Fonksiyonlu Araç Veri yolu (Multi-Functional Vehicle Bus)
OOB	: Torba Dışı (Out-of-Bag)
PdM	: Öngörücü Bakım (Predictive Maintenance)
PRE	: Kesinlik (Precision)
REC	: Duyarlılık (Recall)
RF	: Rastgele Orman (Random Forest)
RNN	: Tekrarlayan Sinir Ağları (Recurrent Neural Network)
RUL	: Kalan Faydalı Ömür (Remaining Useful Life)
SMOTEEN	: Synthetic Minority Oversampling Technique-Edited Nearest Neighbours
SPEC	: Özellik Seçimi (Spectral Feature Selection)
SU	: Simetrik Belirsizlik (Symmetric Uncertainty)
SVM	: Destek Vektör Makinesi (Support Vector Machine)
XGBC	: Aşırı Gradyan Artırma Sınıflandırıcısı (Extreme Gradient Boosting Classifier)
XGBoost	: Aşırı Gradyan Artırma (eXtreme Gradient Boosting)



1. GİRİŞ

1.1. Tezin Konusu

Bu tez çalışması TÜBİTAK tarafından desteklenen 1711- 3227015 numaralı “Araç Verilerinin Yapay Zekâ ve Makine Öğrenmesi Yöntemleri ile Kestirimci Bakım Tahminleme ve Akıllı Yedek Parça Depo Yönetimi Projesi [1] kapsamında, projenin bir parçası olarak TEMSA MD9 Otobüslerine ait CAN Bus verileri kullanılarak gerçekleştirilmiştir.

Bu çalışmada CAN Bus verileri kullanılarak araçlardaki belli alarmların makine öğrenmesi teknikleri ile kestirim performansı araştırılmış, çizge tabanlı bir öznitelik seçme aracı geliştirilerek düzeltici, önleyici, kestirimci ve proaktif bakım yaklaşımlarının iyileştirilmesine yardımcı olacak şekilde öznitelikler arasındaki ilişkilerin görsel olarak yorumlanması imkânı sağlanmıştır.

1.2. Motivasyon

Mobley’e göre bakım maliyetleri toplam üretim maliyetlerinin (sektöre göre değişmekle beraber) %15’i ile %60’ı arasında önemli bir kısmını oluşturmaktadır [2]. Bakım, otobüs işletmeleri için de filo yönetiminde önemli bir maliyet unsuru olması [3] [4], enerji verimliliğinin sağlanması, karbon emisyonunun kontrol edilmesi, varlıkların değerinin korunması [5] ve karşılaşılan arızaların hizmet kesintisine yol açması sebebiyle müşteri memnuniyeti bakımından hayati derecede önemlidir. Endüstri 4.0 çağında, bakım stratejilerinin makine öğrenmesi yöntemleri ile optimize edilmesi hem maliyetleri düşürme hem de iş sürekliliğini sağlama açısından etkili olmaktadır [6] [7]. Bu çalışmada, elektrikli otobüslerden elde edilen CAN Bus verileri üzerinde makine öğrenmesi teknikleri ve öznitelik analizi uygulanarak, araçlardaki potansiyel arızalarla ilişkili alarmların önceden tahmin edilmesi hedeflenmektedir. Bunun yanında, elektrikli otobüsler çok sayıda bileşen ve sensör içeren ve büyük veri üreten karmaşık sistemlerdir. Bu durum, bileşenlerin birbirleriyle olan ilişkilerini anlamlandırmayı ve arıza alarmlarının ilişkili olduğu bileşenlerin tespitini zorlaştırmaktadır [8]. Bu çalışmada, gelişmiş öznitelik seçimi ve model açıklanabilirlik teknikleri kullanılarak bu karmaşıklık ele alınmıştır. Bu sayede, sadece

kestirim yapmakla kalınmayıp, aynı zamanda bu kestirimlerin arkasındaki nedenleri anlayarak sistemin işleyişi ve tasarımı hakkında değerli bilgiler elde edilmiştir.

1.3. Veriye Dayalı Bakımın İmkanları

Sistem ve cihazlarda elektronikleşmenin derinleşmesi, sistemlerde (gerek ürünlerde gerekse üretim sistemlerinde) anlık veri üretimini ve bu verilerin gerçek zamanda işlenmesini mümkün kılmaktadır [9]. Veri analizi ve makine öğrenmesi teknikleri düzeltici bakım ve önleyici bakım da dahil olmak üzere özellikle kestirimci bakımı ve proaktif bakımı çok daha etkin bir şekilde uygulama imkânı sağlamaktadır [10].

Gelişmiş endüstriyel elektronik, işlemci ve denetleyiciler [9] [11], IoT, bulut bilişim, yapay zeka, büyük veri ve sensör teknolojileri sayesinde bir sistemin işleyişi sırasında karşılaşılan anomaliler, sorunlar, siber saldırılar ve kullanıcıların kullanım tarzları tespit edilebilmekte ve bunlar hakkında kestirim yapılması mümkün olmaktadır [12] [13] [14]. Sistemin farklı özellikleri ve davranış şekillerine ilişkin verilerin öznitelik seçimi ve makine öğrenmesi teknolojileri ile işlenmesi sistemlerin işleyişi hakkında içgörü sağlamakta, sistemin iç işleyişi konusunda dışarıdan kolaylıkla görülemeyen iç dinamiklerin anlaşılması arıza teşhisi ve hata analizini iyileştirmekte, sistemin güvenilirliğinin iyileştirilmesine açık alanları işaret etmekte ve tasarım değişikliklerine yönelik çıktılar üretebilmektedir [15] [16] [17] .

Sistemin davranışları üzerinde kestirim yapabilme ve proaktif bakım faaliyetlerinde bulunma, tarihsel verilere, istatistiksel çıkarım yöntemlerine ve mühendislik yaklaşımlarına dayalı tahmin araçlarının geliştirilmesini gerektirmektedir [18].

1.4. Sorun

Makine öğrenmesi modellerinin başarımı büyük ölçüde kullanılan verinin kalitesine, öznitelik seçim sürecinin etkinliğine, probleme uygun modellerin seçilmesine ve makine öğrenmesi yaşam döngüsünün disiplinli bir şekilde uygulanmasına bağlı olmaktadır. Endüstriyel çalışmalarda kaliteli veriye erişim problemleri ile sıklıkla karşılaşılmaktadır [19] [7]. Eksik bilgiler, zaman serisinin oluşturulamaması, bakım tarihçelerinin olmaması, veri setlerindeki yapısal farklılıklar ve veri dönüşüm kayıpları bu başlık altında değerlendirilebilir. Veri boyutunun fazla olması (Örn. yaklaşık 900 özellik, 300 GB veri seti büyüklüğü), sistemin karmaşıklığı ile baş edebilmek ve model

sonuçlarını yorumlayabilmek için gerekli olan sistemin tasarımına ve iç dinamiklerine ilişkin alan uzmanlığına erişim zorlukları da dikkate alınması gereken risklerdir. Alan uzmanlığına erişim zorlukları, veri analizi sonuçlarının yorumlanmasını, ilgili faktörlerin derinlemesine anlaşılmasını, öznitelik seçimi ile ilgili hipotezlerin veya eyleme geçirilebilir içgörülerin formüle edilmesini kısıtlamaktadır [20] [21] [22] [23] [24].

1.5. Yaklaşım

Bu tez çalışmasında, elektrikli otobüslere yönelik bir kestirimci bakım (PdM) sistemi geliştirilmesi projesinin bir parçası olarak, CAN Bus verileri kullanılarak araçlardaki belli alarmların makine öğrenmesi teknikleri ile kestirim performansı araştırılmıştır. Çalışmada, proje yürütülürken, iki yıl süresince, beş farklı araçtan elde edilen ve farklı zamanlarda makine öğrenmesi sürecine aktarılan CAN Bus verileri kullanılmış olup, bu durumun yarattığı risklerle baş edebilmek ve daha etkin şekilde uzman görüşü alabilmek için çizge tabanlı bir öznitelik seçimi aracı geliştirilmiş, değişkenler arasındaki istatistiksel ilişkiler görselleştirilmiş ve makine öğrenmesi yaşam döngüsü olabildiğince disiplinli bir şekilde, iteratif ve çapraz doğrulamalara izin verecek bir şekilde uygulanmıştır [20]. Bu şekilde otobüslere ait veri setlerindeki farklılıkların modelleme performansı üzerindeki etkilerinin incelenebilmesi, çizge tabanlı öznitelik seçimi aracının çıktılarının ve modellerin açıklanabilirlik sonuçlarının sistemin iç dinamiklerine ilişkin içgörü sağlaması hedeflenmiştir. Literatürde sıklıkla kullanılan Filtreleme (Filter), Sarmalayıcı (Wrapper), Gömülü (Embedded) öznitelik seçimi yaklaşımları çoklu bileşen etkileşimlerini yakalamada Hibrit yaklaşımlara kıyasla daha başarısız olup iteratif olarak uzun sürede uygulanabilen ve modele bağımlı yaklaşımlardır [25] [26] [27] [28]. Bu çalışma kapsamında oluşturulan çizge tabanlı öznitelik seçimi aracı sayesinde bileşenlerin çoklu seviyede ilişkilerini yakalamak ve gözlemlenebilir hale getirmek amaçlanmıştır. Bu şekilde öznitelik seçimi aşamasının karmaşıklığı ve modele bağımlılığı azaltılmaya, yorumlanabilirliği ise arttırılmaya çalışılmıştır.

1.6. Tezin Yapısı

Bu tez çalışmasında, elektrikli otobüslerde kestirimci bakım için makine öğrenmesi ve çizge tabanlı öznitelik seçimi tekniklerinin kullanılması üzerine odaklanılmıştır.

Bunun yanında açıklanabilirlik teknikleri kullanılarak model başarımları analiz edilmiş ve sistem işleyişi için öngörüler elde edilmiştir. Tezin yapısı, Şekil 1.1'de gösterilen makine öğrenmesi yaşam döngüsü ve kullanılan tekniklere göre düzenlenmiştir.

Çalışma, birkaç ana bölümden oluşmaktadır. İlk bölüm olan **Giriş** kısmında, tezin konusu, bu araştırmayı motive eden unsurlar, veri odaklı bakımın sunduğu fırsatlar, araştırmacının çözmeyi amaçladığı sorunlar ve bu sorunlara yaklaşım ele alınmıştır.

Literatür Taraması bölümünde, bakım yaklaşımları, bakım yaklaşımlarının evrimi, kestirimci bakım (PdM) kavramları ve araçlarda kullanılan CAN Bus iletişim protokolü üzerine detaylı bir inceleme yapılmıştır. Bu bölümde ayrıca, öznitelik seçimi teknikleri ve çizge tabanlı öznitelik seçimi yöntemlerine dair kapsamlı bir literatür incelemesi sunulmuştur.

Ön Bilgiler bölümünde, bakım kavramları, CAN Bus iletişim protokolü, öznitelik seçimi yöntemleri ve çizge tabanlı topluluk tespiti algoritmalarının yanı sıra, kestirimci bakımda kullanılan makine öğrenmesi modelleri hakkında temel bilgiler verilmiştir.

Veri Seti bölümünde, elektrikli otobüslerden toplanan CAN Bus verileri tanıtılıp, veri toplama sürecine, veri setinin özelliklerine, istatistiksel analizlere (Kategorik değişkenler için Cramér's V testi, sürekli değişkenler için Pearson korelasyon analizi ve ANOVA F-testi) ve ham veri üzerinde gerçekleştirilen anomali analizine yer verilmiştir.

Çizge Tabanlı Öznitelik Seçimi bölümünde, istatistiksel ilişkilerin çizge yapısında görselleştirilmesi ve topluluk tespiti algoritmaları (InfoMap, Louvain, Fast Greedy, Leiden) kullanılarak daha genelleştirilmiş bir yaklaşım ile ayırt edici özellik alt kümelerinin belirlenmesi açıklanmıştır. SPEC skoru, modifiye edilmiş k-Shell skoru gibi metriklerin kullanılarak en ayırt edici öznitelikler seçilmesine ve ham veri üzerinde gerçekleştirilen boyut azaltımı yaklaşımına yer verilmiştir.

Eğitim ve Test Veri Seti Hazırlığı kısmında, özniteliklerin düzenlenmesi, verilerin yeniden etiketlenmesi ve dengeli veri kümelerinin oluşturulması işlemleri açıklanmıştır.

Model Eğitimi ve Açıklanabilirliği bölümünde, Destek Vektör Makinesi (SVM), Rastgele Orman (RF) ve XGBoost gibi farklı makine öğrenmesi modellerinin eğitimi, hiperparametre optimizasyonu ve LIME (Local Interpretable Model-Agnostic

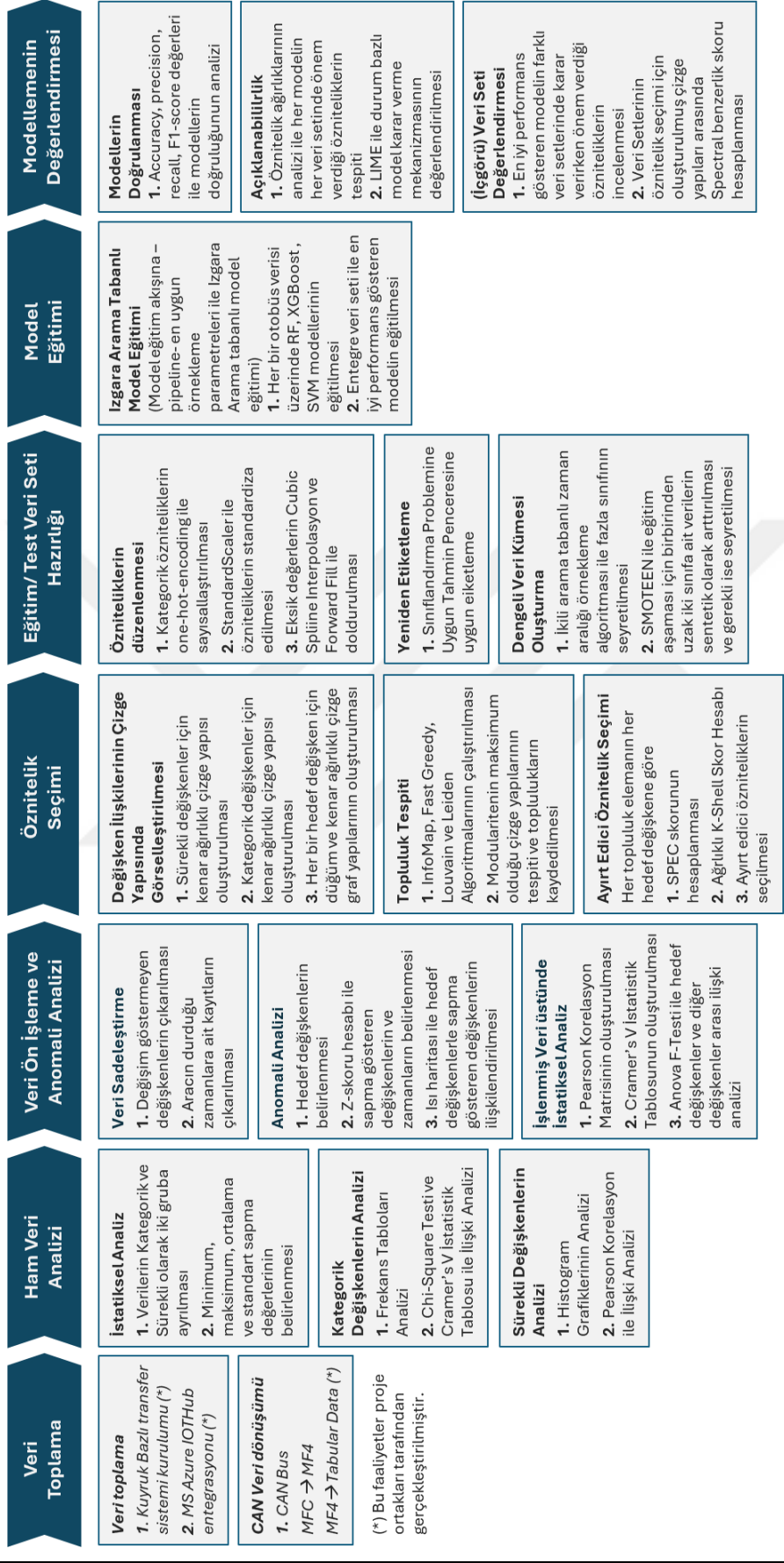
Explanations) aracının kullanımı ile modellerin açıklanabilirliği üzerine odaklanılmıştır.

Tezin **Değerlendirme** bölümünde, geliştirilen modellerin ve oluşturulan makine öğrenmesi yaşam döngüsünün performansının değerlendirilmesine ve sonuçların yorumlanmasına yer verilmiştir.

Gelecek Çalışmalar bölümünde ise, mevcut çalışmanın bulgularına dayalı olarak gelecekte yapılabilecek araştırmalar için öneriler sunulmuştur.



Tez çalışmasına uyarlanmış makine öğrenmesi yaşam döngüsü (ML pipeline) ve kullanılan teknikler_aie240724



Şekil 1.1: Araştırmada kullanılan makine öğrenmesi yaşam döngüsü ve kullanılan teknikler.

2. LİTERATÜR TARAMASI

Endüstriyel sistemlerde karşılaşılan arızalar iş kesintilerine, maliyet artışlarına ve güvenlik risklerine yol açmaktadır. Bu nedenle, sistemlerin işleyişini izlemek ve potansiyel arızaları önceden tahmin etmek büyük önem taşımaktadır. Son yıllarda kestirimci bakım (Predictive Maintenance - PdM) konusunda çok fazla çalışma yapıldığı görülmektedir [2].

Elektronik kontrol sistemleri, sensörler, IoT, bulut bilişim gibi endüstriyel sistemlerin bir bileşeni haline gelen teknolojiler, veri madenciliği, büyük veri, makine öğrenmesi, derin öğrenme teknikleri sistem bileşenlerinden elde edilen verilerin geçmiş zamanlı veya gerçek zamanlı olarak analiz edilmesini mümkün kılmaktadır. Bu şekilde sistem veya bileşenlerinde karşılaşılabilecek arızaları tahmin etmek, arızaların kök sebeplerini bulmak, planlı bakım süreçlerini optimize etmek ve sistemlerin bakım gereksinimlerini belirleyen içsel özelliklerini iyileştirebilmek mümkün olmaktadır.

Literatür taramasında, elektrikli otobüslere yönelik CAN Bus verileri ile gerçekleştirilen kestirimci bakım çalışmalarına odaklanılmıştır. Bu kapsamda elektrikli otobüs, filo yönetimi, CAN Bus verisi kullanımı veya uygulanacak makine öğrenmesi yaşam döngüsü ile ilgili benzer karakteristikler taşıyan çalışmalar da dikkate alınmıştır. Çalışmanın risklerine yönelik olarak çizge tabanlı öznitelik seçimi ve farklı makine öğrenmesi modellerinin performansları ve açıklanabilirlikleri incelenmiştir.

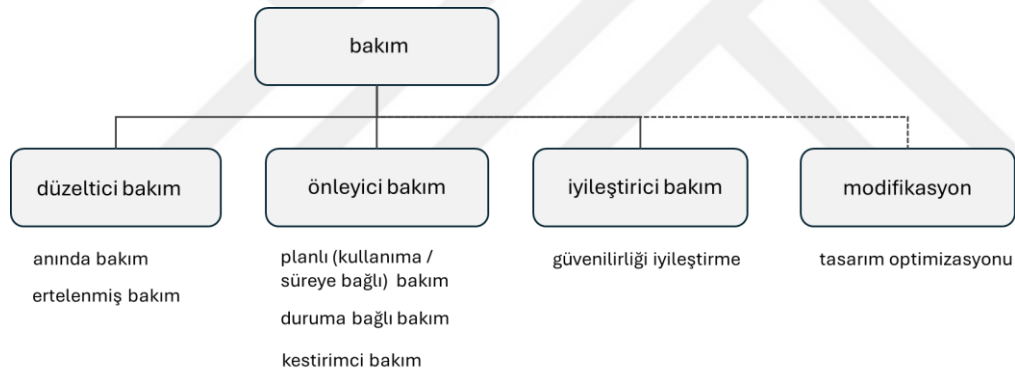
Bu bölümde, öncelikle bakım ve bakım stratejileri, CAN Bus ve öznitelik seçimi gibi temel kavramlar literatürde yer verilen açıklamaları ile ele alınarak, literatürde bakım yaklaşımları ve makine öğrenimi teknikleri bakımından öne çıkan çalışmalara yer verilmiştir.

2.1. Bakım ve Bakım Stratejileri

Çalışmada, öncelikle sistemlerin asgari maliyet ve asgari iş kesintisi ile performanslı ve güvenli bir şekilde işletiminde, bakımın önemi [2], bakım yaklaşımları (stratejileri) [25] [26], elektronikleşme ve veri kullanımına dayalı olarak bakım stratejilerindeki (Bakım 1.0'dan Bakım 4.0'a) gelişim [27] [28] [29] incelenmiştir.

Mobley [2]'e göre bakım, toplam üretim maliyetlerinin %15 ila %60'ını oluşturabilecek önemli bir gider kalemidir. Bakımın yetersiz veya etkisiz yapılması durumunda, beklenmedik ekipman arızaları, üretim kesintileri, maliyet artışları ve hatta güvenlik sorunları ortaya çıkabilir. Bu nedenle, Mobley'in çalışması, bakımın sadece bir maliyet unsuru olarak görülemeyeceğini, aynı zamanda üretim süreçlerinin verimliliğini ve güvenliğini sağlamak bakımından, özellikle endüstriyel ortamlardaki önemini vurgular.

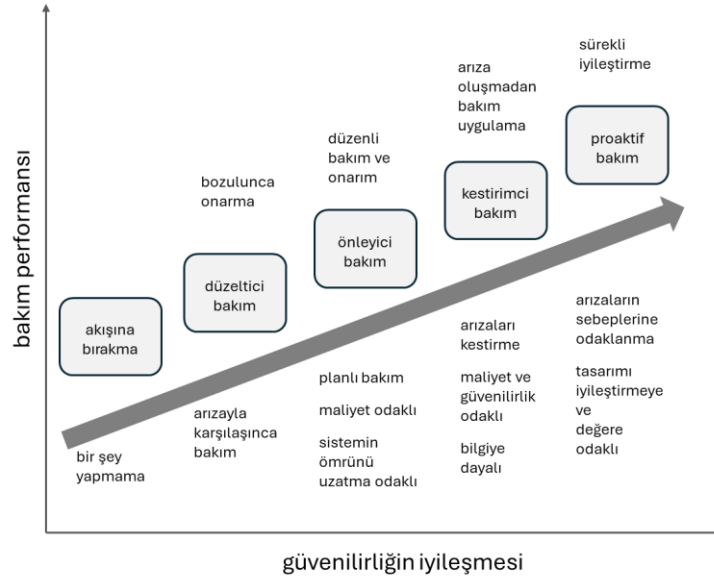
Avrupa Standardı EN 13306:2017'ye [25] göre, “Bakım, bir ögenin yaşam döngüsü boyunca, gerekli işlevi yerine getirebileceği bir durumda tutulması veya bu duruma geri getirilmesi amacıyla gerçekleştirilen tüm teknik, idari ve yönetsel eylemlerin bir kombinasyonudur.” Bu norma göre bakım yaklaşımları Şekil 2.1’ de görüldüğü üzere arıza ile karşılaşma anına göre, düzeltici (reaktif) bakım, önleyici bakım, kestirimci bakım, proaktif (iyileştirici ve modifikasyon) bakım şeklinde dört başlıkta ele alınmaktadır.



Şekil 2.1: Bakım stratejileri ref: EN 13306:2017.

Endüstri 4.0’a geçişle beraber bakım sürecinin gittikçe daha fazla sayısallaştığı; kabiliyetleri artan ve maliyetleri düşen elektronik sistemlerin tüm bakım yaklaşımları için daha fazla bilgi sunduğu ve akıllı bakım stratejilerine yönelimi artırdığı görülmektedir. [29]

Son 50 yılda bakım stratejilerinin gelişimi Bakım 1.0 dan Bakım 4.0’a uzanan bir şekilde tanımlanabilmekte [27] ve EN13306:2017 de yer alan bakım türlerinin Novakowski tarafından bahsedilen bakım stratejileri [30] Şekil 2.2’ de verilen bakım yaklaşımları dönüşüm grafiği ile örtüşmektedir.



Şekil 2.2: Bakım yaklaşımlarının dönüşümü.

Literatürde bakım stratejilerinin sınıflandırılmasına yönelik başka bir referans model çalışması Villiers tarafından [29], Ermonts-Holley'den uyarlanan yaklaşımdır. Bu yaklaşıma göre "reaktif (düzeltici) ve önleyici bakım", sistemle ilgili her hangi bir veri kullanılmaması sebebiyle ikisi birlikte ilk yaklaşımı oluşturmaktadır. Bundan sonraki sınıflandırmalar, sistem bileşenlerinin durumlarının izlendiği "duruma dayalı - izleme" ile sistemin tamamının ve bileşenler arasındaki ilişkilerin izlendiği "duruma dayalı - teşhis" yaklaşımlarıdır. Bunu takip eden yaklaşım sistemin arıza olasılıklarının tahmin edildiği ve olası çözümlerin önerilebildiği "kestirimci bakım" ve en gelişmiş bakım stratejisi olan akıllı bakımdır. Aynı dokümanda "akıllı bakım"ın Henke ve arkadaşları tarafından "dijital uygulamaları kullanarak ilgili mevcut üretim sistemini dikkate alarak, bakım önlemlerinin teknik ve ekonomik etkinliğini en üst düzeye çıkarma hedefiyle öğrenme odaklı, kendi kendini düzenleyen bakımdır" şeklinde tanımlandığı görülmektedir. Bakım stratejilerinin gelişmişliği veri kullanıma paralellik göstermektedir.

2.2. CAN Bus

Bakımın temel amacı, sistemlerin güvenilirliğini ve performansını en üst düzeye çıkarırken, aynı zamanda maliyetleri ve iş kesintilerini en aza indirmektir [2]. Bu amaçla, farklı bakım stratejileri geliştirilmiştir. Literatürde görüldüğü üzere veri kullanımına dayalı olarak düzeltici ve önleyici (planlı) bakımdan kestirimci ve proaktif bakım stratejilerine doğru bir evrim yaşanmaktadır. Bu evrimin merkezinde, modern

araç ve makinelerde yaygın olarak kullanılan elektronikleşme, sensör teknolojileri ve CAN Bus gibi endüstriyel haberleşme protokolleri yer almaktadır. CAN Bus otomotiv başta olmak üzere endüstride yaygın olarak kullanılmaktadır. 1980'li yıllarda Bosh tarafından geliştirilen bu seri haberleşme protokolü 1986 yılında ISO standardı olarak yayınlanmıştır [31]. Araçların çeşitli bileşenleri ve sensörleri arasında gerçek zamanlı veri alışverişini sağlayarak, sistemin sağlık durumu hakkında zengin bir bilgi kaynağı sunmaktadır. Bu verilerin makine öğrenmesi ve veri madenciliği teknikleri ile analizi, potansiyel arızaların erken tespitini ve bakım işlemlerinin optimize edilmesini mümkün kılmaktadır. Böylece, CAN Bus verileri, kestirimci bakım stratejilerinin uygulanmasında kritik bir rol oynayarak, endüstriyel sistemlerin daha verimli ve güvenilir bir şekilde çalışmasını sağlamaktadır. CAN Bus yıllar içerisinde daha hızlı ve esnek kabiliyetler kazanarak Çizelge 2.1'de görüleceği üzere endüstriyel bir standart ailesine dönüşmüştür.

Çizelge 2.1: CAN standartları.

ISO 11898_1:2015 Road vehicles — Controller area network (CAN) Part 1: Data link layer and physical signaling Ed.2
ISO 11898_3:2006 Road vehicles — Controller area network (CAN) Part 3: Low speed, fault tolerant, medium dependent interface
ISO 11898_4:2004 Road vehicles — Controller area network (CAN) Part 4: Time triggered communication
ISO 11898_5:2007 Road vehicles — Controller area network (CAN) Part 5: High speed medium access unit with low power mode
ISO 11898_6:2013 Road vehicles — Controller area network (CAN) Part 6: High speed medium access unit with selective wake up functionality
ISO 16845_1:2016 Road vehicles — Controller area network (CAN) conformance test plan Part 1: Data link layer and physical signaling
ISO 15765_2:2024 Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) Part 2: Transport protocol and network layer services
ISO 15765_5:2023 Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) Part 5: Specification for an in-vehicle network connected to the diagnostic link connector

Bu çalışma kapsamında CAN verisi ile üzerinde makine öğrenmesi ve derin öğrenme teknikleri uygulanan çeşitli (arıza teşhisi, anomali tespiti, enerji optimizasyonu, kalan cihaz ömrü kestirimi vb. amaçlı) araştırmalar incelenmiştir [19] [32] [33].

2.3. Öznitelik Seçimi

Kestirimci bakımın başarısı, karmaşık araç sistemlerinden elde edilen büyük miktardaki CAN Bus verilerinin etkili bir şekilde analiz edilmesine bağlıdır. Bu noktada, öznitelik seçimi, ilgili bilgileri ayıklamak ve makine öğrenimi modellerinin performansını optimize etmek için kritik bir rol oynar [34]. Öznitelik seçimi teknikleri, CAN Bus verilerindeki karmaşık ilişkileri ortaya çıkararak, sadece kestirim doğruluğunu artırmakla kalmaz, aynı zamanda sistemin davranışı ve potansiyel arıza

modları hakkında daha derin bir anlayış sağlar. Öznitelik seçimi, modellerin doğruluğunu ve hesaplama verimliliğini artırmak için etkisiz özelliklerin çıkarıldığı bir veri işleme faaliyeti olarak özetlenebilir [35]. Makine öğrenimi modellerinin performansını artırmak için literatürde Filtreleme (Filter), Sarmalayıcı (Wrapper) , Gömülü (Embedded) ve Hibrit yöntemler öne çıkmaktadır [36].

Hibrit yöntemler sınıfında olan çizge tabanlı yöntemler altta yatan topoloji yapısını ve oluşturulan özellik çizgesindeki düğümler arasındaki karmaşık ilişkileri kullanılabilir kılmada oldukça başarılı olmaktadır [37]. Projenin karakteristiği ve riskleri dikkate alınarak öznitelik seçimi konusuna özel önem verilmiştir.

Bhuyan ve arkadaşları tarafından [38],yapılan çalışmada karşılıklı bilgi (MI) ve doğrusal korelasyon katsayılarını (LCC) kullanarak sınıflandırma için en uygun özellik analitik olarak seçilerek, doğrusal ve doğrusal olmayan bağımlı veri özelliklerini dikkate alınmaktadır.

Peng ve arkadaşları ise tıbbi bir uygulama için iki aşamalı sarmalayıcı içeren daha doğru sonuç verecek bir modeli önermektedirler [39].

Potharaju ve arkadaşı tarafından yapılan çalışmada [40], azaltılmış özellik kümesini seçmek için çizge oluşturarak Simetrik Belirsizlik (SU) ve Korelasyon Katsayısı (CCE) kavramına dayanan yeni bir metodoloji önerilmiştir. Önerilen metodoloji ile önerilen özelliklerin CCE'leri ölçülerek özelliklerin en yüksek SU puanı dikkate alınarak sonlu sayıda grupta (kümeleme) yapılmaktadır. Her gruptan maksimum SU değerine sahip bir özellik seçilmekte ve aynı gruptaki diğer özellikler göz ardı edilmektedir.

Zhang tarafından yapılan çalışmada [41] yüksek dereceli korelasyonlardan özellik seçmeye odaklanan, ilgili özellikler arasındaki üçüncü ve hatta daha yüksek dereceden bağımlılıkları dikkate alabilen ve ilgili özellik alt kümesinin optimum boyutunu yakalayabilen, sırasıyla Çizge Tabanlı Bilgi Teorik Özellik Seçimi ve Hiper grafik Tabanlı Bilgi Teorik Özellik Seçimi olarak adlandırılan iki denetimli özellik seçimi yaklaşımı sunulmaktadır.

Schroeder ve arkadaşları tarafından yapılan çalışmada [42], hem özelliklerin önemini hem de korelasyonunu aynı anda dikkate alan yeni bir çizge tabanlı özellik seçim filtresi önerilmektedir. Çizge tabanlı özellik seçim filtresi, çizgenin maksimum kliklerine bir derecelendirme fonksiyonu uygulayarak bir alt küme önermektedir.

Değerlendirme, farklı temel özellik seçimi yaklaşımları ile önerilen yaklaşım arasında çoklu makine öğrenimi algoritmalarının ve veri kümelerinin doğruluğunun karşılaştırılmasına dayanmaktadır. Sonuçlar, önerilen yaklaşımın mevcut yaklaşımlara kıyasla yaklaşık %69'unda en yüksek doğruluğu sağladığını ve özellik sayısını azalttığını göstermektedir

Bu çalışmada kullanıldığımız çizge tabanlı hibrit öznitelik seçim modeli, Das ve arkadaşları tarafından yapılan çalışmada [43] önerilen çizge tabanlı teknik kullanan hibrit öznitelik seçim algoritmasına benzerlik göstermektedir. Das ve arkadaşları tarafından önerilen algoritma, temel olarak Özellik İlişkilendirme Haritası (FAM) kavramını kullanılmaktadır. Özellik alt kümesini üretmek için minimal tepe örtüsü ve maksimal bağımsız kümenin çizge teorisi ilkeleri kullanılmaktadır. Bu algoritma hem denetimli hem de denetimsiz sınıflandırma için geçerlidir. Önerilen algoritmanın performansı, çeşitli denetimli ve denetimsiz özellik seçim algoritmaları ile karşılaştırılmış ve daha iyi olduğu görülmüştür. Goswami ve arkadaşları tarafından yapılan diğer bir çalışmada [44] Özellik İlişkilendirme Haritası (FAM) ve tepe tırmanma (hill climbing) yöntemleri birleştirilerek (FAMHC) hibrit bir özellik seçim algoritması önerilmektedir.

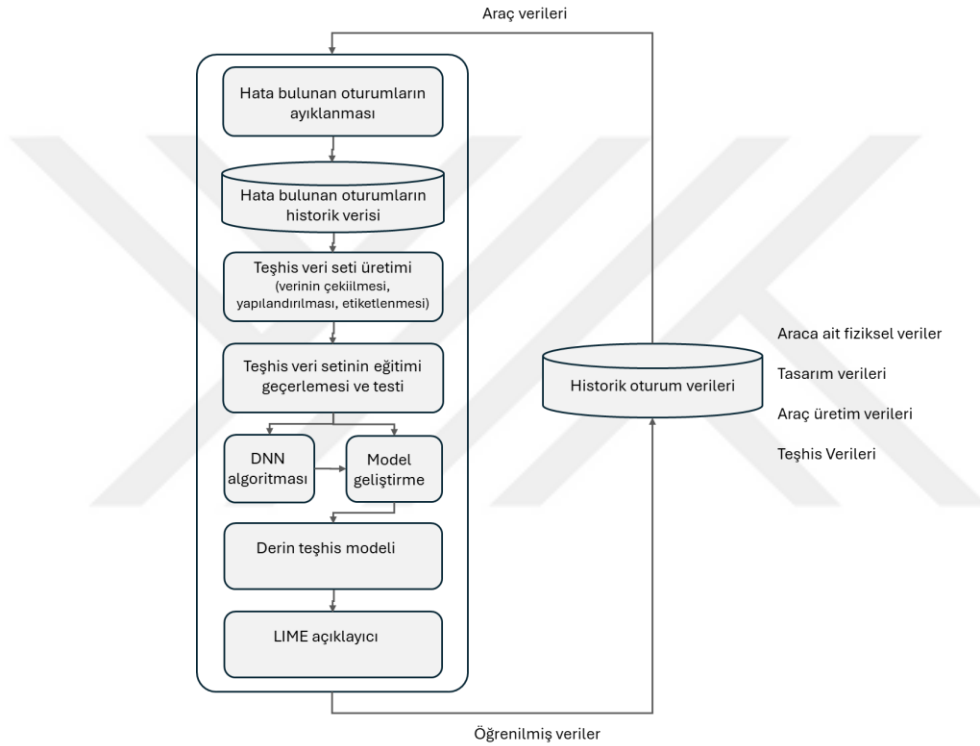
Li ve arkadaşlarınca [45] genellikle çevrim dışı analiz amacıyla kullanılan CAN verilerinin yansırı, çevrim içi arıza teşhisi için Çok Fonksiyonlu Araç Veriyolu (MVB) dalga karakteristiğinden çıkartılan öznitelikler ve karar ağacı yöntemi kullanıldığı görülmektedir.

2.4. Bakım Yaklaşımları ve Makine öğrenimi

Literatürde özellikle son yıllarda makine öğrenmesinin bakım süreçlerinin iyileştirilmesi amacıyla kullanıldığı çok sayıda çalışma yapıldığı gözlenmektedir. Yapılan çalışmaların önemli bir kısmı kestirimci bakım ile ilgili olmakla beraber, çalışmaların çıktıları itibarıyla düzeltici, önleyici ve proaktif bakım yaklaşımlarını da desteklediği görülmektedir.

Düzeltilici bakıma (Bakım 1.0) yönelik makine öğrenmesi çalışmalarına örnek olarak; makine öğrenmesi algoritmaları ile tekrarlayan arıza kalıplarını tespit ederek müdahale süreçlerini iyileştirmek, hatanın yerini tespit etmek, arıza oluşumuna sebep olan faktörlere yönelik kök neden analizi yapabilmek mümkün olmaktadır. Saha ve

arkadaşları tarafından yapılan çalışmada [19] yol ve arazi araçlarındaki sorunların etkili bir şekilde teşhis edilmesi ve hızlı bir şekilde çözülmesi amaçlanmıştır. Çalışmada yeni bir konsept olarak tanımlanan “bilişsel diyagnostik” (cognitive diagnostic) çözümüne ulaşmak için yapay zekâ (AI) Birleşik Diyagnostik Hizmetleri (UDS-Evrensel Teşhis Servisleri) ile birlikte kullanılmıştır. Bu yaklaşımla, makine öğrenimi, geleneksel teşhis protokollerine entegre edilerek hem düzeltici hem de kestirimci bakıma destek verilmesi; azaltılmış arıza süresi ile iyileştirilmiş müşteri memnuniyeti hedeflenmiştir. Çalışmanın akış şeması Şekil 2.3’te görülmektedir.



Şekil 2.3: Saha ve arkadaşlarının [19] arıza teşhis (tahmini) akış şeması.

Çalışmada Motor Yönetim Sistemi (EMS) için arıza tahmininin (teşhis) yapılması hedeflenmiştir. Bu amaçla aracın CAN Bus üzerinden sensör verileri hex kodda toplanarak veri dönüşümü yapılmıştır. İlgili sensör verileri çıkarılmış ve motorun çalışmasının dinamik doğasını yansıtan bir zaman serisi oluşturulmuştur. CAN Bus üzerinden farklı frekanslarda alınan sensör bilgileri uyumlandırılmıştır. Sonraki adımda, sensör verileri iki kategoriye manuel olarak etiketlenmiştir. Bu etiketli veriler daha sonra tahmine dayalı modelleme için Çok Katmanlı Algılayıcı Sınıflandırıcı (MLPC) ve Extreme Gradient Boosting Classifier (XGBC) modellerine beslenmiştir. Her iki modelin Motor Yönetim Sistemi arızalarını tahmin etme performansı karşılaştırılmıştır. MLPC ve XGBC'nin eğitimi ve testi, veri kümesi her seferinde

rastgele 8:2 eğitim ve test kümelerine bölünerek on iterasyon üzerinden gerçekleştirilmiştir. MLPC ortalama %89 olmak üzere %87 ile %91 arasında değişen doğruluk gösterirken, XGBC ortalama %91 olmak üzere %87 ile %94 arasında doğruluk göstermiştir. Çalışmanın değerlendirme kısmında modellerin daha büyük veri kümeleri üzerinde eğitilerek sistemin performansı ve davranışı hakkında içgörüler sunulabileceği ve böylece araç sistemleri üzerinde (bakım stratejilerinde) daha fazla iyileştirme ve geliştirmenin önünün açılabilceği belirtilmiştir.

Önleyici Bakıma (Bakım 2.0) yönelik makine öğrenmesi çalışmaları ile ilgili olarak; makine öğrenimi modelleri ile ekipmanların kullanım ömrünü ve bakım gereksinimlerini tahmin edebilmek mümkün görünmektedir. Öznitelik seçimi ile, hangi değişkenlerin bakım ihtiyaçlarını en iyi şekilde tahmin ettiğini belirlemek ve mevcut kullanıma veya süreye dayalı bakım planlarını optimize edebilme imkânı sağlanmaktadır. Bu şekilde gereksiz bakım işlemlerinin önlenmesi ve bakım maliyetlerinin düşürülmesi, ekipmanların kullanım ömrünün uzatılması, planlı bakım faaliyetleri sayesinde operasyonel kesintilerin minimize edilmesi mümkündür.

Mardukova ve arkadaşları tarafından [46], CAN Bus veri analizi ile ilgili çalışmalar (i) sınıflandırma ve regresyon teknikleri aracılığıyla bir aracın gelecekteki kullanım seviyesini tahmin etmeye yönelik çalışmalar, (ii) kümeleme teknikleri kullanarak benzer özelliklere sahip araçları belirleme ve (iii) anomali tespit yöntemlerine dayalı olarak belirli araç bileşenlerinin arızalarını belirleme şeklinde sınıflandırılmıştır. Bu çalışmada bir iş makinesi filusunda ertesi günkü kullanım seviyesini ve bakıma kalan süreyi tahmin etmek ve planlı bakım takvimlemesini optimize edebilmek hedeflenmiştir. Çalışmada inşaat araçlarından elde edilen CAN Bus verilerine makine öğrenimi teknikleri uygulanmıştır. Filodaki her bir araç için bir sonraki gün kullanım seviyesi ve bir sonraki önleyici bakıma kadar kalan günlerin tahmini amaçlanmıştır. Makine öğrenimi modellerinin eğitimini engelleyen geçmiş veri eksikliği ve önerilen çözümlerin genelliğini sınırlayan filo araçlarının yüksek heterojenliği gibi iki araştırma sorunu ele alınmıştır. Çeşitli şantiyelerde toplanan gerçek araç kullanım verileri üzerinde elde edilen sonuçlar; (i) Doğrusal regresyon modellerinin temel yöntemlerden daha iyi performans göstermekte beraber kullanılan araç modellerinin çeşitliliğinden olumsuz etkilendiğini, (ii) Random Forest ve Gradient Boosting gibi topluluk yöntemlerinin, yeterli miktarda eğitim verisi olduğunda araç başına tahminlerde etkili olduğunu, (iii) zamansal araç kullanım tekrarlarını tanımlayan

bağlamsal özelliklerin, tek değişkenli model performansını iyileştirmeye yardımcı olduğunu göstermiştir. Makalede bu çalışmanın sonunda uygulamanın inşaat firması tarafından kullanıma alındığı belirtilmektedir.

Kestirimci bakıma (Bakım3.0 / Bakım 4.0) yönelik makine öğrenmesi çalışmaları ile ilgili olarak; sistem üzerinden elde edilen gerçek zamanlı sensör verileri ve geçmiş bakım kayıtları kullanılarak arıza tahmin modelleri oluşturulabilmektedir. Makine öğrenimi algoritmaları arızaların önceden tahmin edilmesini ve sonraki bakım zamanı kestirimini yapabilmekte, bakım maliyetlerinin ve ekipmanların beklenmedik duruş sürelerinin en aza inmesini sağlayabilmektedir. Arızanın kestirimi için verilerdeki paternlerin tespit edilmesi, arıza koşullarının belirlenmesi ve arızanın gerçek zamanlı olarak tanımlanması gerekmektedir [47].

Bu çalışmaların bir kısmı belirli kullanım alanlarına özgü projeler iken, bir kısmı ise daha genel amaçlı projeler olarak göze çarpmaktadır. Bu projelerde genel olarak denetimli makine öğrenmesi tekniklerinin kullanıldığı, veri kalitesinde ve miktarında sorunlarla karşılaşıldığı [4], [5], çalışmaların genellikle elektronik kontrol biriminin arıza teşhisi [4], cihaz ömrü veya sonraki bakım için gereken süre kestirimi [31] [32], yakıt tüketimi kestirimi [5], otobüs tekerlek basıncı izlenmesi [33] veya elektrikli otobüslerde enerji tüketimi optimizasyonu [20] gibi daraltılmış hedeflere odaklı olarak gerçekleştirildiği görülmektedir. Bu tez çalışmasında hedef değişkenler (üretici tarafından) sürücü konsol alarmları, fren alarmları ve batarya alarmları olarak belirlenmiştir.

Kestirimci bakım konusunda yürütülen çalışmalara başka bir örnek olarak; Züfle ve arkadaşları [12] tarafından endüstriyel makinelerin arıza süresi tahmini için uçtan uca genel bir PdM metodolojisi önerilmektedir. Belirli makineler için özel olarak tasarlanmış diğer arıza süresi tahmin yaklaşımlarının aksine, önerilen metodoloji evrensel bir sensör işleme ve integral değerlere dayalı özellik çıkarma yaklaşımını benimsemiştir. Çalışma özellik dönüşümü, özellik seçimi, ayarlanabilir hedef, sınıf etiketleme ve farklı makine öğrenme modellerinin hiperparametre optimizasyonu ile eğitimini içermektedir. Önerilen metodoloji sensör bilgisinden arıza zamanı tahminine kadar uçtan uca bir yaklaşım sağlamaktadır.

Çınar ve arkadaşları tarafından yapılan tarama çalışmasına [48] göre, kestirimci bakım amaçlı makine öğrenmesi çalışmalarında en çok kullanılan algoritmalar SVM, RF ve

ANN'dir. Bunlar arasında en yaygın kullanılan ise rastgele orman (RF) algoritması olarak belirtilmektedir.

Kestirimci bakım konusunda ana amaçlardan biri izlenen bileşenin Kalan Faydalı Ömrünü (RUL) tahmin etmektir. Bunun için sistemin mevcut işleyişi ve geçmiş operasyon profili göz önüne alınarak bir arıza meydana gelmeden önce ne kadar zaman kaldığının tahmin edilmesi gerekmektedir. Kim ve arkadaşı [49] tarafından yapılan çalışmada RUL değerini ve sistemin sağlık durumunu eş zamanlı olarak tahmin etmek için bir Çoklu Öğrenme (MTL -Multi Task Learning) modeli oluşturulmuştur. MTL birden fazla görevden gelen birleşik bilgileri yansıtmak için büyük ağlar gerektirdiği ve yüksek hesaplama karmaşıklığı dezavantajı olduğu için CNN mimarisi ile gerçekleştirilmiştir.

Literatür taramasında makine öğrenmesi tekniklerinin bakım stratejilerindeki önemini ve bu alandaki yaklaşımların sınırlılıklarını açıkça ortaya koymuştur. Bunun yanında, çizge tabanlı öznitelik seçimi, makine öğrenmesi ve açıklanabilirlik yöntemlerinin kestirimci bakım süreçlerinde etkin bir şekilde kullanılabileceği vurgulanmıştır. Ancak, literatürdeki çalışmalar incelendiğinde, bu yöntemlerin birleşik bir şekilde gerçek büyük ve karmaşık bir üzerinde nasıl uygulanacağına dair detaylı bir rehber bulunmamıştır. Bu bağlamda, bu tez çalışması, literatürdeki bilgi birikimini temel alarak, elektrikli otobüslerde kestirimci bakım için çizge tabanlı öznitelik seçimi, makine öğrenmesi ve model açıklanabilirliği yöntemlerinin bir araya getirildiği genelleştirilmiş bir yaklaşım geliştirmeyi hedeflemiştir. Aşağıdaki bölümde, bu yaklaşıma dair temel bilgiler ve bu çalışmada kullanılan yöntemler detaylandırılmıştır.

3. ÖN BİLGİ

Bu bölümde, tezin devamında kullanılan yöntemlerin ve kavramların daha iyi anlaşılması için gerekli olan temel teknik bilgiler sunulmuştur. Kestirimci bakımın temelini oluşturan bakım stratejileri ve bu stratejilerin endüstriyel evrimi hakkında genel bir bakış sağlanarak çalışmanın veri kaynağı olan CAN Bus haberleşme protokolü, veri analizi sürecinde kullanılan öznitelik seçimi yöntemleri ve çizge tabanlı topluluk tespiti algoritmaları detaylı bir şekilde açıklanmıştır. Son olarak, kestirimci bakımda sıklıkla kullanılan ve bu tez çalışmasında oluşturulan makine öğrenmesi modellerinden Rastgele Orman (RF), Extreme Gradient Boosting (XGBoost) ve Destek Vektör Makinesi (SVM) hakkında bilgi verilmiştir.

3.1. Bakım

Avrupa Standardı EN 13306:2017'ye [25] göre, “Bakım, bir öğenin yaşam döngüsü boyunca, gerekli işlevi yerine getirebileceği bir durumda tutulması veya bu duruma geri getirilmesi amacıyla gerçekleştirilen tüm teknik, idari ve yönetsel eylemlerin bir kombinasyonudur.” Bu norma göre bakım yaklaşımları Şekil 2.1’ de görüldüğü üzere arıza ile karşılaşma anına göre, düzeltici (reaktif) bakım, önleyici bakım, kestirimci bakım, proaktif (iyileştirici ve modifikasyon) bakım şeklinde dört başlıkta ele alınmaktadır.

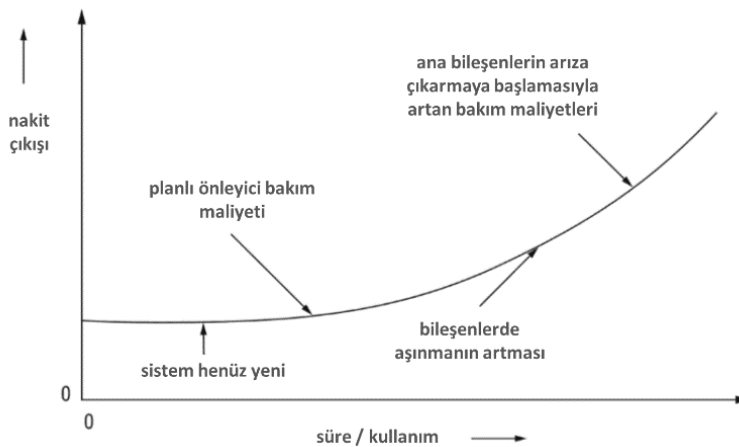
3.1.1. Düzeltici bakım

Bu yaklaşıma göre bir bileşen arıza yaptığında bakım görevinin uygulanması söz konusudur. Düzeltici bakım “run to failure” veya reaktif yaklaşım olarak da isimlendirilmekte olup, temel prensip bozulana kadar sisteme dokunmamaktır. Bu yaklaşımın iş kesintisi, olası sorunun daha büyük ve maliyetli sorunları tetiklemesi, beklenmeyen zamanda ortaya çıkıp pahalı bir çaba gerektirmesi gibi dezavantajları bulunmaktadır [7]. Arıza ortaya çıktıktan sonra düşük maliyetle, doğru ve hızlı bir teşhis ve düzeltme uygulanabilmesi bu yaklaşımın öne çıkan performans kriterleridir.

3.1.2. Önleyici bakım

Önleyici bakım, arıza ve iş kesintisi olasılığını azaltmak için düzenli ve rutin olarak gerçekleştirilen periyodik bakımı ve duruma dayalı bakımı kapsamaktadır. Bu şekilde beklenmedik arızaların önlenmesi için ekipman hala çalışır durumdayken önleyici bakım işlemi gerçekleştirilmektedir. Önleyici bakım stratejisi, düzeltici bakım ile kestirime dayalı bakım arasında yer alan yaygın olarak kullanılan bir yaklaşımdır. [26] Önleyici bakım kapsamındaki periyodik bakımın etkinliği süre veya kullanım miktarına bağlı olarak rutin bakım periyotlarının optimum şekilde ayarlanmasına bağlıdır. Diğer bir önleyici bakım uygulaması duruma dayalı bakımdır. Buna göre sistemin durumu izlenerek veya belli göstergelere ve arıza işaretlerine bakarak bakım uygulanmaktadır.

Periyodik bakımda endüstriyel makineler gibi sistemler düzenli aralıklarla bakımdan geçirilirken, duruma dayalı bakımda belirli göstergeler için eşik değerler tanımlanmakta ve bu değerler aşıldığında bakım işlemleri başlatılmaktadır [50]. Periyodik bakımın genellikle personel ve malzeme israfına yol açtığı, birçok durumda bakımın ertelenebilir olduğu belirtilmektedir. Mobley fabrikalarda bakım maliyetlerinin işletme maliyetlerinde önemli bir pay aldığını, sektöre göre değişmekle birlikte ürün maliyetinin (gıda sektöründe %15, daha ağır üretimlerde %60'a varan) önemli bir kısmını oluşturduğunu belirtmekte ve sistemler yaşlanıp yıprandıkça önleyici bakım maliyetlerinin arttığına işaret etmektedir [2].



Şekil 3.1: Önleyici bakımda tipik maliyet [2].

Duruma bağlı bakımda ekipmanın performansını ve durumunu izlemek için durum izleme araçları ve teknikleri kullanılmakta, belirli eşikler veya parametreler ihlal

edildiğinde bakım gerçekleştirilmektedir. Ancak duruma bağlı bakım değerlerini tanımlamak için yüksek düzeyde uzman bilgisi gerekmektedir [51].

3.1.3. Kestirimci bakım

Kestirimci bakımda daha önceki izleme verileri analiz edilerek ve önceki makine arızalarından ders çıkarılarak bir bileşenin ne zaman bozulacağı tahmin edilerek, arıza oluşmadan, en doğru zamanda ve asgari işletim maliyetiyle bakım yapılması mümkün olmaktadır. Gerçek zamanlı bir analiz ve izleme işlemi, sistem bileşeninin performansıyla ilgili tüm değerlerin kontrol edilmesine olanak tanımaktadır.

Endüstride kestirimci bakım uygulamalarına yönelimin maliyetleri %30'a kadar azaltılabileceği ve önleyici bakıma kıyasla arızaları %75'e kadar azaltılabileceği ifade edilmektedir [29]. Kestirimci ve proaktif bakım gerçek zamanlı algılama ve kapsamlı veri toplama gerektirmekte olup çoğunlukla durum tabanlı bakımın evrimsel bir adımı olarak değerlendirilebilir [52]. Kestirimci bakımın başarısı bir anlamda duruma dayalı bakımın (sensör, veri toplama, veri işleme değerlendirme kabiliyetlerine) yetkinliğine bağlıdır.

3.1.4. Proaktif bakım

Proaktif bakım iyileştirici veya modifiye edici bakım gibi de sınıflandırılabilir. Bu yaklaşım, sistem üzerinden gerçek zamanda elde edilen verilerin gerek gerçek zamanda gerekse historik analizi sonucu, bakım ihtiyacını azaltmaya ve yönetmeye yönelik iyileştirmelerin sistem tasarımı üzerinde ve diğer kök sebepler üzerinde sürdürülebilir bir şekilde uygulanmasına dayalıdır.

3.2. CAN Bus

CAN Bus başta otomotiv olmak üzere denizcilik ve endüstri sektörlerinde bir sistem veya platformdaki cihazların ve sensörlerin güvenilir ve verimli bir şekilde veri alışverişinde bulunmasını sağlayan seri bir iletişim protokolüdür. CAN Bus, verimliliği ve esnekliği nedeniyle modern otomotiv endüstrisinin ayrılmaz bir parçası haline gelmiştir. Bir araç içindeki çeşitli Elektronik Kontrol Üniteleri (ECU'lar) ve sensörler arasındaki iletişimi kolaylaştırmak için kullanılır ve çok sayıda alt sistemin sorunsuz entegrasyonunu sağlamaktadır. CAN Bus'ın otomotiv sistemlerindeki temel

uygulamalarından bazıları motor kontrolü, güvenlik sistemleri, gövde kontrolü ve bilgi-eğlence sistemlerini içermektedir.

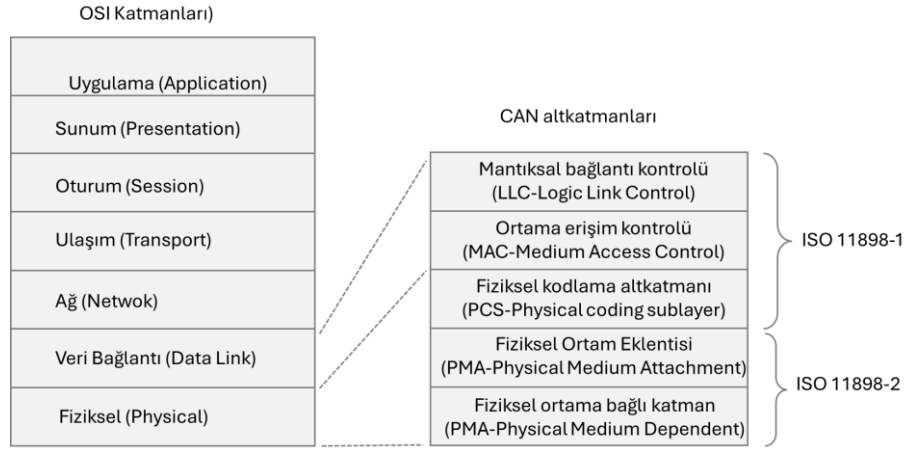
CAN Bus ortak bir iletişim protokolü kullanarak kablolamayı basitleştirmekte ve aracın elektronik mimarisinin karmaşıklığını azaltarak daha düşük ağırlık, daha düşük üretim maliyetleri ve daha kolay teşhis sağlamaktadır.

CAN Bus ilk olarak 1980 yılında Bosch tarafından otomotiv uygulamaları için tasarlanmış, 1993 yılında ISO 11898 uluslararası bir standart olarak yayınlanmıştır Bkz Tablo-XX . 2004 yılında, geleneksel CAN Bus ağlarından daha yüksek veri hızları ve daha büyük yükler sağlayan CAN FD'nin (Esnek Veri Hızı) tanıtımı yapılmıştır. 2015 yılında, ISO 16845:2015' hem klasik CAN hem de CAN FD protokollerini uygulayan cihazlar için bir uygunluk test planı olarak benimsenmiştir.



Şekil 3.2: CAN Bus'ın gelişimi.

CAN; (i) çok yöneticili öncelik tabanlı veri yolu erişimine (multi-master priority-based bus access), (ii) yıkıcı olmayan içerik tabanlı uyuşmaya (arbitration), (iii) çoklu yayın (broadcast) ve kabul filtrelemesi ile çok noktaya yayın özelliklerine, (iv) uzaktan veri talebi, (v) yapılandırma esnekliği, (vi) ağ çapında veri tutarlılığı, (vii) hata algılama özelliği, (viii) uyuşmazlık olan, onaylanmayan veya iletim sırasında hatalar nedeniyle yok olan çerçevelerin otomatik olarak yeniden iletimi özelliklerine sahiptir. [31]



Şekil 3.3: OSI katmanları ile CAN katmanları arasındaki ilişki.

3.3. Öznitelik Seçimi

Makine öğrenimi modellerinin performansını artırmak için literatürde Filtreleme (Filter), Sarmalama (Wrapper), Gömülü (Embedded) ve Hibrit Yöntemler başta olmak üzere birçok özellik seçimi yöntemi üzerinde durulmuştur. Bu yöntemlerle, verinin karmaşıklığını azaltarak, modellerin aşırı uyumunu (overfitting) önlemek ve modelin genelleme yeteneğini arttırmak amaçlanmaktadır. Ancak, hangi özellik seçimi yönteminin kullanılacağı, veri setine, modelin türüne, hesaplama kaynaklarına ve zamana bağlıdır. Bununla beraber her yöntemin kendine özgü avantajları ve dezavantajları bulunmaktadır [53].

Filtreleme yöntemlerinde özelliklerin önemini belirlemek için istatistiksel ölçütler kullanılmaktadır. Bu ölçütler, özelliklerin hedef değişkenle ya da ikili ilişkilerini değerlendirmeye yardımcı olmaktadır. Basittirler ve genellikle elde edilen sonuçların anlaşılması kolaydır, ancak bu yöntemler sadece istatistiksel tabanlı değerlendirme yapmaktadırlar ve çoklu değişkenlerin arasındaki etkileşimler dikkate alınmamaktadır [54].

Sarmalama yöntemlerinde ise rastgele seçilen bir özellik kümesi ile belirlenen makine öğrenmesi modeli eğitilmekte ve performansı değerlendirilmektedir. Değerlendirme sonrası ilk başta seçilen özellik kümesi üzerinden ekleme veya çıkarma işlemi yapılmaktadır. Bu süreç maksimum özellik sayısına ulaşıncaya kadar ya da model performansında herhangi bir artış gözlenmediği ana kadar iteratif olarak sürdürülmektedir. Bu yöntemle kullanılan model bazında en ayırt edici özellik seçimi

gerçekleştirilse bile süreç genelleştirilemez, modele bağımlıdır ve hesaplama açısından maliyetlidir [55].

Gömülü yöntemlerde ise, öznitelik seçimi modelin eğitim aşmasının gömülü bir parçası olarak gerçekleştirilir. Tüm öznitelikler makine öğrenmesi modeline girdi olarak verilir ve modelin eğitim süreci boyunca özelliklerin katsayıları veya ağırlıkları düzenli olarak güncellenir. Bu süreçte, model için ayırt edici olamayan özelliklerin katsayıları azalırken önemli özelliklerin katsayıları artar. Eğitim sonunda modelin kararında etkili olan ve olmayan değişkenler belirlenebilir ancak, bu yöntemlerde özellik seçimi modelin tahmin başarımına doğrudan bağlı olmakla beraber modelin karar mekanizmasını takip etmek ve görselleştirmek zorlaşır ve kapalı kutu gerçekleşen özellik seçim süreci doğrulanamaz [55].

Bu yöntemlerin yanı sıra son çalışmalarda özellik seçimi kapsamında Genetik Algoritmalar ve Karınca Kolonisi optimizasyonu kullanılmaktadır. Bu yöntemlerin özellikle büyük ve karmaşık veri setlerinde etkili olabildiği ve farklı problem türlerine uyarlanabildiği görülmüştür [56] [44]. Ancak, bu yöntemlerin hesaplama maliyetleri yüksektir ve yerel optimuma takılma riski bulunmaktadır.

Literatür araştırmasında kompleks ve karmaşık verilerinin analizinde, genellikle Hibrit Özellik Seçimi yaklaşımlarından biri olan Çizge Tabanlı özellik seçimi (Graph-Based Feature Selection) öne çıktığı görülmektedir [37] [43] [44]. Çizge tabanlı özellik seçimi, veriler arasındaki karmaşık ilişkileri çizge teorisi prensipleri ile temsil etmektedir. Veri setindeki her bir özellik bir düğüm (node) olarak temsil edilirken, özellikler arasındaki ilişkiler ise düğümlerin arasındaki kenarlar (edge) ile ifade edilmektedir. Bu yöntem, özellikle elektrikli otobüslerin CAN Bus verisi gibi yüksek boyutlu ve karmaşık veri setlerinde, özellikler arasındaki karmaşık ve doğrusal olmayan ilişkilerin tespit edilmesinde başarılıdır. Bunun yanında, özelliklerin önemini ve aralarındaki ilişkileri görselleştirmeyi dayalı bir yöntem olduğu için yorumlanması kolaydır. Gürültü verilerine karşı daha dayanıklıdır, büyük ve farklı türden veri içeren veri setlerine uygulanabilirler.

Bu tez çalışmasında referans elektrikli otobüsten toplanan yaklaşık 900 tanımsız potansiyel öznitelik içeren 300 GB büyüklüğünde ham veri setindeki özellikler arasında en ilgili olanların (ön) seçimini yapabilmek, modelin karmaşıklığını azaltmak, eğitim verisindeki gürültüyü azaltmak, aşırı uyumun önüne geçmek, model

yorumlanabilirliğini artırmak, verideki temel ilişkileri ve etki faktörlerini görselleştirerek uzman görüşü alınabilir hale getirmek, proje kapsamında sınırlı olan işlemci kaynağını etkin kullanabilmek için hibrit [27] [28] ve çizge tabanlı bir öznitelik seçimi uygulanmıştır.

3.3.1. Çizge yapısı üzerinde topluluk tespiti

Bu çalışmada, değişkenler arasındaki istatistiksel ilişkilerin ağırlıklı çizge yapısında temsil edilmiş ve sonrasında alakalı özellik alt kümelerinin tespiti için InfoMap (optimizasyon), Fast Greedy (açgözlü) , Louvain ve Leiden (sezgisel) topluluk tespit algoritmaları kullanılmış ve elde edilen toplulukların kalitesi, modülerite ölçümü ile değerlendirilmiştir.

3.3.2. Modülerite ölçümü

Modülerlik (Modularity), bir çizgenin topluluk yapısının ne kadar iyi tanımlandığının bir ölçüsüdür. Başka bir deyişle, bir çizgenin düğümlerinin topluluklara ayrılmasının ne kadar anlamlı olduğunu değerlendirir. Yüksek modülerlik, topluluk içindeki bağlantıların yoğun, topluluklar arasındaki bağlantıların ise seyrek olduğunu gösterir. [57]

Modülerite, bir ağdaki topluluk yapısını ölçmek Eşitlik (3.1) ile gösterildiği gibi hesaplanır:

$$Q = \left(\frac{1}{2m} \right) \sum_{ij} \left[A_{ij} - \frac{(k_i * k_j)}{2m} \right] \delta(c_i, c_j) \quad (3.1)$$

Burada;

Q: Modülerite Skoru

A_{ij}: Düğüm i ve düğüm j arasındaki kenarın ağırlığı (ağırlıksız grafikte 1,yoksa 0)

k_i: Düğüm i'nin derece (toplam bağlantı sayısı veya ağırlığı)

m: Grafikteki toplam kenar sayısı veya toplam kenar ağırlığı (ağırlıklı grafikte)

δ(c_i, c_j): Düğüm i ve düğüm j'nin aynı toplulukta olup olmadığını gösteren delta fonksiyonu (aynı toplulukta ise 1,değilse 0)

Ağırlıklı çizgelerde ise kenarların ağırlıkları, düğümler arasındaki bağlantıların gücünü temsil ettiği için Modülerlik hesabı Eşitlik (3.2)' de görüldüğü gibi güncellenir aşağıdaki güncellemeler yapılır.

$$Q = \left(\frac{1}{2W} \right) \sum_{ij} \left[w_{ij} - \frac{(s_i * s_j)}{2W} \right] \delta(c_i, c_j) \quad (3.2)$$

Burada;

w_{ij} : Düğüm i ve düğüm j arasındaki kenarın ağırlığı.

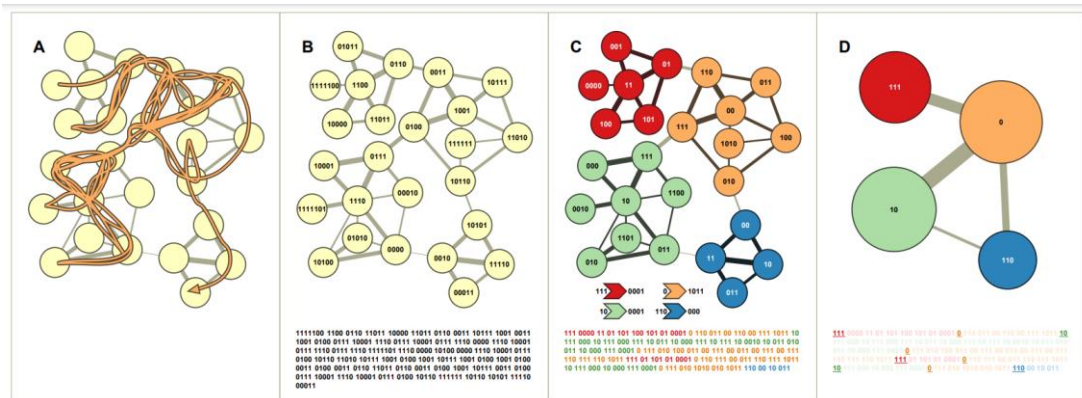
s_i : Düğüm i 'nin ağırlıklı derece (yani, düğüm i 'ye bağlı kenarların toplam ağırlığı).

W : Grafikteki toplam kenar ağırlığı.

Diğer terimler: Ağırlıksız grafikteki ile aynıdır. Modülerlik değeri -1 ile 1 arasında değişir. Pozitif değerler, çizge yapısının anlamlı topluluklara sahip olduğunu gösterirken, 0 değeri rastgele bir topluluk yapısını temsil eder. Negatif değerler ise topluluk yapısının beklenenden daha zayıf olduğunu gösterir.

3.3.2.1. InfoMap

InfoMap algoritması, bilgi akışı teorisine dayanan bir topluluk tespit algoritmasıdır. Bu süreçte, çizge yapısı üzerinde yüksek ağırlıklı kenarları tercih eden rastgele yürüyüşler simüle edilerek her düğüme, rastgele yürüyüşlerin hangi toplulukta daha sık bulunduğunu gösteren bir kod atanır. Bu kodlar, düğümlerin hangi topluluğa ait olduğunu belirlemek için kullanılır. Daha sonrasında toplulukları optimize etmek için Minimum Description Length (MDL) prensibi kullanılır. Bu prensip ile hem topluluk yapısını tanımlamak için gereken kod uzunluğunu hem de çizge yapısının bu kodlarla ifade edilmesi için gereken bilgi miktarını en aza indirmek amaçlanır. Böylece, çizge üzerindeki bilgi akışı en verimli şekilde kodlanır ve topluluk yapıları en anlamlı biçimde ortaya çıkar. [58]



Şekil 3.4: Infomap çalışma prensibi [58].

- (A) Rastgele Yürüyüşün Tanımlanması:** Bu aşamada, bir çizge üzerindeki rastgele bir yürüyüşün izini tanımlamak amaçlanmıştır. Burada, turuncu çizgi, rastgele bir yürüyüşün örnek bir yolunu göstermektedir. Bu yürüyüş sırasında, önemli yapılar benzersiz isimlerle tanımlanır.
- (B) Temel Kodlama Yaklaşımı:** Burada, çizgedeki her düğüme benzersiz bir isim verilerek, rastgele yürüyüşün tanımlanması hedeflenmiştir. Gösterilen Huffman kodu, her düğümü verimli bir şekilde adlandırmak için kullanılan bir yöntemdir. Çizge yapısının altında gösterilen 314 bitlik bilgi, A'daki örnek yürüyüşü tanımlar. Örneğin, sol üst köşedeki ilk düğüm 1111100 olarak kodlanmıştır, ikinci düğüm 1100 vb. Kodlar, yürüyüş sırasında ziyaret edilen düğümleri tanımlar.
- (C) İki Seviyeli Tanımlama:** Bu süreçte, rastgele yürüyüşün iki seviyeli bir tanımı yapılmıştır. Bu yöntemde, büyük kümeler benzersiz isimler alır ancak kümeler içindeki düğümler tekrar kullanılan isimlerle belirtilir. Bu yaklaşım, çizge yapısını daha kısa bir kodla tanımlamayı sağlar (Şekil 3.1'deki çizge yapısı için ortalama %32 daha kısa bir tanım sağlanmıştır). Modülleri adlandıran kodlar ve her modülden çıkışı belirtmek için kullanılan kodlar, çizge yapısının altında okların solunda ve sağında gösterilmiştir. Bu kodlama ile A'daki yürüyüş, 243 bit ile tanımlanabilir. İlk üç bit (111), yürüyüşün kırmızı modülde başladığını belirtir; 0000 kodu ise yürüyüşün ilk düğümünü belirtir.
- (D) Modül Tabanlı Tanımlama:** Bu aşamada, çizge yalnızca modül isimleri kullanılarak tanımlanır. Bu yaklaşım, çizge yapısının optimal kaba taneli bir gösterimini sağlar. Burada, modüller içindeki konumlar belirtilmez; sadece modüller arasındaki geçişler gösterilir. Bu şekilde, çizge yapısı daha basit ve genel bir şekilde tanımlanabilir.

3.3.2.2. Fast greedy

Fast Greedy algoritması, açgözlü bir algoritmadır, her iterasyondaki amacı çizge yapısının modülerliğini (modularity) en üst düzeye çıkarmaya çalışarak çizgedeki toplulukları belirlemektir. Bu doğrultuda, birbirine ağırlığı yüksek kenarlarla bağlı düğümleri aynı toplulukta tutmaya çalışır. [59]

Algoritma 3.1: Fast Greedy algoritma adımları

1.Başlangıç Durumu:

Her düğüm başlangıçta kendi başına bir topluluk olarak kabul edilir.

Bu durumda, çizge yapısında 'n' düğüm varsa, başlangıçta 'n' topluluk vardır.

2.Modülerite Artışı Hesaplama:

Algoritmanın her adımında, iki topluluğu birleştirmenin modüleriteyi ne kadar artıracağı hesaplanır.

Ağırlıklı çizge yapılarında modülerite değişimi Eşitlik (3.3) ile hesaplanır

$$\Delta Q = \left(\frac{1}{2W}\right) \left[2e_{AB} - \left(\frac{(a_A + a_B)^2}{2W} + \frac{a_A^2}{2W} + \frac{a_B^2}{2W} \right) \right] \quad (3.3)$$

Burada;

W: Çizge yapısındaki toplam kenar ağırlığı

e_{AB}: Topluluk A ve B arasındaki kenarların toplam ağırlığı

a_A ve a_B: Sırasıyla topluluk A ve B'nin toplam kenar ağırlıkları (düğüm dereceleri)

3.Topluluk Birleştirme

Her adımda, modüleritede en büyük artışı sağlayan iki topluluk birleştirilir.

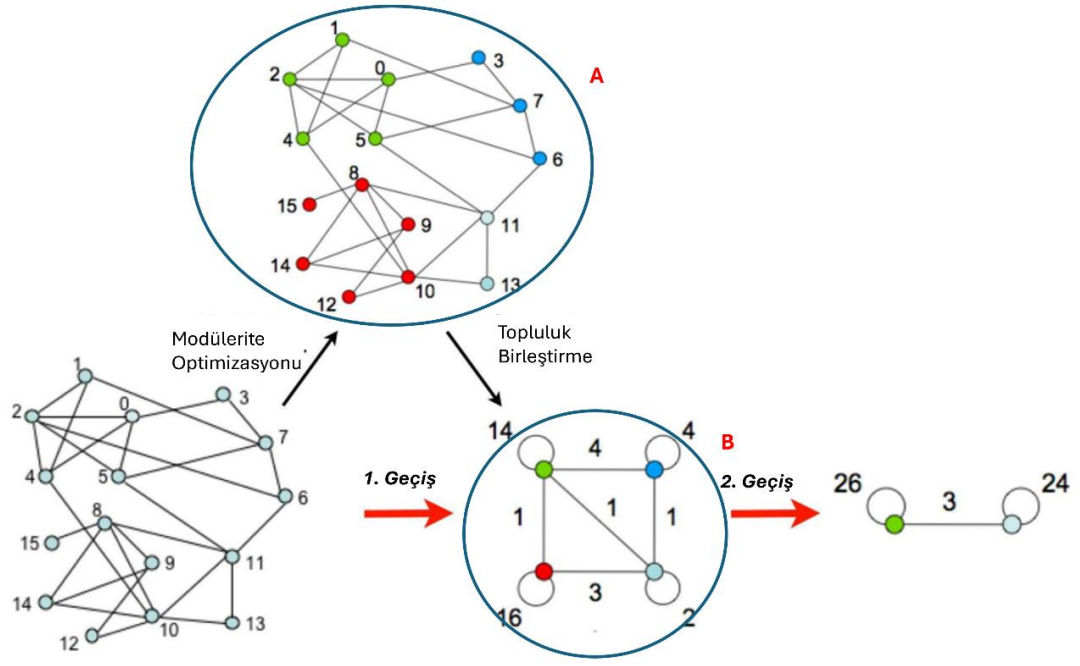
Bu işlem, topluluklar tek bir topluluk haline gelene veya daha fazla birleştirme modüleriteyi artırmaz hale gelene kadar devam eder.

4.Tekrarlama

Adım 2 ve 3,tüm topluluklar tek bir toplulukta birleşene veya daha fazla birleştirme modüleriteyi artırmaz hale gelene kadar tekrarlanır.

3.3.2.3. Louvain algoritması

Fast Greedy Algoritması sezgisel bir algoritmadır, Louvain algoritmasına benzer olarak bir çizge yapısındaki toplulukları , modüleriteyi optimize ederek tespit etmeyi amaçlar. [60]



Şekil 3.5: Louvain algoritması çalışma prensibi [60].

1. Modülerite Optimizasyonu (İlk Geçiş):

Başlangıç Durumu

İlk olarak, her düğüm kendi başına bir topluluk olarak kabul edilir.

Düğüm Taşıma

Her düğüm, komşu düğümlerin topluluklarına taşınarak modüleritedeki değişim Eşitlik (3.3)'te verildiği gibi hesaplanır ve düğüm, modüleriteyi en çok artıran topluluğa taşınır. Şekil 3.2'de, Çizge A'da, farklı renklerle gösterilen topluluklar, modüleritenin optimize edildiği bu aşamanın sonucudur.

Bir düğüm i'nin topluluk C'ye taşınması durumunda modüleritedeki değişimi (ΔQ) Eşitlik (3.3)'te verildiği gibi hesaplanır.

$$\Delta Q = \left[\frac{\sum_{in} C + k_{i,in}}{2m} - \left(\frac{\sum_{tot} C + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in} C}{2m} - \left(\frac{\sum_{tot} C}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \quad (3.3)$$

Burada;

$\sum_{in}$: Topluluk C içindeki kenarların toplam ağırlığı

$k_{\{i\}}$: düğüm i'nin toplam kenar ağırlığını;

$k_{\{i,in\}}$: düğüm i'nin topluluk C'deki düğümlerle olan kenarlarının toplam ağırlığını;

m : çizgedeki toplam kenar ağırlığını ifade eder.

2. Topluluk Birleştirme (İlk Geçiş)

Modülerite optimizasyonu tamamlandıktan sonra, bulunan topluluklar bir araya getirilir ve bir sonraki iterasyonda kullanılmak üzere daha basit yapıda yeni bir topluluk çizgesi (Şekil 3.2'de Çizge B) oluşturulur.

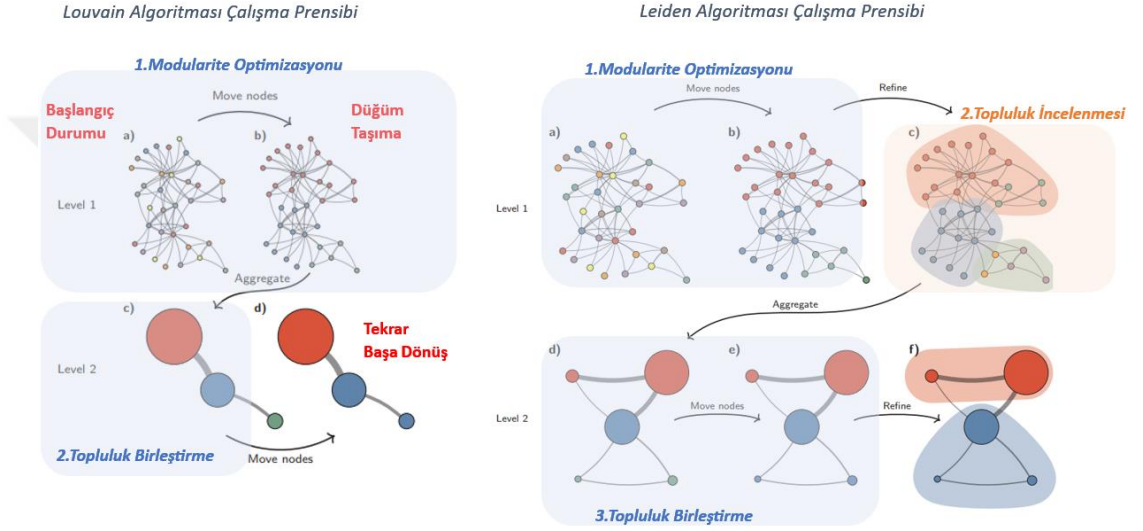
3. İkinci Geçiş (2nd Pass) ve Tekrarlama

Adım 1 ve 2 (Modülerite Optimizasyonu ve Topluluk Birleştirme) tekrarlanarak modülerite değeri daha da arttırılır ve yeni çizge (Şekil 3.2'de Çizge C) üzerinden daha büyük topluluklar tanımlanır.

Bu iki aşama (modülerite optimizasyonu ve topluluk birleştirme) modülerite artışı mümkün olana kadar tekrarlanır.

3.3.2.4. Leiden

Leiden algoritması sezgisel bir algoritmadır ve Louvain algoritmasının geliştirilmiş bir versiyonudur. Louvain algoritması Leiden algoritmasının aksine toplulukların bağlanabilirliğini (connectivity) garanti etmez. Bundan dolayı, bir topluluk içindeki düğümler her zaman birbirine bağlantılı olmayabilir veya zayıf bağlarla bir arada olabilir. Bu durum, toplulukların bölünmesine veya anlamsız toplulukların oluşmasına neden olabilir. Leiden algoritmasında ise toplulukların bağlanabilirliğini artırmak için Topluluk Rafinesi / İncelenmesi adımını içerir. [61]



Şekil 3.6: Louvain ve Leiden algoritmalarının çalışma prensiplerinin karşılaştırması [61].

Algoritma 3.3: Leiden algoritma adımları.

1. Modülerite Optimizasyonu:

Başlangıç Durumu (a): İlk olarak, her düğüm kendi başına bir topluluk olarak kabul edilir.

Düğüm Taşıma (b): Her düğüm, komşu düğümlerin topluluklarına taşınarak modüleritedeki değişim hesaplanır ve düğüm, modüleriteyi en çok artıran topluluğa taşınır.

2. Topluluk Rafinesi/İncelenmesi (c):

Elde edilen toplulukların içindeki düğümlerin tamamen bağlantılı (connected) olup olmadığı kontrol edilir. Eğer bir topluluk tamamen bağlantılı değilse, düğümler, mevcut topluluklarından alınıp rastgele diğer topluluklara yeniden atanır ve Modülerite Optimizasyonu işlemleri tekrarlanarak topluluklar iki veya daha fazla alt topluluğa ayrılır.

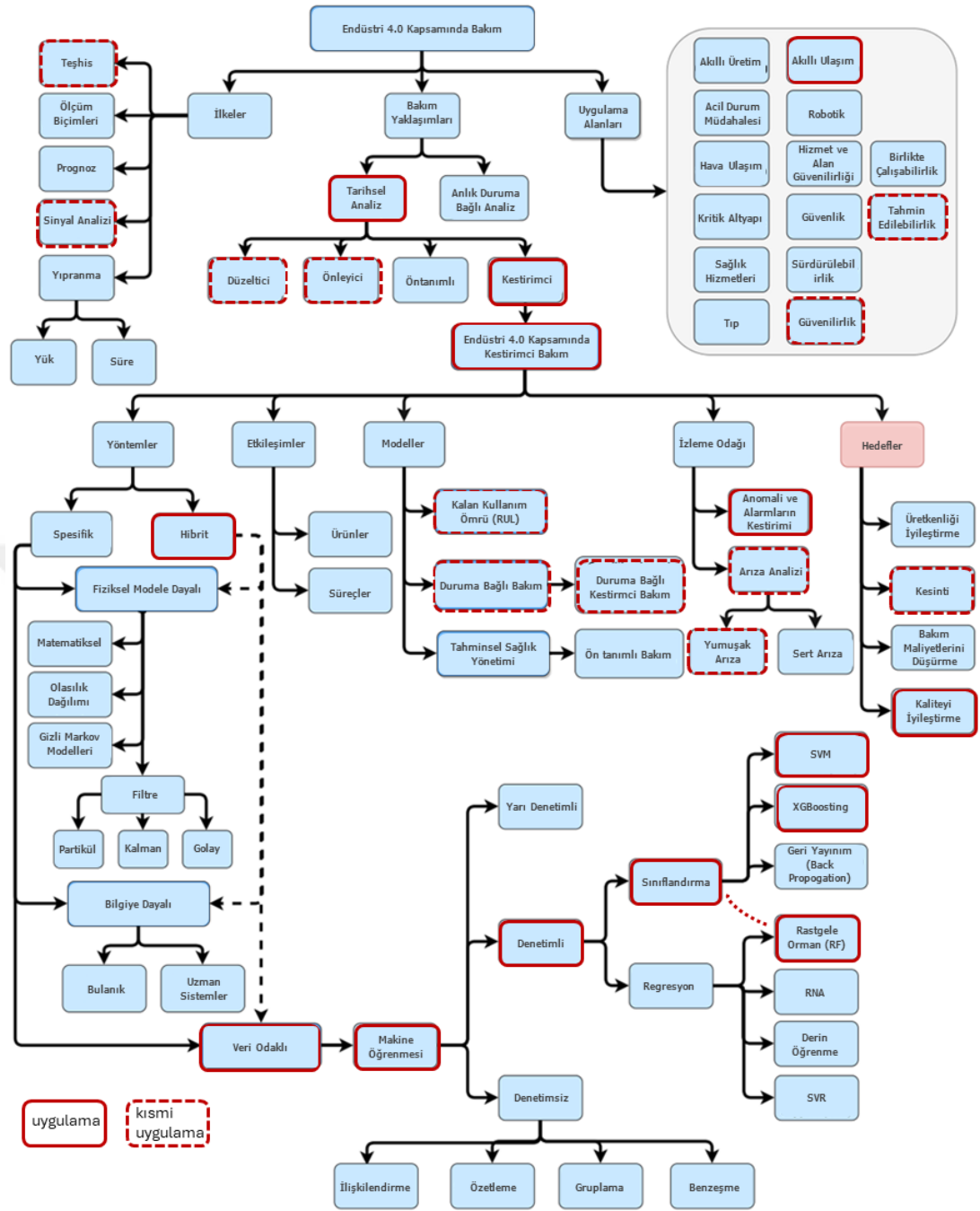
3. Topluluk Birleştirme ve Tekrarlama (d-e-f)

Topluluklar birleştirilerek yeni bir topluluk ağı oluşturulur. Bu yeni ağ üzerinde yeniden modülerite optimizasyonu yapılır ve düğümlerin ince ayarı tekrar uygulanır.

Bu aşamalar, modüleritede daha fazla artış sağlanamayınca kadar tekrarlanır.

3.4. Kestirimci Bakım ve Makine Öğrenmesi

Tiago ve arkadaşları tarafından [51] Endüstri 4.0 kestirimci bakım konusunda yapılan sistematik literatür taramasında geliştirilen taxonomy üzerinde, bu tezde uygulanan veya kısmen uygulanan kavramlar kırmızı ile işaretlenmiş ve uyarlanmış Şekil 3.4'de gösterilmiştir.



Şekil 3.7: Endüstri 4.0 Kestirimci bakım taksonomisi [51] ve bu tez çalışmasında uygulanan teknikler.

3.4.1. Rastgele orman (RF)

Leo Breiman ve Adele Cutler tarafından geliştirilen ve tek bir sonuca ulaşmak için birden fazla karar ağacının çıktısını birleştiren, yaygın olarak kullanılan bir makine öğrenimi algoritmasıdır [62] [63] [64]. Hem sınıflandırma hem de regresyon problemlerini ele aldığı için kullanım kolaylığı ve esnekliği benimsenmesini sağlamıştır. Paralel topluluk yöntemine bir örnektir. Bir dizi karar ağacı oluşturarak

çalışır. Eğitim ve sonuçlar, bireysel ağaçların ortalama tahmininden faydalanır. Paralel yöntemlerde amaç, temel öğrenciler arasındaki bağımsızlıktan yararlanmaktır, çünkü hata ortalama ile önemli ölçüde azaltılabilir. Aksi takdirde, sıralı topluluk yöntemlerinde, temel öğrenciler sırayla oluşturulur ve amaç aralarındaki bağımlılıktan yararlanmaktır. Genel performans, daha önce yanlış etiketlenmiş örneklerin daha yüksek yüklerle tartılmasıyla artırılabilir.

Rastgele orman algoritması bir karar ağaçları topluluğundan oluşur ve topluluktaki her ağaç, bootstrap örneği olarak adlandırılan bir eğitim setinden değiştirilerek alınan bir veri örneğinden oluşur. Bu eğitim örneğinin üçte biri, torba dışı (oob) örnek olarak bilinen ve daha sonra geri dönülecek test verisi olarak ayrılır. Daha sonra özellik torbalama yoluyla başka bir rastgelelik örneği enjekte edilerek veri kümesine daha fazla çeşitlilik eklenir ve karar ağaçları arasındaki korelasyon azaltılır. Problemin türüne bağlı olarak, tahminin belirlenmesi değişecektir. Bir regresyon görevi için, bireysel karar ağaçlarının ortalaması alınacak ve bir sınıflandırma görevi için, çoğunluk oyu (yani en sık kategorik değişken) tahmin edilen sınıfı verecektir. Son olarak, oob örneği daha sonra çapraz doğrulama için kullanılır ve bu tahmini kesinleştirir.

Model parametreleri:

- **n_estimators (Ağaç Sayısı):** Modelde kullanılacak karar ağacı sayısını belirler. Genellikle daha fazla ağaç, daha iyi performans sağlar, ancak hesaplama maliyetini artırır.
- **max_depth (Maksimum Derinlik):** Karar ağaçlarının maksimum derinliğini sınırlar. Daha derin ağaçlar daha karmaşık modeller oluşturur, ancak aşırı öğrenmeye yol açabilir.
- **min_samples_split (Bölünme İçin Minimum Örnek Sayısı):** Bir düğümün bölünebilmesi için gereken minimum örnek sayısını belirler. Daha yüksek değerler, daha küçük ağaçlar oluşturur ve aşırı öğrenmeyi önlemeye yardımcı olabilir.
- **min_samples_leaf (Yaprak İçin Minimum Örnek Sayısı):** Bir yaprak düğümde bulunması gereken minimum örnek sayısını belirler. Daha yüksek değerler, daha küçük ağaçlar oluşturur ve aşırı öğrenmeyi önlemeye yardımcı olabilir.

- **max_features (Bölünmelerde Kullanılacak Maksimum Özellik Sayısı):** Her bir bölünmede dikkate alınacak maksimum özellik sayısını belirler. Daha düşük değerler, daha fazla rastgelelik sağlar ve aşırı öğrenmeyi önlemeye yardımcı olabilir.
- **bootstrap (Bootstrap Örnekleme):** Her bir karar ağacının eğitimi için bootstrap örnekleme kullanılıp kullanılmayacağını belirler. Bootstrap örnekleme, veri setinden rastgele örnekler seçerek farklı ağaçların farklı veri alt kümeleri üzerinde eğitilmesini sağlar.

Avantajları

- **Azaltılmış aşırı uyum riski:** Karar ağaçları, eğitim verilerindeki tüm örneklerle sıkıca uyma eğiliminde olduklarından aşırı uyum riski taşırlar. Bununla birlikte, rastgele bir ormanda yeterli sayıda karar ağacı olduğunda, ilişkisiz ağaçların ortalaması genel varyansı ve tahmin hatasını düşürdüğünden sınıflandırıcı modele aşırı uyum sağlamayacaktır.
- **Esneklik sağlar:** Rastgele orman hem regresyon hem de sınıflandırma görevlerini yüksek doğrulukla yerine getirebildiğinden, veri bilimcileri arasında popüler bir yöntemdir. Özellik torbalama, verilerin bir kısmı eksik olduğunda doğruluğu koruduğu için rastgele orman sınıflandırıcısını eksik değerleri tahmin etmek için de etkili bir araç haline getirir.
- **Özelliğin önemini belirlemek kolaydır:** Rastgele orman, değişken önemini veya modele katkısını değerlendirmeyi kolaylaştırır. Özelliğin önemini değerlendirmenin birkaç yolu vardır. Gini önemi ve safsızlıktaki ortalama azalma (MDI) genellikle belirli bir değişken hariç tutulduğunda modelin doğruluğunun ne kadar azaldığını ölçmek için kullanılır. Bununla birlikte, ortalama azalma doğruluğu (MDA) olarak da bilinen permütasyon önemi başka bir önem ölçüsüdür. MDA, oob örneklerindeki özellik değerlerinin rastgele permütasyonu ile doğruluktaki ortalama düşüşü tanımlar.

Dezavantajları

- **Yavaşlık:** Rastgele orman algoritmaları büyük veri setlerini işleyebildiğinden, daha doğru tahminler sağlayabilirler, ancak her bir karar ağacı için veri hesapladıkları için verileri işlemek yavaş olabilir.

- **Kaynak gerektirme:** Rastgele ormanlar daha büyük veri setlerini işlediğinden, bu verileri depolamak için daha fazla kaynağa ihtiyaç duyacaktır.
- **Karmaşıklık:** Tek bir karar ağacının tahminini yorumlamak, bu ağaçlardan oluşan bir ormana kıyasla daha kolaydır.

3.4.2. Extreme gradient boosting yöntemi (XGBoost)

Gradient boosting, regresyon ve sınıflandırma görevlerinde kullanılan bir makine öğrenimi tekniğidir. GBM’de zayıf öğreniciler kullanılarak hataları azaltan iteratif bir süreç sonunda güçlü bir öğrenici oluşturmak amaçlanır. Tipik olarak karar ağaçları zayıf tahmin modelleri topluluğu şeklinde bir tahmin modeli verir. Gradyan takviyeli ağaçlar modeli, diğer takviye yöntemlerinde olduğu gibi aşamalı bir şekilde oluşturulur, modelinin temel amacı, regularization terimleri içeren kayıp fonksiyonu minimize etmektir [65].

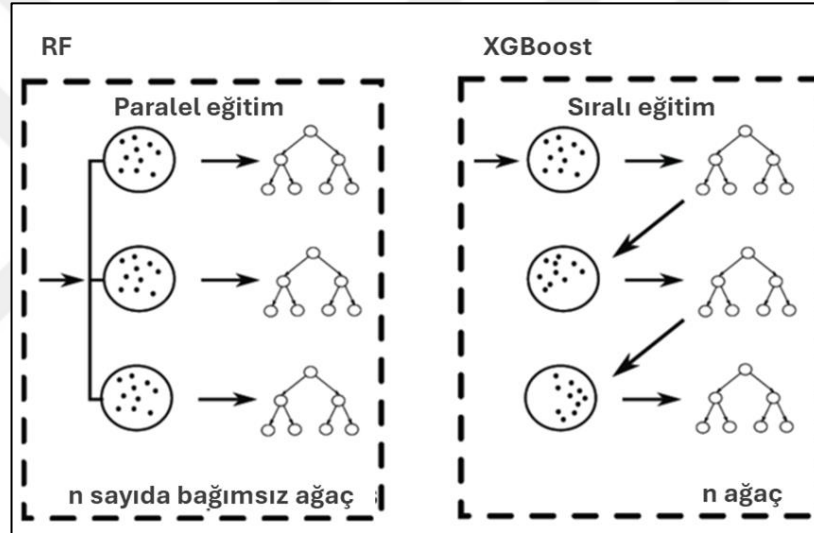
Süreç başlangıç ağacı ile başlar, hedef değişkenin değeri ile ağacın tahmin ettiği değer arasındaki farka göre hesaplanan hatayı azaltmak için bir sonraki ağaç oluşturulur ve bu ağaç kendinden önceki ağacın tahmin ettiği hatanın negatif gradyanını tahmin ederek hatayı düzeltmeye yardımcı olur. Yeni ağaçlar eklenerek hatanın azalması sağlanır.

Model parametreleri:

XGBoost'un öne çıkan parametreleri :

- **n_estimators (Ağaç Sayısı):** Modelde kullanılacak karar ağacı sayısını belirler. Genellikle daha fazla ağaç, daha iyi performans sağlar, ancak hesaplama maliyetini artırır.
- **max_depth (Maksimum Derinlik):** Karar ağaçlarının maksimum derinliğini sınırlar. Daha derin ağaçlar daha karmaşık modeller oluşturur, ancak aşırı öğrenmeye yol açabilir.
- **learning_rate (Öğrenme Oranı):** Her bir ağacın modelin genel tahminine katkısını kontrol eder. Düşük öğrenme oranları, daha fazla ağaç gerektirir ancak aşırı öğrenmeyi önlemeye yardımcı olabilir.

- **subsample (Alt Örnekleme Oranı):** Her bir ağaç için kullanılacak örneklerin oranını belirler. Daha düşük oranlar, daha fazla rastgelelik sağlar ve aşırı öğrenmeyi önlemeye yardımcı olabilir.
- **colsample_bytree (Ağaç Başına Özellik Örnekleme Oranı):** Her bir ağaç için kullanılacak özelliklerin oranını belirler. Daha düşük oranlar, daha fazla rastgelelik sağlar ve aşırı öğrenmeyi önlemeye yardımcı olabilir.
- **reg_alpha (L1 Regularizasyonu):** Ağaçların karmaşıklığını kontrol etmek için L1 regularizasyonu uygular.
- **reg_lambda (L2 Regularizasyonu):** Ağaçların karmaşıklığını kontrol etmek için L2 regularizasyonu uygular.



Şekil 3.8: Random Forest ve XGBoost yöntemlerinin karşılaştırılması [66].

Rastgele ormanda, N adet bağımsız öğrenici (ağaç), girdi veri kümesinin rastgele alt kümelerinden paralel olarak oluşturulur. Gradyan Boost modelinde ise, bir sonraki ağaç topluluğun önceki ağaçlarından oluşturulur ve ona eklenir.

Çizelge 3.1: Random Forest ve XGboost farklılıkları.

Random Forest ve XGBoost Model Farklılıkları		
Özellik	Random Forest	XGBoost
Ağaç Oluşturma	Paralel olarak ağaçlar oluşturur.	Ardışık olarak ağaçlar oluşturur, bir önceki ağacın hatalarını düzeltmeye çalışır.
Bölme Kriteri	Gini impurity veya entropy gibi bilgi kazancı temelli kriterler kullanır.	Gradyanlar ve Hessian'lar gibi ikinci dereceden türev bilgisi kullanarak daha hassas bölmeler yapar.
Regularizasyon	Aşırı öğrenmeyi önlemek için daha az regularizasyon seçeneği sunar.	L1 ve L2 regularizasyonu ile aşırı öğrenmeyi daha etkili bir şekilde önler.
Hız	Genellikle XGBoost'tan daha yavaştır.	Genellikle Random Forest'tan daha hızlıdır, özellikle büyük veri setlerinde.
Hiperparametre Optimizasyonu	Daha az sayıda hiperparametreye sahiptir.	Daha fazla sayıda hiperparametreye sahiptir, bu da daha fazla ayarlama imkanı sunar ancak optimizasyon sürecini zorlaştırabilir.
Eksik Veri İşleme:	Eksik verileri doldurmak için özel bir mekanizma kullanmaz.	Eksik verileri otomatik olarak işleyebilir.

Avantajlar

- **Yüksek Doğruluk:**, Büyük veri setleri ve karmaşık özelliklerle çalışırken yüksek doğruluk sağlar.
- **Genelleme Yeteneği:** GDoğru hiperparametreler seçildiğinde, yeni verilere de iyi genelleme yapabilir.
- **Çok Yönlülük:** Hem sınıflandırma hem de regresyon problemlerinde kullanılabilir.
- **Direnç:** Veri setindeki gürültüye karşı oldukça dirençlidir.
- **Azaltılmış Hata:** Diğer makine öğrenme algoritmalarına göre daha az hata yapar.

Dezavantajlar

- **Aşırı Öğrenme:** Yeterli düzenleme yapılmazsa, eğitim verilerine aşırı uyum sağlayabilir ve aşırı öğrenme yapabilir.
- **Hiperparametre Seçimi:** XGBM'nin performansı, doğru hiperparametrelerin seçilmesine bağlıdır. Yanlış seçilmiş hiperparametreler, performansı önemli ölçüde etkileyebilir.

- **Hesaplama Gücü:** Hesaplama gücü açısından oldukça yoğun bir algoritmadır ve büyük veri kümelerinde kullanılması zor olabilir.

3.4.3. Destek vektör makinesi (SVM)

Destek Vektör Makinesi (SVM), hem sınıflandırma hem de regresyon problemlerini çözmek için kullanılan bir makine öğrenimi algoritmasıdır [67]. Genellikle sınıflandırma için kullanılır. SVM, verileri farklı sınıflara ayırmak için en iyi çizgiyi veya hiper düzlemi bulmaya çalışır.

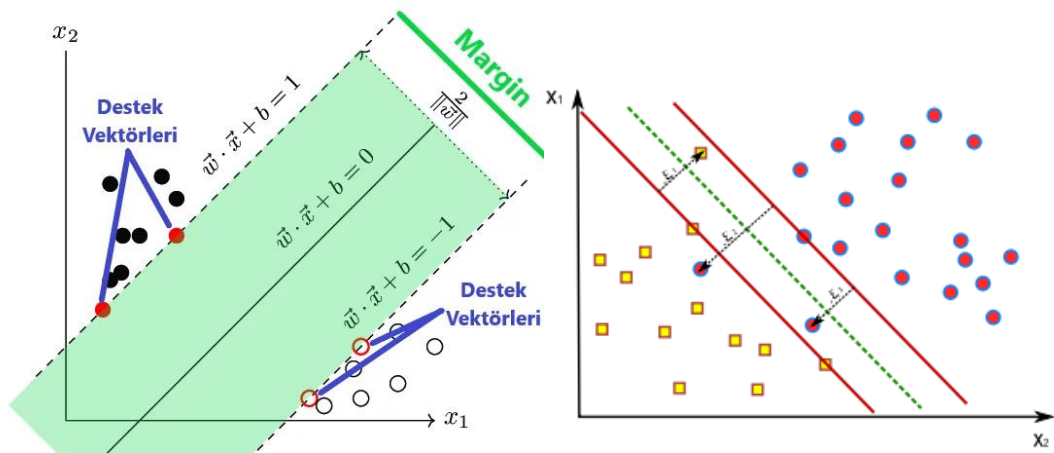
İki sınıfı birbirinden ayıran çizgiye veya düzleme "hiper düzlem" denir. Hiper düzlem, verilerin yer aldığı uzayı ikiye böler.

Destek vektörleri, hiper düzleme en yakın olan veri noktalarıdır. Bu noktalar, hiper düzlemin konumunu belirler.

Hiper düzlem ile destek vektörleri arasındaki mesafe marj şeklinde adlandırılır. SVM, bu marjı maksimize etmeye çalışır, böylece sınıflar arasındaki ayrım en belirgin hale gelir.

Doğrusal ve doğrusal olmayan SVM den bahsedilebilir. Eğer veriler doğrusal olarak ayrılabilirse, SVM, sınıfları ayıran en iyi çizgiyi bulur. Eğer veriler doğrusal olarak ayrılmıyorsa, SVM, verileri daha yüksek boyutlu bir uzaya dönüştürür ve burada doğrusal olarak ayırır. Bu dönüşüm için çekirdek fonksiyonları kullanılır (örneğin, RBF çekirdeği).

SVM resim, metin, biyolojik veri kategorizasyonunda kullanılabilir.



Şekil 3.9: SVM algoritması [67].

Temel kavramlar ve formüller:

- **Hiper Düzlem (Hyperplane)**

Hiper düzlem, iki sınıfı birbirinden ayıran doğrusal bir sınırlayıcıdır. İki boyutlu bir uzayda bir doğru, üç boyutlu bir uzayda ise bir düzlemdir.

- **Marjin (Margin)**

Marjin, iki sınıfın en yakın noktalarına olan mesafedir. SVM'nin amacı, bu marjini maksimize etmektir.

- **Destek Vektörleri**

Destek vektörleri, Destek Vektör Makineleri (SVM) modelinde, iki sınıfı ayıran hiper düzleme en yakın noktalar olarak tanımlanır. Bu vektörler, SVM'nin karar sınırını belirlemede kritik rol oynar ve marjini maksimize etmeye yardımcı olur.

- **Soft Margin (Yumuşak Marjin) SVM**

Soft Margin SVM (Yumuşak Marjinli Destek Vektör Makineleri), orijinal SVM algoritmasının bir uzantısıdır ve doğrusal olarak tamamen ayrılamayan veri setleri için kullanılır. Orijinal SVM (hard margin SVM), veri noktalarını mükemmel bir şekilde ayıran bir hiperdüzlem bulmaya çalışır ve bu nedenle aykırı değerlere karşı hassastır. Soft margin SVM ise, bazı hatalara izin vererek daha esnek bir karar sınırı oluşturur ve bu sayede aykırı değerlere daha dayanıklı hale gelir.

Soft margin SVM'in amacı, Eşitlik (3.4)'de verilen optimizasyon problemini çözmektir:

$$OP = \text{minimize} \left(\frac{1}{2} |w|^2 + C \sum \xi_i \right) \quad (3.4)$$

Burada;

w: Hiperdüzlemin ağırlık vektörüdür.

C: Hata terimi üzerindeki ceza katsayısıdır.

ξ_i : Her bir veri noktası için hata payıdır (slack variable).

Bu optimizasyon probleminin kısıtları ise Eşitlik (3.5) ve Eşitlik (3.6) ile gösterilmiştir:

$$K1 = y_i(w^T x_i + b) \geq 1 - \xi_i \quad (3.5)$$

$$K2 = \xi_i \geq 0 \quad (3.6)$$

Bu kısıtlar, her bir veri noktasının doğru tarafta ve mümkün olduğunca hiperdüzlemin doğru tarafında olmasını sağlar. Hata payı (ξ_i), bir veri noktasının ne kadar yanlış sınıflandırıldığını gösterir.

Model parametreleri:

- **C (Regularization Parameter):** C parametresi, modelin karmaşıklığı ile eğitim verilerine ne kadar uyum sağlayacağı arasındaki dengeyi kontrol eder. Düşük C değerleri, daha geniş bir karar sınırı oluşturarak daha basit bir model elde edilmesini sağlar. Yüksek C değerleri ise, daha karmaşık bir model oluşturarak eğitim verilerine daha iyi uyum sağlamayı hedefler.
- **kernel (Çekirdek Fonksiyonu):** SVM'ler, doğrusal olarak ayıramayan veri setlerini sınıflandırmak için çekirdek fonksiyonları kullanır. Çekirdek fonksiyonu, veri noktalarını daha yüksek boyutlu bir uzaya dönüştürerek doğrusal olarak ayrılabilir hale getirmeye çalışır. Yaygın olarak kullanılan çekirdek fonksiyonları arasında linear, polynomial, radial basis function (RBF) ve sigmoid bulunur.
- **gamma (RBF Çekirdeği Parametresi):** RBF çekirdeği kullanıldığında, gamma parametresi karar sınırının ne kadar esnek olacağını belirler. Düşük gamma değerleri, daha geniş ve pürüzsüz bir karar sınırı oluştururken, yüksek gamma değerleri, daha dar ve karmaşık bir karar sınırı oluşturur.
- **degree (Polynomial Çekirdeği Parametresi):** Polynomial çekirdeği kullanıldığında, degree parametresi polinomun derecesini belirler. Yüksek dereceli polinomlar, daha karmaşık karar sınırları oluşturabilir.

Avantajları:

Yüksek boyutlu verilerde etkilidir.

Veriler arasındaki ayrımın net olduğu durumlarda iyi çalışır.

Dezavantajları:

Büyük veri setlerinde yavaş çalışır.

Parametrelerin doğru şekilde ayarlanması önemlidir.

Ön bilgi bölümünde bakım stratejileri detaylandırılarak açıklanmış, çizge tabanlı öznelik seçimi ve makine öğrenmesi yöntemlerinin temel prensipleri detaylı bir şekilde ele alınmıştır. Bu temel bilgiler, çalışmanın dayandığı teorik çerçeveyi oluşturarak, tezin ilerleyen bölümlerinde kullanılan yöntemler ve analiz teknikleri için bir altyapı sunmuştur. Bir sonraki Veri Seti bölümünde, bu teorik çerçeveyi pratiğe dökerek, çalışmanın temelini oluşturan veri setinin ayrıntılı analizi gerçekleştirilmiştir. Bölümde, ham veriden başlayarak, verinin toplanması, özetlenmesi analiz edilmesi ve sadeleştirilmesi süreçlerine yer verilmiştir. Veri setinin yapısı ve içeriği hakkında kapsamlı bilgi sunulmuş, verinin nasıl toplandığı ve işlendiği detaylandırılmış ve verinin analiz edilmesi sürecinde, anomali tespitine yönelik adımlar ele alınmıştır.



4. VERİ SETİ

4.1. Ham Veri

Çalışmaya girdi olan otobüs CAN Bus verileri, 29.09.2021 ile 23.06.2023 tarihleri arasında, beş farklı otobüsten gerçek zamanlı olarak toplanmış projede görev alan yüklenici tarafından 1sn. periyotlu, tabular formatlı zaman serilerine dönüştürülmüştür. Bu veriler, otobüslerin CAN Bus’ından elde edilen paket bilgilerini, çeşitli operasyonel parametreleri ve performans ölçümlerini içermektedir.

4.1.1. Veri seti özeti

Her bir otobüsten toplanan ve işlenen veriler benzersiz bir id ile etiketlenmiştir. Bununla beraber verilerin gönderim tarihleri ve büyüklükleri değişiklik göstermektedir. Her bir veri seti farklı sayıda .csv dosya formatındaki veri paketlerinden oluşmaktadır. Bu veri paketlerindeki değişken (sütun) sayısı tutarlı olmamakla beraber otobüslerden saniyelik toplanan ham CAN Bus verilerini ve zaman damgalarını içermektedir. Veri setlerinin analizi ve kapsamını belirleyebilmek için, ilk olarak tüm veri paketleri taranmış ve her veri setinin içerdiği eşsiz değişkenler listelenmiştir, bununla beraber farklı formatlarda bulunan zaman damgaları ISO 8601 formatlarını tanıma yeteneğine sahip isoparse fonksiyonu kullanılarak “%Y-%m-%d %H:%M:%S” formatına çevrilmiştir. Daha sonra, çalışma kapsamında hedeflenen alarmların aktif olduğu zaman aralıkları ve sıklıkları tespit edilmiştir.

Çizelge 4.1’de, beş farklı otobüsün CAN Bus verilerinden oluşturulan veri setlerinin veri setlerinin tanımlayıcı bilgileri bulunmaktadır.

Çizelge 4.1: Çalışma kapsamında toplanan veri setleri.

Veri Seti Adı	İlk Gönderim Tarihi	Veri Seti Büyüklüğü	Değişken Sayısı	Ortalama Hedef Değişken Sıklığı	Araç Hareket Tarihi
3F2551C1 (Referans Veri Seti)	26.06.2023	80.2 GB	902	974.5	30.09.2021- 26.05.2023
5E3DA3A9	10.07.2023	79.4 GB	872	123.2	29.09.2021- 23.06.2023
6BF5FA70	17.07.2023	26.3 GB	875	30.2	30.09.2021- 10.11.2022
920CB615	17.08.2023	24.8 GB	838	70	01.10.2021- 20.06.2023
EF818714	25.08.2023	81 GB	893	970	13.12.2021- 20.06.2023

Çizelge 4.2: Çalışma kapsamında belirlenen hedef değişken (Alarm) listesi.

Değişken Adı	Değişken Çeşidi	Değişken Tanımı
CAN1.EBC1.ABS_EBSAmberWarningSignal (AWS)	Diagnostik Alarm	Aracın elektronik fren sistemi ve güvenlik uyarı sistemleri ile ilgili belirli bir durumu veya uyarıyı tanımlar.
CAN1.VMU1.Dsh_System_Warning (DSW)	Genel Alarm	Aracın gösterge panelinde (dashboard) sistem uyarısı olduğunu belirten bir sinyali temsil eder. Bu tür uyarılar, araçtaki çeşitli sistemlerin durumları hakkında sürücüyü bilgilendirir ve potansiyel sorunlara dikkat çeker.
CAN1.VMU1.Dsh_System_Fault (DSF)	Genel Alarm	Aracın gösterge panelinde (dashboard) sistem hatası olduğunu belirten bir sinyali temsil eder.
CAN1.BMS_Status_Message.PDU_FAULT (PDU)	Diagnostik Alarm	Aracın Batarya Yönetim Sistemi (BMS) ile ilgili bir hata durumunu belirten sinyali temsil eder. Bu sinyal, batarya sisteminde meydana gelen ciddi bir arızayı veya anormalliği bildirir.
CAN1.Battery_Status.BMS_SOC¹(BMS-SOC)	Diagnostik Alarm	A batarya yönetim sistemindeki (Battery Management System - BMS) batarya şarj durumu (State of Charge - SOC) bilgisini temsil eder. Bu sinyal, bataryanın mevcut enerji kapasitesini ve ne kadar enerji depoladığını belirtir.

¹ CAN1.Battery_Status.BMS_SOC değişkeni sürekli bir değişkendir ancak bu çalışma kapsamında, değişkenin %10 üzerindeki değeri 0 %10 altındaki değerini 1 olarak tekrardan etiketlenerek kategorik bir değişkene çevrilmiştir.

Çalışma kapsamında "3F2551C1" veri seti, diğer veri setlerine kıyasla daha fazla eşsiz değişken içerdiği ve tahmini üzerinde çalışılacak hedef değişken sıklığı diğer veri setlerine oranla daha yüksek olduğu için uzman görüşleri doğrultusunda referans veri seti olarak kabul edilmiştir. Bu veri seti üzerinde veri analizi ve özellik seçimi için adımları gerçekleştirilerek sonuçlar diğer veri setlerine genelleştirilmiştir.

4.1.2. Veri toplama

Bu başlık altında, veri setlerinin oluşturulma aşamasının, elektrikli otobüslerden veri toplaması süreci detaylandırılmıştır. Kuyruk bazlı transfer sistemi (queue) kullanımı, veri tutarlılığı, Azure IOTHub entegrasyonu, geçici ara bellek kullanımı, hata yönetimi ve takibi, sistem gözlemleme ekranları gibi konularda uygulanan yöntemler ve elde ettiğimiz sonuçlar aktarılacaktır. Ayrıca, elektrikli araçlardan gelen geçmiş dönem verilerinin işlenmesi ve "MFC" formatından "MF4" formatına dönüştürülmesi ile ilgili ayrıntılara da değinilecektir.

4.1.2.1. Gerçek zamanlı toplanan veriler

Elektrikli otobüslerden gelen gerçek zamanlı verileri toplamak için kuyruk tabanlı transfer sistemi geliştirilmiştir. Bu sayede veri akışı yönetilebilir ve işlenebilir hale gelmiştir. Toplanan tüm veriler, veri kaybını önlemek amacıyla kuyruk sistemine ek olarak veri tabanında da güvenli bir şekilde saklanmıştır.

Transfer sistemi aracılığıyla gerçek zamanlı veriler, üzerlerinde ön işlem gerçekleştirilmek amacıyla Azure IOTHub'a transfer edilmiştir. Bu transfer sürecinde gerçek zamanlı veriler, Azure IOTHub'a aktarılmadan önce kuyruk sisteminde geçici olarak ara belleğe alınarak güvenli bir veri transferi gerçekleştirmek amaçlanmıştır, bununla beraber belirli bir zaman dilimi içerisinde işlenen veriler daha sonra kuyruk sisteminden çıkartılmıştır, bu şekilde kaynakların etkin kullanımı sağlanmıştır.

Veri transferi sırasında herhangi bir sorun oluşması durumunda, hatalar kayıt altına alınarak incelenebilir hale getirilmiştir. Bu sayede hata takibi kolaylaştırılarak sorunların hızlı bir şekilde çözülmesi sağlanmıştır.

Kuyruk sisteminde çalışan işlemlerin durumu, iş yükü yüzdeleri ve tamamlanma durumu gibi metriklerin yanı sıra grafiksel bilgiler de geliştirilen transfer sisteminde yer alan ekranlar aracılığıyla izlenebilmiştir. Bu sayede veri toplama ve işleme sürecinin genel sağlık durumu hakkında anlık bilgi sahibi olunabilmektedir.

4.1.2.2. Geçmiş dönem verileri

Elektrikli otobüslerden toplanan ve transfer edilen veriler, veri analizinde ve alarm tahmin model eğitimlerinde kullanılmak üzere aşağıdaki adımlar izlenerek hazır hale getirilmiştir.

"MFC" Formatından "MF4" Formatına Dönüşüm: Araç verileri, araçların bilgi üreten log sistemlerinden milisaniye seviyesinde detaylar içeren analiz için tercih edilmeyen "MFC" formatında saklanmaktadır. Dolayısıyla verilerin işlenip analiz edilebilir hale getirilmesi gerekmektedir. Bu amaçla, "MFC" biçimindeki veriler, analiz süreçleri için daha uygun olan "MF4" formatına dönüştürülmüştür.

MF4 Verilerinin Tablo Haline Dönüştürülmesi: "MF4" formatındaki veriler, farklı ".DBC" (Veritabanı Konteyneri) dosyaları aracılığıyla analiz için tabular data formatına dönüştürülmüştür. Bunun yanında, her bir ".DBC" dosyası, CAN Datasından elde edilen ilgili hex kod bloklarının belirli değişkenlerle eşleştirilmesini sağlayan bilgiler içermektedir. Bu şekilde dönüştürülen verilerin anlamlı hale gelmesi ve analiz edilebilir duruma getirilmesi sağlanmıştır.

4.2. Ham Veri Analizi

Şehir içi elektrikli otobüse ait referans veri seti 3F2551C1 üzerinde kapsamlı bir veri analizi gerçekleştirilmiştir. Bu analiz, verilerin genel özelliklerini anlamak, potansiyel problemleri belirlemek ve verilerden anlamlı bilgiler çıkarmak amacıyla yapılmıştır.

4.2.1. İstatiksel analiz

Veri analizi sürecinde ilk olarak, verilerin istatistiksel özellikleri incelenmiştir. Ham veride bulunan 902 özelliğin her biri için veri tipi, minimum ve maksimum değer, dağılım, ortalama ve sapma değerleri hesaplanarak değişkenlerin merkez eğilimleri ve dağılımları hakkında bilgi edinilmiştir. Değişkenlerin ortalama değerleri ile medyan değerleri kıyaslanarak dağılımı sağa ya da sola çarpık olan değişkenler belirlenmiştir. Bununla beraber standart sapması yüksek olan, ortalama etrafında geniş bir dağılıma sahip veriler tespit edilmiştir. Değişkenler, kategorik ve sürekli olmak üzere ikiye ayrılmıştır. Bu ayrım yapılırken veri tiplerine ve her değişkenin benzersiz değer sayısı dikkate alınmıştır. Integer ya da String tipinde olan ve benzersiz değer sayısı 20'nin altında kalan her bir değişken kategorik değişken olarak sınıflandırılmıştır.

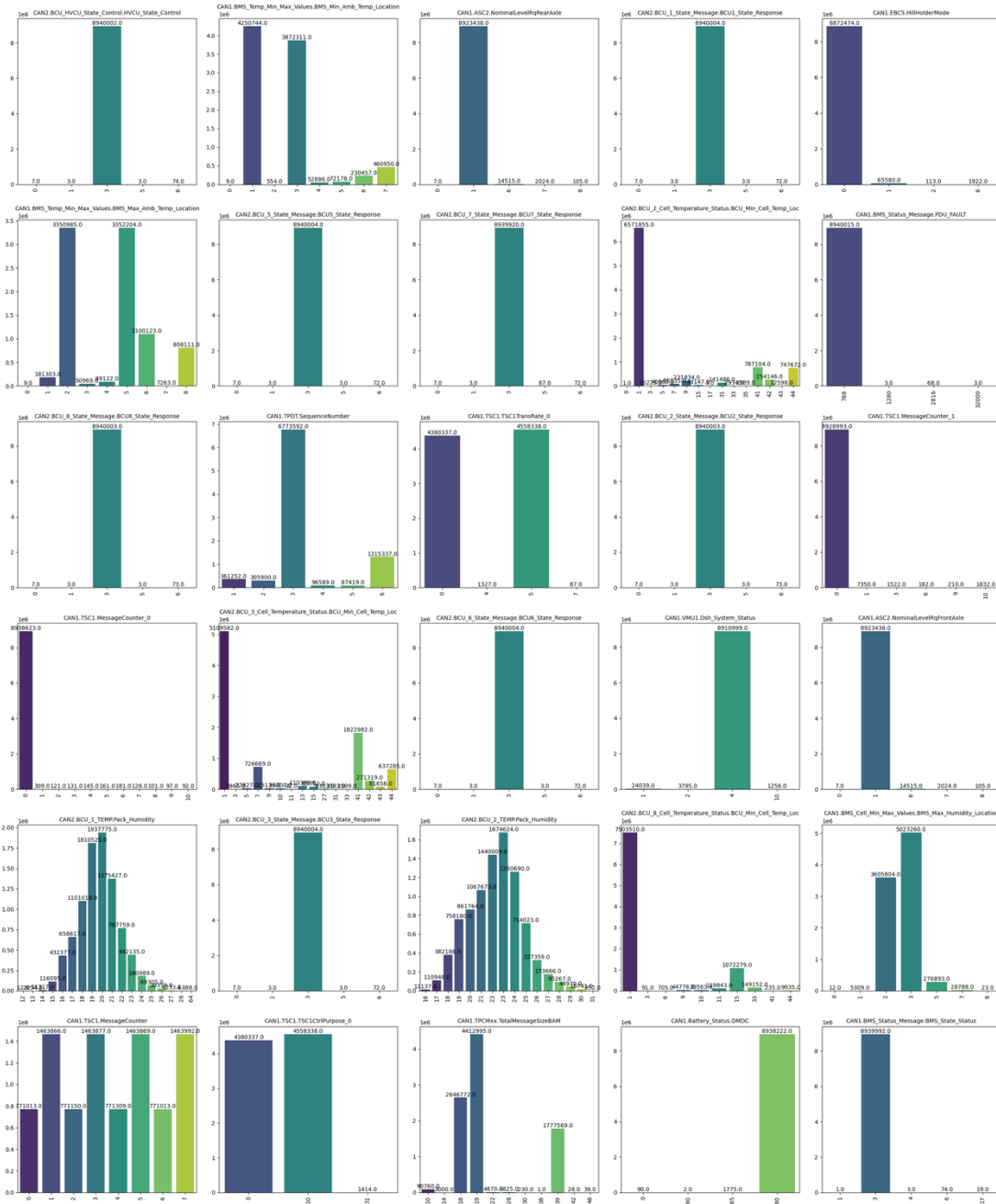
Çizelge 4.3: 902 Değişkenin istatistiksel analiz tablosu (ilk 25 değişken).

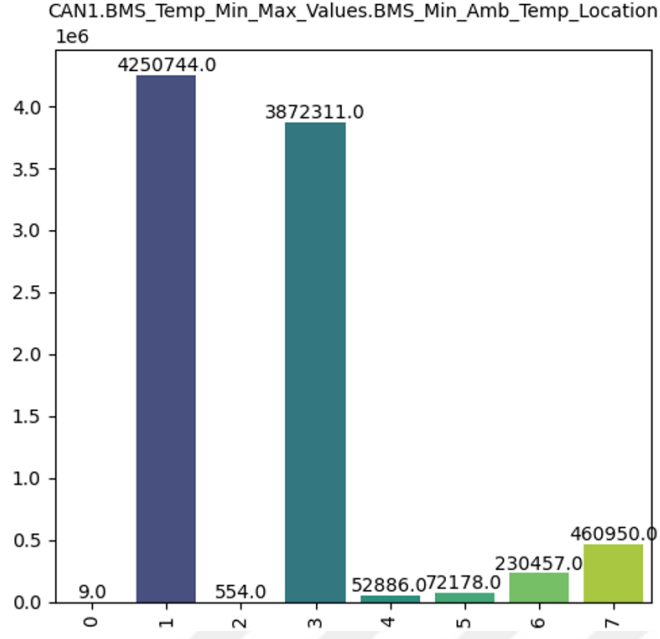
	Değişken Adı	Veri Tipi	Değişken Dağılımı	min	max	Kategorik/Sürekli	Tek Değer İlerme	Ortalama	Standart Sapma
0	CAN1.DCU_PSU_1.Clamp15_Status_14	<class 'numpy.float64'>	[CAN1.DCU_PSU_1.Clamp15_Status_14': [0: 0.0, ...	0	1	categorical	no	0.985636	0.018061
1	CAN1.TSC1.OverrideCtrlModePriority_4	<class 'numpy.float64'>	[CAN1.TSC1.OverrideCtrlModePriority_4': [0: 0...	0	0	categorical	yes	0	0
2	CAN2.BCU_4_Fault_Message.OverTempFaultAICharge	<class 'numpy.float64'>	[CAN2.BCU_4_Fault_Message.OverTempFaultAIChar...	0	0	categorical	yes	0	0
3	CAN2.BCU_8_Fault_Message.InternalMemoryFault	<class 'numpy.float64'>	[CAN2.BCU_8_Fault_Message.InternalMemoryFault...	0	0	categorical	yes	0	0
4	CAN2.BCU_3_Cell_Total_Status.BCU_Total_Current	<class 'numpy.float64'>	[CAN2.BCU_3_Cell_Total_Status.BCU_Total_Curre...	-250	96.846	continuous	no	-1.787287	9.762908
5	CAN1.ASC2.DamperStiffnessRqLift_TagWle	<class 'numpy.float64'>	[CAN1.ASC2.DamperStiffnessRqLift_TagWle': [0...	0	0	categorical	yes	0	0
6	CAN1.BMS_Cel_Min_Max_Values.BMS_Max_Cel_Volt...	<class 'numpy.float64'>	[CAN1.BMS_Cel_Min_Max_Values.BMS_Max_Cel_Vo...	1	255.5	continuous	no	5.8095	0.960925
7	CAN2.BCU_7_State_Message.Balance_Enable	<class 'numpy.float64'>	[CAN2.BCU_7_State_Message.Balance_Enable': [0...	0	1	categorical	no	0.256837	0.014712
8	CAN1.EBC1.ASRBrakeCtrlActive	<class 'numpy.float64'>	[CAN1.EBC1.ASRBrakeCtrlActive': [0: 0.0, 1: 1...	0	1	categorical	no	0.000001	0.000028
9	CAN2.BCU_7_TEMP.Middle_Cell_1_Terminal_Temp	<class 'numpy.float64'>	[CAN2.BCU_7_TEMP.Middle_Cell_1_Terminal_Temp...	-5	35	continuous	no	18.757935	0.089043
10	CAN2.BCU_1_Fault_Message.InternalMemoryFault	<class 'numpy.float64'>	[CAN2.BCU_1_Fault_Message.InternalMemoryFault...	0	0	categorical	yes	0	0
11	CAN1.AC.FunctionInstance	<class 'numpy.float64'>	[CAN1.AC.FunctionInstance': [0: 0.0, 1: 'nan'...	0	0	categorical	yes	0	0
12	CAN1.VMU1.Dsh_System_Fault	<class 'numpy.float64'>	[CAN1.VMU1.Dsh_System_Fault': [0: 0.0, 1: 1.0...	0	1	categorical	no	0.001113	0.001245
13	CAN2.BCU_1_State_Message.BCU1_Contactor_Status	<class 'numpy.float64'>	[CAN2.BCU_1_State_Message.BCU1_Contactor_Sta...	0	1	categorical	no	0.979251	0.018908
14	CAN2.BCU_4_Battery_Status.BCU_SOC	<class 'numpy.float64'>	[CAN2.BCU_4_Battery_Status.BCU_SOC': [0: 0.0...	0	100	categorical	no	78.674348	0.691819
15	CAN2.TSC1.TSC1CtrlPurpose_0	<class 'numpy.float64'>	[CAN2.TSC1.TSC1CtrlPurpose_0': [0: 0.0, 1: 'n...	0	0	categorical	yes	0	0
16	CAN1.EBC5.FoundationBrakeUse	<class 'numpy.float64'>	[CAN1.EBC5.FoundationBrakeUse': [0: 0.0, 1: 1...	0	1	categorical	no	0.073563	0.184968
17	CAN1.Battery_Status.DMCC	<class 'numpy.float64'>	[CAN1.Battery_Status.DMCC': [0: 0.0, 1: 7.0...	0	104	categorical	no	25.614155	1.549936
18	CAN1.ETC1.TransInputShaftSpeed	<class 'numpy.float64'>	[CAN1.ETC1.TransInputShaftSpeed': [0: 0.0, 1...	0	3034.622	continuous	no	459.918705	312.51563
19	CAN2.BCU_3_TEMP.Positive_Terminal_Cell_Temp	<class 'numpy.float64'>	[CAN2.BCU_3_TEMP.Positive_Terminal_Cell_Temp...	-5	39	continuous	no	17.541698	0.144908
20	CAN1.TSC1.TSC1CtrlPurpose	<class 'numpy.float64'>	[CAN1.TSC1.TSC1CtrlPurpose': [0: 0.0, 1: 10.0...	0	31	categorical	no	30.845141	0.000306
21	CAN2.BCU_5_Fault_Message.InternalMeasurementFault	<class 'numpy.float64'>	[CAN2.BCU_5_Fault_Message.InternalMeasurement...	0	0	categorical	yes	0	0
22	CAN1.VDC1.ROPBrakeCtrlActive	<class 'numpy.float64'>	[CAN1.VDC1.ROPBrakeCtrlActive': [0: 0.0, 1: '...	0	0	categorical	yes	0	0
23	CAN2.BCU_1_TEMP.Pack_Ambient_Temp	<class 'numpy.float64'>	[CAN2.BCU_1_TEMP.Pack_Ambient_Temp': [0: -6.0...	-6	32	continuous	no	18.301474	0.065781
24	CAN2.BCU_4_State_Message.BCU4_Contactor_Status	<class 'numpy.float64'>	[CAN2.BCU_4_State_Message.BCU4_Contactor_Sta...	0	1	categorical	no	0.980875	0.018971
25	CAN1.ASC2.AutomaticTractionHelp	<class 'numpy.float64'>	[CAN1.ASC2.AutomaticTractionHelp': [0: 0.0, 1...	0	0	categorical	yes	0	0

4.2.2. Kategorik değişkenlerin incelenmesi

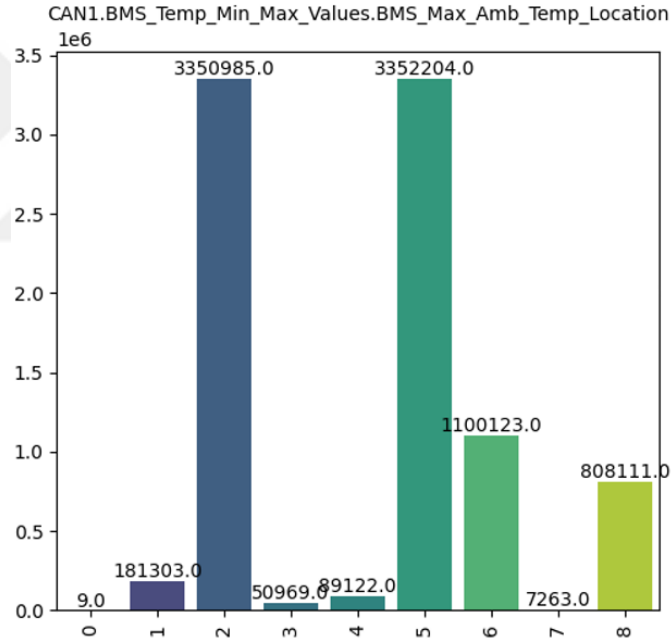
Kategorik değişkenlerin dağılımlarını ve kategorilerin sıklıklarını gözlemlemek için birden fazla sınıftan oluşan kategorik değişkenler üzerinden frekans tabloları (Çizelge 4.4.) oluşturulmuştur. Değişkenlerin büyük bir bölümünde bir sınıfın diğer sınıflara göre sayıca büyük bir üstünlük gösterdiği gözlemlenmiştir. Birçok sınıfa sahip olan ve belirli bir dağılım gösteren “sürekli kategorik değişkenler” tespit edilmiştir.

Çizelge 4.4: Kategorik değişkenlerin frekans tablosu.





Şekil 4.1: Batarya paketlerinin minimum sıcaklık gösterme frekansı.



Şekil 4.2: Batarya paketlerinin maksimum sıcaklık gösterme frekansı.

4.2.2.1. Chi-Square testi ve Cramer's V istatistik tablosu ile ilişki analizi

Kategorik değişkenler arasındaki ilişkiyi incelemek ve bu ilişkilerin gücünü ölçmek için Cramér's V istatistiği kullanılmıştır. Cramér's V, iki kategorik değişken arasındaki ilişki gücünü ölçen bir istatistiktir ve Chi-Square testine dayanmaktadır. Chi-Square testi, iki kategorik değişkenin bağımsız olup olmadığını test etmek için kullanılır. Ancak, Chi-Square testi yalnızca ilişki olup olmadığını belirlerken, Cramér's V bu ilişkinin gücünü ölçmektedir. Cramér's V istatistiği ile, iki kategorik değişken

arasındaki ilişkinin gücü 0 ile 1 arasında bir değerle ifade edilir. 0 değeri, değişkenler arasında ilişki olmadığını, 1 değeri ise mükemmel bir ilişki olduğunu göstermektedir. Cramér's V hesaplaması Eşitlik (4.1) ile gösterildiği şekildedir.

$$V = \sqrt{\left(\frac{(x^2/n)}{(\min(k-1, r-1))} \right)} \quad (4.1)$$

Burada;

x^2 : Chi-Square Değeri

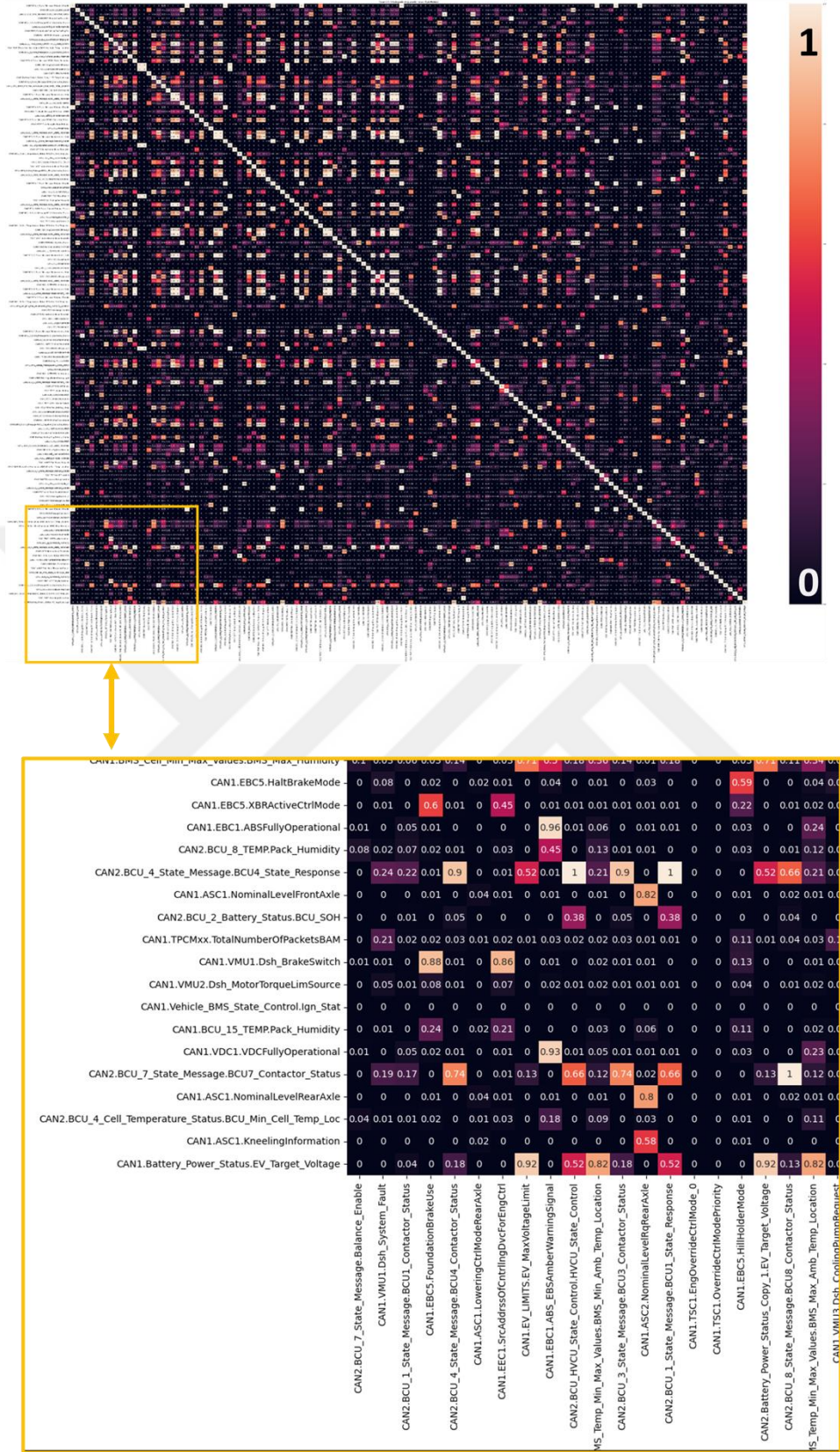
n : Toplam Örnek Sayısı

k : Değişken(sütun) Sayısı

r : Kayıt(satır) Sayısı' dır

Bu süreçte değişkenler arasındaki ilişkiyi belirlemek için öncelikle çapraz tablolar (contingency tables) oluşturulmuş, bu tablolardan elde edilen Chi-Square değerleri kullanılarak Cramér's V katsayıları hesaplanmış ve son olarak kategorik değişkenler arasındaki ilişkinin gücünü yansıtan ilişki matrisi oluşturulmuştur. Matristeki değerler incelenerek, hangi değişken çiftleri arasında daha güçlü ilişkiler olduğu belirlenebilmiştir.

Oluşturulan İlişki Matrisi Şekil 4.3 de görülmektedir. Şekilde 145 x 145 Bir matris olduğu görülmektedir. Matriste birbirleriyle güçlü ilişki içinde olan değişkenler açık renk hücreler (1.00) ile, zayıf ilişki içinde olan değişkenler koyu renk hücreler (0.00) ile gösterilmektedir. Oluşturulan matris, her bir kategorik değişken çifti arasındaki ilişkinin gücünü görselleştirmekte olup, aracın 2 yıllık tüm hareketini kapsayan zamanlarda (aracın durduğu ve hareket halinde olan tüm zaman aralıkları) kategorik değişkenler arasındaki ilişkilerde bir değişim olup olmadığını incelemek amacıyla kullanılmıştır.



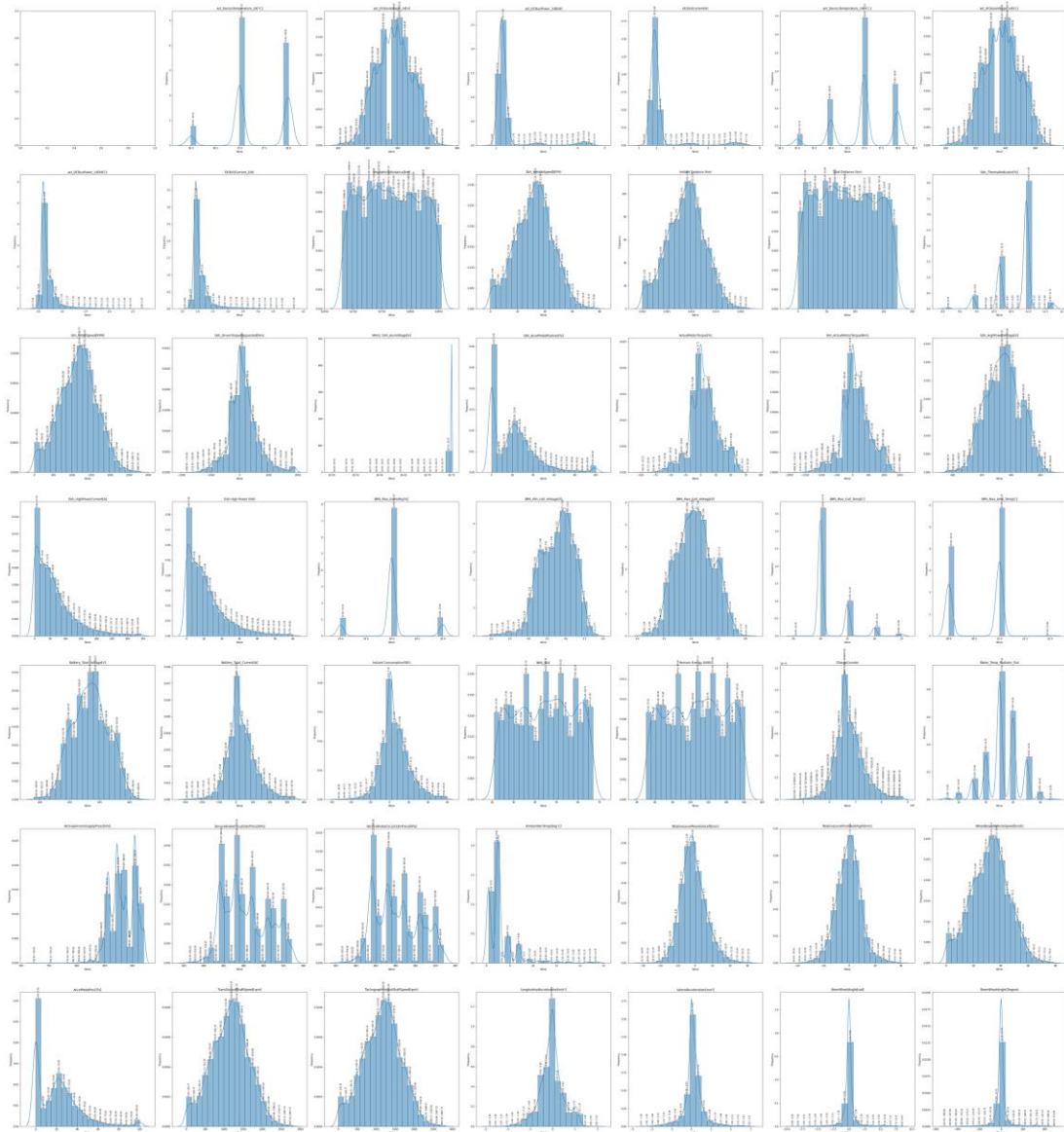
Şekil 4.3: Cramér's V kategorik değişkenler arası ilişki.

4.2.3. Sürekli değişkenlerin analizi

4.2.3.1. Sürekli değişken dağılım grafikleri

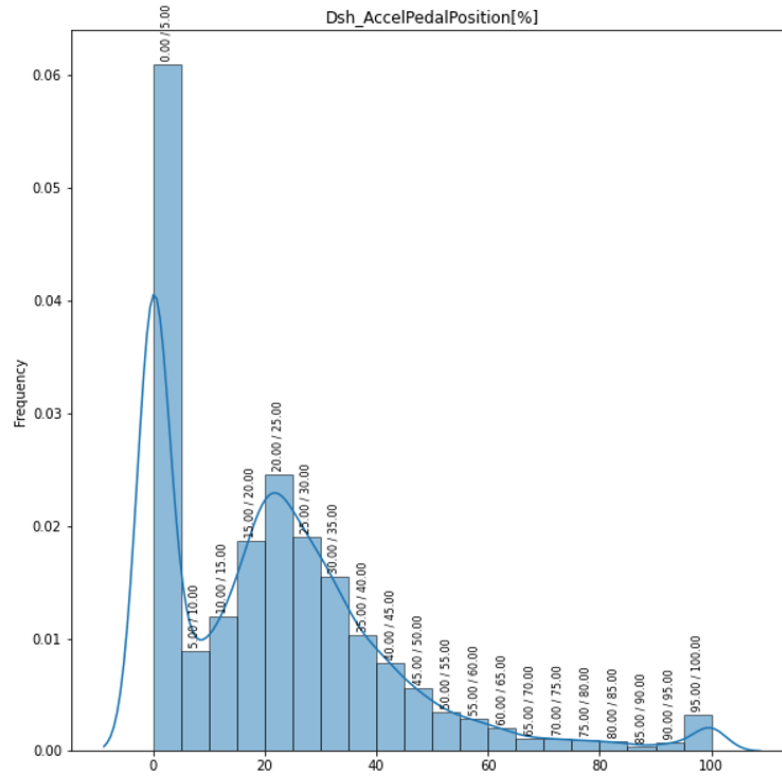
Sürekli veriler içerisinde standart sapması en yüksek 48 değişkenin histogram grafikleri çizilerek dağılımları incelenmiş, normal dağılımdan sapmaları belirlenmiştir. Sağa veya sola çarpık dağılım sergileyen değişkenler incelenmiş, bazı değişkenlerin ortalama değerlerinin medyan değerlerinden farklı olduğu ve verilerin belirli bir yönde yoğunlaştığı tespit edilmiştir.

Çizelge 4.5: Standart sapması yüksek olan 48 sürekli değişkenin histogram grafikleri.

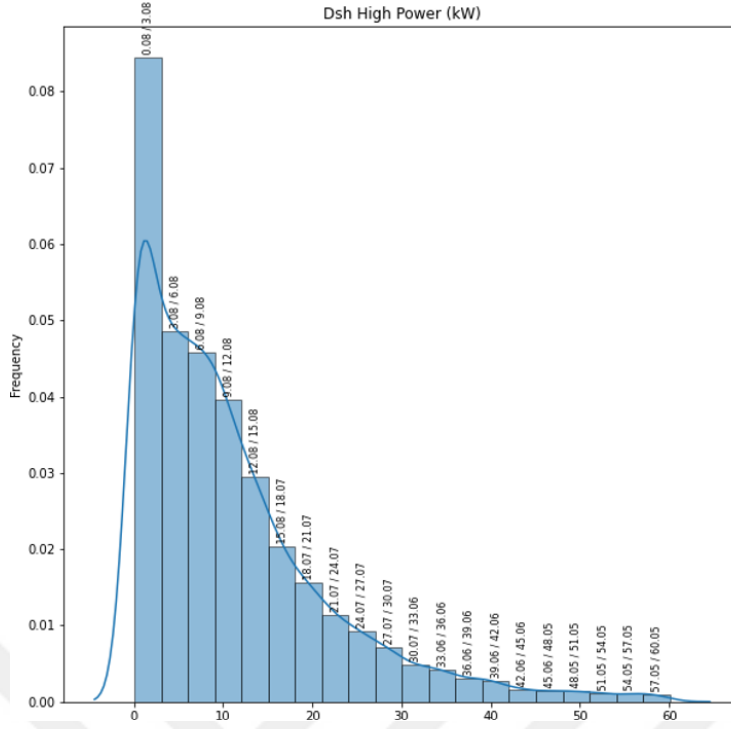


Sürekli değişkenlerin dağılım analizine örnek olarak, aracın hızlanma pedal pozisyonunun ne kadar basıldığını gösteren ve genellikle yüzdelik (%0 ile %100

arasında) bir değer olarak ifade edilen Dsh_AccelPedalPosition[%] ve aracın yüksek güç tüketimini ifade eden, yüksek güç gerektiren durumlarda (örneğin, hızlanma, yüksek hızda seyir, yokuş çıkma gibi) ne kadar enerji harcadığını gösteren Dsh High Power (kW) değişkenlerinin dağılımı Şekil 4.4 ve Şekil 4.5’deki gibi oluşturularak uzman görüşüne sunulmuştur. Bu değişkenlerin sağa çarpık dağılım sergilediği ve bunun nedeninin otobüsün çoğunlukla düşük güç tüketimiyle çalışmış ve yüksek güç tüketimi gerektiren durumlarla nadiren karşılaşmış olduğu belirlenmiştir.



Şekil 4.4: Dsh_AccelPedalPosition [%] değişkenin dağılım grafiği.



Şekil 4.5: Dsh high power (kW) değişkenin dağılım grafiği.

4.2.4. Sürekli değişkenler arasındaki ilişkinin pearson korelasyon analizi

Sürekli değişkenler arasındaki ilişkiyi incelemek ve bu ilişkilerin yönünü ve gücünü ölçmek için Pearson korelasyon katsayısı kullanılmıştır. Pearson korelasyon katsayısı, iki sürekli değişken arasındaki doğrusal ilişkinin yönünü ve gücünü -1 ile +1 arasında bir değerle ifade etmektedir. -1 değeri, değişkenler arasında mükemmel negatif doğrusal bir ilişki olduğunu, +1 değeri ise mükemmel pozitif doğrusal bir ilişki olduğunu ve 0 değeri ise ilişki olmadığını göstermektedir. Pearson korelasyon katsayısı hesaplanışı Eşitlik (4.2) ile gösterildiği şekildedir.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.2)$$

Burada;

r : Korelasyon Katsayısı

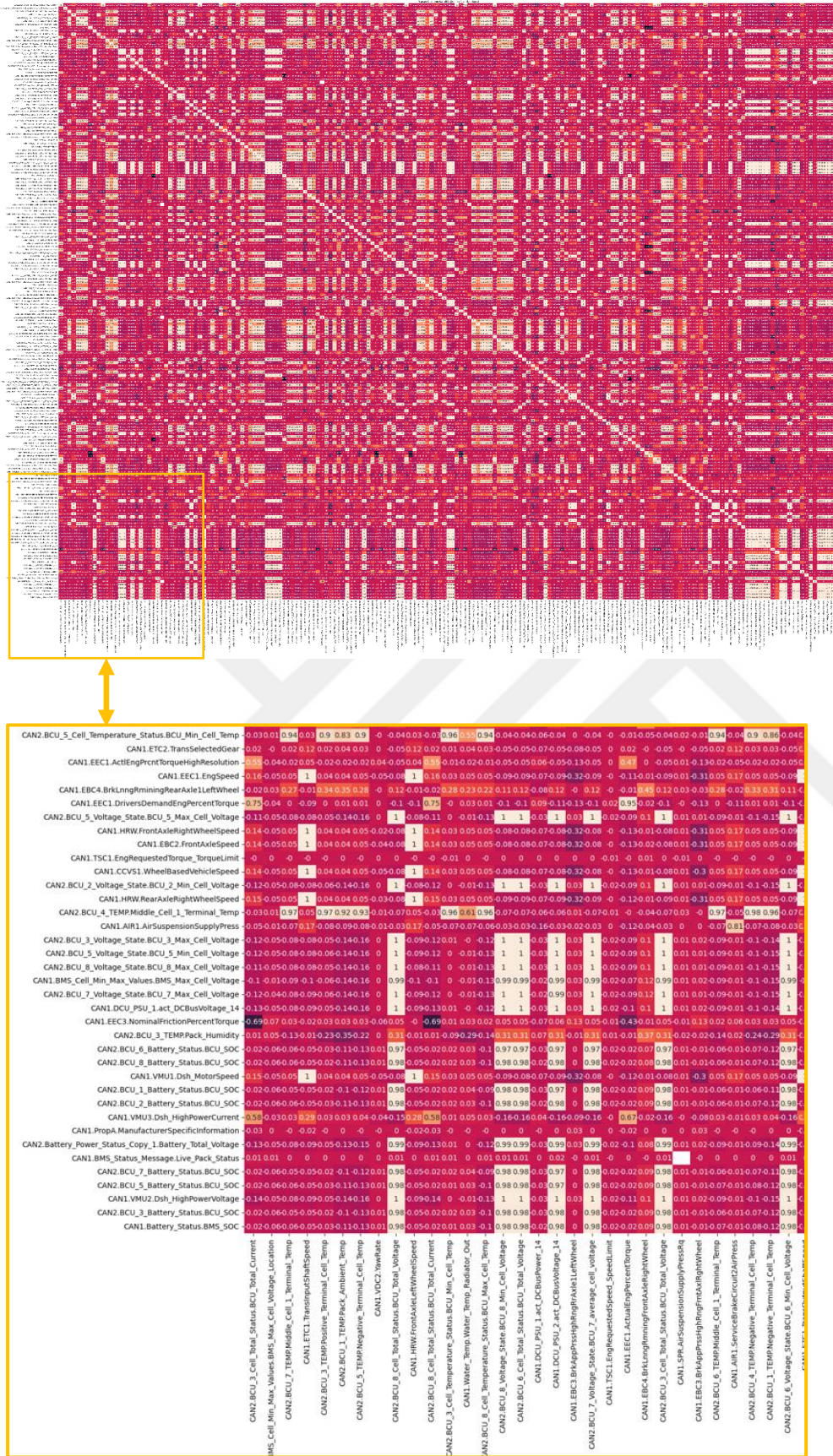
x_i ve y_i : x ve y değişkenlerinin i -inci gözlemleri

$\bar{x}$ ve $\bar{y}$: x ve y değişkenlerinin ortalamaları.

Bu süreçte, Pearson korelasyon katsayısı hesaplaması yapılarak sürekli değişkenler arasındaki ilişki matrisi oluşturulmuştur. Matristeki değerlerin pozitif olması,

değişkenlerin eş zamanlı olarak arttığını veya azaldığını, negatif olması ise bir değişkenin artarken diğerinin azaldığını göstermektedir. Değerin mutlak değerinin yüksek olması, ilişkinin güçlü olduğunu gösterirken düşük olması ise ilişkinin zayıf olduğunu göstermektedir.

Şekil 4.6’da 190 x 190 boyutunda Pearson Korelasyon katsayılarından oluşturulmuş bir matris olduğu görülmektedir. Matriste birbirleriyle güçlü ve pozitif yönde ilişki içinde olan değişkenler açık renk hücreler (1.00) ile, güçlü ve negatif yöne ilişki içinde olan değişkenler koyu renk hücreler (-1.00) ile gösterilmektedir. Oluşturulan matris, her bir sürekli değişken çifti arasındaki ilişkinin yönünü ve gücünü görselleştirmekte olup, aracın 2 yıllık tüm hareketini kapsayan zamanlarda (aracın durduğu ve hareket halinde olan tüm zaman aralıkları) sürekli değişkenler arasındaki ilişkilerde bir değişim olup olmadığını incelemek amacıyla kullanılmıştır.



Şekil 4.6: Sürekli değişkenlerin korelasyon matrisi.

4.3. Veri Ön İşleme ve Anomali Analizi

Elektrikli otobüse ait CAN veri seti, yüksek hacimli ve karmaşık bir yapıya sahiptir. Bu nedenle, verilerin makine öğrenimi modellerine uygun hale getirilmesi için kapsamlı bir veri ön işleme süreci gerçekleştirilmiştir. Bu süreçte, ham veri analizinden yararlanılarak, ham veri üzerinde sadeleştirme ve dönüştürme işlemleri uygulanarak kapsayıcı değişken kümesi oluşturulmuştur. Bu aşamada, verilerin kalitesini artırmak ve ayırt edici özellik seçiminin karmaşıklığını azaltarak modellerin tahmin performansını iyileştirmek ve anlamlı sonuçlar elde etmek amaçlanmıştır.

4.3.1. Veri Sadeleştirme

Referans olarak alınan, **3F2551C1** veri seti 80.2 GB olup, 35607 adet csv uzantılı veri paketi içermektedir. Verinin karmaşık yapısı ve yüksek hacmi, analiz ve modelleme süreçlerinde işlem yükünü artırarak, hatalı sonuçlar üretme riskini yükseltmekte ve süreçlerin verimliliğini düşürmektedir.

Bu süreçte ilk olarak, veri setindeki değişken (sütun) sayısını azaltmak hedeflenmiştir. Bu kapsamda, ham veri analizi sürecinde gerçekleştirilen istatistiksel analizler incelenerek sapma göstermeyen sürekli değişkenler belirlenmiştir. Benzer yaklaşımla, kategorik değişkenlerin analiz sürecinde oluşturulan frekans tabloları incelenerek tüm kayıtlarda tek bir kategorik değer içeren değişkenler belirlenmiştir. Bu değişkenler, aracın hareket sürecinde herhangi bir değişim göstermedikleri için modelin tahmin doğruluğunu veya performansını artırmaya yönelik anlamlı bilgi sağlamayacağı düşünüldükçe uzman görüşü alınarak veriden çıkartılmıştır. Ayrıca, uzman görüşleri doğrultusunda, CAN Bus (frame) bilgileri içeren değişkenler, aracın çalışma sürecinden ziyade iletişim protokollerine ilişkin mesaj boyutu, doğrulama kodları (checksum), bayrak (flag) ve benzeri teknik parametreleri içerdiği için, tahmin modelinin performansını etkilemeyeceği düşünüldükçe veri setinden çıkarılmıştır.

Bu süreçte ikinci olarak, veri setindeki kayıt (satır) sayısını azaltarak öznitelik analizi ve model performansı için veriyi kalitesinin iyileştirilmesi amaçlanmıştır. Bu kapsamda, aracın durduğu zaman aralıklarına ait veriler, ayırt edici özniteliklerin belirlenme sürecine ve tahmin modellerinin öğrenme sürecine anlamlı bir katkı sağlayamayacağı ve hatta modellerinin performansını olumsuz yönde etkileyebileceği düşünüldükçe (uzman görüşü alınarak) veri setinden çıkartılmıştır. Aracın durduğu aralıklar tespit edilirken, uzman görüşü doğrultusunda seçilen

Dsh_VehicleSpeed[KPH] değişkeni ve sürekli değişkenler üzerinde yapılan korelasyon analizi baz alınarak bu değişkene yüksek derecede pozitif korele olan Dsh_ActualMotorSpeed [RPM] ve CAN1.CCVS1.WheelBasedVehicleSpeed değişkenleri dikkate alınmıştır. Bu değişkenlerin 0 olduğu aralıklar aracın durduğu aralıklar olarak kabul edilerek, bu aralıklardaki kayıtlar veriden filtrelenmiştir.

4.3.2. Anomali Analizi

Elektrikli araçlarda, sistemlerin karmaşıklığı ve bileşenlerin birbirleriyle olan kompleks etkileşimleri, beklenmeyen durumların ortaya çıkması halinde etkileyen ve etkilenen bileşenlerin tespitini ve analizini zorlaştırmaktadır. Bu beklenmeyen durumların, aracın performansını, güvenliğini ve ömrünü olumsuz yönde etkileme ihtimali oldukça yüksektir. Bu nedenle, çalışmanın bu aşamasında, sadeleştirilmiş CAN verileri üzerinde anomali analizi gerçekleştirilerek, aracın yaklaşık 2 yıllık hareket süreci içerisinde performansını etkilemiş olabilecek değişken değişimlerini analiz etmek amaçlanmıştır. Bu analizde, kritik durumlarda aktif hale gelen, çalışma kapsamında belirlenen hedef alarmlar ile normal çalışma koşullarından sapma gösteren kullanıcı sensör verileri, batarya yönetim sistemi verileri ve diğer araç bileşenlerinden gelen bildirileri içeren değişkenler arasındaki ilişkiler incelenmiştir.

4.3.2.1. Hedef alarmların analizi

İlk olarak, bu analiz kapsamında, uzman görüşü ile aracın Fren, Şanzıman ve Batarya sistemlerinin durumu hakkında bilgi veren ve bu sistemlerin bileşenlerinde gözlemlenen kritik değişimlerde aktif hale gelen hedef değişkenlerin 5 saniye ve üzerinde aktif olduğu zaman dilimleri tespit edilmiş ve kaydedilmiştir.

4.3.2.2. Değişkenlerin sapma gösterdiği zamanların belirlenmesi

Bu adımda, alarmların dışındaki değişkenlerin ilişkili oldukları araç bileşenler incelenerek değişkenler Sistem Değişkenleri ve Batarya Değişkenleri olmak üzere iki gruba ayrılmıştır. Daha sonrasında her iki gruptaki değişkenlerin cuDF ve cuPy kütüphanelerinden yararlanılarak GPU üzerinde Z-skoru hesaplamaları gerçekleştirilmiştir. Değişkenlerin sapma gösterdiği zaman aralıkları tespit edilirken, Z-skor mutlak değerinin 3'ten büyük olduğu zaman aralıklarına ait veri noktaları diğer zamanlara ait veri noktalarına kıyasla daha anlamlı bir şekilde farklılık gösterdiği için

değişkenlerin bu veri noktalarının zaman dilimlerinde sapma gösterdiği kabul edilmiştir [68]. Z-Skoru hesaplanışı Eşitlik (4.3) ile gösterildiği şekildedir.

$$Z = \frac{(X-\mu)}{\sigma} \quad (4.3)$$

Burada:

X: Veri noktası

μ : Veri setinin ortalaması

(σ): Veri setinin standart sapmasıdır

Bu yöntemde, her bir değişken için ortalama ve standart sapma hesaplanarak, her bir veri noktasının bu değerlere göre ne kadar uzakta olduğu standartlaştırılmış bir puan (Z-skoru) ile ifade edilmiştir. Verilerin normal dağıldığı varsayımına dayanarak ve verilerin büyük çoğunluğunun (yaklaşık %99.7'si) ortalama ± 3 standart sapma içinde yer alacağını [69] yayınından referans alarak, bu sınırın dışındaki değerler anomali olarak kabul edilmiştir.

4.3.2.3. Hedeflenen alarmlar anomali gösteren değişkenlerin ilişkilendirilmesi

Bu süreçte hedeflenen alarmların aktif olduğu her bir 5 saniye ve üzeri zaman diliminde, Z-Skoru Belirlenen Sistem ve Batarya Değişkenlerinin ± 3 standart sapma içinde bir değere sahip olup olmadıkları gözlemlenmiş zaman diliminin yüzde kaçında değişkenlerin Z-skoru 3'ten büyük hesaplanmıştır.

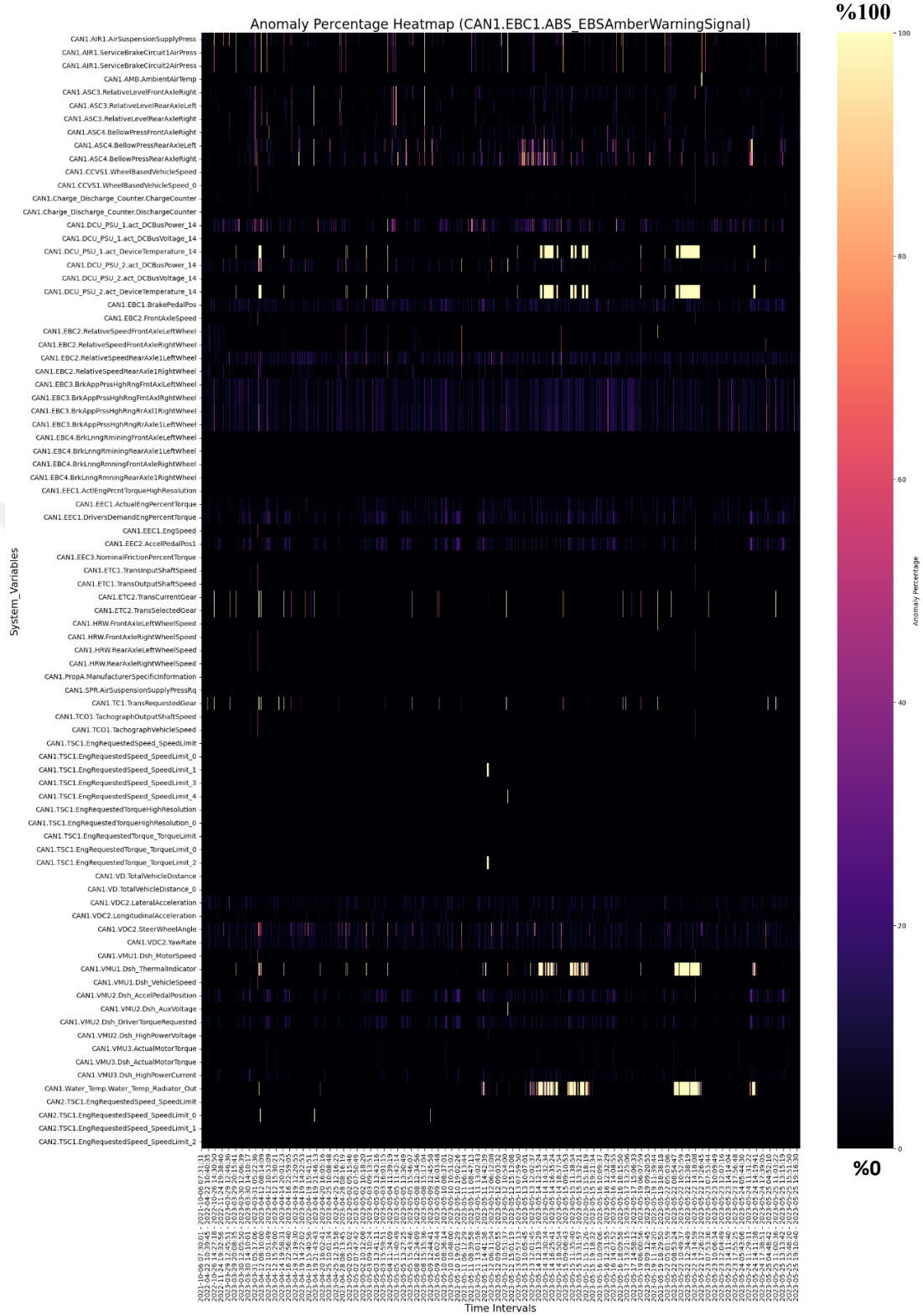
Daha sonrasında, her bir hedef değişkeninin aktif olduğu zaman aralıklarında, Sistem ve Batarya değişkenlerinin % kaç oranla ± 3 standart sapmanın üstünde değer gösterdiği tablolaştırılmış ve sıcaklık haritası üzerinde görselleştirilmiştir. Elde edilen sonuçlar, hangi değişkenlerin hangi hedef alarmları ile ilişkili olduğunu ve bu değişkenlerin kritik değişim gösterme olasılıklarının ne olduğunu belirlemek için kullanılmıştır.

4.3.2.4. Sonuçlar ve gözlemler

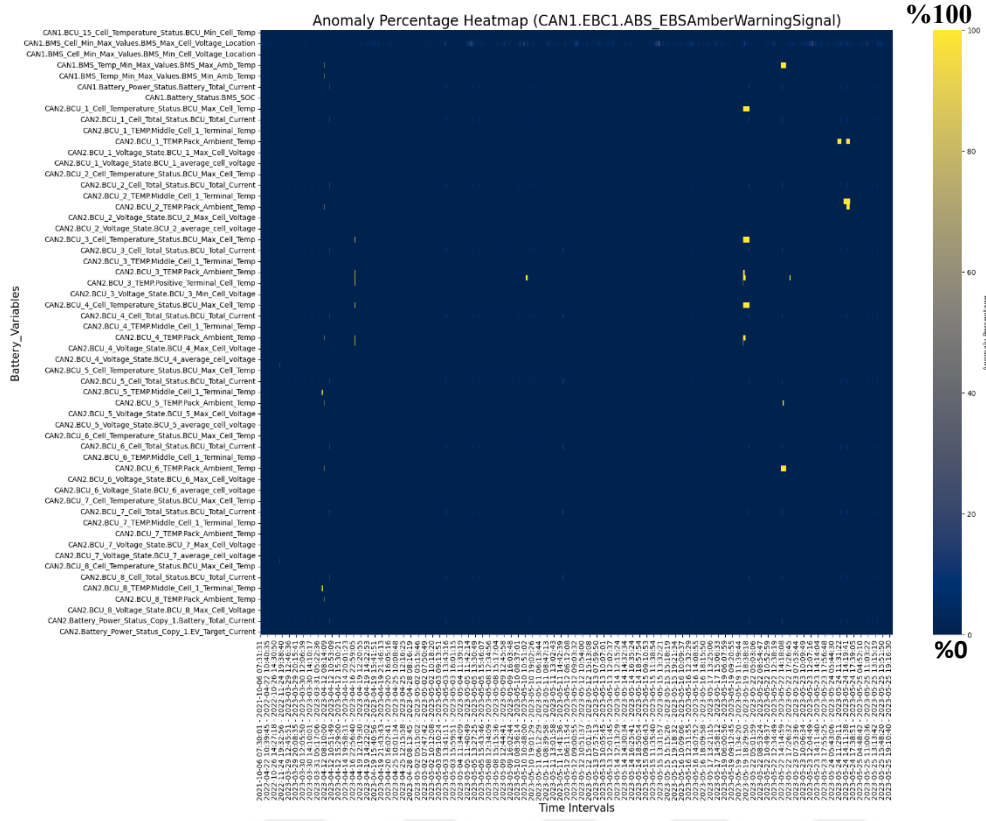
Hedef alarlara ait sıcaklık haritalarının x ekseninde, hedeflenen alarmların aktif olduğu 5sn'den uzun zaman aralıkları bulunmaktayken, y ekseninde bu aralıklarda anomali gösteren Sistem ve Batarya değişkenlerinin listesi bulunmaktadır. Haritadaki

her bir hücre ise Sistem ve Batarya Değişkenlerinin yüzde kaç oranda hedef alarmların aktif olduğu zamanda anomali gösterdiğini temsil eder. Açık Renkli hücreler değişkenlerin anomali frekansının yüksek olduğunu koyu renkli hücreler ise anomali frekansının düşük olduğunu belirtir.

Yapılan analizler sonucunda Şekil 4.7 ve Şekil 4.8’de görüldüğü üzere Amber Warning (Fren Sistemi) alarmlarının batarya değişkenlerinden daha çok sistem değişkenleriyle ilişki içinde olduğu gözlemlenmiştir. Alarmların aktive olduğu periyotların belli bir kısmında Dsh_Thermal_Indicator ,Water_Temp_Radiator_Out ve Device_Temperature değişkenleri yüksek sıklıkta anomali gösterdiği, aracın hareketine ve balata kalınlığına bağlı DCBusPower, RelativeSpeedFrontAxleLeft / RightWheel ve BrkLnngRmningFrontAxleLeft / RightWheel değişkenlerinin alarmların aktive olduğu neredeyse her periyotta düşük sıklıkta anomali gösterdiği tespit edilmiştir. Bununla beraber Dsh_Thermal_Indicator ,Water_Temp_Radiator_Out değişkenlerinin yüksek sıklıkta anomali gösterdiği alarm periyotlarında aynı zamanda BCU_Max_Cell_Temp, Terminal_Temp gibi batarya paketlerinin sıcaklığı ile alakalı olan değişkenlerin de yüksek sıklıkta anomali gösterdiği gözlemlenmiştir

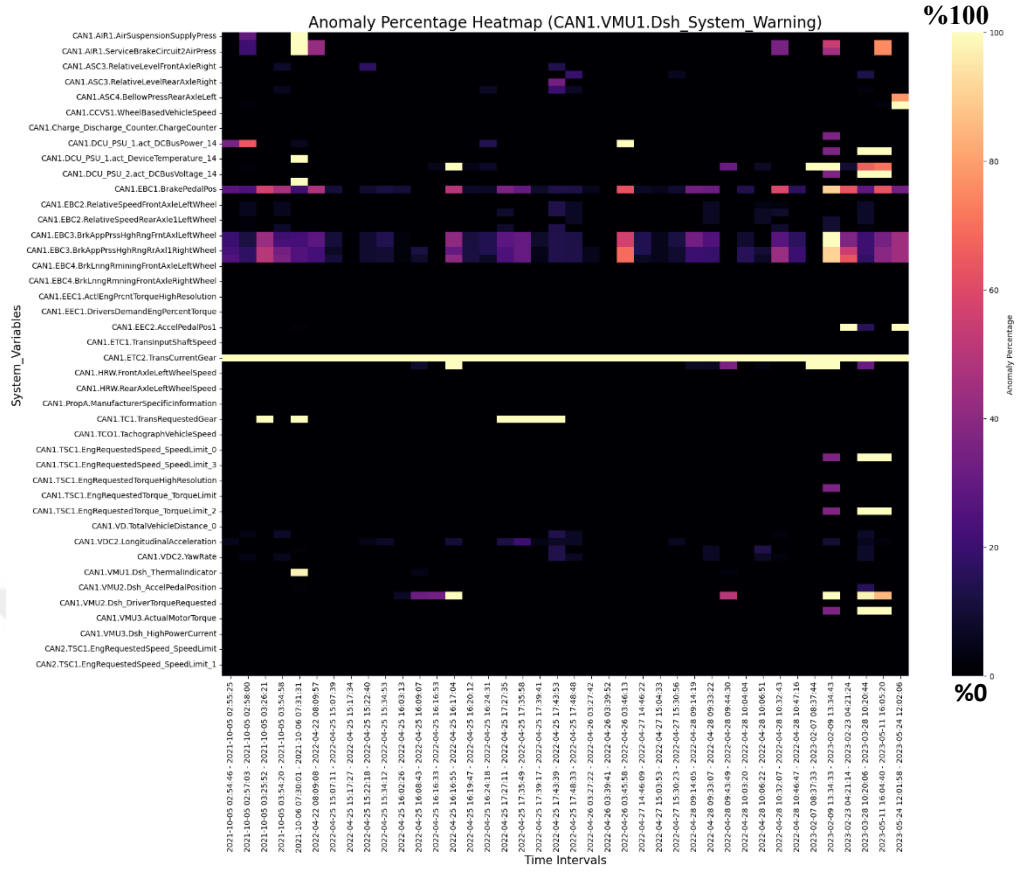


Şekil 4.7: Amber warning alarminın sistem değişkenleri anomali analizi.

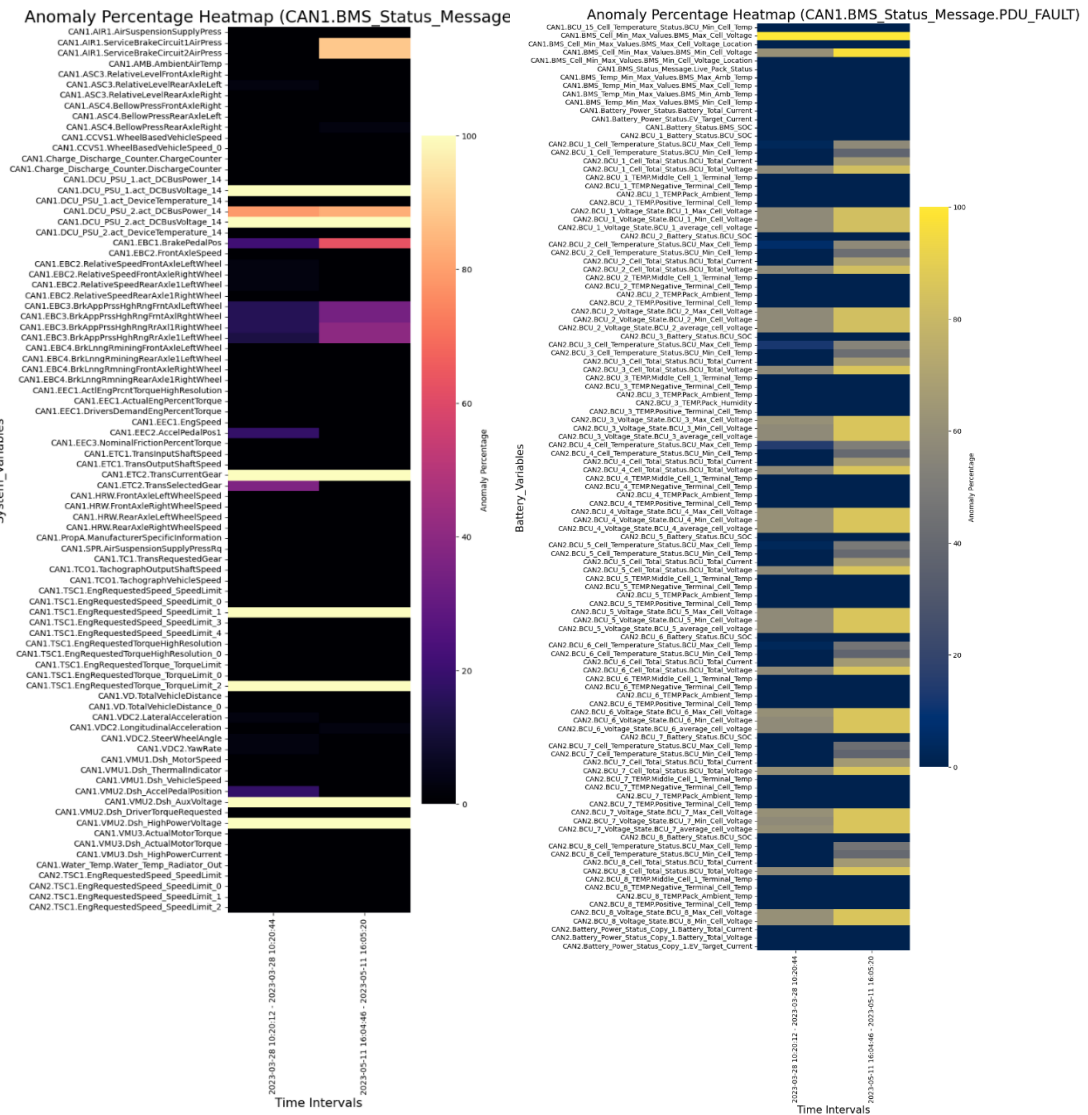


Şekil 4.8: Amber warning alarımının batarya değişkenleri anomali analizi.

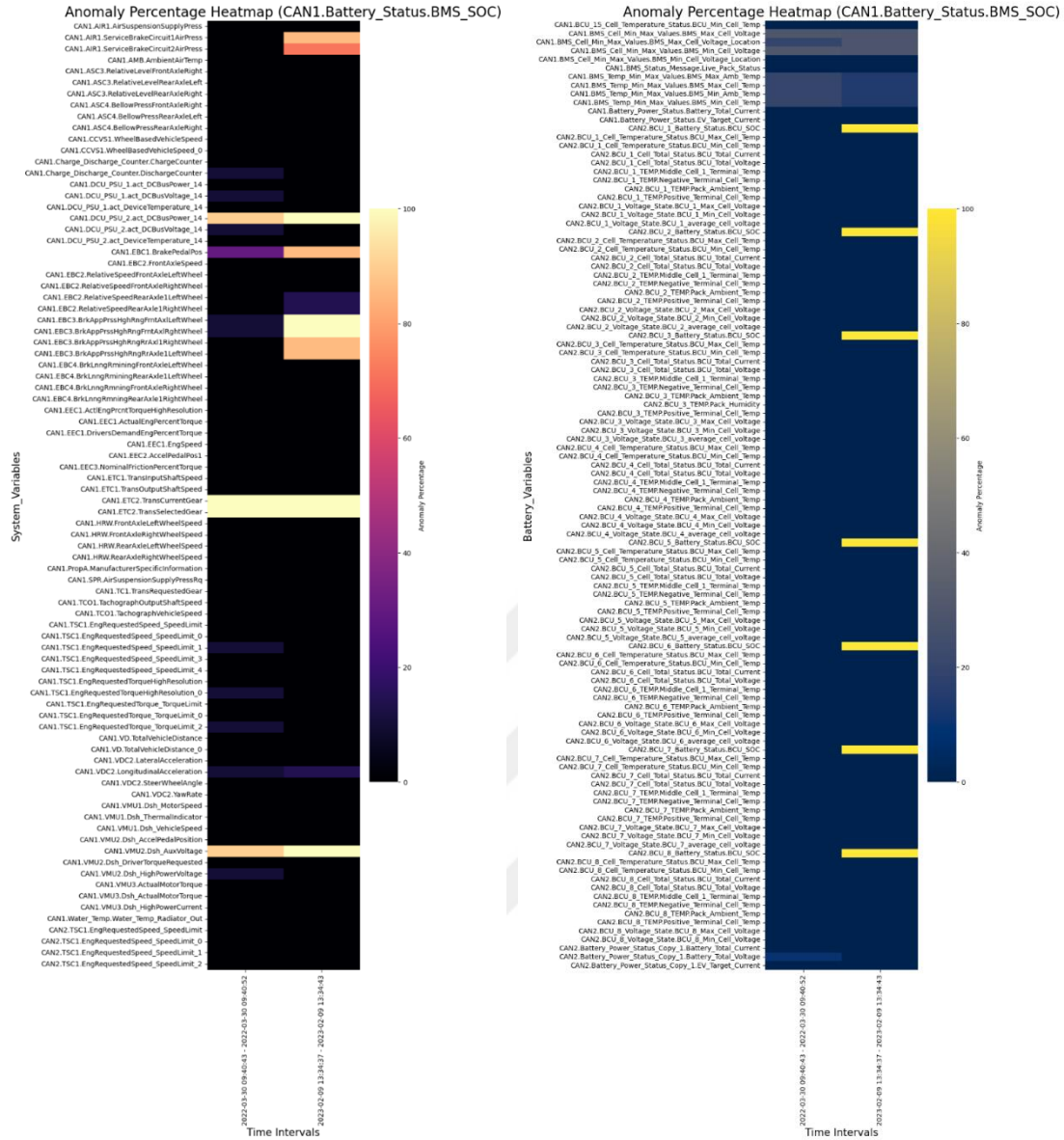
Sürücü konsoluna yansıtılan uyarı ve hataların ilişki olduğu Dsh_System_Warning ve Dsh_System_Fault genel alarmlarının aktive olduğu periyotlar incelendiğinde alarmların neredeyse eş olduğu aynı periyotlarda aynı değişkenlerle ilişkili oldukları gözlemlenmiştir. Şekil 4.9 ve 4.10’da görüldüğü üzere iki alarımında aktive olduğu periyotların bazılarında batarya değişkenleri ile bazılarında ise sistem değişkenleri ile ilişkili olduğu, aracın fren, balata kalınlığı ve batarya bileşenlerinin değişkenlerinin belli zaman periyotlarında yüksek sıklıkta anomali gösterdiği görülmüştür. Alarmların TransCurrentGear değişkeninin anomali durumları ile yüksek derecede korele olacak şekilde aktive olduğu tespit edilmiş, alarmların aktive olmasının TransCurrentGear değişkeninin anomali göstermesini tetiklediği konu uzmanlarca doğrulanmıştır.



Şekil 4.9: DSH system warning alarminın sistem değişkenleri anomali analizi.



Şekil 4.11: PDU fault alarımının sistem ve batarya değişkenleri anomali analizi.



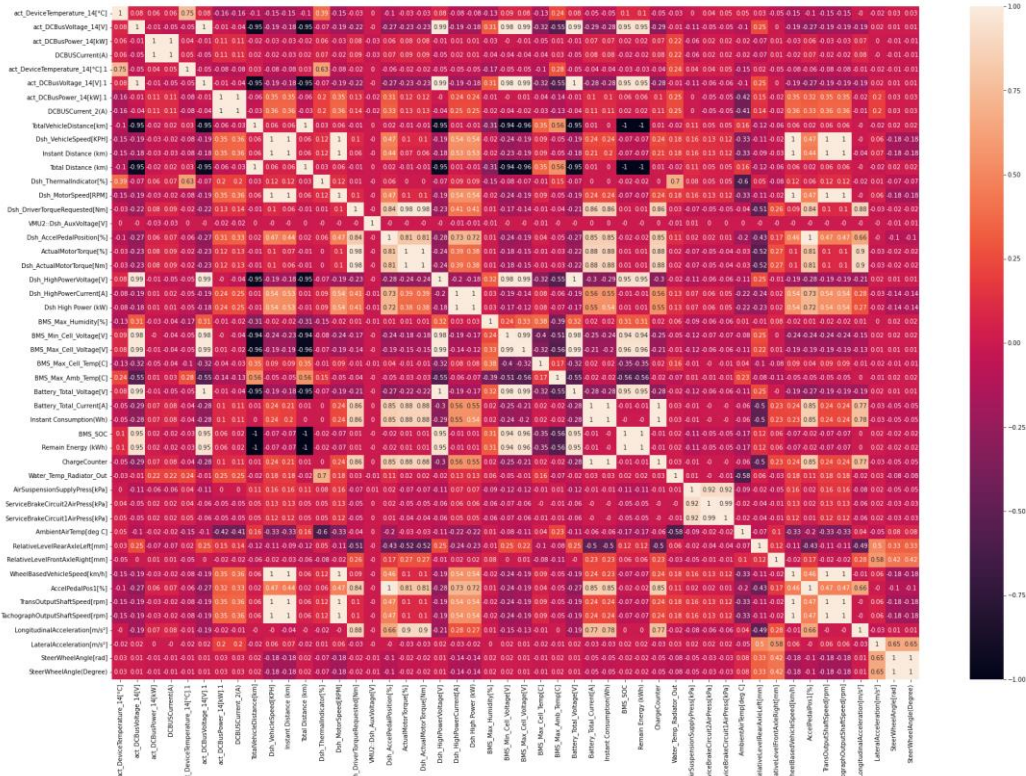
Şekil 4.12: BMS SOC alarmı sistem ve batarya değişkenleri anomali analizi.

4.3.3. Sadeleştirilmiş veri üzerinde yapılan istatistiksel analizler

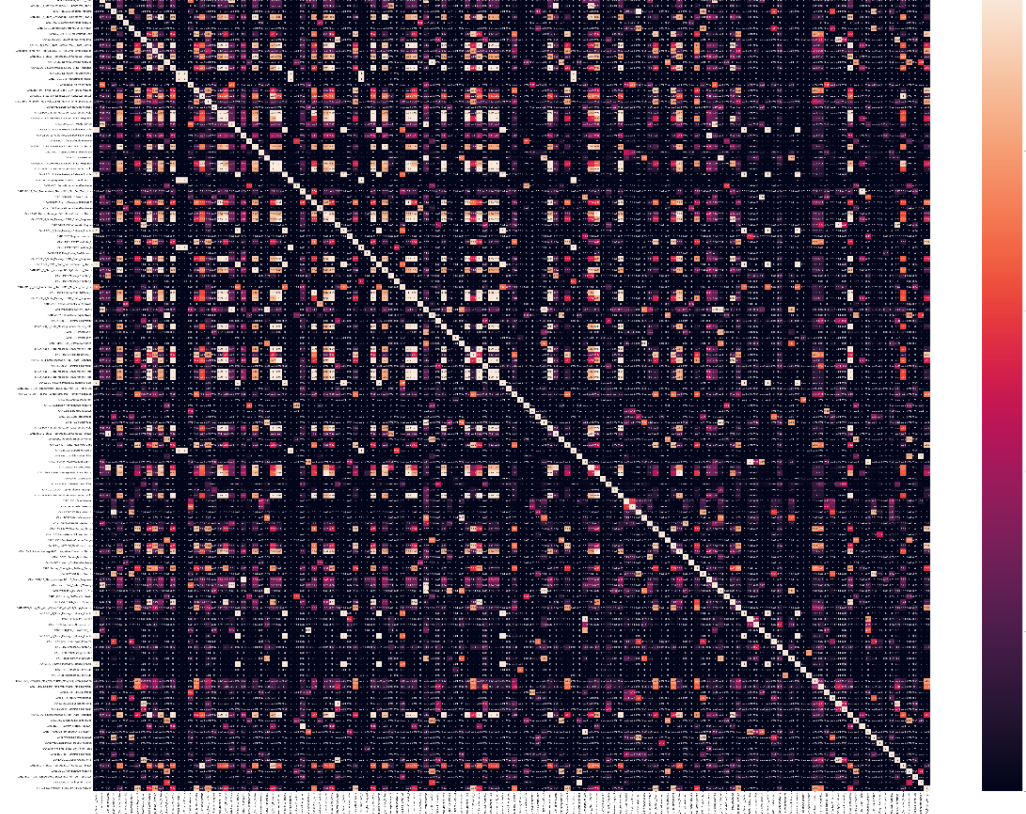
Bu aşamada, ham veri üzerinde yapılan sadeleştirme işlemleri kapsamında güncellenen ve sadece aracın hareket ettiği zaman dilimlerini kapsayan veri setine ait değişkenlerin kendi aralarındaki ve hedeflenen alarmlarla olan istatistiksel ilişkileri incelenmiş ve görselleştirilmiştir.

Bu kapsamda, ilk olarak ham veri üzerinde gerçekleştirilen analizler, sadeleştirilmiş veri üzerinde tekrarlanmıştır. Sadeleştirme işlemleri ile filtrelenen sürekli değişkenler için Pearson korelasyon katsayısı tekrardan hesaplanmış ve korelasyon matrisi oluşturulmuştur. Kategorik değişkenler arasındaki ilişkiler içinse ise Cramer's V istatistiği hesaplanmış ve ilişki matrisi oluşturulmuştur.

Çizelge 4.6: Sadeleştirilmiş sürekli değişkenlerin pearson korelasyon matrisi.



Çizelge 4.7: Sadeleştirilmiş kategorik değişkenlerin cramer's V ilişkisi matrisi.



Daha sonrasında, hedeflenen alarmların ikili (binary) yapıda olması ve diğer değişkenlerin hem sürekli hem sıralı (ordinal) değişkenlerden oluşması nedeniyle, alarmların diğer değişkenlerle olan ilişkilerini belirlemek için Anova F-testi kullanılmış ve ilişkilerin Anova F-Skor Tabloları oluşturulmuştur. Bu test alarmların sınıfları (alarm durumu; aktive-1:aktive değil-0) içerisinde, sürekli değişkenlerin ortalamalarında anlamlı farklılıklar olup olmadığını belirlemeye yardımcı olmuştur. Anova F-Testinin Adımları Algoritma 4.1’de bulunan, Eşitlik (4.4-4.8) ile gösterildiği şekildedir.



Algoritma 4.1: Anova F-testinin adımları.

1.Hipotezlerin Kurulması:

Null Hipotez(H0):

Tüm grupların ortalamaları eşittir, gruplar arasında anlamlı bir fark yoktur.

$$H0: \mu_1 = \mu_2 = \dots = \mu_k \quad (4.4)$$

Alternatif Hipotez(H1):

En az bir grup diğerlerinden anlamlı olarak farklıdır.

H1: En az bir μ_i diğerlerinden farklıdır.

2.Varyansların Hesaplanması:

Gruplar Arası Varyans Ölçümü:

Farklı grupların ortalamaları arasındaki fark ölçülür.

$$SSB = \sum_{i=1}^k n_i (\bar{X}_i - \bar{X})^2 \quad (4.5)$$

Grup İçi Varyans Ölçümü:

Her bir grubun kendi içindeki veri noktaları arasındaki fark ölçülür.

$$SSW = \sum_{i=1}^k \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2 \quad (4.6)$$

3. F İstatistiğinin Hesaplanması:

F- İstatistiği, gruplar arası varyansın grup içi varyansa oranı olarak hesaplanır.

$$F = \frac{MSB}{MSW} \quad (4.7)$$

$$MSB = \frac{SSB}{k-1} \text{ ve } MSW = \frac{SSW}{N-k} \quad (4.8)$$

Burada,

SSB: Gruplar arası kareler toplamı (Sum of Squares Between Groups)

SSW: Grup içi kareler toplamı (Sum of Squares Within Groups)

MSB: Gruplar arası ortalama kare (Mean Square Between Groups)

MSW: Grup içi ortalama kare (Mean Square Within Groups)

k: Grup sayısı

N: Toplam gözlem sayısı

n_i: i'inci gruptaki gözlem sayısı

$\bar{X}_i$: i' inci grubun ortalaması

$\bar{X}$: Tüm grupların genel ortalaması

X_{ij}: i' inci gruptaki j' inci gözlem

Çizelge 4.8: Sürekli değişkenler Anova F-Skor tablosu (İlk 25 değişken).

Index	Sürekli DeğişkenAdı	AmberWarningAlarmı		DSH_Sytem_FaultAlarmı		DSH_System_WarningAlarmı		PDU_FaultAlarmı		BMS_SOCAlarmı	
		Anova F-Skor	Normalize F-Skor	Anova F-Skor	Normalize F-Skor	Anova F-Skor	Normalize F-Skor	Anova F-Skor	Normalize F-Skor	Anova F-Skor	Normalize F-Skor
0	CAN2.BCU_3_Cell_Total_Status.BCU_Total_Current	610.0451629	0.000321789	569.1234619	0.000849186	638.2562568	0.000756201	685.0054332	3.76239E-05	25.26704303	8.65654E-05
1	CAN1.BMS_Cell_Min_Max.Values.BMS_Max_Cell_Voltage_Location	21366.6371	0.011270695	9.363670308	1.31199E-05	85.56692336	0.000101378	125.2252993	6.878E-06	1041.825511	0.003569353
2	CAN2.BCU_7_TEMP.Middle_Cell_1_Terminal_Temp	729492.738	0.384800523	264.4799851	0.000374782	224.3056201	0.000265754	26.07160154	1.43198E-08	0.017392662	5.86052E-08
3	CAN1.ETC1_TransInputShaftSpeed	367.0398222	0.000193608	850.721117	0.001205861	1057.366727	0.001252754	59.07445913	3.24466E-06	270.65181	0.000827268
4	CAN2.BCU_3_TEMP.Positive_Terminal_Cell_Temp	991357.5047	0.522931713	460.3145318	0.000652405	409.846055	0.00048558	40.43768929	2.22104E-05	1.441206708	4.93667E-05
5	CAN2.BCU_1_TEMP.Pack_Ambient_Temp	1173561.285	0.619042486	548.3861415	0.00077276	501.9639367	0.00059472	45.19220281	2.48218E-06	5.017290519	1.71888E-05
6	CAN2.BCU_5_TEMP.Negative_Terminal_Cell_Temp	974518.403	0.514049246	120.2524425	0.00017032	92.35242028	0.000109418	52.68895832	2.89394E-06	0.356484325	1.22035E-06
7	CAN1.VDC2_YawRate	69.34416495	3.69E-05	2.99E+00	4.08E-03	3.97E+00	4.70E-06	5.78E+00	3.17E-07	2.72E+00	9.30E-06
8	CAN2.BCU_8_Cell_Total_Status.BCU_Total_Voltage	26308.0305	0.013877234	0.369440019	3.69078E-07	0.054755503	6.47909E-08	2053.972932	0.000112814	5.674854243	1.94414E-05
9	CAN1.HRW.FrontAxeLeftWheelSpeed	189.2926877	9.98E-05	1.05E+03	1.48E-03	1.28E+03	1.51E-03	2.73E+02	1.50E-05	2.92E+02	1.00E-03
10	CAN2.BCU_8_Cell_Total_Status.BCU_Total_Current	571.7726572	0.000301801	574.1719508	0.000813814	611.3097795	0.000724271	653.638198	3.59011E-05	25.04412697	8.58016E-05
11	CAN2.BCU_3_Cell_Temperature_Status.BCU_Min_Cell_Temp	600077.9258	0.316535432	161.2896078	0.000228467	140.0032863	0.000165874	294.1341315	1.61553E-05	0.032482205	1.10303E-07
12	CAN1.Water_Temp.Water_Temp_Radiator_Out	1895768.561	1	13.50262564	1.88972E-05	1.740864557	2.06247E-06	3.523043068	1.93498E-07	63.97820497	0.000219192
13	CAN2.BCU_8_Cell_Temperature_Status.BCU_Max_Cell_Temp	778964.54	0.410896431	0.343117233	3.31762E-07	3.612464416	4.27962E-06	368.1938979	2.0223E-05	14.08500515	4.82551E-05
14	CAN2.BCU_8_Voltage_State.BCU_Min_Cell_Voltage	24420.46689	0.012881562	0.142755356	4.7721E-08	6.97845E-05	0	2080.385779	0.000114265	6.12153173	2.09718E-05
15	CAN2.BCU_6_Cell_Voltage_State.BCU_Total_Voltage	26127.89076	0.013782212	0.662604426	0.000795E-07	0.192057486	2.27464E-07	2044.986768	0.000112321	5.435671041	1.86219E-05
16	CAN1.DCU_PSU_1_act_DCbusPower_14	245.5150404	0.000125903	10.85551014	1.52345E-05	47.66155708	5.64687E-05	259.4739217	1.42516E-05	217.3358138	0.000744604
17	CAN1.DCU_PSU_2_act_DCbusVoltage_14	24738.18862	0.013049157	844.0883377	0.001196458	1056.177225	0.001251344	20138.58117	0.001106112	138.9318872	0.000475988
18	CAN1.EBC3.BrakingPressureRightAxeLeftWheel	881.0557754	0.000464745	1578.687991	0.002237855	1733.718078	0.002054085	168.6823	9.26467E-06	220.1578137	0.000754272
19	CAN2.BCU_7_Voltage_State.BCU_7_average_cell_voltage	26449.54066	0.013951879	0.657978016	7.78121E-07	0.188799127	2.23604E-07	2056.787745	0.000112969	5.399938171	1.84995E-05
20	CAN1.TSC1_EngRequestedSpeed_SpeedLimit	125.8510476	6.64E-05	4.89E-01	5.38E-07	5.23E-01	6.20E-07	9.96E-02	5.47E-09	3.35E-02	1.14E-07
21	CAN1.EBC1.ActualEngPercentTorque	143.2067327	7.59E-05	6.17E+02	8.75E-04	6.68E+02	7.89E-04	4.07E+01	2.24E-06	1.77E+01	6.08E-05
22	CAN1.EBC2.BrakingPressureFrontAxeRightWheel	1185703.24	0.625447252	330.4540208	0.00046831	310.469873	0.00036784	115.5621057	6.34724E-06	12.14308297	4.16019E-05
23	CAN2.BCU_3_Cell_Total_Status.BCU_Total_Voltage	26373.90509	0.013911982	0.622059181	7.27201E-07	0.17272249	2.04557E-07	2045.921513	0.000112372	5.483307119	1.87815E-05
24	CAN1.SPR.AirSuspensionSupplyPressure	17.19214538	9.09E-06	2.25E+02	3.24E-04	1.97E+02	2.34E-04	1.48E+01	8.11E-07	4.17E+00	1.43E-05

Çizelge 4.9: Kategorik değişkenler anova F-Skor tablosu (İlk 25 değişken).

Index	Kategorik Özellik Adı	AmberWarningAlarmı		DSH_Sytem_FaultAlarmı		DSH_System_WarningAlarmı		PDU_FaultAlarmı		BMS_SOCAlarmı	
		Anova F-Skor	Normalize F-Skor	Anova F-Skor	Normalize F-Skor	Anova F-Skor	Normalize F-Skor	Anova F-Skor	Normalize F-Skor	Anova F-Skor	Normalize F-Skor
0	CAN2.BCU_7_State_Message.Balance_Enable	801.6020458	7.34E-05	1.52E+00	1.74E-08	1.64E+00	4.33E-09	9.86E-02	1.51E-10	1.01E-01	1.06E-07
1	CAN2.BCU_1_State_Message.BCU1_Contactor_Status	24857.16837	0.000227472	33149.22146	0.000380938	42289.77624	0.000111814	385451.0361	0.00059062	4192.718046	0.004380765
2	CAN1.EBC3.FoundationBrakeLike	1089.925326	9.97E-06	6.42E+02	7.38E-06	7.02E+02	1.85E-06	2.21E+02	3.39E-07	1.14E+01	1.19E-05
3	CAN1.TSC1_CtrlCtrlPurpose	5.640100105	5.19E-08	1.09E-02	1.21E-10	1.19E-02	2.89E-11	6.03E-04	9.11E-13	6.03E-04	6.21E-10
4	CAN2.BCU_4_State_Message.BCU4_Contactor_Status	228.1033964	2.09E-06	6.10E+05	7.01E-03	7.75E+05	2.05E-03	2.01E+07	3.08E-02	5.58E+04	5.83E-02
5	CAN2.BCU_2_Fault_Message.InternalMeasurementFault	0.067316809	5.05E-10	1.27E-04	1.46E-12	7.28E+03	1.92E-05	8.28E-06	0.00E+00	8.50E-06	0.00E+00
6	CAN1.ASC1.LoweringCtrlModeRearAxe	0.012166867	0	0.39309466	4.51734E-09	0.424164074	1.11894E-09	0.02536277	3.91161E-11	0.026226453	2.73939E-08
7	CAN1.EBC1_SrcAddressCtrlCtrlDncForEngCtrl	57.18968218	5.23E-07	6.79E-02	7.77E-10	2.37E-01	6.23E-10	7.48E-02	1.15E-08	5.95E-01	6.22E-07
8	CAN1.VMU1.Dsh_SystemOverTemp	102.2581454	9.39E-07	4.75E-02	5.49E-10	5.13E-02	1.33E-10	3.09E-03	4.72E-12	3.17E-03	3.30E-08
9	CAN1.ASC2.KneelingCtrlRearAxe	0.471218001	4.20E-09	8.92E-04	1.02E-11	9.62E-04	0.00E+00	5.79E-05	7.61E-14	5.59E-05	5.33E-11
10	CAN1.EV_LIMITS.EV_MaxVoltageLimit	0.404365849	3.59E-09	2.64E-03	3.04E-11	4.39E+04	1.16E-04	1.85E-05	2.83E-04	7.86E-05	8.01E-01
11	CAN2.BCU_HVOU_State_Control.HVOU_State_Control	230.7254955	2.11E-06	4.89E+05	5.57E-03	4.08E+05	1.08E-03	6.84E+07	1.05E-01	1.89E+04	1.98E-02
12	CAN1.BMS_Temp_Min_Max.Values.BMS_Min_Amb_Temp_Location	9731.920976	8.91E-05	1.17E+01	1.34E-07	2.37E+01	6.27E-08	3.34E+00	5.11E-09	1.31E+00	1.37E-06
13	CAN2.BCU_3_State_Message.BCU3_Contactor_Status	228.1033964	2.09E-06	6.10E+05	7.01E-03	7.75E+05	2.05E-03	2.01E+07	3.08E-02	5.58E+04	5.83E-02
14	CAN1.ASC2.NominalLevelRearAxe	7.57261004	6.92E-08	2.11E+00	2.43E-08	2.28E+00	6.03E-09	1.37E-01	2.10E-10	4.88E+00	5.10E-06
15	CAN2.BCU_1_State_Message.BCU1_State_Response	224.3021759	2.05E-06	4.82E+05	5.54E-03	3.93E+05	1.04E-03	5.39E+07	8.26E-02	1.03E+04	1.07E-02
16	CAN1.TSC1_EngOverrideCtrlMode_0	6.54560825	5.98E-08	8.02E-02	9.22E-10	1.77E-02	4.43E-11	5.77E-01	8.85E-10	3.99E-01	4.17E-07
17	CAN1.TSC1_OverrideCtrlModePriority	6.699260083	6.12E-08	7.95E-02	9.13E-10	1.81E-02	4.52E-11	5.78E-01	8.85E-10	3.98E-01	4.16E-07
18	CAN1.EBC3.HillHolderMode	4432.519474	4.08E-05	2.91E+06	3.34E-02	2.63E+06	6.95E-03	3.68E-01	5.64E-10	3.52E+01	3.67E-05
19	CAN2.Battery_Power_State_Copy_1.EV_Target_Voltage	0.402927101	3.58E-09	0.00E+00	0.00E+00	4.39E+04	1.16E-04	1.85E-05	2.83E-04	7.86E-05	8.01E-01
20	CAN2.BCU_8_State_Message.BCU8_Contactor_Status	92.14503653	8.43E-07	3.26E+05	3.74E-03	4.11E+05	1.09E-03	5.50E+06	8.43E-03	3.06E+04	3.20E-02
21	CAN1.BMS_Temp_Min_Max.Values.BMS_Max_Amb_Temp_Location	101918.7013	9.33E-04	5.77E+02	6.63E-06	6.61E+02	1.75E-06	1.43E+00	2.19E-09	6.37E+01	6.66E-05
22	CAN1.VMU3.Dsh_CoolingPumpRequest	74.06017568	6.78E-07	2.23E+05	2.56E-03	3.31E+05	8.75E-04	2.49E+04	3.82E-05	5.27E+03	5.50E-03
23	CAN2.BCU_2_State_Message.Measurement_Hold	258.5474706	2.37E-06	5.10E+05	5.80E-03	5.64E+05	1.49E-03	6.53E+08	1.00E+00	5.84E+04	6.10E-02
24	CAN2.BCU_5_TEMP.Pack_Humidity	1079545.745	9.88E-03	3.24E+01	3.72E-07	2.67E+01	7.06E-08	7.86E+01	1.20E-07	4.54E+00	4.74E-06

Veri Seti bölümünde, ham verinin toplanması, özetlenmesi, analiz edilmesi, sadeleştirilmesi ve anomali tespiti gibi süreçler detaylı bir şekilde ele alınmıştır. Bir sonraki Çizge Tabanlı Öznitelik Seçimi bölümünde, veri seti boyutunu azaltmak ve model performansını iyileştirmek amacıyla, geliştirilen çizge tabanlı öznitelik seçimi yöntemi açıklanmıştır. Bu kapsamda, değişkenler arasındaki istatistiksel ilişkiler çizge yapısında görselleştirilmiş ve çizge yapısı üzerinde topluluk tespiti algoritmaları (Leiden, Louvain, InfoMap, Fast Greedy) kullanılarak, alakalı özellik alt kümeleri belirlenmiştir. Daha sonrasında, çizge teorisi prensipleri kullanılarak SPEC skoru ve K-Shell ayrıştırma yöntemi gibi metrikler hesaplanarak ayırt edici öznitelikler tespit edilmiştir. Bu yöntemle, veri setinin boyutunu etkin bir şekilde azaltmak ve kestirimi gerçekleştirecek modellerin performansını arttırmak hedeflenmiştir.

5. ÇİZGE TABANLI ÖZNİTELİK SEÇİMİ

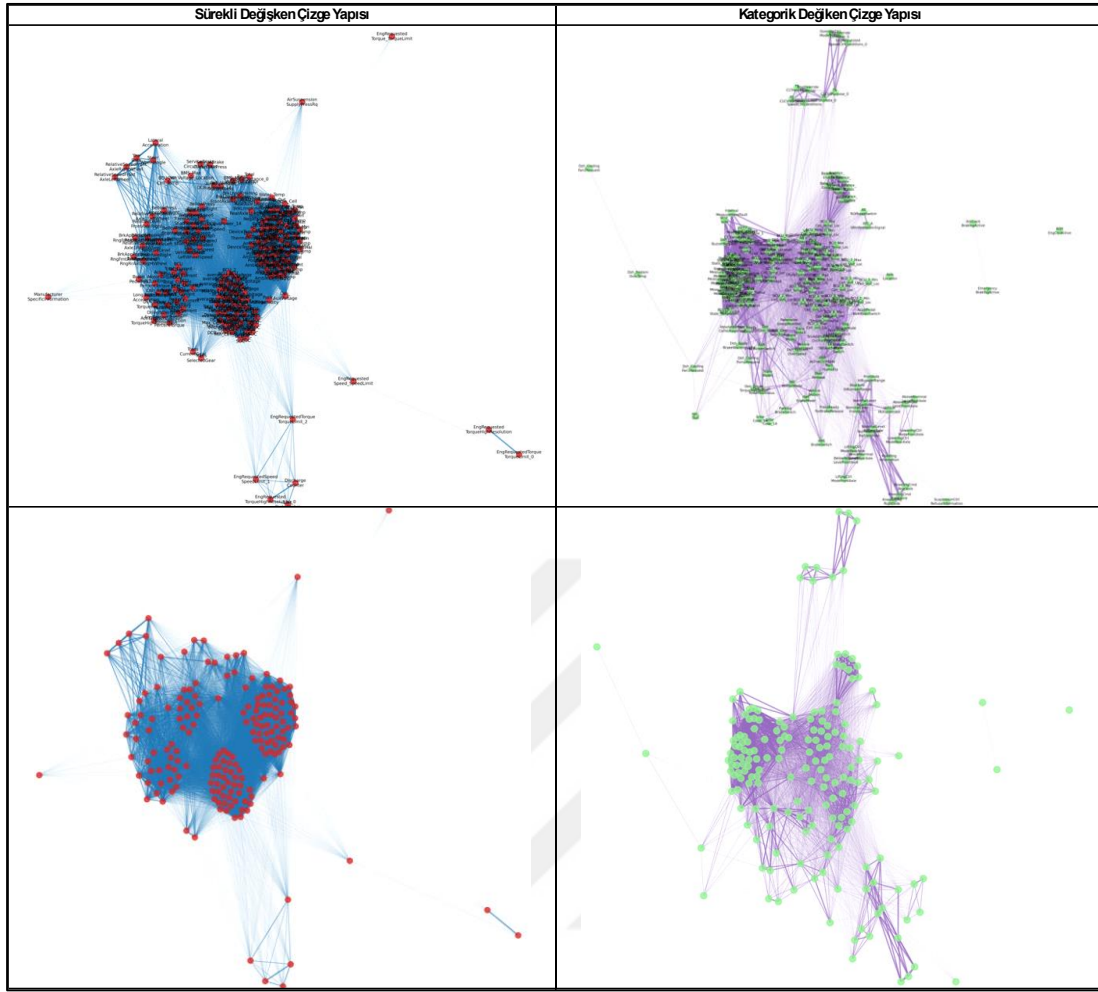
Bu çalışma kapsamında, istatistik tabanlı filtreleme yöntemleri ile çizge prensiplerini birleştiren bir özellik seçimi algoritması geliştirilmiş ve sadeleştirilmiş Can Bus verileri üzerinden ayırt edici özellikler (öznitelikler) belirlenmiştir. Algoritma, her hedeflenen alarm için düğüm ve kenar ağırlıklarının istatistiksel ölçütlerle belirlendiği ağırlıklı çizgeler oluşturarak, sezgisel, açgözlü ve optimizasyon tabanlı topluluk tespit algoritmaları ile optimum çizge yapısını oluşturmuş ve tespit edilen toplulukların temsil ettiği Alakalı Özellik Altkümelerini (Relevant Feature Subsets) belirlemiştir. Son olarak, her bir Alakalı Özellik Altkümesi üzerindeki düğümlerin SPEC (Spectral Feature Selection) skoru ve [37] çalışmasından yola çıkarak düğüm ağırlıklarının etkisinin de gözetildiği modifiye edilmiş k-Shell skoru hesaplanmış ve en bilgilendirici, ayırt edici özellikler seçilmiştir.

5.1. Değişkenler Arası İstatistiksel İlişkilerin Çizge Yapısında Görselleştirilmesi

Bu süreçte ilk olarak, sadeleştirilmiş veri üzerinde gerçekleştirilen istatistiksel analizler (Pearson Korelasyon Matrisi, Cramer's V İstatistik Tablosu ve Anova F-Skor Hesaplamaları) temel alınarak, kategorik ve sürekli değişkenlerin ilişkilerini görselleştirecek Şekil 5.1 ve Şekil 5.2'de detaylıca gösterilen çizge yapıları oluşturulmuştur.

Oluşturulan çizge yapılarında, her bir düğüm (node) bir değişkeni temsil ederken, düğümler arasındaki kenarlar (edge) değişkenler arasındaki istatistiksel ilişkinin gücünü ifade etmektedir. Sürekli değişkenler arasındaki ilişkiler Pearson korelasyon katsayıları ile, kategorik değişkenler arasındaki ilişkiler ise Cramer's V katsayıları ile belirlenmiştir. Kenarların kalınlığı ilişki büyüklüğü ile doğru orantılı olarak ayarlanmış, böylece daha güçlü ilişkiler daha belirgin bir şekilde görselleştirilmiştir. Aynı zamanda, düğümler arasındaki mesafe ilişki büyüklüğü ile ters orantılı olarak ayarlanmış, böylece güçlü ilişkili değişkenler çizge yapısı üzerinde birbirine daha yakın konumlandırılmıştır.

Çizelge 5.1: Sürekli ve kategorik değişkenlerin ilişki çizge yapıları.

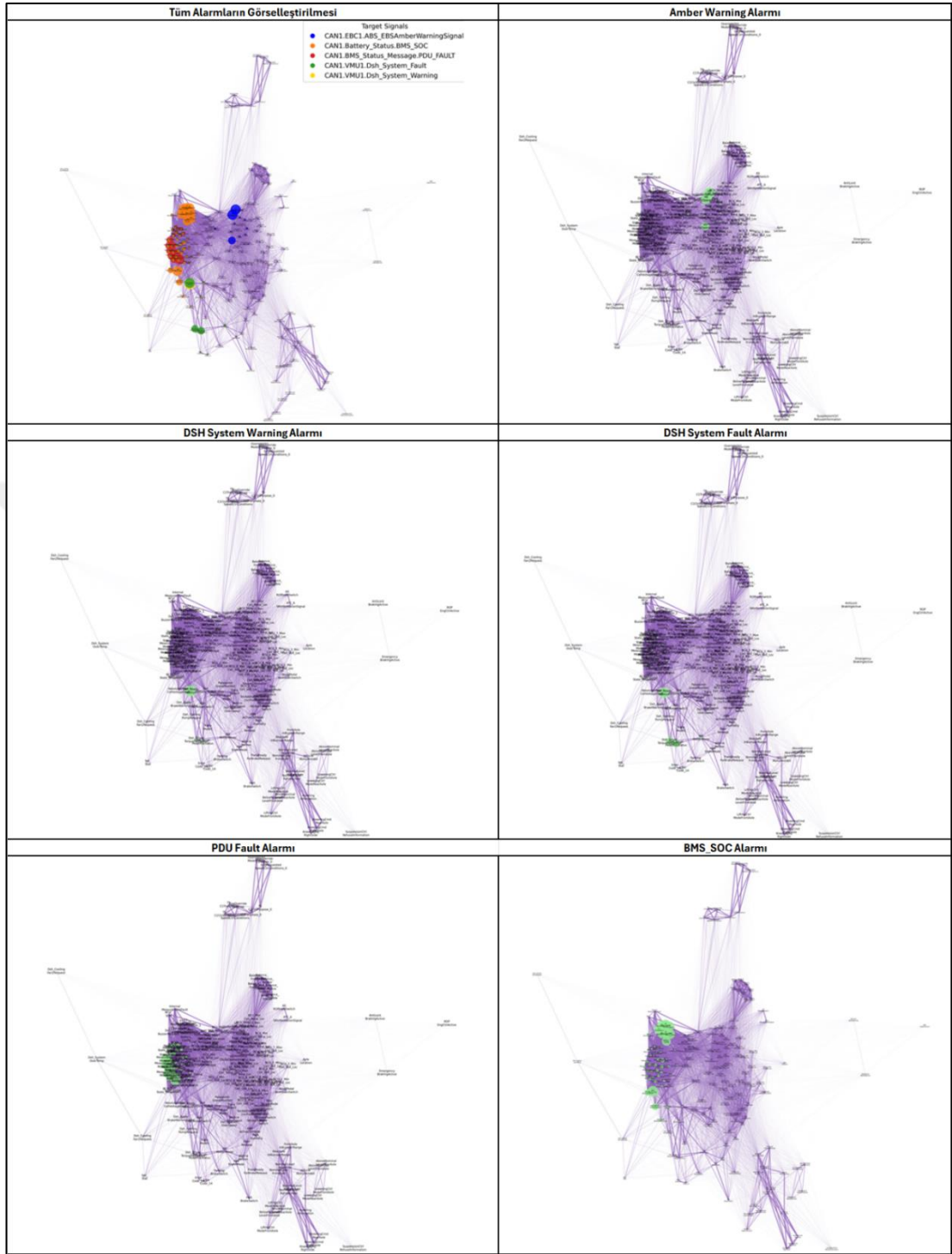


Her bir hedeflenen alarmanın diğer değişkenlerle olan ilişkisi Anova F-testi ile hesaplanarak kategorik ve sürekli değişkenlerin hedef değişkenlerle olan ilişkileri düğümlerin büyüklükleri ile temsil edilmiştir. Daha büyük düğümler, alarmlarla daha güçlü bir ilişkiye sahip değişkenleri temsil ederken, daha küçük düğümler daha zayıf ilişkileri temsil etmektedir.

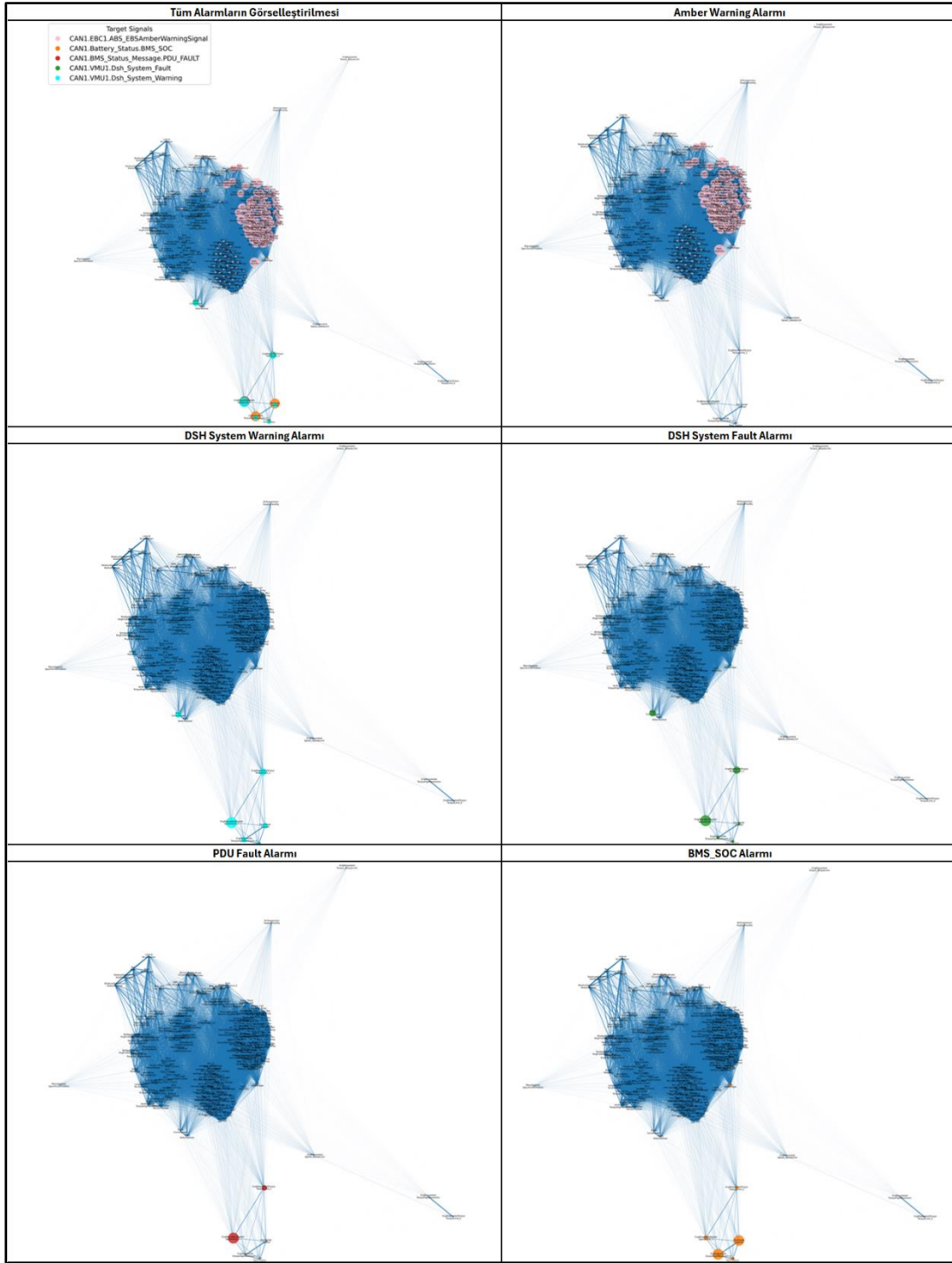
Bu sayede, değişkenlerin hem birbirleriyle hem de alarmlarla olan istatistiksel ilişkileri tek bir çizge yapısı üzerinde görselleştirilmiş, değişkenler arası ilişkilerin bütünsel bir haritası oluşturulmuştur.

Bu görselleştirme sayesinde, değişkenler arasındaki ilişkilerin karmaşık yapısı daha kolay anlaşılır hale gelmiştir. Özellikle, çizge üzerinde yoğunlaşan bölgeler ve belirgin kenarlar güçlü ilişkileri ve potansiyel etkileşimleri işaret etmektedir. Bu sayede, ilgili özellik alt kümeleri genel olarak belirlenmiş ve yorumlanabilmiştir.

Çizelge 5.2: Kategorik değişken ve hedeflenen alarm ilişkilerinin görselleştirilmesi.



Çizelge 5.3: Sürekli değişken ve hedeflenen alarm ilişkilerinin görselleştirilmesi.



5.2. Çizge Yapısı Üzerinde Topluluk Tespiti

Bu aşamada, oluşturulan **ağırlıklı** çizge yapıları üzerinde alakalı özellik alt kümelerinin tespiti için InfoMap (optimizasyon), Fast Greedy (açgözlü) , Louvain ve Leiden (sezgisel) topluluk tespit algoritmaları kullanılmış ve elde edilen toplulukların

kalitesi, modülerite ölçümü ile değerlendirilmiştir. Çizge yapılarının üstünde topluluk tespiti algoritmaları çalıştırırken, geleneksel yöntemlerde olduğu gibi doğrudan bir eşik değeri atanıp bu değerin altındaki kenarların (edge) çizgeden silinmesi yerine, daha dinamik ve optimizasyon odaklı ve uygulanabilirliği yüksek bir yaklaşım benimsenmiştir.

Bu yaklaşımda 0.1'den 0.95'e kadar 0.05 adımlarla artan eşik değerleri kullanılarak, her adımda eşik değerinin altındaki kenarlar çizgelerden çıkarılmıştır. Ardından, Leiden, InfoMap, Fast Greedy ve Louvain gibi farklı topluluk tespit algoritmaları uygulanmış ve elde edilen topluluk yapıları modülerite (modularity) ölçütüne göre değerlendirilmiştir. En yüksek modülerlik değerini veren algoritma ve eşik değeri kombinasyonu seçilerek, çizgeler üzerindeki topluluklar belirlenmiştir. Bu sayede, değişkenler arasındaki ilişkilerin daha iyi anlaşılması ve alakalı alt grupların (toplulukların) ortaya çıkarılması sağlanmıştır. Elde edilen topluluklar, özellik seçimi sürecinde daha sonra kullanılmak üzere saklanmıştır.

Çizelge 5.4: Topluluk tespiti algoritmaların performansı.

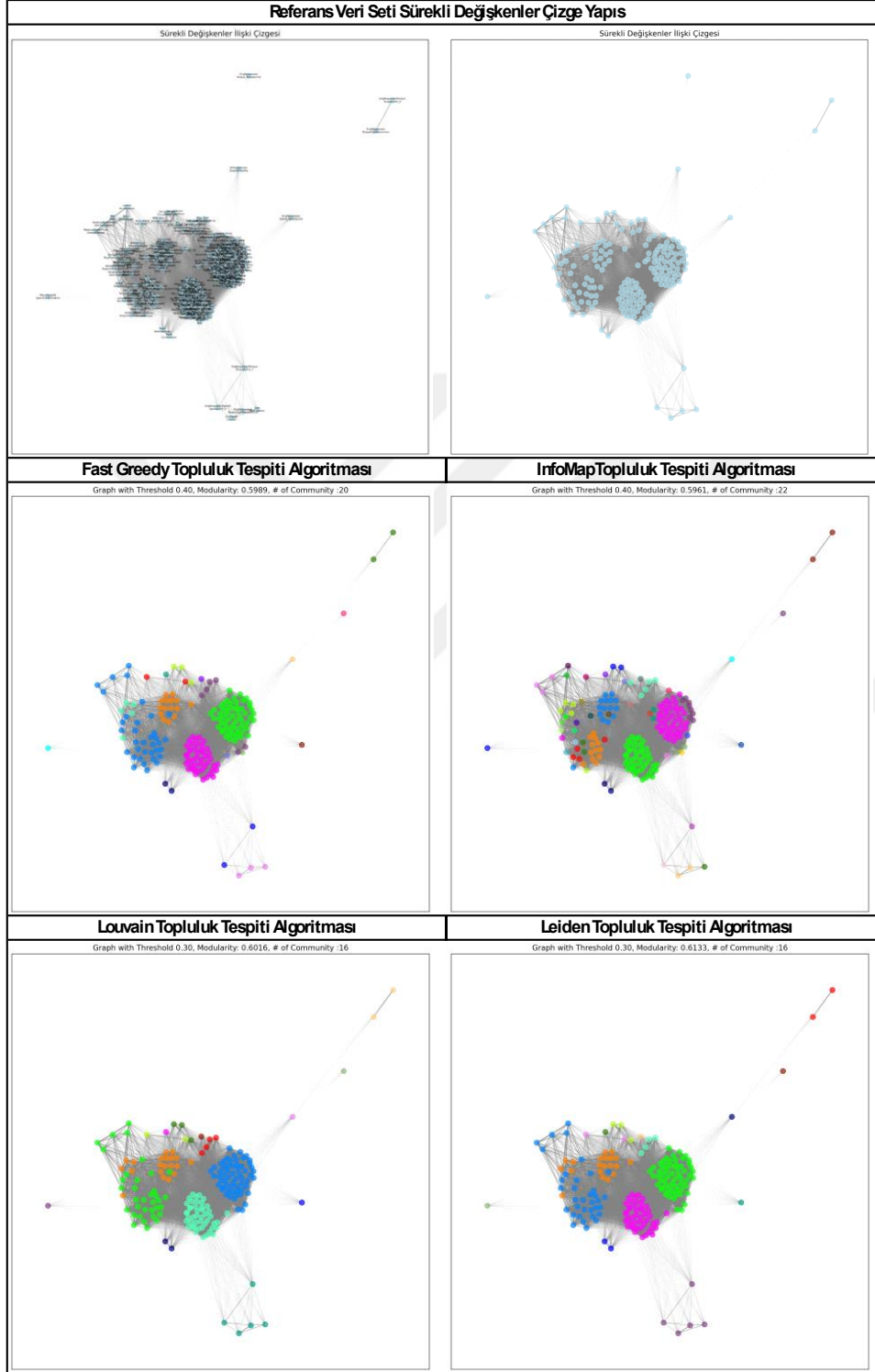
Sürekli Değişkenler Grafiği ve Algoritma Performansları			
Topluluk Tespiti Algoritmaları	En İyi Sonuç Veren Eşik Değeri	Modülerite Skoru	Topluluk Sayısı
InfoMap Algoritması	0.40	0.5961	22
Fast Greedy Algoritması	0.35	0.5999	16
Louvain Algoritması	0.30	0.6016	16
Leiden Algoritması	0.30	0.6133	16

Kategorik Değişkenler Grafiği ve Algoritma Performansları			
Topluluk Tespiti Algoritmaları	En İyi Sonuç Veren Eşik Değeri	Modülerite Skoru	Topluluk Sayısı
InfoMap Algoritması	0.95	0.6222	102
Fast Greedy Algoritması	0.95	0.6210	101
Louvain Algoritması	0.95	0.6222	102
Leiden Algoritması	0.95	0.6216	102

Sürekli değişkenler çizgesinde, 0.30 eşik değeri altında kalan kenarların silinmesi ve Leiden algoritmasının uygulanması sonucunda 0.6133 ile en iyi modülerite skoruna sahip ve 16 topluluktan oluşan yeni bir çizge yapısı oluşturulmuşken, kategorik değişkenler çizgesinde 0.95 eşik değeri altında kalan kenarların silinmesi ve Louvain veya InfoMap algoritmalarının kullanılmasıyla 0.6222 ile en iyi modülerite skoruna sahip ve 102 topluluktan oluşan yeni bir çizge oluşturulmuştur. Bu durum kategorik

değişkenlerin sürekli değişkenlere göre daha heterojen bir yapıda olup ve sürekli ve kategorik değişkenlerin farklı ilişki örüntülerine sahip olduğunu göstermiştir.

Çizelge 5.5: Sürekli değişkenler çizge yapısı ve topluluk tespiti algoritmaları.



Algoritma 5.1: Geliştirilen dinamik eşik değerleri topluluk tespiti algoritması.

Algorithm 1 Community Detection with Dynamic Thresholds

```

1: function COMMDETECTDYNTHRESHOLDS(Graph  $G$ )
2:    $best\_modularity \leftarrow -1$ 
3:    $best\_community\_structure \leftarrow None$ 
4:    $best\_threshold \leftarrow None$ 
5:    $best\_algorithm \leftarrow None$ 
6:    $thresholds \leftarrow [0.1, 0.15, 0.2, \dots, 0.95]$ 
7:    $algorithms \leftarrow ['Leiden', 'InfoMap', 'FastGreedy', 'Louvain']$ 
8:   for each  $threshold \in thresholds$  do
9:      $G\_thresholded \leftarrow REMOVEEDGES(G, threshold)$ 
10:    for each  $algo \in algorithms$  do
11:       $communities \leftarrow COMMUNITYDETECTION(G\_thresholded, algo)$ 
12:       $modularity \leftarrow MODULARITY(G\_thresholded, communities)$ 
13:      if  $modularity > best\_modularity$  then
14:         $best\_modularity \leftarrow modularity$ 
15:         $best\_community\_structure \leftarrow communities$ 
16:         $best\_threshold \leftarrow threshold$ 
17:         $best\_algorithm \leftarrow algo$ 
18:      end if
19:    end for
20:  end for
21:  return  $best\_community\_structure, best\_threshold, best\_algorithm$ 
22: end function

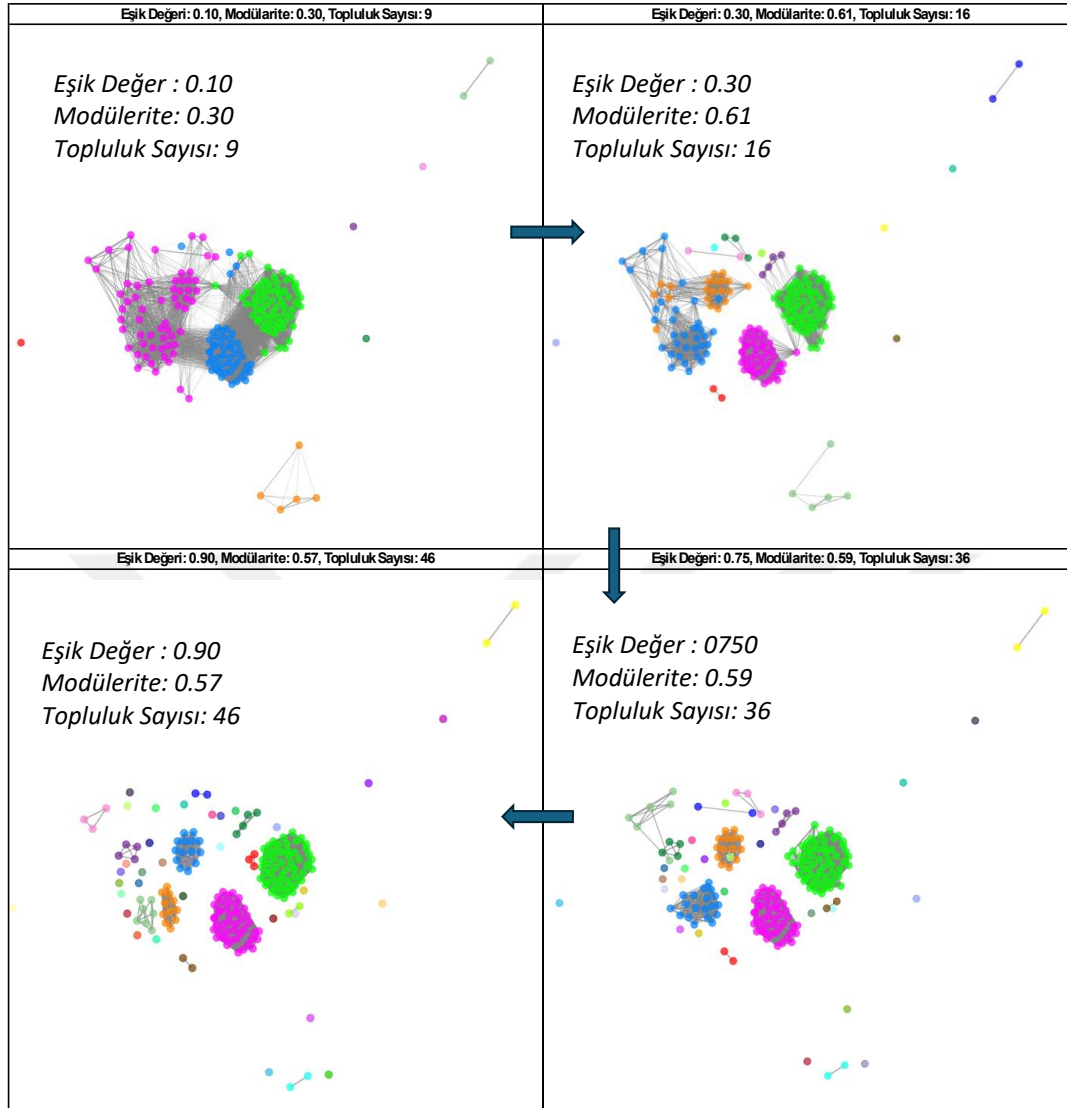
23:
24: function REMOVEEDGES(Graph  $G$ , float  $threshold$ )
25:    $new\_graph \leftarrow \text{Copy of } G$ 
26:   for each  $(u, v, weight) \in G.edges(data='weight')$  do
27:     if  $weight < threshold$  then
28:        $new\_graph.remove\_edge(u, v)$ 
29:     end if
30:   end for
31:   return  $new\_graph$ 
32: end function

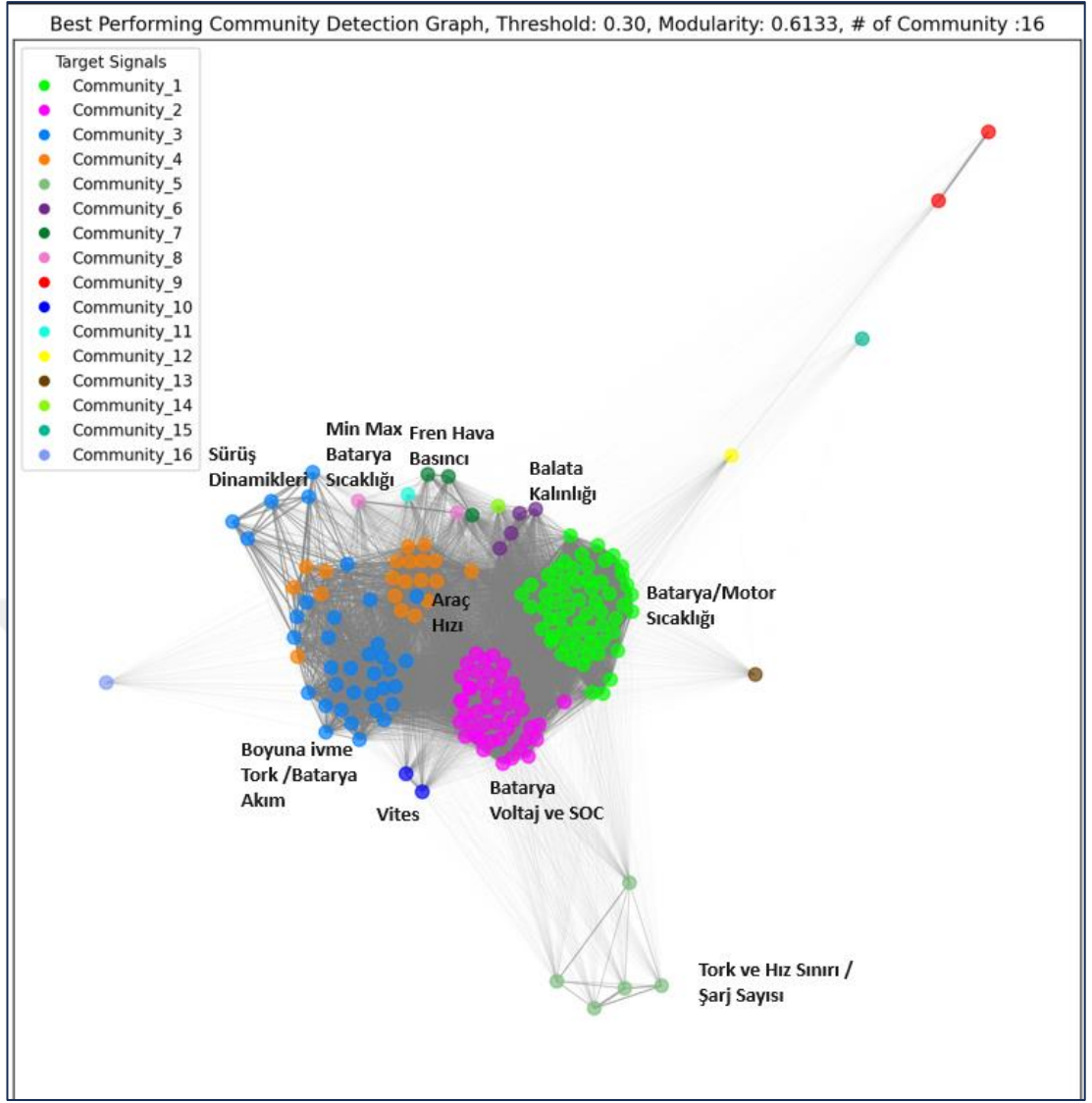
33:
34: function COMMUNITYDETECTION(Graph  $G$ , string  $algorithm$ )
35:   if  $algorithm == 'Leiden'$  then
36:     return APPLYLEIDEN( $G$ )
37:   else if  $algorithm == 'InfoMap'$  then
38:     return APPLYINFOMAP( $G$ )
39:   else if  $algorithm == 'FastGreedy'$  then
40:     return APPLYFASTGREEDY( $G$ )
41:   else if  $algorithm == 'Louvain'$  then
42:     return APPLYLOUVAIN( $G$ )
43:   else
44:     raise ValueError("Unknown algorithm")
45:   end if
46: end function

47:
48: function ModularityGraph  $G$ , communities
49:    $Q \leftarrow 0$ 
50:    $m \leftarrow \text{total weight of all edges in } G$ 
51:   for each community in communities do
52:      $L_c \leftarrow \text{total weight of edges within community}$ 
53:      $D_c \leftarrow \text{total sum of weights of edges attached to nodes in community}$ 
54:      $Q \leftarrow Q + \left( \frac{L_c}{m} - \left( \frac{D_c}{2m} \right)^2 \right)$ 
55:   end for
56:   return  $Q$ 
57: end function

```

Çizelge 5.6: Geliştirilen algoritmanın çalışma akışı kesitleri.





Şekil 5.3: Leiden algoritması topluluk tespiti.

5.3. Ayırt Edici Özniteliklerin Seçimi

Bu süreçte, sürekli değişkenlerde 0.30 eşik değeri ve Leiden algoritması uygulanarak, kategorik değişkenlerde 0.95 eşik değeri ve InfoMap algoritması uygulanarak elde edilen çizge yapılarındaki topluluklar içerisinde bulunan değişkenlerin, hedeflenen alarmlarla olan ilişkileri ve birbirleriyle olan ilişkileri göz önüne alınarak SPEC (Spectral Feature Selection) skoru ve modifiye edilmiş K- Shell skoru hesaplanmış ve en ayırt edici özellikler belirlenmiştir.

5.3.1. SPEC skorunun hesaplanması

Spectral Feature (SPEC) skoru, özniteliklerin topluluklar içindeki önemini ve alarmlarla olan ilişkisini değerlendirmek için kullanılmıştır. Bu skor, özniteliklerin

topluluklar içindeki diğer özniteliklerle olan benzerliği ve alarmlarla olan ilişkiyi dikkate alarak hesaplanmıştır.

Bu aşamada ilk olarak, topluluğun alt çizge yapısının düğüm ve kenarlar ilişkilerini içeren Laplacian matrisi oluşturulmuştur. Laplacian Matrisi Eşitlik (5.1)'deki gibi tanımlanmıştır.

$$L = D - A \quad (5.1)$$

Burada;

D: Çizgenin her düğümünün komşu sayısını içeren diagonal bir matrisidir

A: Çizgenin komşu (adjacency)matrisidir.

$$SPEC(x_i) = \sum_{j=1}^k \lambda_j (u_j^T x_i)^2 \times F_i \quad (5.2)$$

Burada;

λ_j : Laplacian matrisinin j-inci özdeğeri

u_j : Laplacian matrisinin j inci özvektörü

x_i : i'inci öznitelik vektörü

F_i : i'inci özniteliğin Anova F-skoru

Daha sonrasında çizgenin yapısal özelliklerini ve düğümler arasındaki ilişkileri temsil eden Laplacian Matrisinin özdeğer ve özvektörleri hesaplanmış ve her topluluktaki özniteliklerin her bir hedef değişken için aldığı Anova F-skor değerleri, topluluk yapısının Laplacian matrisinin öz vektörlerine dahil edilmiştir. Bu aşamada, her değişkenin her bir hedef değer için aldığı F-skor değerleri vektörleştirilerek, Laplacian matrisin özvektörleriyle çarpılmış ve özvektörleri F-skorlarıyla ağırlıklandırarak, değişkenlerin hem çizge yapısındaki konumlarını hem de hedef değerlerle olan ilişkilerini yansıtan yeni özvektörler elde edilmiştir. Bu yeni ağırlıklandırılmış özvektörler üzerinden her bir ilgili bileşenin karelerinin toplamı hesaplanarak topluluktaki özelliklerin hedef değişkenler için SPEC Skoru hesaplanmıştır. Bu skor sayesinde, hangi özelliklerin topluluk içindeki diğer özelliklerle güçlü bir ilişkisi olduğunu ve hedef değişkenle anlamlı bir bağlantıya sahip olduğu tespit edilmiştir. SPEC Skoru Eşitlik (5.2) ile gösterildiği şekilde hesaplanmıştır.

5.3.2. K-Shell ayrıştırma yöntemi

SPEC Skorun yanında, k-Shell Ayrıştırma yöntemi kullanılarak, topluluktaki her bir özelliğin her bir alarmı için k-Shell skoru hesaplanmıştır. K-Shell Ayrıştırma yöntemi, bir çizgedeki düğümlerin **merkeziliğini** ve önemini belirlemek için kullanılan bir algoritmadır. K-Shell Skoru, düğümlerin topolojik önemini ve bağlantı yoğunluğunu yansıtmaktadır.

K-Shell Ayrıştırma (k-Core Decomposition), Algoritma 5.2’de detaylandırıldığı gibi bir çizge yapısındaki düğümleri, bağlantı yoğunluklarına göre katmanlara ayırır. Daha sonrasında, düğümleri iteratif olarak çizgeden çıkararak, her bir düğümün ait olduğu en derin katmanı belirler.

Algoritma 5.2: K-Shell ayrıştırma yöntemi adımları.

1. Başlangıç Durumu

Tüm düğümler başlangıçta çizgede bulunur

Her düğümün derecesi (bağlantı sayısı) hesaplanır

2. Düşük Dereceli Düğümleri Çıkarma:

Derecesi 1 olan düğümler bulunur

Bu düğümler komşu düğümlerinin dereceleri de azaltarak çıkarılır

3. İterasyon:

Derecesi 1 olan düğümler çıkarıldıktan sonra,

Yeni derecesi 1 olan düğümler tekrar bulunur ve çıkarılır.

Bu süreç, derecesi 2 olan düğümler için tekrarlanır ve tüm dereceler için devam ederek tüm düğümler çıkarılana kadar sürer.

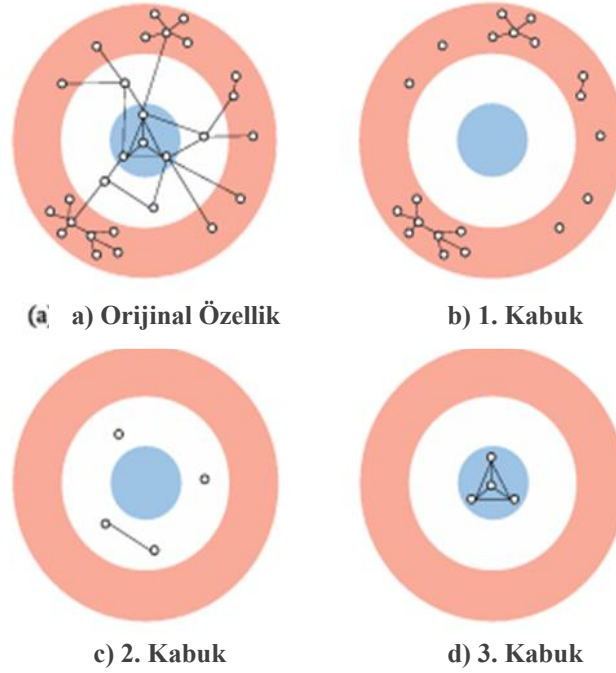
4. Katmanların Belirlenmesi:

Eğer:

Bir düğüm en az k dereceye sahipse ve k dereceli komşuları da çıkarılmışsa,

Düğüm k-shell'in bir parçası olarak kabul edilir.

K-shell skoru, her düğümün ait olduğu en yüksek k-shell'dir



Şekil 5.4: Bir özellik çizge yapısının 3 kabuklu katmanlı yapısı [37].

5.3.2.1. Modifiye edilmiş K-Shell skorunun hesaplanması

Bu aşamada ilk olarak topluluktaki özelliklerin alarmlarla ilişkisini temsil eden Anova F-skor değerleri, düğümlerin ağırlıkları olarak kabul edilmiş ve [37] çalışmasından yola çıkarak topluluklar içindeki her bir özelliğin her bir alarm için modifiye edilmiş k-Shell skoru Eşitlik (5.3) ile gösterildiği gibi hesaplanmıştır. Modifiye edilmiş K-Shell skoru ile, topluluk özelliklerinin hem kendi ağırlıkları hem de komşularının ve bağlantı ağırlıkları hesaba katılmış ve özelliklerin topluluk içindeki önemi hem alarmlarla olan ilişkilerine hem de topluluktaki merkezi konumuna bakılarak değerlendirilmiştir.

$$k'_{\{\eta\}} = \left[\sqrt{k_{\{\eta\}} W_{\{V_{\{\eta\}}\}} \sum_{\{k_{\{\eta\}}\}} W_{\{E_{\{\eta,\xi\}}\}} \right] \quad (5.3)$$

Burada;

$k_{\{\eta\}}$: Düğümün derecesi,

$W_{\{V_{\{\eta\}}\}}$: Düğümün ağırlığı,

$W_{\{E_{\{\eta,\xi\}}\}}$: Düğüm (η) ile bağlı diğer düğümler arasındaki kenarların ağırlıklarının toplamıdır

5.3.3. Topluluklardan ayırt edici özniteliklerin belirlenmesi

Bu aşamada ise her topluluktaki özniteliklerin hedeflenen alarm için maksimum SPEC ve k-Shell skoruna sahip öznitelikler seçilmiş ve iki farklı potansiyel ayırt edici öznitelik kümesi oluşturulmuştur. Maksimum k-Shell skorlarının seçilmesiyle elde edilen özellik kümesinin, maksimum SPEC skorunun belirlenmesiyle elde edilen özellik kümesine kıyasla daha az özellik içerdiği, bunun en büyük nedeninin ise k-Shell ayırıştırma yönteminin az sayıda özellik içeren topluluklarda düşük performans sergilemesinden kaynaklı oluşu gözlemlenmiştir. Ancak topluluklar içerisinde seçilen aynı ve maksimum SPEC skoruna sahip özellikler arasından en ayırt edici özellik bulunmaya çalışılırken bu özelliklerim k-Shell skoruna bakılarak içlerinden diğer topluluk elemanlarıyla daha çok ilişki içinde olan, en merkezi konuma sahip özellik seçilmiş ve tek bir ayırt edici özellik kümesi oluşturulmuştur.

Daha sonra oluşturulan ayırt edici özellik kümesi alan uzmanları tarafından değerlendirilip onaylanmış ve talepleri üzerine ayırt edici özellik kümesinde bulunan ve 8 farklı batarya paketine özgü özellikler, araçların yapısına göre değişiklik gösterebileceğinden kaynaklı olarak çıkarılmıştır.

Çizelge 5.7: SPEC skoru ve K-Shell skoruna göre oluşturulan özellik kümeleri.

Sürekli Değişkenler SPAC Skoruna Göre Seçilen Ayırt edici Öznitelikler	Sürekli Değişkenler K-Shell Skoruna Göre Seçilen Ayırt edici Öznitelikler
Community No	Selected Features According to SPAC Values
Community 1	CAN1.Water_Temp.Water_Temp_Radiator_Out CAN1.VMU1.Dsh_ThermalIndicator CAN1.VMU2.Dsh_AuxVoltage
Community 2	CAN1.BMS_Cell_Min_Max_Values.BMS_Max_Cell_Voltage
Community 3	CAN1.VMU3.Dsh_HighPowerCurrent CAN1.ASC4.BellowPressRearAxleLeft
Community 4	CAN1.DCU_PSU_2.act_DCBusPower_14 CAN1.EBC1.BrakePedalPos
Community 5	CAN1.TSC1.EngRequestedTorqueHighResolution_0 CAN1.TSC1.EngRequestedSpeed_SpeedLimit_1 CAN1.TSC1.EngRequestedTorque_TorqueLimit_2
Community 6	CAN1.EBC4.BrakeLiningFrontAxleRightWheel CAN1.EBC4.BrakeLiningFrontAxleLeftWheel CAN1.VD.TotalVehicleDistance
Community 7	CAN1.AIR1.AirSuspensionSupplyPress CAN1.AIR1.ServiceBrakeCircuit2AirPress
Community 8	CAN1.DCU_PSU_1.act_DCBusPower_14
Community 9	(-)
Community 10	CAN1.ETC2.TransCurrentGear
Community 11	CAN1.BMS_Cell_Min_Max_Values.BMS_Max_Cell_Voltage_Location
Community 12	(-)
Community 13	(-)
Community 14	CAN1.BMS_Cell_Min_Max_Values.BMS_Min_Cell_Voltage_Location
Community 15	(-)
Community 16	(-)

Community No	Selected Features According to K-Shell Values
Community 1	CAN1.Water_Temp.Water_Temp_Radiator_Out CAN2.BCU_7_TEMP.Positive_Terminal_Cell_Temp CAN1.VMU2.Dsh_AuxVoltage CAN2.BCU_4_TEMP.Pack_Ambient_Temp
Community 2	CAN1.BMS_Cell_Min_Max_Values.BMS_Max_Cell_Voltage
Community 3	CAN1.ASC4.BellowPressRearAxleLeft
Community 4	CAN1.DCU_PSU_2.act_DCBusPower_14
Community 5	CAN1.TSC1.EngRequestedSpeed_SpeedLimit_1 CAN1.TSC1.EngRequestedTorqueHighResolution_0
Community 6	CAN1.EBC4.BrakeLiningFrontAxleRightWheel
Community 7	(-)
Community 8	(-)
Community 9	(-)
Community 10	(-)
Community 11	(-)
Community 12	(-)
Community 13	(-)
Community 14	(-)
Community 15	(-)
Community 16	(-)

Tüm bu aşamalar sonunda, başlangıçta 902 değişkenden oluşan ham veri seti üzerinde yapılan analizler ve çizge tabanlı özellik seçimi sonucunda alarm tahmininde kullanılacak 52 ayırt edici belirlenmiştir.

Çizelge 5.8: Kararlaştırılan öznelik listesi.

index	Kategorik Öznelikler Listesi	Sürekli Öznelikler Listesi
0	CAN1.EV_LIMITS.EV_MaxCurrentLimit	CAN1.Water_Temp.Water_Temp_Radiator_Out
1	CAN1.TSC1.TSC1TransRate_1	CAN1.VMU1.Dsh_ThermalIndicator
2	CAN2.Battery_Power_Status_Copy_1.EV_Target_Voltage	CAN1.VMU2.Dsh_AuxVoltage
3	CAN1.EV_LIMITS.EV_MaxVoltageLimit	CAN1.BMS_Cell_Min_Max_Values.BMS_Max_Cell_Voltage
4	CAN1.EV_LIMITS.EV_MaxPowerLimit	CAN1.DCU_PSU_2.act_DCBusPower_14
5	CAN1.TSC1.TSC1CtrlPurpose_1	CAN1.TSC1.EngRequestedTorqueHighResolution_0
6	CAN1.BMS_Status_Message.BMS_Contactor_Status	CAN1.TSC1.EngRequestedSpeed_SpeedLimit_1
7	CAN2.BCU_HVCU_State_Control.Contactor_Status	CAN1.TSC1.EngRequestedTorque_TorqueLimit_2
8	CAN1.Battery_Status.DMDC	CAN1.EBC4.BrLnngrmningFrontAxleRightWheel
9	CAN1.VMU1.Dsh_ReadyToDrive	CAN1.EBC4.BrLnngrmningFrontAxleLeftWheel
10	CAN1.VMU1.Dsh_System_Status	CAN1.VD.TotalVehicleDistance
11	CAN1.VDC1.VDCFullyOperational	CAN1.AIR1.AirSuspensionSupplyPress
12	CAN1.EBC1.ABSFullyOperational	CAN1.AIR1.ServiceBrakeCircuit2AirPress
13	CAN1.VDC1.VDCInformationSignal	CAN1.DCU_PSU_1.act_DCBusPower_14
14	CAN1.BMS_Status_Message.HVCU_Pos_Contactor_Status	CAN1.ETC2.TransCurrentGear
15	CAN1.BMS_Status_Message.HVCU_Negative_Contactor_Status	CAN1.BMS_Cell_Min_Max_Values.BMS_Max_Cell_Voltage_Location
16	CAN1.DCU_PSU_1.ErrorCode_14	CAN1.BMS_Cell_Min_Max_Values.BMS_Min_Cell_Voltage_Location
17	CAN1.DCU_PSU_2.ErrorCode_14	
18	CAN1.VMU2.Dsh_DriverTorqueReqStatus	
19	CAN1.VMU2.Dsh_AccelPedalPosStatus	
20	CAN1.VMU3.Dsh_CoolingPumpRequest	
21	CAN1.ETC7.TransReadyForBrakeRelease	
22	CAN1.BMS_Cell_Min_Max_Values.BMS_Max_Humidity_Location	
23	CAN1.VMU3.Dsh_ApplyBrakeWarningLight	
24	CAN1.ASC1.DoorRelease	
25	CAN1.TCO1.VehicleMotion	
26	CAN1.EBC1.ASRHillHolderSwitch	
27	CAN2.Battery_Config_Msg.Battery_Config	
28	CAN1.BMS_Cell_Min_Max_Values.BMS_Max_Humidity	
29	CAN1.TSC1.EngOverrideCtrlMode_1	
30	CAN1.BMS_Status_Message.BMS_State_Status	
31	CAN2.BCU_HVCU_State_Control.HVCU_State_Control	
32	CAN1.VMU3.Dsh_BuzzerRequest	
33	CAN1.TSC1.EngOverrideCtrlMode	

Çizge Tabanlı Öznelik Seçimi bölümünde, değişkenler arasındaki istatistiksel ilişkiler çizge yapısında görselleştirilmiş, topluluk tespiti algoritmaları kullanılarak öznelikler arasında topluluklar belirlenmiş ve bu topluluklardan ayırt edici öznelikler seçilmiştir. Özellikle SPEC skoru ve K-Shell ayrıştırma yöntemi gibi çizge teorisi prensiplerine dayanan metrikler kullanılarak veri seti boyutunun azaltılması ve daha verimli ve etkili modellerin geliştirilmesi hedeflenmiştir. Bir sonraki Eğitim ve Test Veri Seti Hazırlığı bölümünde, seçilen özneliklerin düzenlenerek eğitim ve test veri setlerinin hazırlanması amaçlanmıştır. Bölümde verilerin uygun şekilde işlenmesi, standardizasyonu ve yeniden etiketlenmesi süreçleri açıklanmıştır. Bu kapsamda gerçekleştirilen veri ön işleme adımları, kübik spline interpolasyon teknikleri ve dengeli veri kümesi oluşturma yöntemleri ele alınmıştır.

6. EĞİTİM VE TEST VERİ SETİ HAZIRLIĞI

Çizge tabanlı öznitelik seçimi sonrası elde edilen özelliklerin kayıtları sadeleştirilmiş veri seti üzerinden çekilerek veri seti üzerinde kapsamlı bir veri işleme ve dönüştürme süreci gerçekleştirilmiştir. Bu süreçte, model eğitime uygun, dengeli ve güvenilir bir veri seti oluşturularak hedef değişkenlerin tahmininde çalıştırılacak makine öğrenmesi modellerin (RF, SVM, XGBoost) etkin ve doğru bir şekilde çalışabilmesi amaçlanmıştır.

İlk olarak, model girdilerinin tutarlılığı ve karşılaştırılabilirliğini sağlamak için belirlenen özniteliklerin kayıtları üzerinde, eksik değerlerin doldurulması (imputation), kategorik değişkenlerin one-hot encoding ile sayısallaştırılması gibi ön işleme adımları ve verilerin standartlaştırılması işlemleri gerçekleştirilmiştir.

İkinci aşamada ise modellerin tahmin yapısına uygun bir veri seti oluşturmak üzere yeniden yapılandırma işlemleri gerçekleştirilmiştir. Öncelikle, hedeflenen alarmların aktivasyon durumlarının erken tahmini için değişkenler üzerinde etiket kaydırma (label shifting) yöntemi uygulanmıştır. Ardından, alarmların aktif olma sıklığının düşük olması nedeniyle ortaya çıkan sınıf dengesizliği problemini çözmek için değişkenlerin aktif olmadığı kayıtlar üzerinden ikili arama tabanlı zaman aralığı örnekleme (binary-search based down-sampling) yöntemi uygulanmıştır. Son olarak, eğitim ve test verileri oluşturulmuş, eğitim verileri üzerinde SMOTEEN (Synthetic Minority Over-sampling Technique for Nominal and Continuous) tekniği ile sentetik örnekler üretilerek sınıf dengesizliği daha da giderilmiş ve eğitim/test veri setleri oluşturulurken katmanlı örnekleme (stratified sampling) yöntemi ile sınıf dağılımlarının korunması sağlanmıştır.

6.1. Özniteliklerin Düzenlenmesi

6.1.1. Veri ön işleme ve standardizasyon

Bu aşamada ilk olarak, özniteliklerin sadeleştirilmiş referans veri seti üzerinden aracın hareket halindeki zaman dilimlerine ait kayıtları toplanmıştır. Sonrasında veri setindeki eksik değerlerin (null değerler) analizi yapılmış ve farklı veri tiplerine uygun stratejiler belirlenerek doldurulmuştur. Sürekli değişkenlerdeki eksik değerler, komşu

değerlerden yararlanarak tahminler üreten kübik Spline İnterpolasyon (cubic spline interpolation) yöntemi ile doldurulurken, kategorik değişkenlerdeki eksiklikler, önceki değerleri tekrarlayarak dolduran İleriye Doğru Doldurma (forward fill imputation) tekniği ile giderilmiştir. Bu sayede, eksik değerlerin neden olabileceği bilgi kaybıyla beraber modellerin performansında görülebilecek olası düşüşlerin önüne geçilmeye çalışılmıştır.

6.1.2. Kübik spline interpolasyon

Spline interpolasyon, verilen veri noktalarından geçen düzgün bir eğri oluşturmak için kullanılan bir yöntemdir. Bu yöntemde, her bir veri noktası arasında farklı bir polinom fonksiyonu tanımlanır ve bu fonksiyonlar birleştirilerek sürekli bir eğri elde edilir. Spline interpolasyonun farklı türleri vardır ve her bir tür farklı bir polinom fonksiyonu kullanır.

En yaygın kullanılan spline interpolasyon türlerinden biri kübik spline interpolasyondur. Kübik spline interpolasyonda, her bir veri noktası arasında üçüncü dereceden bir polinom fonksiyonu kullanılır. Bu fonksiyonlar, birinci ve ikinci türevlerinin sürekliliği sağlanacak şekilde birleştirilir. Kübik spline interpolasyon formülü Eşitlik (6.1) ile gösterildiği şekilde hesaplanır.

$$S_{i(x)} = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (6.1)$$

Burada,

$S_{i(x)}$: x_i ve x_{i+1} noktaları arasındaki eğriyi temsil eden polinom fonksiyonudur.

a_i, b_i, c_i, d_i polinom katsayılarıdır

$$S_{i(x_i)} = y_i \text{ (eğri veri noktasından geçer)} \quad (6.2)$$

$$S_{i(x_{i+1})} = y_{i+1} \text{ (eğri veri noktasından geçer)} \quad (6.3)$$

$$S'_{i(x_{i+1})} = S'_{i+1}(x_{i+1}) \text{ (birinci türev sürekliliği)} \quad (6.4)$$

$$S''_{i(x_{i+1})} = S''_{i+1}(x_{i+1}) \text{ (ikinci türev sürekliliği)} \quad (6.5)$$

Eşitlik (6.2.-6.5) ile gösterilen koşulları kullanılarak, her bir polinom fonksiyonunun katsayıları hesaplanabilir ve böylece tüm veri noktalarından geçen düzgün bir eğri elde edilebilir [70].

Eksik değerlerin doldurulmasıyla elde edilen veri seti üzerinde, makine öğrenimi modellerinin performansını optimize etmek ve özelliklerin ölçeklerini dengelemek amacıyla çeşitli dönüştürme işlemleri gerçekleştirilmiştir. Sürekli özellikler, StandardScaler kullanılarak standartlaştırılmış ve böylece ortalamaları sıfıra, standart sapmaları bir'e getirilerek farklı ölçeklerdeki özelliklerin model üzerindeki etkisinin dengelenmesi sağlanmıştır. Batarya paketlerinin lokasyonunu temsil eden sayısal (numeric) değişkenler ise, nominal ölçekli oldukları ve aralarında bir sıralama ilişkisi bulunmadığı için, one-hot encoding yöntemiyle dönüştürülmüştür. Bu dönüşümle, her bir lokasyon kategorisi için ayrı bir ikili (binary) değişken oluşturulmuştur. Nominal ve sıralı (ordinal) değişkenler ise, önceden label encoding ile sayısallaştırıldığı için bu aşamada herhangi bir dönüşüme tabi tutulmamış ve orijinal halleriyle korunmuştur.

6.2. Yeniden Etiketleme

Bu süreçte, alarmların aktivasyonlarının önceden tahmin edilmesinde sınıflandırma tabanlı denetimli makine öğrenimi (supervised ML) modelleri kullanılmasına karar verildiğinden, alarm sınıflarının etiketlenmesinde belirli değişiklikler yapılmıştır. İlk olarak, Marwin Zufle ve arkadaşlarının çalışmasında [71] bahsedilen etiketleme stratejilerinden yola çıkarak, araç hareketlerinden alarmların aktivasyon periyotları belirlenmiş ve bu periyotların başlangıç, bitiş ve süre bilgileri kaydedilmiştir. Daha sonra, alarmların ne kadar süre önce tahmin edilmesi planlanıyorsa, buna göre bir tahmin penceresi (prediction window) oluşturulmuştur. Her alarmın aktivasyon periyodunun başlangıç zamanından, belirlenen tahmin penceresi kadar geriye gidilmiş ve pencere içinde kalan zaman damgalı kayıtlar yeniden 1 (aktivasyon gözlemlendi) olarak etiketlenmiştir. Son olarak, modelin alarmların aktivasyon gösterdiği aralıklardan ziyade öncesini tahmin etmesini sağlamak amacıyla, aktivasyon özelliği gösterilen periyotlar veriden çıkartılmış sadece pencere işlemi sonrası yeniden etiketlenen veriler bırakılmıştır. Bu yaklaşım, literatürde Olay Tabanlı Etiketleme (event-based labeling) veya Tetikleme Tabanlı Etiketleme (trigger-based labeling) olarak adlandırılmaktadır ve alarmların tahmin edilebilirliğini artırmak amacıyla yaygın olarak kullanılmaktadır.

6.3. Dengeli Veri Kümesi Oluşturma

Bu aşamada, yaklaşık iki yıllık araç hareketi süresince alarmların aktivasyon sıklığı oldukça az olması nedeniyle ortaya çıkan dengesiz veri seti problemiyle başa çıkmak için özel bir yaklaşım benimsenmiştir. Her ne kadar yeniden etiketleme sürecinde, gerçek aktivasyon periyotlarının başlangıç zamanından tahmin penceresi boyutu kadar geriye gidilip, aktivasyon öncesi kayıtlar 1 olarak yeniden sınıflandırılrsa da ve aslında bu şekilde veride (1) olarak etiketlenen kayıt sayısı arttırılmış olsa da hala alarmların aktif olduğu durumların, aktif olmadığı durumlara kıyasla oldukça az sayıda olduğu gözlemlenmiştir. Bu sorunu çözmek için, hedef değişkenlerin aktif olmadığı (0) kayıtlar üzerinde, zaman serisinin yapısını koruyan ve ikili arama (binary search) tabanlı bir zaman aralığı seyreltme (time interval down-sampling) yöntemi geliştirilmiştir. Bu yöntem ile, verinin zaman serisi yapısı korunarak alarmların aktivasyon göstermediği kayıt sayısının, aktivasyon gösterdiği kayıt sayısına olabildiğince yaklaştırılması amaçlanmıştır.

Bu yöntemle, veri seti, belirlenen zaman aralığına göre bölünerek gruplaştırılmış ve gruptan rastgele örnekler alınarak aktivasyon gözlenmeyen kayıt sayısı seyreltilerek, aktivasyon gösterilen kayıt sayısına ne kadar yaklaşıldığı hesaplanmıştır. Aktivasyon gösteren kayıt boyutuna en çok yaklaştıran zaman aralığını belirlemek için ise çalışma zamanı (runtime) açısından verimli olan bir ikili arama algoritması kullanılmıştır. İlk olarak, algoritmanın arama yapacağı zaman aralıklarının penceresini belirlemek için minimum ve maksimum zaman değerleri belirlenmiş ve aramanın gerçekleşeceği pencere oluşturulmuştur. Daha sonra, algoritmaya en yakın olması gereken hedef veri boyutu olarak alarmin aktivasyon gösterdiği kayıt sayısı verilmiştir. İkili arama algoritması çalıştırılarak, belirlenen periyot penceresinden ideal zaman aralığı bulunmuş ve bu değere göre aktivasyon göstermeyen kayıtlar üzerinde seyreltme işlemi gerçekleştirilmiştir.

Algoritma 6.1: Geliştirilen ikili arama tabanlı zaman aralığı seyreltme algoritması.

Algorithm 2 Time Interval Undersampling

Require: Data *data*, Time Column *time_column*, Interval *interval*, Number of Samples *num_sample*

Ensure: Undersampled Data *X*, Target Variable *y*

- 1: Set $df \leftarrow data$
- 2: Set $df[target_name] \leftarrow y$
- 3: Set $df \leftarrow df.set_index(time_column)$
- 4: Initialize *undersampled_data* $\leftarrow []$
- 5: **for all** groups *g* in $df.resample(interval)$ **do**
- 6: **if** length of *g* > *num_sample* **then**
- 7: Append $g.sample(num_sample)$ to *undersampled_data*
- 8: **else**
- 9: Set $updated_sample \leftarrow lengthofg - num_sample$
- 10: Append $g.sample(updated_sample)$ to *undersampled_data*
- 11: **end if**
- 12: **end for**
- 13: Set $undersampled_df \leftarrow pd.concat(undersampled_data)$
- 14: Set $undersampled_X \leftarrow undersampled_df.drop([target_name], axis = 1)$
- 15: Set $undersampled_y \leftarrow undersampled_df[target_name]$
- 16: **return** *undersampled_X*, *undersampled_y*

Algorithm 3 Find Optimal Interval

Require: Data *data*, Target Size *target_size*, Minimum Interval *min_interval*, Maximum Interval *max_interval*

Ensure: Best Interval *best_interval*

- 1: Set *left* $\leftarrow min_interval$
- 2: Set *right* $\leftarrow max_interval$
- 3: Set *best_interval* $\leftarrow max_interval$
- 4: **while** *left* $\leq right$ **do**
- 5: Set $mid \leftarrow \frac{left + right}{2}$
- 6: Set *undersampled_data*, *-* $\leftarrow$ Time Interval Undersampling(*data*, 'times-tamps', *midT*)
- 7: **if** length of *undersampled_data* $\geq target_size$ **then**
- 8: Set *best_interval* $\leftarrow mid$
- 9: Set *left* $\leftarrow mid + 1$
- 10: **else**
- 11: Set *right* $\leftarrow mid - 1$
- 12: **end if**
- 13: **end while**
- 14: **return** *best_interval*

6.3.1. Eğitim ve test veri setlerinin oluşturulması

Oluşturulan dengeli veri ver seti üzerinden, modelin performansını değerlendirmek ve genelleme yeteneğini test etmek için eğitim ve test veri setleri oluşturulmuştur. Bu ayırım, verinin %70'i eğitim, %30'u test olacak şekilde gerçekleştirilmiştir. Veri setinin dengesini korumak ve her iki veri setinin de aynı dağılımdan gelen aktif alarm durumları içermesi için, tabakalı örnekleme (stratified sampling) yöntemi kullanılmıştır. Bu yöntem, veri setini alarm etiketlerine göre alt gruplara ayırarak, her bir alt gruptan oransal olarak örneklem seçilmesini ve böylece eğitim ve test veri setlerinin aktif alarm dağılımlarının birbirine benzer olmasını sağlamıştır. Bu sayede,

her iki veri setinde de etiketlerin orantılı temsili sağlanarak, modelin genel performansının ve genelleme kabiliyetinin artırılması amaçlanmıştır.

Eğitim ve Test Veri Seti Hazırlığı bölümünde, veri setinin özniteliklerinin düzenlenmesi, veri ön işleme ve standardizasyon adımları, kübik spline interpolasyon teknikleri, yeniden etiketleme ve dengeli veri kümesi oluşturma süreçleri ayrıntılı bir şekilde ele alınmıştır. Bir sonraki Modellerin Eğitilmesi ve Açıklanabilirliği bölümünde, modellerin eğitilmesi ve bu modellerin genel ve kayıt bazında açıklanabilirliği üzerinde durulmuştur. Bu kapsamda, model eğitimi, hiperparametre optimizasyonu süreçlerini ve SMOTEEN gibi dengeleme tekniklerinin yanı sıra çeşitli makine öğrenmesi modellerinin eğitimi gerçekleştirilmiştir. Ayrıca, Random Forest (RF) ve XGBoost modellerinin öznitelik önemini (feature importance) nasıl belirlediği detaylandırılmıştır. Bunun yanında, LIME (Local Interpretable Model-Agnostic Explanations) aracı ile belli kayıtlar üzerinden modellerin performansı değerlendirilip, karar mekanizmaları analiz edilerek alarm karakteristikleri üzerinden öngörüler elde edilmiştir.

7. MODELLERİN EĞİTİLMESİ VE AÇIKLANABİLİRLİĞİ

Bu aşamada alarmların aktif olduğu durumların sınıflandırılması için SVM, RF ve XGBoost modelleri eğitim ve test veri setleri ile eğitilerek model performansları değerlendirilmiştir. Modellerin eğitimi sürecinde, sınıf dengesizliğinin giderilmesi ve modellerin performansının artırılması amacıyla hedef değişkenlerin aktif olduğu (1) kayıtların sayısını artırarak veri setini dengelemeyi hedefleyen SMOTEEN ("Synthetic Minority Oversampling Technique-Edited Nearest Neighbours) tekniği kullanılmıştır [72]. Bunun yanında, modellerin en iyi performansı gösterdiği hiper parametreleri ve SMOTEEN tekniğinin en uygun parametrelerini belirlemek için GridSearchCV ile kapsamlı bir hiperparametre optimizasyonu (Hyperparameter Tuning) gerçekleştirilmiştir. Bu optimizasyon sürecinde, modellerin parametreleri ve SMOTEEN fonksiyonunda kullanılan `k_neighbors` ve `enn_n_neighbors` parametreleri sistematik olarak taranarak en iyi sonuçları veren kombinasyonlar belirlenmiştir.

Model eğitiminin ardından, modellerin karar verme mekanizmalarını anlamak ve hangi özelliklerin tahminlerde daha etkili olduğunu belirlemek için açıklanabilirlik analizleri yapılmıştır. Bu analizlerde, her bir modelin özniteliklere verdiği ağırlıklar incelenmiş ve LIME (Local Interpretable Model-Agnostic Explanations) yöntemi kullanılarak modellerin tahminlerinde hangi özniteliklere daha çok odaklandığı tespit edilmiştir [73]. Bu sayede, modellerin karar süreçleri daha şeffaf hale getirilmiş ve sonuçların yorumlanma kabiliyeti arttırılmıştır.

7.1. Model Eğitimi ve Hiperparametre Optimizasyonu

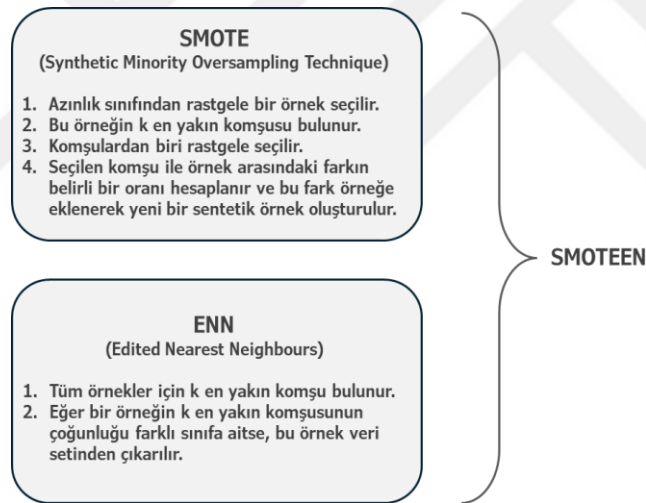
Bu çalışmada hedef değişkenlerin aktivasyon durumlarının kestirimi için güçlü ve yaygın olarak kullanılan üç makine öğrenmesi sınıflandırıcısı olan Destek Vektör Makineleri (SVM), Rastgele Orman (RF) ve Extreme Gradient Boosting (XGBoost) modelleri kullanılmıştır. Bu süreçte, SMOTEEN örnekleme yöntemi ve model eğitimi tek bir yapı altında birleştirmek için bir pipeline (boru hattı) mimarisi kullanılmıştır. Bu yaklaşım hiperparametre optimizasyonunu kolaylaştırarak hem modelin hiperparametrelerini hem de SMOTEEN yönteminin parametrelerini eş

zamanlı optimize edilebilmiş, modellerin ve örnekleme yönteminin en iyi kombinasyonunu bularak, modellerin performansının artmasında yardımcı olmuştur.

7.1.1. SMOTEEN

SMOTEEN (Synthetic Minority Oversampling Technique-Edited Nearest Neighbours), makine öğrenmesi alanında kullanılan bir veri örnekleme (resampling) yöntemidir. Özellikle sınıf dengesizliği probleminin olduğu veri setlerinde, azınlık sınıfına ait örnek sayısını artırarak sınıflar arasındaki dengeyi sağlamaya çalışır. SMOTEEN, SMOTE (Synthetic Minority Oversampling Technique) ve ENN (Edited Nearest Neighbours) yöntemlerinin birleşiminden oluşur.

SMOTEEN, önce SMOTE yöntemiyle azınlık sınıfına sentetik örnekler ekler, ardından ENN yöntemiyle yanlış sınıflandırılmış örnekleri temizler. Bu sayede, hem azınlık sınıfının temsil oranı artırılır hem de gürültülü örnekler veri setinden çıkarılarak modelin performansı iyileştirilir. [72]



Şekil 7.1: SMOTEEN yönteminin çalışma mekanizması.

SMOTEEN parametreleri:

- **sampling_strategy:** Azınlık sınıfının hedef örnek sayısını veya çoğunluk sınıfına göre oranını belirler.
- **k_neighbors:** SMOTE için kullanılan komşu sayısıdır.
- **n_neighbors:** ENN için kullanılan komşu sayısıdır.

- **n_neighbors**: SMOTE'de sentetik örnek oluşturulurken dikkate alınacak komşu sayısıdır (varsayılan değeri k_neighbors ile aynıdır).

7.1.2. Modellerin eğitimi

7.1.2.1. Referans otobüse ait veri seti ile eğitim

Bu çalışma kapsamında, her bir alarmin 20 dk. önceden kestirimi amaçlanmıştır. Öznitelik seçimi yapılmış olan referans 3F2551C1 veri setinden oluşturulmuş eğitim ve test setleri ve Çizelge 7.1'de yer alan parametre uzayı kullanılarak Rastgele Orman, Destek Vektör Makinesi ve XGBoost Modelleri eğitilmiştir.

Çizelge 7.1: Örnekleme parametreleri ve hiperparametre uzayı.

SMOTEEN Parametre Uzayı	
smote_k_neighbors	[1, 3, 5]
enn_n_neighbors	[3, 5, 7]
SVM Parametre Uzayı	
classifier_C	[0.1, 1, 10]
classifier_gamma	[1, 0.1, 0.01]
classifier_kernel	['rbf', 'poly', 'sigmoid']
RF Parametre Uzayı	
n_estimators	[100, 200]
max_depth	[None, 10, 20]
min_samples_split	[2, 5]
min_samples_leaf	[1, 2]
XGBoost Parametre Uzayı	
n_estimators	[100, 200]
max_depth	[3, 6, 10]
learning_rate	[0.01, 0.1, 0.2]
subsample	[0.8, 1.0]
colsample_bytree	[0.8, 1.0]
eval_metric	['auc', 'logloss']

Model eğitimi sonuçları Çizelge 7.2'de yer almaktadır. Sonuçlar incelendiğinde modellerin yüksek performans sergilediği gözlenmiş ve aşırı öğrenme durumunu test etmek için diğer otobüslerden elde edilen verilerin kullanılması kararlaştırılmıştır.

Çizelge 7.2: Referans veri seti üzerinde model başarımları.

CAN1.EBC1.ABS_EBSAmberWarningSignal				
	ACC	PRE	REC	F1
SVM	0.99	0.97	0.98	0.99
RF	0.98	0.96	0.97	0.98
XGBoost	0.99	0.98	0.99	0.98
CAN1.VMU1.Dsh_System_Fault				
	ACC	PRE	REC	F1
SVM	0.98	0.99	0.98	0.99
RF	0.99	0.99	1.00	1.00
XGBoost	1.00	0.99	1.00	0.99
CAN1.VMU1.Dsh_System_Warning				
	ACC	PRE	REC	F1
SVM	0.98	0.99	0.98	0.99
RF	0.99	0.99	1.00	1.00
XGBoost	1.00	0.99	1.00	0.99
CAN1.BMS_Status_Message.PDU_FAULT				
	ACC	PRE	REC	F1
SVM	0.99	1.00	1.00	1.00
RF	0.99	1.00	1.00	1.00
XGBoost	0.99	1.00	1.00	1.00
CAN1.Battery_Status.BMS_SOC				
	ACC	PRE	REC	F1
SVM	0.97	0.97	0.99	0.98
RF	1.00	0.99	1.00	1.00
XGBoost	1.00	1.00	1.00	1.00

7.1.2.2. Diğer otobüslere ait veri setleri ile eğitim

Öncelikle diğer otobüs verilerinde hedeflenen alarmların görülme sıklığı Çizelge 7.3'te görüldüğü üzere incelenmiştir. Diğer Otobüslerin veri setlerinde DSH System Fault ve PDU Fault alarmlarının aktive olduğu kayıtların bulunmadığı, diğer alarmlarda ise referans veri setine kıyasla daha az kayıt olduğu görülmektedir.

Çizelge 7.3: Otobüs verilerinde hedeflenen alarmların sıklığı.

Veri Setlerine Göre Hedef Değişken Sıklığı					
	3F2551C1 (Ref.)	5E3DA3A9 - 0	6BF5FA70 - 1	920CB615- 2	EF818714- 4
CAN1.EBC1.ABS_EBSAmberWarningSignal	4700	562	136	350	4819
CAN1.VMU1.Dsh_System_Fault	57	1	0	0	1
CAN1.VMU1.Dsh_System_Warning	83	41	10	0	21
CAN1.BMS_Status_Message.PDU_FAULT	6	0	0	0	0
CAN1.Battery_Status.BMS_SOC	25	12	5	0	9
Ortalama	974.20	123.2	30.2	70.00	970.00

Diğer otobüslere ait veri setlerinden referans veri seti kullanılarak seçilen özneliklere karşılık gelen kayıtların çekilmesiyle her bir otobüs için eğitim ve test verisi oluşturulmuştur.

Referans veri seti ile eğitilen modeller diğer otobüs verileri ile test edilmiştir. Sonuçlar Çizelge 7.4'te görülmekte olup, referans otobüs verisi ile eğitilen modellerin diğer otobüslere ait alarmların aktif olduğu durumları kestiriminde başarısız olduğu görülmüştür.

Çizelge 7.4: Referans otobüs verileri ile test sonuçları.

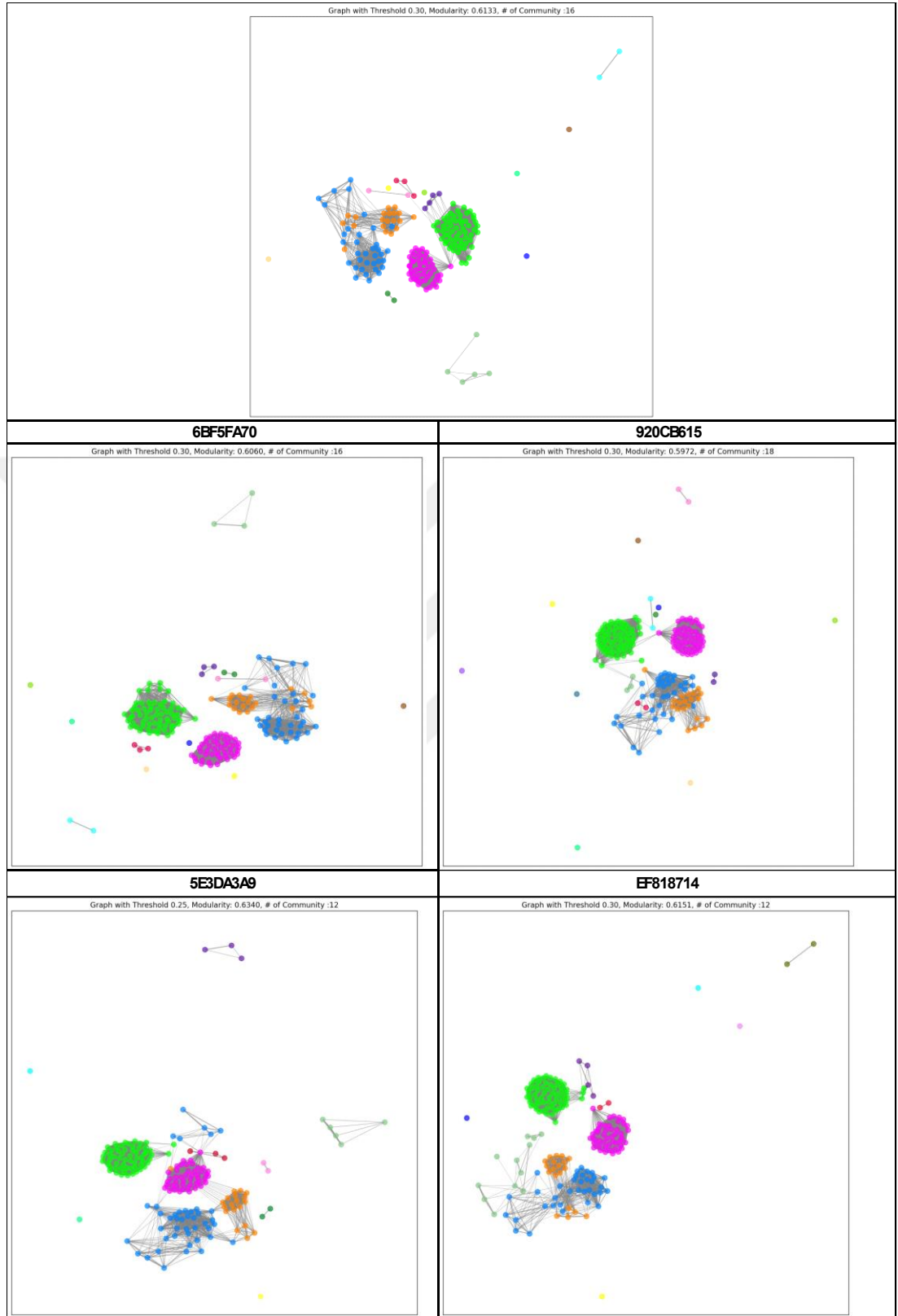
CAN1.EBC1.ABS_EBSAmberWarningSignal		
	ACC	REC
Veri Seti 5E3DA3A9	0.58	0.08
Veri Seti 6BF5FA70	0.51	0.00
Veri Seti 920CB615	0.50	0.00
Veri Seti EF818714	0.62	0.1
CAN1.VMU1.Dsh_System_Fault		
	ACC	REC
Veri Seti 5E3DA3A9	(-)	
Veri Seti 6BF5FA70	(-)	
Veri Seti 920CB615	(-)	
Veri Seti EF818714	(-)	
CAN1.VMU1.Dsh_System_Warning		
	ACC	REC
Veri Seti 5E3DA3A9	0.56	0.06
Veri Seti 6BF5FA70	0.52	0.00
Veri Seti 920CB615	(-)	
Veri Seti EF818714	0.54	0.05
CAN1.BMS_Status_Message.PDU_FAULT		
	ACC	REC
Veri Seti 5E3DA3A9	(-)	
Veri Seti 6BF5FA70	(-)	
Veri Seti 920CB615	(-)	
Veri Seti EF818714	(-)	
CAN1.Battery_Status.BMS_SOC		
	ACC	REC
Veri Seti 5E3DA3A9	0.55	0.03
Veri Seti 6BF5FA70	0.50	0.00
Veri Seti 920CB615	(-)	
Veri Seti EF818714	0.51	0.00

Bu durum karşısında diğer otobüslere ait ham verilerin referans otobüs verisi ile benzerlik durumu kontrol edilmiştir. Referans ham veri seti ile diğer otobüslere ait ham veri setlerindeki ortak değişkenlerin istatistiksel analizleri (Pearson Korelasyon

Matrisi, Cramer's V İstatistik Tablosu ve Anova F-Testi) yapılmış ve değişken ilişkilerinin temsil edildiği çizge yapıları oluşturulmuştur. Bu çizge yapıları üzerinde topluluk algoritmaları çalıştırılarak modülerite skoru yüksek, alakalı özellik kümelerini içeren rafine çizge yapıları (Çizelge 7.5) elde edilmiştir.



Çizelge 7.5: Otobüslere ait rafine edilmiş çizge yapıları.



Bu çalışmaya paralel olarak, diğer otobüslere ait ham verilerin referans otobüse ait ham veriye olan benzerlik seviyesini incelemek için Spektral Benzerlik Skorları hesaplanmış olup, Çizelge 7.6’da verilmiştir.

Çizelge 7.6: Diğer otobüslerin referans otobüse benzerliğinin spektral benzerlik tablosu.

	Referans Veri Setine Spektral Benzerlik Skoru
Veri Seti 6BF5FA70	253.3
Veri Seti 920CB615	196.03
Veri Seti 5E3DA3A9	186.66
Veri Seti EF818714	166.4

Çizelge 7.5’te yer alan her bir otobüse ait rafine çizge yapıları ile Çizelge 7.6’da bulunan Spektral Benzerlik Skorları birlikte değerlendirildiğinde, otobüslerin veri karakteristiklerinin farklı olduğu ancak veri sayısı yüksek olan otobüslerin referans veri setine daha çok benzediği görülmüştür.

7.1.2.3. Genelleştirilmiş model eğitimi

Otobüslerin veri setleri (değişkenleri arasındaki ilişkilerin) farklılığı tespit edildikten sonra referans veri setinin genelleştirme yeteneğini arttırmak için, referans otobüsün ve diğer otobüslerin eğitim setleri aynı oranda birleştirilerek yeni bir eğitim veri seti oluşturulmuştur. Hesaplama verimliliği ve yorumlanabilirliği yüksek olan Rastgele Orman (RF) Modeli seçilerek, yeni oluşturulan eğitim veri seti ile eğitilmiştir. Çizelge 7.7’de görüldüğü üzere eğitilen model her otobüsün test verisi üzerinde %90 üzerinde başarı yakalayarak iyi bir performans sergilemiştir.

Çizelge 7.7: Genelleştirilmiş model eğitimi sonuçları

CAN1.EBC1.ABS_EBSAmberWarningSignal		
	ACC	REC
Veri Seti 5E3DA3A9	0.98	0.98
Veri Seti 6BF5FA70	0.99	1.00
Veri Seti 920CB615	1.00	1.00
Veri Seti EF818714	0.96	0.92
CAN1.VMU1.Dsh_System_Fault		
	ACC	REC
Veri Seti 5E3DA3A9	(-)	
Veri Seti 6BF5FA70	(-)	
Veri Seti 920CB615	(-)	
Veri Seti EF818714	(-)	
CAN1.VMU1.Dsh_System_Warning		
	ACC	REC
Veri Seti 5E3DA3A9	0.90	0.87
Veri Seti 6BF5FA70	0.99	0.98
Veri Seti 920CB615	(-)	
Veri Seti EF818714	0.95	89
CAN1.BMS_Status_Message.PDU_FAULT		
	ACC	REC
Veri Seti 5E3DA3A9	(-)	
Veri Seti 6BF5FA70	(-)	
Veri Seti 920CB615	(-)	
Veri Seti EF818714	(-)	
CAN1.Battery_Status.BMS_SOC		
	ACC	REC
Veri Seti 5E3DA3A9	0.99	0.99
Veri Seti 6BF5FA70	0.97	0.96
Veri Seti 920CB615	(-)	
Veri Seti EF818714	0.90	0.95

7.2. Modellerin Açıklanabilirliği

Modellerin eğitimi ve değerlendirilmesi sonrası, alarmların aktivasyon durumlarının tahmininde kullanılan SVM, RF ve XGBoost modellerinin karar mekanizmalarını anlamak ve tahminlerde hangi özelliklerin daha etkili olduğunu belirlemek amacıyla açıklanabilirlik analizleri gerçekleştirilmiştir. Bu analizler kapsamında, her bir alarm için eğitilen modellerin tahminlerini yorumlamak ve hangi özelliklere dayalı kararlar verdiğini anlamak için LIME (Local Interpretable Model-Agnostic Explanations) yöntemi kullanılmıştır.

Bunun yanında, RF ve XGBoost modelleri için yerleşik olarak sunulan özellik önemi (feature importance) skorları hesaplanmış ve LIME sonuçlarıyla karşılaştırılarak farklı açıklanabilirlik yöntemlerinin sonuçlarının tutarlılığı ve güvenilirliği değerlendirilmiştir. Özellik önemi skorları, modelin genel olarak hangi özelliklere daha çok ağırlık verdiğini gösterirken; LIME, belirli bir örnek için modelin kararını etkileyen özellikleri ve bu özelliklerin etkileşimini ortaya koymaktadır.

7.2.1. LIME (Local Interpretable Model-Agnostic Explanations)

LIME (Local Interpretable Model-Agnostic Explanations), karmaşık makine öğrenimi modellerinin tahminlerini açıklamak için kullanılan Model agnostik bir yöntemdir, modelden bağımsız çalışır ve herhangi bir sınıflandırma veya regresyon modeli için kullanılabilir. Bu yöntem, modelin tahminlerini yerel olarak açıklar, belirli bir örnek için modelin kararını etkileyen özellikleri ve bu özelliklerin etkileşimini analiz eder ve görselleştirir.

Algoritma 7.1: LIME' algoritma adımları

1. Örnek Seçim:

Açıklanacak örnek seçilir.

2. Pertürbasyon:

Seçilen örneğin etrafında küçük değişiklikler (pertürbasyonlar) yapılarak yeni örnekler oluşturulur

3. Model Tahminleri

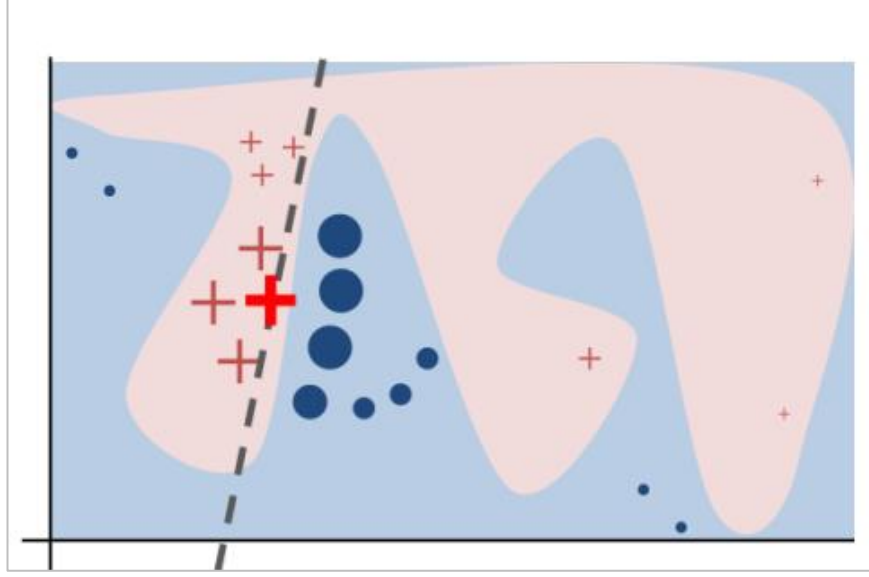
Oluşturulan örnekler üzerinde orijinal model çalıştırılarak tahminler elde edilir.

4. Yerel Model Eğitimi:

Pertürbe edilmiş örnekler ve model tahminleri kullanılarak açıklanabilir bir model (doğrusal regresyon veya karar ağacı) eğitilir.

4. Yerel Model ile Açıklama:

Eğitilen açıklanabilir model, orijinal modelin seçilen örnek için tahminini açıklamak için kullanılır.



Şekil 7.2: LIME çalışma prensibi.

Şekil 7.2: LIME'in karmaşık modellerin tahminlerini yerel düzeyde nasıl açıklayabileceğini göstermektedir. LIME, açıklanmak istenen veri noktasının etrafında yeni örnekler oluşturarak ve bu örnekler üzerinde orijinal modelin tahminlerini alarak basit ve yorumlanabilir bir model oluşturur. Bu basit model, sadece açıklanan veri noktasının etrafında geçerli olan bir açıklama sunar [73].

Arka Plan (Mavi/Pembe Bölge): Siyah kutu modelinin karmaşık karar sınırlarını temsil eder. Bu sınırlar, doğrusal olmayan ve karmaşık yapıda olup, basit bir modelle tam olarak açıklanamaz.

Kalın Kırmızı Çarpı (Açıklanan Örnek): Açıklanmak istenen veri noktasıdır. Bu nokta, LIME'in hedef aldığı ve açıklama üretmek istediği örnektir.

Örneklenen Veriler (Mavi ve Kırmızı Noktalar): LIME, açıklanmak istenen veri noktasının etrafında rastgele örnekler üretir. Bu örnekler, orijinal model f (orijinal modelin sınıflandırıcı fonksiyonu) üzerinde çalıştırılır ve tahminler alınır.

Örneklerin Ağırlıkları (Nokta Boyutları): Üretilen örnekler, açıklanmak istenen veri noktasına yakınlıklarına göre ağırlıklandırılır. Görselde, yakın olan noktalar daha büyük (daha ağır) gösterilmiştir. Bu ağırlıklandırma, açıklamanın lokal (yerel) olmasını sağlar.

Kesik Çizgi (Yerel Açıklama Modeli): Bu çizgi, LIME tarafından oluşturulan ve orijinal modelin f sınıflandırıcı fonksiyonunu açıklanan örnek etrafında yaklaşık olarak

temsil eden doğrusal modeldir. Bu model, sadece yerel olarak (örnek etrafında) sadıktır ve küresel olarak (genel) orijinal modeli iyi temsil edemez.

7.3. Modellerin Genel Yorumlanabilirliği

Diğer Otobüs verileri ile referans veri setindeki verilerin farklılığının Spektral Benzerlik Skoru ile tespit edilmesi sonrası her otobüs her bir alarm tahmini için Rastgele Orman ve XGBoost Modelleri üzerinde ayrı ayrı eğitilmiştir ve her modelin karar verme sürecinde en çok önem verdiği ilk 5 öznelik, modellerin performansı ile beraber Çizelge 7.8 ve Çizelge 7.9’da görüldüğü üzere kaydedilmiştir. Rastgele Orman modelinde modelin önem verdiği öznelikler, eğitim sırasında öznelikler üzerinde her bir karar ağacında hesaplanan Gini Azalması (Gini Impurity Reduction) ve Bilgi Kazancı (Information Gain) değerlerinin ortalaması alınarak belirlenmiştir. XGBoost Modelinde ise modelin önem verdiği öznelikler belirlenirken, her bir özelliğin modelde kaç kez kullanıldığını gösteren, özelliğin tüm ağaçlarda kaç kez bölünme noktası olarak kullanıldığını ifade eden Ağırlık (Weight) değeri kullanılmıştır. Elde edilen önemli öznelikler incelendiğinde, aslında modellerin hedeflenen alarmlardan ve modelden bağımsız olarak ortak özneliklere bakarak karar verdiği gözlemlenmiştir. Bu öznelikler, (i) balata kalınlığı öznelikleri (BrkLnngRmningFrontAxleRight/LeftWheel), (ii) batarya öznelikleri (BMS_Max_Cell_Voltage, BMS_Max_Humidity), (iii) motor sıcaklığı ve soğutma sistemi öznelikleridir (Water_Temp_Radiator_Out, Dsh_ThermalIndicator). Bu öznelikler aynı zamanda anomali analizi aşamasında alarmların aktif olduğu periyotlarda yüksek frekansta anomali gösteren değişkenler olarak dikkat çekmişti. Bunun yanında özneliklerin ilişkili olduğu araç bileşenlerinin Çizelge 4.2’de yer alan hedeflenen alarmların araç bileşen tanımları ile uyumlu olduğu görülmüştür. Ancak her ne kadar modeller karar verirken ortak özneliklere baksalar bile, Çizelge 7.4’te görüldüğü üzere, herhangi bir alarm sinyali için herhangi bir otobüs verisi ile eğitilmiş model, diğer otobüs verilerinde alarmların aktif olduğu kayıtları tahmin etmekte düşük performans göstermektedir.

Eğer Çizelge 7.7’de görüldüğü gibi genelleştirilmiş model kullanılır ise bu sorunun önüne geçilmekte ve tahminlerde yüksek performans elde edilmektedir. Bu durum otobüsler arasındaki araç bileşen ilişkilerinin benzer ancak otobüslerin farklı karakteristikleri olmasından kaynaklı olabileceği düşünülmüştür.

7.4. Modellerin Kayıt Bazında Yorumlanabilirliği

Genelleştirmiş Rastgele Orman Modeli üzerinde her bir alarmın 3 tane aktif olan durumu 1 tane aktif olmayan durumu tahmininde LIME çalıştırılarak, modellerin alarmların spesifik kayıtlarında nasıl karar verdiği açıklanmıştır.

Çizelge 7.10'da bulunan çıktılar incelendiğinde Amber Warning Fren ve güvenlik sistemi alarmının karar verme sürecinde motorun aşırı ısındığını gösteren Water_Radiator_out özniteliğinin ayırt edici öznitelik olduğu görülmüştür. Bu öznitelik rastgele seçilen tüm örneklerde model kararını büyük oranda etkileyen bir özniteliktir. Bunun yanında, BMS_SOC alarmının daha çok batarya sistemine bağlı değişkenlerden etkilendiği, modelin karar verirken BMS_Max_Cell_Voltage, BMS_Max_Humidity ve BMS_Max_Humidity_Location gibi değişkenlere daha çok önem verdiği tespit edilmiştir. DSH System Warning alarmı genel bir alarm olmasının da etkisiyle modelin her örnekte aracın farklı bileşenlerinden etkilenecek karar verdiği gözlenmiştir.

Çizelge 7.8: Alarm tahminlerinin LIME ile açıklanması.

Alarm	Amber Warning Alarm	DSH-System Warning Alarm	BMS SOC Alarm	Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
Alarm	Amber Warning Alarm	DSH-System Warning Alarm	BMS SOC Alarm	Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
Alarm	Amber Warning Alarm	DSH-System Warning Alarm	BMS SOC Alarm	Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
Alarm	Amber Warning Alarm	DSH-System Warning Alarm	BMS SOC Alarm	Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
Alarm	Amber Warning Alarm	DSH-System Warning Alarm	BMS SOC Alarm	Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC
				Class 1 Instance of CAN1 Battery Status BMS SOC

Çizelge 7.9: Rastgele orman modelinin performansı ve öznitelik önemi.

CAN1.Battery_Status.BMS_SOC Rastgele Orman Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BMS_Max_Humidity	Water_Temp_Radiator_Out□		BrkLnngRmningFrontAxeLeftWheel	BrkLnngRmningFrontAxeLeftWheel
BrkLnngRmningFrontAxeRightWheel	BrkLnngRmningFrontAxeRightWheel		Water_Temp_Radiator_Out□	BrkLnngRmningFrontAxeRightWheel
BrkLnngRmningFrontAxeLeftWheel	BrkLnngRmningFrontAxeLeftWheel		BMS_Max_Cell_Voltage	BMS_Max_Cell_Voltage
Water_Temp_Radiator_Out	BMS_Max_Cell_Voltage		BrkLnngRmningFrontAxeRightWheel	Water_Temp_Radiator_Out□
Dsh_ThermalIndicator	Dsh_ThermalIndicator		Dsh_ThermalIndicator	Dsh_ThermalIndicator
ACC: 1.00	ACC: 0.99		ACC: 1.00	ACC: 1.00
PRE: 1.00	PRE: 0.99		PRE: 1.00	PRE: 1.00
REC: 0.99	REC: 1.00		REC: 1.00	REC: 1.00
F1: 1.00	F1: 0.99		F1: 1.00	F1: 1.00

CAN1.EBC1.ABS_EBSAmberWarningSignal Rastgele Orman Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BrkLnngRmningFrontAxeLeftWheel	Water_Temp_Radiator_Out□	BrkLnngRmningFrontAxeLeftWheel	BMS_Max_Cell_Voltage	Water_Temp_Radiator_Out□
BrkLnngRmningFrontAxeRightWheel	BMS_Max_Cell_Voltage	BrkLnngRmningFrontAxeRightWheel	Water_Temp_Radiator_Out	BrkLnngRmningFrontAxeRightWheel
BMS_Max_Humidity	Dsh_ThermalIndicator	act_DCBusPower_14	BrkLnngRmningFrontAxeLeftWheel	BrkLnngRmningFrontAxeLeftWheel
BMS_Max_Cell_Voltage	BMS_Max_Humidity	BMS_Max_Cell_Voltage	BrkLnngRmningFrontAxeRightWheel	BMS_Max_Humidity
Water_Temp_Radiator_Out	BrkLnngRmningFrontAxeLeftWheel	BMS_Max_Humidity	BMS_Max_Humidity	BMS_Max_Cell_Voltage
ACC: 0.99	ACC: 0.99	ACC: 0.99	ACC: 1.00	ACC: 1.00
PRE: 1.00	PRE: 0.99	PRE: 0.98	PRE: 1.00	PRE: 0.99
REC: 0.97	REC: 0.98	REC: 0.99	REC: 1.00	REC: 0.98
F1: 0.99	F1: 0.99	F1: 0.99	F1: 1.00	F1: 1.00

CAN1.VMU1.Dsh_System_WarningRastgele Orman Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BMS_Max_Humidity	Water_Temp_Radiator_Out		BrkLnngRmningFrontAxeLeftWheel	BrkLnngRmningFrontAxeLeftWheel
BrkLnngRmningFrontAxeRightWheel	BrkLnngRmningFrontAxeLeftWheel		BMS_Max_Humidity	BrkLnngRmningFrontAxeRightWheel
BrkLnngRmningFrontAxeLeftWheel	BMS_Max_Cell_Voltage		BrkLnngRmningFrontAxeRightWheel	BMS_Max_Humidity
BMS_Max_Cell_Voltage	Dsh_ThermalIndicator		Water_Temp_Radiator_Out	Water_Temp_Radiator_Out
Water_Temp_Radiator_Out	BMS_Max_Humidity		Dsh_ThermalIndicator	BMS_Max_Cell_Voltage
ACC: 0.99	ACC: 1.00		ACC: 1.00	ACC: 1.00
PRE: 0.99	PRE: 0.99		PRE: 0.99	PRE: 0.99
REC: 1.00	REC: 0.99		REC: 0.99	REC: 0.98
F1: 0.99	F1: 1.00		F1: 1.00	F1: 1.00

CAN1.VMU1.Dsh_System_Fault Rastgele Orman Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BrkLnngRmningFrontAxeRightWheel				
BrkLnngRmningFrontAxeLeftWheel				
BMS_Max_Humidity				
BMS_Max_Cell_Voltage				
Water_Temp_Radiator_Out				
ACC: 0.99				
PRE: 0.99				
REC: 1.00				
F1: 0.99				

CAN1.BMS_Status_Message.PDU_FAULT Rastgele Orman Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BMS_Max_Humidity				
BrkLnngRmningFrontAxeLeftWheel				
act_DCBusPower_14				
BMS_Max_Cell_Voltage				
BrkLnngRmningFrontAxeRightWheel				
ACC: 1.00				
PRE: 1.00				
REC: 1.00				
F1: 1.00				

Çizelge 7.10: Xgboost modelinin performansı ve öznitelik önemi.

CAN1.Battery_Status.BMS_SOC XGBoost Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BrkLnngRmningFrontAxeRightWheel	BMS Max_Cell_Voltage		Water_Temp_Radiator_Out	BMS Max_Cell_Voltage
BMS Max_Humidity	Dsh_ThermalIndicator		BMS Max_Cell_Voltage□	BrkLnngRmningFrontAxeRightWheel
Water_Temp_Radiator_Out	Water_Temp_Radiator_Out□		Dsh_ThermalIndicator	BrkLnngRmningFrontAxeLeftWheel
BrkLnngRmningFrontAxeLeftWheel	BMS Max_Humidity□		act_DCBusPower_14	Water_Temp_Radiator_Out□
BMS Max_Cell_Voltage	BrkLnngRmningFrontAxeLeftWheel□		BrkLnngRmningFrontAxeLeftWheel	BMS Max_Humidity□
ACC: 0.99	ACC: 0.99		ACC: 1.00	ACC: 1.00
PRE: 0.99	PRE: 0.98		PRE: 0.98	PRE: 1.00
REC: 1.00	REC: 1.00		REC: 1.00	REC: 1.00
F1: 0.99	F1: 0.99		F1: 0.99	F1: 1.00

CAN1.EBC1.ABS_EBSAmberWarningSignal XGBoost Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BrkLnngRmningFrontAxeLeftWheel	BMS Max_Cell_Voltage	BMS Max_Cell_Voltage	BMS Max_Cell_Voltage	Water_Temp_Radiator_Out□
Water_Temp_Radiator_Out□	Water_Temp_Radiator_Out	BrkLnngRmningFrontAxeRightWheel	Water_Temp_Radiator_Out	BMS Max_Cell_Voltage
BMS Max_Humidity□	BMS Max_Humidity	act_DCBusPower_14	BMS Max_Humidity	BrkLnngRmningFrontAxeLeftWheel
BrkLnngRmningFrontAxeRightWheel	BMS Max_Humidity_Location	Water_Temp_Radiator_Out	Dsh_ThermalIndicator	BMS Max_Humidity
BMS Max_Cell_Voltage	BrkLnngRmningFrontAxeRightWheel	ServiceBrakeCircuit2AirPress	BrkLnngRmningFrontAxeLeftWheel	Dsh_ThermalIndicator
ACC: 0.98	ACC: 1.00	ACC: 1.00	ACC: 0.99	ACC: 0.99
PRE: 0.97	PRE: 0.99	PRE: 0.99	PRE: 0.98	PRE: 0.98
REC: 1.00	REC: 1.00	REC: 1.00	REC: 0.99	REC: 1.00
F1: 0.98	F1: 1.00	F1: 0.99	F1: 0.99	F1: 0.99

CAN1.VMU1.Dsh_System_Warning XGBoost Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BrkLnngRmningFrontAxeLeftWheel	BMS Max_Cell_Voltage		BMS Max_Cell_Voltage	BrkLnngRmningFrontAxeRightWheel
BrkLnngRmningFrontAxeRightWheel	Dsh_ThermalIndicator		BrkLnngRmningFrontAxeLeftWheel	BMS Max_Cell_Voltage
BMS Max_Humidity	Water_Temp_Radiator_Out		Water_Temp_Radiator_Out	BrkLnngRmningFrontAxeLeftWheel
BMS Max_Cell_Voltage	BMS Max_Humidity_Location		BMS Max_Humidity	BMS Max_Humidity
Dsh_ThermalIndicator□	BMS Max_Humidity		act_DCBusPower_14	Water_Temp_Radiator_Out
ACC: 0.98	ACC: 0.99		ACC: 0.99	ACC: 0.99
PRE: 0.97	PRE: 0.99		PRE: 0.99	PRE: 0.98
REC: 0.99	REC: 1.00		REC: 1.00	REC: 1.00
F1: 0.98	F1: 0.99		F1: 0.99	F1: 0.99

CAN1.VMU1.Dsh_System_Fault XGBoost Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BrkLnngRmningFrontAxeLeftWheel□				
BMS Max_Humidity				
BrkLnngRmningFrontAxeRightWheel				
BMS Max_Cell_Voltage				
Dsh_ThermalIndicator□				
ACC: 0.98				
PRE: 0.97				
REC: 0.99				
F1: 0.98				

CAN1.BMS_Status_Message.PDU_FAULT XGBoost Tahmin Sürecinde Öznitelik Önem Sırası				
Reference	Data1	Data2	Data4	Data0
BMS Max_Humidity				
BrkLnngRmningFrontAxeLeftWheel				
act_DCBusPower_14				
BMS Max_Cell_Voltage				
BrkLnngRmningFrontAxeRightWheel				
ACC: 1.00				
PRE: 1.00				
REC: 1.00				
F1: 1.00				

8. DEĞERLENDİRME

Bu tez çalışmasında kestirimci bakım başta olmak üzere düzeltici, önleyici ve proaktif bakım stratejilerini destekleyecek bir makine öğrenmesi çalışması gerçekleştirilmiştir.

Ham veri paketleri üzerinde kapsamlı ön işleme teknikleri uygulanmış; Pearson korelasyon analizi, Cramer's V istatistik tablosu ve Anova F-testi gibi istatistiksel filtreleme yöntemleri ve InfoMap, Leiden, Louvain, Fast Greedy gibi optimizasyon tabanlı topluluk tespiti algoritmaları kullanılarak, hibrit çizge tabanlı öznitelik seçim aracı geliştirilmiştir. Geliştirilen araç kullanılarak özellik altkümeleri belirlenmiş, özelliklerin ilişkisel konumu ve hedeflenen alarmlarla ilişkileri değerlendirilerek ayırt edici öznitelikler seçilmiş ve boyut azaltımı yapılmıştır. Destek Vektör Makinesi (SVM), Rastgele Orman (RF), eXtreme Gradient Boosting (XGBoost) uygulanarak performansları değerlendirilmiştir. Sınıf dengesizliğinin modellerin performansındaki etkisini minimize etmek için SMOTEEN (Synthetic Minority Oversampling Technique-Edited Nearest Neighbours) ve ikili arama tabanlı zaman aralığı örnekleme (binary search based time interval down-sampling) yöntemleri uygulanmıştır. Her modelin en iyi ayırt edici sınıflandırıcı uzayını bulmak için Izgara Arama (Grid Search) ve Rastgele Arama (Random Search) tabanlı hiperparametre ayarlama (Hyperparameter tuning) yapılmıştır. Geliştirilen modellerin performansları geçerleme (validation) metrikleri esas alınarak değerlendirilmiştir. En iyi performans gösteren modellerin kestirimleri özniteliklerin ağırlıkları incelenerek ve LIME (Local Interpretable Model-Agnostic Explanations) aracı kullanılarak açıklanmaya çalışılmış, kestirimleri tetikleyen öznitelikler belirlenmiş ve sistemin işleyişi (tasarımı) ile ilgili içgörüler elde edilmiştir.

Projede elektrikli otobüslere yönelik belirlenen alarmların kestirim hedeflerine ulaşmaya çalışırken, sahadan alınan veri setlerindeki farklılıklar, veri kalitesi, veri boyutunun büyüklüğü, otobüs sisteminin karmaşıklığı, değişkenler arasındaki ilişkileri yorumlama zorlukları, uzman görüşüne erişim zorlukları ve sınırlı hesaplama kapasitesi gibi gerçek hayatta karşılaşılan tipik makine öğrenmesi riskleri yönetilmeye çalışılmıştır.

Projedeki riskleri azaltmak için makine öğrenimi yaşam döngüsü iteratif ve çapraz kontrollere izin verecek şekilde planlanmış ve disiplinli bir şekilde uygulanmıştır. Veri setleri analizinde pek çok farklı teknik birlikte kullanılmış, elde edilen sonuçların doğrulamalarına özen gösterilmiş ve çizge tabanlı bir öznitelik seçme aracı geliştirilerek öznitelikler arasındaki ilişkilerin görsel olarak yorumlanması imkânı sağlanmıştır. Alarmların aktif olduğu durumlar 20.dk öncesinden, otobüse özgü eğitilen modellerde ve tüm otobüslerin verisi ile eğitilmiş genelleştirilmiş modelde, en az 0.97 doğrulukla kestirilebilmektedir. Bunun yanında farklı modellerin performansları validasyon metrikleri ile ölçülmüş, LIME ve Öznitelik Önem Analizi ile karar verme süreçleri açıklanmaya çalışılmıştır.



9. GELECEK ÇALIŞMALAR

Bu çalışmada endüstriyel verilerden (sahadan) kaynaklı kısıtlar ve zorluklar olduğu görülmüştür. Gelecek araştırmalarda, daha geniş ve zengin veri setleriyle tez çalışmasında kullanılan yöntemlerin daha başarılı sonuçlar vereceği değerlendirilmektedir. Ayrıca, proje gereksinimlerine göre uyarlanan makine öğrenmesi yaşam döngüsü ve kullanılan tekniklerin benzer karakteristikteki endüstriyel projelerde de kullanılabileceği düşünülmektedir.

Makine öğrenimi ile kestirimci bakım alanında yapılacak gelecek çalışmalarda, veri toplama ve hazırlama süreçlerinin iyileştirilmesinin model performansını doğrudan etkileyeceği öngörülmektedir. Araç verilerinin bütünlük içinde ve yeterli miktarda toplanması, veri kalitesinin sağlanması ve veri setinin CAN Bus dışı verilerle zenginleştirilmesi, modelin gerçek dünya koşullarını daha iyi öğrenmesini sağlayacaktır. Veri setinin dengeli oluşturulması ve model çıktılarının kontrol edilmesiyle modelin güvenilirliği artacaktır. Sistemlerin konfigürasyon farklılıkları ile veri setleri arasındaki izlenebilirlik bilgisinin korunması ise sonuçların yorumlanabilirliğini kolaylaştıracaktır.

Öznitelik seçimi ve analizi süreçlerinde çizge tabanlı yöntemlerin kullanımı, verinin daha derinlemesine anlaşılmasını sağlayacaktır. Büyük ve karmaşık çizge tabanlı öznitelik temsillerinde ise hipergraflar ve maksimum clique gibi tekniklerin daha etkili çözümler sunabileceği öngörülmektedir.

Makine öğrenimi yaşam döngüsünün proje risklerine göre planlanması ve disiplinli bir şekilde uygulanması, başarılı bir kestirimci bakım sistemi geliştirmenin temelini oluşturacaktır. Sistemlerin ölçeklenebilirliği, gerçek zamanda işleme ve mevcut bakım iş akışlarıyla entegrasyon gibi faktörler, gerçek dünya uygulamalarında karşılaşılabilecek zorlukların üstesinden gelinmesine yardımcı olacaktır.

Gerçek dünyaya ait yeterli verinin olduğu durumlarda LSTM, RNN, CNN, GRU gibi derin öğrenme modellerin eğitilmesi veya önceden eğitilmiş modellerin kullanılması durumunda kestirim başarımının artırılabilceği, voting, stacking ve bagging gibi ensemble teknikleri ile model genelleştirilebilirliği yüksek sistemlerin oluşturulabileceği düşünülmektedir. Verideki boşlukların doldurulmasında istatistiki

tekniklerin yanında K-Nearest Neighbors (KNN) ve Multiple Imputation by Chained Equations (MICE) gibi teknikler kullanılması modellerin performansını olumlu yönde etkileyecektir.

Alakalı değişkenlerin ve özneliliklerin tespiti için ideal çizge yapısının oluşturulmasında bu çalışma kapsamında gerçekleştirilen literatür taramasında da görüldüğü gibi genetik algoritma, tepe tırmanma (hill climbing), karınca kolonisi (ant colony), gri kurt (gray wolf) gibi sezgisel optimizasyon algoritmalarının kullanımı ile kestirim başarımının arttırılabileceği öngörülmektedir. Bunun yanında, çizge tabanlı derin öğrenme modeli (GNN - Graph Neural Network), ham CAN Bus verisi ile eğitilebilir ve hem her bir otobüs hem de bütün otobüs filosunu ele alan entegre bir bakım konsepti çalışılabilir. Bu yaklaşıma ek olarak, zaman içinde değişen veri setleri için dinamik çizge analizleri ve temporal graph network'ler kullanılabilir. Farklı seviyelerdeki ilişkilerin keşfi için hiyerarşik çizge yapıları da dikkate alınabilir.

Sistemlerdeki beklenmedik durumları tespit etmek için istatistiksel analizlerin yanında son yıllarda birçok çalışmada yer alan Isolation Forest, One-Class SVM, autoencoder, GAN gibi algoritmalar denenebilir.

Denetimli kestirimci bakım modellerinin yerine Q-learning, Deep Q-Network gibi pekiştirmeli öğrenme algoritmaları entegre edilebilir. Bu sayede bakım stratejileri optimize edilebilir. Bu değişikliğe ek olarak, farklı otobüs filoları arasında transfer öğrenme yöntemleri kullanılarak modellerin genelleştirilebilirliği artırılabilir.

Bu çalışmanın gelecek araştırmalara ilham vermesi ve kestirimci bakım alanında daha etkili ve verimli çözümlerin geliştirilmesine yardımcı olması umulmaktadır. Özellikle çizge tabanlı yöntemlerin ve derin öğrenme tekniklerinin daha fazla kullanılmasının bu alanda önemli ilerlemelere yol açabileceği, araç bakım süreçlerinin daha verimli ve etkili bir şekilde yönetilmesine olanak sağlayabileceği öngörülmektedir. Ayrıca, farklı veri türlerinin entegrasyonu (multimodal veri entegrasyonu) ve dağıtık sistemlerde çizge algoritmalarının çalıştırılması gibi konular da gelecekteki araştırmalar için potansiyel alanlar olarak görülmektedir.

KAYNAKLAR

- [1] **TÜBİTAK**, «TÜBİTAK BİLGEM,» TÜBİTAK YZE, 2024. [Çevrimiçi]. Available: <https://bilgem.tubitak.gov.tr/ye-akilli-uretim-sistemleri-alaninda-kabul-edilen-projeler/>.
- [2] **R. K. Mobley**, An Introduction to Predictive Maintenance, Butterworth-Heinemann (Elsevier Science), 2002.
- [3] **H. Raposo, J. T. Farinha, L. Ferreira ve D. Galar**, «An integrated econometric model for bus replacement and determination of reserve fleet size based on predictive maintenance.,» *Eksploatacja i Niezawodnosc – Maintenance and Reliability*, pp. 358-368, 2017.
- [4] **L. Ambriško ve K. Teplická**, «Proactive Maintenance as a Tool of Optimization for Vehicle Fleets, in Terms of Economic and Technical Benefits,» *Acta Polytechnica Hungarica* , pp. 235-249, 2021.
- [5] **E. F. Villar**, Machine Learning Maintenance Costs Prediction Model for Heavy-duty Alternative Fuel and Diesel Vehicles, West Virginia University, 2024.
- [6] **A. Theissler, J. Pérez-Velázquez, M. Kettelgerdes ve G. Elger**, «Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry,» *Reliability Engineering & System Safety_Elsevier*, November 2021.
- [7] **S. Cinieri**, Predictive maintenance of industrial vehicles based on supervised machine learning techniques, Torino: Politecnico Di Torino , 2019.
- [8] **M. S. Hossan ve B. Chowdhury**, «Data-Driven Fault Location Scheme for Advanced Distribution Management Systems,» *IEEE*, 2019.
- [9] **S. Bradbury ve B. Carpizo**, «Digitally enabled reliability: Beyond,» McKinsey&Company, 2018.
- [10] **A. D. Scaife**, «Improve predictive maintenance through the application of artificial intelligence: A systematic review,» *Results in Engineering*, cilt Volume 21, 2024.
- [11] **V. Klathae**, «The Predictable Maintenance 4.0 by Applying Digital Technology: A Case Study of Heavy Construction Machinery,» *Review of Integrative Business and Economics Research*, cilt Vol. 8, 2019.
- [12] **I. Errandonea ve S. Arrizabalaga**, «Digital Twin for maintenance: A literature review,» *Computers in Industry*, 2020.

- [13] **A.-Q. O. Gbadamosi**, «An Internet Of Things Enabled System For Real-Time,» University of the West of England, Bristol, 2023.
- [14] **S. Hosseini**, «A Comprehensive IoT-Enabled Predictive Maintenance Framework A Case Study of Predictive Maintenance for a Printing Machine,» *Mathematics and Computer Science*, 2021.
- [15] **A. Turnbull**, «Applications of machine learning in diagnostics and prognostics of wind turbine high speed generator failure,» University of Strathclyde, 2021.
- [16] **Y. Zhao, T. Li ve X. Zhang**, «Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future,» *Renewable and Sustainable Energy Reviews*, cilt 109, 2019.
- [17] **A. Diez-Olivan, J. D. Ser, D. Galar ve B. Sierra**, «Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0,» *Information Fusion*.
- [18] **M. Nacchia, F. Fruggiero, A. Lambiase ve K. Bruton**, «A systematic mapping of the advancing use of machine learning techniques for predictive maintenance in the manufacturing sector,» *Applied Industrial Technologies*, 12 March 2021.
- [19] **D. Saha, S. Mukherjee, A. Ankit ve A. Jadhav**, «Cognitive Diagnostics in Automotive using ML,» %1 içinde *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, India, 2024.
- [20] **V. S. Victor, A. Schmeißer, H. Leitte ve S. Gramsch**, «Machine learning-based optimization workflow of the homogeneity of spunbond nonwovens with human validation,» *ArXiv*, 15 April 2024.
- [21] **Srimrudhula**, «What are the top 10 challenges in data mining, and how do you overcome them?,» medium.com, 3 June 2024. [Çevrimiçi]. Available: <https://medium.com/@srimrudhula786/what-are-the-top-10-challenges-in-data-mining-and-how-do-you-overcome-them-91754aadf168>.
- [22] **S. Martins**, «Exploring The Benefits And Limitations Of Automated Machine Learning,» Gradient Insight, 4 December 2023. [Çevrimiçi]. Available: <https://gradientinsight.com/exploring-the-benefits-and-limitations-of-automated-machine-learning/>.
- [23] **T. Lede**, «Exploratory Data Analytics: Unveiling Insights for Data Enthusiasts,» medium.com, 26 February 2024. [Çevrimiçi]. Available: <https://medium.com/@thomas.lede.21/exploratory-data-analytics-unveiling-insights-for-data-enthusiasts-3aea84ae3c86>.
- [24] **A. Analytics**, «Challenges of predictive analytics and how to overcome them,» Amplitude Analytics, [Çevrimiçi]. Available: <https://amplitude.com/explore/analytics/what-predictive-analytics>.

- [25] **European Norm**, *EN 13306:2017 Maintenance - Maintenance Terminology*, Brussel: European Committee for Standardization, 2017.
- [26] **Fiix Software**, «What is preventive maintenance», Rockwell Automation, [Çevrimiçi]. Available: <https://fiixsoftware.com/maintenance-strategies/preventative-maintenance/>. [Erişildi: 15 July 2024].
- [27] **T. Nowakowski, A. Tubis ve S. Werbińska-Wojciechowska**, «Evolution of Technical Systems Maintenance Approaches – Review and a Case Study,» %1 içinde *Intelligent Systems in Production Engineering and Maintenance* (pp.161-174), Wrocklaw, 2019.
- [28] **A. Patil, G. Soni, A. Prakash ve K. Karwasra**, «Maintenance strategy selection: a comprehensive review of current paradigms and solution approaches.,» *The International journal of quality and reliability management*, pp. vol. 39, no. 3, pp. 675-703, 2022.
- [29] **P.-R. H. Villiers**, Smart Maintenance System for Inner City, Matieland, South Africa: Stellenbosch University <https://scholar.sun.ac.za>, 2022.
- [30] **T. Nowakowski, A. Tubis ve S. Werbińska-Wojciechowska**, «Evolution of technical systems maintenance approaches—review and a case study,» *Intelligent Systems in Production Engineering and Maintenance*, January 2019.
- [31] **ISO**, ISO 11898-1 Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling, Vernier, Ceneva: ISO, 2015.
- [32] **V. Tanksale**, «Anomaly Detection for Controller Area Networks Using Long Short-Term Memory,» *IEEE Open Journal of Intelligent Traffic Systems*, 8 December 2020.
- [33] **S. S. Salunkhe, S. Pal, A. Agrawal, R. Rai, S. S. S. Mole ve B. M. Jos**, «Energy optimization for CAN bus and media controls in electric vehicles using deep learning algorithms,» *The Journal of Supercomputing* (2022) , p. 8493–8508, 10 January 2022.
- [34] **M. İfraz, A. Aktepe ve S. Ersöz**, «Demand forecasting of spare parts with regression and machine learning methods: Application in a bus fleet,» *Journal of Engineering Research*, cilt 11, 2023.
- [35] **X. Dai ve Z. Gao**, «From Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis,» *IEEE*, cilt 9, 2013.
- [36] **P. Dhal ve C. Azad**, «A comprehensive survey on feature selection in the various fields of machine learning,» *Applied Intelligence* , p. 4543–4581, 23 July 2021.

- [37] **F. Cheng, C. Zhou, X. Liu, Q. Wang, J. Qiu ve L. Zhang**, «Graph-Based Feature Selection in Classification: Structure and Node Dynamic Mechanisms,» *IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE*, pp. 1314-1328, August 2023.
- [38] **H. K. Bhuyan, M. Saikiran, M. Tripathy ve V. Ravi**, «Wide-ranging approach-based feature selection for classification,» *Multimedia Tools and Applications (2023)* , p. 23277–23304, 2023.
- [39] **H. Peng, F. Long ve C. Ding**, «Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy,» *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 2005.
- [40] **S. P. Potharaju ve M. Sreedevi**, «Correlation Coefficient Based Feature Selection Framework Using Graph Construction,» *Journal of Science*, pp. 775-787, 16 April 2018.
- [41] **Z. Zhang**, Feature Selection from Higher Order Correlations, York: University of York, 2012.
- [42] **D. T. Schroeder, K. Styp-Rekowski, F. Schmidt, A. Acker ve O. Kao**, «Graph-based Feature Selection Filter Utilizing Maximal Cliques,» %1 içinde *Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Milan, 2019.
- [43] **A. K. Das, S. Goswami, A. Chakrabarti ve B. Chakraborty**, «A new hybrid feature selection approach using feature association map for supervised and unsupervised classification,» *Expert Systems With Applications*, pp. 81-94, 1 December 2017.
- [44] **S. Goswami, A. K. Das, P. Guha, A. Tarafdar, S. Chakraborty, A. Chakrabart ve B. Chakraborty**, «An approach of feature selection using graph-theoretic heuristic and hill climbing».
- [45] **L. W. Ye Li, Z. Jia, H. Wang ve L. Xu**, «Online Fault Diagnosis of Multi-function Vehicle Bus Based on Decision Tree,» *International Conference on Electrical and Information Technologies for Rail*, pp. 36-44, 2021.
- [46] **D. Markudovaa, S. Mishraa, L. Caglieroa, L. Vassioa, M. Melliaa, E. Baralis, L. Salvatori ve R. Loti**, «Preventive maintenance for heterogeneous industrial vehicles with incomplete usage data,» *Computers in Industry*, p. <https://doi.org/10.1016/j.compind.2021.103468>, September 2021.
- [47] **K. S. Kiangala ve Z. Wang**, «Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts,» *International Journal of Advanced Manufacturing Technology*, pp. 3251 - 3271, 1 August 2018.

- [48] **Z. M. Çınar, A. A. Nuhu, Q. Zeeshan, O. Korhan, M. Asmael ve B. Safaei**, «Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0,» *Sustainability*, 5 October 2020.
- [49] **T. S. Kim ve S. Y. Sohn**, «Multitask learning for health condition identification and remaining useful life prediction: deep convolutional neural network approach,» *Journal of Intelligent Manufacturing*, pp. 2169 - 2179, December 2021.
- [50] **M. Züfle, J. Agne, J. Grohmann, I. Dörtoluk ve S. Kounev**, «A Predictive Maintenance Methodology: Predicting the Time-to-Failure of Machines in Industry 4.0,» %1 içinde *IEEE 19th International Conference on Industrial Informatics (INDIN)*, Palma de Mallorca, Spain, 2021.
- [51] **T. Zonta, C. A. Costa, R. R. Righi, M. J. Lima, E. S. Trindade ve G. P. Li**, «Predictive maintenance in the Industry 4.0: A systematic literature review,» *Computers & Industrial Engineering*, 2020.
- [52] **C. Franciosi, V. Di Pasquale, R. Iannone ve S. Miranda**, «Multi-stakeholder perspectives on indicators for sustainable maintenance performance in production contexts: An exploratory study.,» *Journal of Quality in Maintenance Engineering*, 27 April 2021.
- [53] **S. B. Kotsiantis**, «Feature selection for machine learning classification problems: a recent overview,» *Springer*, 2011.
- [54] **B. Venkatesh ve J. Anuradha**, «A Review of Feature Selection and Its Methods,» *sciendo*, cilt 19, 2019.
- [55] **E. Tuv ve A. Borisov**, «Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination,» *Journal of Machine Learning Research*, 2009.
- [56] **M. Z. Ali, A. Abdullah, A. M. Zaki, F. H. Rizk, M. M.Eid ve E. M. El-Kenway** «Advances and Challenges in Feature Selection Methods: A Comprehensive Review,» *Journal of Artificial Intelligence and Metaheuristics*, pp. 67-77, 2018 January 2024.
- [57] **M. E. J. Newman ve M. Girvan**, «Finding and evaluating community structure in networks,» *PHYSICAL REVIEW E*, 26 February 2004.
- [58] **M. Rosvall ve C. T. Bergstrom**, «Maps of random walks on complex networks reveal community structure,» *PubMed*, pp. 1118-1123, 29 January 2008.
- [59] **A. Clauset, M. E. J. Newman ve C. Moore**, «Finding community structure in very large networks,» *Physical Review E*, 9 August 2004.
- [60] **V. D. Blondel, J.-L. Guillaume, R. Lambiotte ve E. Lefebvre**, «Fast unfolding of communities in large networks,» *Journal of Statistical Mechanics: Theory and Experiment*, 9 October 2008.

- [61] **V. Traag, L. Waltman ve N. V. Eck**, «From Louvain to Leiden: guaranteeing well-connected communities,» *Scientific Reports* , 26 March 2019.
- [62] **L. Breiman**, «Random Forests,» *Machine Learning*, pp. 5-32, 2001.
- [63] **O. A. Montesinos-López, A. Montesinos ve J. Crossa**, «Random Forest for Genomic Prediction,» *Multivariate Statistical Machine Learning Methods for Genomic Prediction* , pp. 633-681, January 2022.
- [64] **IBM**, «What is random forest,» IBM, [Çevrimiçi]. Available: <https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,Decision%20trees>. [Erişildi: 20 June 2024].
- [65] **T. Chen ve C. Guestrin**, «XGBoost: A Scalable Tree Boosting System,» *Knowledge Discovery and Data Mining*, 9 March 2016.
- [66] **P. Kowalek, H. Loch-Olszewska ve J. Szwabiński**, «Classification of diffusion modes in single-particle tracking data: Feature-based versus deep-learning approach,» *Physical Review E*, September 2019.
- [67] **J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua ve A. Lopez**, «A comprehensive survey on support vector machine classification: Applications, challenges and trends,» *Neurocomputing*, pp. 189-215, 30 September 2020.
- [68] **V. Barnett ve T. Lewis**, «TEST STATISTICS,» %1 içinde *Outliers in Statistical Data*, 1994, pp. 38-47.
- [69] **T. L. Vic Barnett**, «Outliers In More Structured Situations,» %1 içinde *Outliers in Statistical Data*, 1978.
- [70] **S. Moritz ve T. Bartz-Beielstein**, «imputeTS: Time Series Missing Value,» *The R Journal*, cilt 9/1, 2027.
- [71] **M. Züfle ve J. Agne**, «A Predictive Maintenance Methodology: Predicting the Time-to-Failure of Machines in Industry 4.0,» *IEEE*, 2021.
- [72] **N. V. Chawla ve K. W. Bowyer**, «SMOTE: Synthetic Minority Over-sampling Technique,» *Journal of Artificial Intelligence Research*.
- [73] **M. T. Ribeiro, S. Singh ve C. Guestrin**, «"Why Should I Trust You?": Explaining the Predictions of Any Classifier,» *ACM*, 2016.