KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

NESNELERİN İNTERNETİ TEMELLİ VERİ KÜMESİNDE SENTETİK AZINLIK AŞIRI ÖRNEKLEME TEKNİĞİ İLE GÜÇLENDİRİLMİŞ MAKİNE ÖĞRENİMİ YAKLAŞIMLARININ PERFORMANS ANALİZİ

ELA KALİFATİ

KOCAELİ 2025

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ

ANABİLİM DALI

YÜKSEK LİSANS TEZİ

NESNELERİN İNTERNETİ TEMELLİ VERİ KÜMESİNDE SENTETİK AZINLIK AŞIRI ÖRNEKLEME TEKNİĞİ İLE GÜÇLENDİRİLMİŞ MAKİNE ÖĞRENİMİ YAKLAŞIMLARININ PERFORMANS ANALİZİ

ELA KALİFATİ

Prof. Dr. Kerem KÜÇÜK
Danışman, Kocaeli Üniv. .....

Doç. Dr. Mehmet Zeki KONYAR
Jüri Üyesi, Kocaeli Üniv. .....

Doç. Dr. Ali Can KARACA
Jüri Üyesi, Yıldız Teknik Üniv. .....

Tezin Savunulduğu Tarih: 04.07.2025

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez/proje çalışmasında,

- Bu tezin/projenin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamasında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmamın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin/projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez/proje çalışması olarak sunmadığımı,

beyan ederim.

☐ Bu tez/proje çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

☑ Bu tez/proje çalışması kapsamında üretilen veri ve bilgiler Kocaeli Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından FBA2024-3752 no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

.....

Ela KALİFATİ

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin/projemin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin/projemin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanımı bana ait olacaktır.
Tezin/projenin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin/projenin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.
Yükseköğretim kurulu tarafından yayınlanan ***"Lisanüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"*** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

☐ Enstitü yönetim kurulu kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.
☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.
☑ Tezim/projem ile ilgili gizlilik kararı verilmemiştir.

.....

Ela KALİFATİ

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, Nesnelerin İnterneti (IoT) cihazlarından elde edilen veriler üzerinde gerçekleştirilen makine öğrenimi tabanlı analiz ve sınıflandırma yöntemleri ele alınmıştır. Çalışma, veri dengesizliği problemini ele almak ve makine öğrenimi modellerinin doğruluğunu artırmak amacıyla geliştirilmiştir. Elde edilen sonuçlar, veri bilimi ve IoT alanındaki çalışmalara katkı sağlamayı hedeflemektedir.

Bu yüksek lisans tezini hazırlama sürecinde, karşılaştığım her türlü zorlukta yanımda olan ve desteğini hiçbir zaman esirgemeyen danışmanım Prof. Dr. Kerem Küçük'e en içten teşekkürlerimi sunarım. Akademik bilgi birikimiyle, rehberliğiyle ve motive edici yaklaşımıyla bu çalışmanın şekillenmesine önemli katkılarda bulunmuştur.

Bu tez çalışması, Kocaeli Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından FBA2024-3752 numaralı proje kapsamında desteklenmiştir. Sağladıkları değerli katkılar için teşekkür ederim. Ayrıca, bu tez çalışması Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından 124E307 numaralı proje kapsamında desteklenmiştir. Sağlanan destekler için TÜBİTAK'a teşekkür ederim.

Bu uzun ve zorlu süreçte her zaman yanımda olan, bana güç ve cesaret veren aileme sonsuz teşekkür ederim. Onların sevgisi, sabrı ve desteği olmasaydı, bu başarıya ulaşmam mümkün olmazdı. Ayrıca, bu tez çalışmasının yürütülmesinde katkıda bulunan ve değerli fikirleriyle destek sağlayan tüm hocalarıma ve çalışma arkadaşlarıma teşekkür ederim. Tezimin hazırlanmasında ihtiyaç duyduğum akademik altyapıyı sağlayan üniversiteme ve bu süreçte yanımda olan tüm dostlarıma da şükranlarımı sunarım. Bu çalışmanın, ilgili alanda yeni ufuklar açmasını ve gelecekte yapılacak çalışmalara ilham kaynağı olmasını temenni ederim.

Haziran - 2025 Ela KALİFATİ

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI ii
ÖNSÖZ VE TEŞEKKÜR iii
İÇİNDEKİLER iv
ŞEKİLLER DİZİNİ vi
TABLOLAR DİZİNİ viii
SİMGELER VE KISALTMALAR DİZİNİ ix
ÖZET x
ABSTRACT xi
1. GİRİŞ 1
2. LİTERATÜR İNCELEMESİ 3
3. KULLANILAN MATERYALLER 6
3.1. Veri Kümesi (TON_IoT) 6
3.1.1. TON_IoT'nin Genel Yapısı 6
3.1.2. TON_IoT'nin Kullanılan IoT Cihazlar 6
3.1.2.1. Fridge (Akıllı Buzdolabı) 6
3.1.2.2. Garage Door (Garaj Kapısı) 7
3.1.2.3. GPS Tracker (Konum İzleyici) 8
3.1.2.4. Motion Light (Hareket Sensörlü Işık) 9
3.1.2.5. Thermostat (Termostat) 9
3.1.2.6. Weather (Hava Durumu Sensörü) 10
3.1.2.7. Modbus 11
3.1.3. TON_IoT Veri Kümesi Dengesizliği 12
3.2. Makine Öğrenimi (ML) Yöntemleri 12
3.2.1. Support Vector Machine (SVM) 12
3.2.2. K-Nearest Neighbor (En Yakın K-Komşu - KNN) 14
3.2.3. Lojistik Regresyon (LR) Algoritması 15
3.2.4. Çok Katmanlı Algılayıcı (MLP) Algoritması 16
3.3. SMOTE Tekniği 18
3.4. Değerlendirme Teknikleri 19
3.4.1. Karmaşıklık Matrisi (Confusion Matrix) 19
3.4.2. Sınıflandırma Ölçütleri (Classification Measures) 20
4. METODOLOJİ VE SONUÇLARIN DEĞERLENDİRMESİ 22
4.1. Ton_Iot Veri Kümesi 22
4.2. Veri Ön İşleme (Data Preprocessing) 23
4.3. SMOTE Uygulaması 24
4.4. Makine Öğrenimi Algoritmalarının Uygulanması 25
5. IOT CİAZLARINA GÖRE SALDIRI TESPİTİ PERFORMANS ANALİZİ 26
5.1. Fridge (Akıllı Buzdolabı Cihazı) 26
5.1.1. KNN Algoritması 27
5.1.2. SVM Algoritması 28
5.1.3. Lojistik Regresyon (LR) Algoritması 30
5.1.4. MLP Algoritması 31

5.2. Garage Door (Akıllı Garaj Kapısı Cihazı) ... 33
5.2.1. KNN Algoritması ... 33
5.2.2. SVM Algoritması ... 35
5.2.3. Lojistik Regresyon (LR) Algoritması ... 36
5.2.4. MLP Algoritması ... 37
5.3. GPS Tracker (Konum Takip Cihazı) ... 38
5.3.1. KNN Algoritması ... 38
5.3.2. SVM Algoritması ... 39
5.3.3. Lojistik Regresyon (LR) Algoritması ... 40
5.3.4. MLP Algoritması ... 41
5.4. Modbus (Endüstriyel Otomasyon Cihazı) ... 42
5.4.1. KNN Algoritması ... 43
5.4.2. SVM Algoritması ... 44
5.4.3. LR Algoritması ... 45
5.4.4. MLP Algoritması ... 46
5.5. Motion Light (Hareket Işığı Cihazı) ... 48
5.5.1. KNN Algoritması ... 48
5.5.2. SVM Algoritması ... 50
5.5.3. LR Algoritması ... 51
5.5.4. MLP Algoritması ... 52
5.6. Thermostat (Akıllı Termostat Cihazı) ... 54
5.6.1. KNN Algoritması ... 55
5.6.2. SVM Algoritması ... 57
5.6.3. LR Algoritması ... 58
5.6.4. MLP Algoritması ... 59
5.7. Weather (Hava Durumu Sensör Cihazı) ... 61
5.7.1. KNN Algoritması ... 62
5.7.2. SVM Algoritması ... 63
5.7.3. Lojistik Regresyon (LR) Algoritması ... 64
5.7.4. MLP Algoritması ... 65
6. GENEL DEĞERLENDİRME VE SONUÇLAR ... 67
6.1. KNN Algoritması Performansı: ... 67
6.2. SVM Algoritması Performansı: ... 68
6.3. Logistic Regression Algoritması Performansı: ... 68
6.4. MLP Algoritması Performansı: ... 69
6.5. Gelecek Çalışmalar ... 70
KAYNAKLAR ... 71
KİŞİSEL YAYIN VE ESERLER ... 74
ÖZGEÇMİŞ ... 75

ŞEKİLLER DİZİNİ

Şekil 3.1. KNN çalışma prensibini 15
Şekil 3.2. SMOTE yöntemi ile sentetik örneklerin oluşturulması..... 19
Şekil 3.3. Karmaşıklık matrisi örneğini 20
Şekil 4.1. Ton IoT veri setinde SMOTE ile makine öğrenimi ile performans analizi süreci akış diyagramı 22
Şekil 4.2. IoT Modbus SMOTE Öncesi ve Sonrası Kategori Dağılımı 24
Şekil 5.1. IoT Akıllı Buzdolab SMOTE Öncesi ve Sonrası Kategori Dağılımı 26
Şekil 5.2. KNN - Fridge için Confusion Matrix 27
Şekil 5.3. SVM - Fridge için Confusion Matrix 28
Şekil 5.4. LR - Fridge için Confusion Matrix 30
Şekil 5.5. MLP - Fridge için Confusion Matrix..... 31
Şekil 5.6. IoT Akıllı Garaj Kapısı SMOTE Öncesi ve Sonrası Kategori Dağılımı 33
Şekil 5.7. KNN - Garage Door için Confusion Matrix 34
Şekil 5.8. SVM - Garage Door için Confusion Matrix 35
Şekil 5.9. LR - Garage Door için Confusion Matrix 36
Şekil 5.10. MLP - Garage Door için Confusion Matrix..... 37
Şekil 5.11. IoT Konum Takip Cihazı SMOTE Öncesi ve Sonrası Kategori Dağılımı 38
Şekil 5.12. KNN - GPS Tracker için Confusion Matrix..... 39
Şekil 5.13. SVM - GPS Tracker için Confusion Matrix..... 40
Şekil 5.14. LR - GPS Tracker için Confusion Matrix 40
Şekil 5.15. MLP - GPS Tracker için Confusion Matrix 41
Şekil 5.16. IoT Modbus SMOTE Öncesi ve Sonrası Kategori Dağılımı 43
Şekil 5.17. KNN - Modbus için SMOTE Öncesi ve Sonrası Kategori Dağılımı 43
Şekil 5.18. KSVM - Modbus için SMOTE Öncesi ve Sonrası Kategori Dağılımı 44
Şekil 5.19. LR - Modbus için SMOTE Öncesi ve Sonrası Kategori Dağılımı 45
Şekil 5.20. MLP - Modbus için SMOTE Öncesi ve Sonrası Kategori Dağılımı..... 47
Şekil 5.21. IoT Hareket Işığı SMOTE Öncesi ve Sonrası Kategori Dağılımı 48
Şekil 5.22. KNN - Motion Light için Confusion Matrix..... 49
Şekil 5.23. SVM - Motion Light için Confusion Matrix..... 50
Şekil 5.24. LR - Motion Light için Confusion Matrix 51
Şekil 5.25. MLP - Motion Light için Confusion Matrix 53
Şekil 5.26. IoT Akıllı Termostat Cihazı SMOTE Öncesi ve Sonrası Kategori Dağılımı 55
Şekil 5.27. KNN - Thermostat için Confusion Matrix 55
Şekil 5.28. SVM - Thermostat için Confusion Matrix 57
Şekil 5.29. LR - Thermostat için Confusion Matrix..... 58
Şekil 5.30. MLP - Thermostat için Confusion Matrix..... 59
Şekil 5.31. IoT Hava Durumu Sensör Cihazı SMOTE Öncesi ve Sonrası Kategori Dağılımı 61
Şekil 5.32. KNN - Weather için Confusion Matrix 62

Şekil 5.33. SVM - Weather için Confusion Matrix 63
Şekil 5.34. LR - Weather için Confusion Matrix 64
Şekil 5.35. MLP - Weather için Confusion Matrix 65

TABLOLAR DİZİNİ

Tablo 3.1. Fridge verilerinin özellikleri ve türleri 7
Tablo 3.2. Saldırı türleri ve dağılımı 7
Tablo 3.3. Garage Door verilerinin özellikleri ve türleri 7
Tablo 3.4. Saldırı türleri ve dağılımı 8
Tablo 3.5. GPS Tracker verilerinin özellikleri ve türleri 8
Tablo 3.6. Saldırı türleri ve dağılımı 8
Tablo 3.7. Motion Light verilerinin özellikleri ve türleri 9
Tablo 3.8. Saldırı türleri ve dağılımı 9
Tablo 3.9. Thermostat verilerinin özellikleri ve türleri 10
Tablo 3.10. Saldırı türleri ve dağılımı 10
Tablo 3.11. Weather verilerinin özellikleri ve türleri 10
Tablo 3.12. Saldırı türleri ve dağılımı 11
Tablo 3.13. Modbus verilerinin özellikleri ve türleri 11
Tablo 3.14. Saldırı türleri ve dağılımı 11
Tablo 4.1. IoT CIHAZLARINA GÖRE SALDIRI TÜRLERI DAĞILIMI 23
Tablo 4.2. IoT MODBUS SALDIRI KATEGORILERI VE SAYISAL DEĞERLERI 24
Tablo 5.1. IoT AKILLI BUZDOLABI SALDIRI KATEGORILERI VE SAYISAL DEĞERLERI TABLOSU 26
Tablo 5.2. IoT GARAJ KAPISI SALDIRI KATEGORILERI VE SAYISAL DEĞERLERI TABLOSU 33
Tablo 5.3. IoT KONUM TAKIP CIHAZI SALDIRI KATEGORILERI VE SAYISAL DEĞERLERI TABLOSU 38
Tablo 5.4. IoT MODBUS SALDIRI KATEGORILERI VE SAYISAL DEĞERLERI 42
Tablo 5.5. IoT Hareket Sensörlü Isık saldırı kategorileri ve sayısal değerleri tablosu 48
Tablo 5.6. IoT Termostat saldırı kategorileri ve sayısal değerleri tablosu 54
Tablo 5.7. IoT Hava Durumu Sensörü saldırı kategorileri ve sayısal değerleri tablosu 61
Tablo 6.1. KNN test kümesi performansı (SMOTE öncesi ve sonrası) 67
Tablo 6.2. SVM test kümesi performansı (SMOTE öncesi ve sonrası) 68
Tablo 6.3. Logistic Regression test kümesi performansı (SMOTE öncesi ve sonrası) 68
Tablo 6.4. MLP test kümesi performansı (SMOTE öncesi ve sonrası) 69

SİMGELER VE KISALTMALAR DİZİNİ

b	:	Bias terimi
e	:	Doğal logaritma tabanı
w	:	Girdi için katsayı (ağırlık)
x	:	Girdi değeri
$\hat{y}$	:	Tahmin edilen çıktı (bir gözlemin "pozitif sınıfa" ait olma olasılığı)

Kısaltmalar

AUC-ROC	:	Area Under the ROC Curve (ROC Eğrisi Altındaki Alan)
DDS	:	Data Distribution Service (Veri Dağıtım Hizmeti)
DRNN	:	Deep Recurrent Neural Network (Derin Tekrarlayan Sinir Ağı)
GBM	:	Gradient Boosting Machine (Gradyan Artırma Makinesi)
IDS	:	Intrusion Detection System (İzinsiz Giriş Tespit Sistemi)
IoT	:	Internet of Things (Nesnelerin İnterneti)
KNN	:	k-Nearest Neighbors (En Yakın k Komşu)
LR	:	Logistic Regression (Lojistik Regresyon)
ML	:	Machine Learning (Makine Öğrenimi)
MLP	:	Multi-Layer Perceptron (Çok Katmanlı Algılayıcı)
SMOTE	:	Synthetic Minority Over-sampling Technique (Sentetik Azınlık Aşırı Örnekleme Tekniği)
SVM	:	Support Vector Machines (Destek Vektör Makineleri)

NESNELERİN İNTERNETİ TEMELLİ VERİ KÜMESİNDE SENTETİK AZINLIK AŞIRI ÖRNEKLEME TEKNİĞİ İLE GÜÇLENDİRİLMİŞ MAKİNE ÖĞRENİMİ YAKLAŞIMLARININ PERFORMANS ANALİZİ

ÖZET

Nesnelerin İnterneti (IoT) cihazlarından toplanan veriler genellikle dengesiz sınıf dağılımı sergilemekte ve bu durum, makine öğrenimi tabanlı saldırı tespit sistemlerinin performansını önemli ölçüde düşürmektedir. Bu çalışmada, sınıf dengesizliği problemini ele almak amacıyla Sentetik Azınlık Örnekleme Tekniği (SMOTE) yöntemi kullanılmış ve dört farklı makine öğrenimi algoritmasının (K-En Yakın Komşu, Destek Vektör Makineleri, Lojistik Regresyon ve Çok Katmanlı Algılayıcı) performansı kapsamlı bir şekilde değerlendirilmiştir. Deneylerde kullanılan ToN_IoT veri seti, yedi farklı IoT cihazından alınan verileri içermekte olup, veriler eğitim, doğrulama ve test olmak üzere üç ayrı küme halinde bölünmüştür. SMOTE uygulandıktan sonra, özellikle Hareket Işığı, GPS Takip ve Garaj Kapısı gibi azınlık sınıfların baskın olduğu cihazlarda doğruluk ve F1-skor değerlerinde önemli iyileşmeler gözlemlenmiştir. Örneğin, Çok Katmanlı Algılayıcı (MLP) algoritması, Garaj Kapısı cihazında doğruluğu %55'ten %97'ye yükseltirken, Lojistik Regresyon (LR) modeli test veri kümesinde %100 doğruluk oranına ulaşmıştır. Bu sonuçlar, SMOTE yönteminin sınıf dengesizliği sorununu etkili bir şekilde azalttığını ve makine öğrenimi modellerinin sınıflandırma performansını belirgin şekilde artırdığını göstermektedir. Çalışmanın bulguları, özellikle IoT tabanlı güvenlik sistemlerinde, veri setindeki azınlık sınıfların tespit yeteneğinin geliştirilmesi açısından önemli katkılar sunmaktadır. Bu araştırma, dengesiz veri setleri üzerinde SMOTE gibi yeniden örnekleme tekniklerinin kullanımının, saldırı tespit sistemlerinin başarısını artırmada etkili bir yöntem olduğunu kanıtlamaktadır. Gelecek çalışmalarda, farklı örnekleme stratejilerinin kombinasyonlarının veya hibrit modellerin performansa etkisi incelenebilir.

Anahtar Kelimeler: K-en Yakın Komşular (KNN), Nesnelerin İnterneti (IoT), Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE), Destek Vektör Makineleri (SVM), Veri Dengesizliği.

PERFORMANCE ANALYSIS OF MACHINE LEARNING APPROACHES ENHANCED WITH SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE ON AN INTERNET OF THINGS-BASED DATASET

ABSTRACT

Data collected from Internet of Things (IoT) devices frequently exhibits significant class imbalance, substantially degrading the performance of machine learning-based intrusion detection systems. This study investigates the effectiveness of Synthetic Minority Over-sampling Technique (SMOTE) in mitigating class imbalance and evaluates the performance of four machine learning algorithms: K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Logistic Regression (LR), and Multilayer Perceptron (MLP). The experimental framework employs the ToN_IoT dataset, comprising network traffic data from seven distinct IoT devices. The dataset is partitioned into training, validation, and test sets to ensure rigorous evaluation. Implementation of SMOTE yields notable improvements in both accuracy and F1-scores, particularly for minority-class devices including Motion Light, GPS Tracker, and Garage Door. The MLP algorithm demonstrates particularly strong performance, improving detection accuracy for Garage Door devices from 55% to 97%, while the Logistic Regression model achieves perfect classification (100% accuracy) on the test set. These results demonstrate SMOTE's efficacy in addressing class imbalance and significantly enhancing classification performance in IoT security applications. The findings highlight the critical importance of data balancing techniques in security contexts where accurate detection of minority-class attack patterns is essential. This research empirically validates SMOTE as an effective preprocessing technique for improving intrusion detection system performance on imbalanced IoT datasets. Future work could investigate hybrid sampling approaches or ensemble methods to further optimize detection capabilities for rare attack types. The study contributes to the growing body of research on machine learning applications in IoT security, particularly in addressing the fundamental challenge of class imbalance.

Keywords: K-Nearest Neighbors (KNN), Internet of Things (IoT), Synthetic Minority Over-sampling Technique (SMOTE), Support Vector Machines (SVM), Data Imbalance.

1. GİRİŞ

Son yıllarda hızla gelişen teknoloji, küresel çapta geniş kapsamlı etkileri olan bir dijital dönüşümü beraberinde getirmiştir. Bu dönüşümün merkezinde, akıllı ev aletlerinden otonom araçlara, endüstriyel ekipmanlardan kentsel altyapıya kadar geniş bir yelpazede yer alan cihazlar arasında veri toplama, paylaşma ve analiz etme olanağı sağlayan Nesnelerin İnterneti (IoT) bulunmaktadır. Bu teknolojilerin kusursuz bir şekilde entegre edilmesiyle IoT, günlük yaşamı yeniden şekillendirmekte ve endüstrilerde devrim yaratmaktadır (Ait Mouha, 2021). Ancak, geniş potansiyeline ve çeşitli uygulama alanlarına rağmen Nesnelerin İnterneti (IoT), önemli zorluklarla karşı karşıyadır. Cihaz uyumluluğu sorunları, siber güvenlik açıkları ve veri yönetiminin karmaşıklığı gibi temel engeller, IoT'nin yaygın şekilde benimsenmesini ve verimli kullanımını sınırlandırmaktadır (Gokhale ve diğ., 2018), (Mazhar ve diğ., 2023). Bu zorluklar arasında, IoT cihazları tarafından toplanan verilerin dengesiz dağılımı önemli bir sorun olarak öne çıkmaktadır. Bu dengesizlik, saldırı tespit sistemlerinin etkinliğini büyük ölçüde olumsuz etkilemekte ve ciddi siber güvenlik riskleri oluşturmaktadır (Farea ve diğ., 2024), (Eren ve Küçük, 2023). Zorlukların üstesinden gelmek için Makine Öğrenmesi (ML) teknikleri, IoT verilerini analiz etme ve sınıflandırmada kritik bir rol oynamaktadır. Özellikle büyük miktarda veriyi işleme ve dengesiz sınıf dağılımlarını yönetme gibi karmaşık görevlerde, IoT sistemlerinin performansını artırmak için etkili çözümler sunmaktadır (Hussain ve diğ., 2020). Ancak, veri kümelerindeki dengesiz sınıf dağılımı, makine öğrenmesi modellerinin az temsil edilen sınıfları doğru şekilde tanımasını engelleyerek model performansını önemli ölçüde etkilemektedir. Bu durum, özellikle saldırı tespiti gibi kritik uygulamalarda yanlış sınıflandırmalara ve güvenlik açıklarına yol açabilmektedir (Adi ve diğ., 2020). Bu sorunun üstesinden gelmek için, sınıf dengesizliğiyle başa çıkma tekniklerinin uygulanması kritik önem taşımaktadır. Bu yöntemler, verideki az temsil edilen sınıfların temsilini artırarak model performansını iyileştirmektedir (Mohindru ve diğ., 2021), (Eid ve diğ., 2024).Bu hedefe ulaşmak için, az temsil edilen sınıflardaki dengesizliği azaltmak amacıyla Sentetik Azınlık Örnekleme Tekniği (SMOTE) gibi veri dengeleme yöntemleri yaygın olarak kullanılmaktadır (Pradipta ve diğ., 2021). SMOTE tekniği, az temsil edilen sınıflara ait örnekler ile bunların en yakın komşuları arasında enterpolasyon yaparak yeni sentetik veri noktaları

üretmekte ve böylece bu sınıfların temsilini artırmaktadır. Bu tekniğin amacı, veri kümelerindeki sınıf dengesizliğini azaltmak ve makine öğrenmesi modellerinin performansını iyileştirmektir (Soe ve diğ., 2019). Bu çalışma, özellikle IoT veri kümelerindeki dengesizlik problemini çözmek için SMOTE'nin etkinliğini değerlendirerek, makine öğrenimi modellerinin sınıflandırma başarısını artırmayı hedeflemektedir.

2. LİTERATÜR İNCELEMESİ

Mohammed ve arkadaşları çalışmalarında (Mohammed ve diğ., 2020), dengesiz bir tıbbi veri setinin sınıflandırma performansını artırmak amacıyla SMOTE yöntemini uygulamışlardır. Bu çalışmada, sınıflandırma performansını etkileyebilecek dengesiz verilerin yeniden örneklenmesi sayesinde model doğruluğunda önemli bir iyileşme sağlandığı gözlemlenmiştir. Sonuçlar, SMOTE'un yeniden örnekleme oranında önemli bir iyileşme sağladığını göstermektedir ve sınıflandırma algoritmalarında daha dengeli sonuçlar elde edilmesini sağlamaktadır .

Blagus ve Lusa (Blagus ve Lusa, 2013), SMOTE yönteminin yüksek boyutlu ve dengesiz veri kümelerindeki etkisini incelemiştir. Çalışmada, gen ekspresyonu gibi yüksek boyutlu veri kümeleri üzerinde SMOTE ve diğer yöntemlerin performansı karşılaştırılmıştır. Sonuçlar, SMOTE'nin düşük boyutlu veri kümelerinde etkili olduğunu, ancak yüksek boyutlu veri kümelerinde sınırlı fayda sağladığını göstermiştir. Özellikle KNN gibi sınıflandırıcılarla, değişken seçimi yapıldıktan sonra SMOTE'nin performansı artırdığı tespit edilmiştir. Bununla birlikte, yüksek boyutlu veri kümelerinde basit undersampling yönteminin genellikle daha etkili olduğu ve sınıflandırıcı doğruluğunu iyileştirdiği vurgulanmıştır.

Dablain ve arkadaşları (Dablain ve diğ., 2022), dengesiz veri kümeleri üzerinde sınıflandırma performansını artırmak amacıyla DeepSMOTE yöntemini geliştirmiştir. Bu yöntem, SMOTE algoritmasını derin öğrenme teknikleriyle birleştirerek, yüksek kaliteli sentetik örnekler üretmekte ve sınıflandırma modellerinin başarısını artırmaktadır. Çalışmada, MNIST, Fashion-MNIST ve CIFAR-10 gibi veri kümeleri üzerinde deneyler gerçekleştirilmiş ve DeepSMOTE'nin GAN tabanlı ve piksel tabanlı yöntemlere kıyasla üstün performans gösterdiği gözlemlenmiştir. Özellikle, düşük boyutlu gömüler oluşturma yeteneği sayesinde sınıf sınırlarını stratejik olarak güçlendirmiştir. Deneysel sonuçlara göre, DeepSMOTE, sınıflandırma doğruluğunu ve dengeyi önemli ölçüde artırmış ve Bayesian Wilcoxon testiyle bu iyileşmenin istatistiksel olarak anlamlı olduğu doğrulanmıştır.

Calp ve arkadaşları (Calp ve Bütüner, 2024), IoT tabanlı ağ cihazlarına yönelik siber

saldırıları tespit etmek amacıyla KNN, RF ve ANN gibi makine öğrenimi algoritmalarını kullanarak bir model önermiştir. Bu çalışmanın sonuçlarına göre, özellikle Random Forest algoritması %99,6 doğruluk oranına ulaşarak en iyi performansı göstermiştir. Bu bulgu, RF algoritmasının IoT tabanlı sistemlerde siber saldırı tespitinde etkili bir yöntem olduğunu önermektedir.

Gad ve diğerleri (Gad ve diğ., 2021), araç içi ağlar üzerinde güvenlik çözümleri geliştirmek amacıyla bir İzinsiz Giriş Tespit Sistemi (IDS) oluşturmuşlardır. Bu sistemin geliştirilmesinde SMOTE ve Chi-kare özellik seçimi teknikleri kullanılmış ve çeşitli makine öğrenimi algoritmaları test edilmiştir. Sonuçlar, XGBoost algoritmasının diğer algoritmalarla karşılaştırıldığında üstün performans gösterdiğini ortaya koymuştur.

Ayrıca, Verma ve Ranga (Verma ve Ranga, 2020), IoT için makine öğrenimi tabanlı izinsiz giriş tespit sistemleri geliştirmiş ve RF, AdaBoost, GBM ve XGBoost gibi algoritmaların performansını değerlendirmiştir. Çalışma sonuçlarına göre, Random Forest doğruluk açısından diğer algoritmalara kıyasla daha iyi performans gösterirken, XGBoost özellikle AUC (Eğri Altındaki Alan) metriklerinde üstünlük sağlamıştır.

Popoola ve diğerleri tarafından yapılan bir çalışmada (Popoola ve diğ., 2021), IoT ağlarında botnet saldırılarını tespit etmek için SMOTE-DRNN algoritması geliştirilmiştir. Bu algoritma, eğitim setlerindeki dengesiz ağ trafiği verilerinin sınıflandırma performansını artırmayı hedeflemektedir. Sonuçlar, SMOTE-DRNN modelinin diğer makine öğrenimi ve derin öğrenme modellerine göre daha yüksek doğruluk, hassasiyet ve F1 puanı sağladığını göstermektedir. Ayrıca, modelin aşırı uyuma ve yetersiz uyuma karşı dayanıklılık gösterdiği belirtilmiştir.

Soe ve arkadaşları (Soe ve diğ., 2019), IoT ortamında DDoS saldırılarını tespit etmek amacıyla basit bir Yapay Sinir Ağı (ANN) modeli geliştirmiştir. Bu çalışmada, veri dengesizliği sorununu çözmek için SMOTE tekniği uygulanmış ve Bot-IoT veri seti kullanılmıştır. Sonuçlar, SMOTE uygulanmasının ardından modelin %100 doğruluk oranına ulaştığını göstermiştir.

Zhang ve arkadaşları (Zhang ve diğ., 2022), dengesiz veri kümelerindeki sınıflandırma

problemini çözmek için SEB-XGB adlı bir model geliştirmiştir. Bu model, azınlık sınıfını güçlendirmek için SVM-SMOTE ve çoğunluk sınıfını dengelemek için EasyEnsemble tekniklerini birleştirmiştir. Çalışmada, Bayes Optimizasyonu ile parametre ayarlamaları yapılarak SEB-XGB'nin, G-mean ve AUC metriklerinde üstün başarı sağladığı gösterilmiştir. Model, kredi kartı ve dolandırıcılık veri kümelerinde XGBoost'a kıyasla daha yüksek doğruluk oranları sunmuştur. Bu çalışma, dengesiz veri kümeleri için etkili bir sınıflandırma yöntemi önermektedir.

Doghramachi ve Ameen (Doghramachi ve Ameen, 2023), IoT ağlarındaki güvenlik açıklarını tespit etmek ve sınıflandırma doğruluğunu artırmak için XGBoost ve SMOTE tekniklerini birleştiren bir yöntem önermiştir. Çalışmada, IoTID20 veri kümesi kullanılarak, dengesiz veri kümeleri üzerinde etkili sınıflandırma sonuçları elde edilmiştir. Model, özellikle ikili ve çok sınıflı sınıflandırmalarda 0.81 doğruluk ve 0.83 geri çağırma oranı ile üstün performans göstermiştir. Ayrıca, önerilen yöntem, önceki çalışmalara kıyasla G-mean ve AUC metriklerinde iyileştirme sağlamıştır. Gelecekteki çalışmalar için dinamik IoT ağlarının simülasyonu ve saldırılara karşı daha dayanıklı sistemler geliştirilmesi önerilmektedir.

Eren ve Kerem tarafından yapılan bir çalışmada (Eren ve Küçük, 2023), IoT cihazları için kullanılan Saldırı Tespit Sistemlerinin (IDS) performansını geliştirmeyi amaçlamaktadır. Modern IoT ortamlarını temsil eden ToN-IoT veri kümesi, özellik ön işleme ve otomatik özellik türetme teknikleriyle optimize edilmiştir. Sonuçlar, özellikle karmaşık saldırı türlerinde Random Forest modeli ile %99.99 ROC-AUC başarısı elde edildiğini göstermektedir. Çalışma, IoT ağ güvenliği için etkili bir çerçeve sunarken, MITM saldırıları gibi az veri içeren senaryolarda iyileştirme gereksinimi vurgulanmıştır.

3. KULLANILAN MATERYALLER

3.1. Veri Kümesi (TON_IoT)

TON_IoT veri kümesi, IoT ve Endüstriyel IoT (IIoT) sistemlerindeki siber güvenlik uygulamalarını değerlendirmek için geliştirilmiş yeni nesil bir veri kümesidir. UNSW Canberra Siber IoT Laboratuvarı'nda gerçekçi bir ağ ortamında toplanan veri kümesi, IoT ve IIoT sensörlerinden, işletim sistemlerinden ve ağ trafiğinden elde edilen çeşitli verileri içermektedir. Çalışmamızda bu veri kümesi seçilmiştir çünkü farklı saldırı türlerini ve normal olayları bir arada sunarak makine öğrenimi algoritmalarının performansını değerlendirmek için kapsamlı bir altyapı sağlamaktadır. (Moustafa, 2021), (Alsaedi ve diğ., 2020).

3.1.1. TON_IoT'nin Genel Yapısı

TON_IoT veri seti, IoT ve IIoT ortamlarında siber güvenlik uygulamalarını test etmek ve değerlendirmek için kapsamlı bir yapı sunar. Veri seti; telemetri verileri, ağ verileri, ve işletim sistemi verileri olmak üzere üç ana bileşenden oluşur.

Bu veri çeşitliliği, gerçek dünyadaki saldırı senaryolarını ve normal operasyonel davranışları kapsayan geniş bir örneklem sunmaktadır. TON_IoT'nin en önemli özelliklerinden biri, farklı saldırı türlerini (örneğin, DDoS, ransomware, injection) ve normal trafik verilerini içermesidir. Veri seti, özellikle IoT cihazları üzerinden elde edilen telemetri verileri ile öne çıkmaktadır.

Çalışmamızda, IoT cihazlarından toplanan bu veriler üzerinde odaklanılacak ve bu cihazlardan alınan saldırı türleri sınıflandırılarak analiz edilecektir.

3.1.2. TON_IoT'nin Kullanılan IoT Cihazlar

Her bir cihazın veri yapısı, özellikleri ve içerdiği saldırı türleri aşağıda açıklanmıştır.

3.1.2.1. Fridge (Akıllı Buzdolabı)

Bu alt bölümde, akıllı buzdolabına ait IoT telemetri verileri incelenmiştir. Aşağıda, buzdolabına ait veri özellikleri ve bu verilerin türleri Tablo 3.1'de açıklanmıştır. Ayrıca,

Tablo 3.2'de bu verilere ilişkin saldırı türleri ve dağılımları detaylandırılmıştır.

Tablo 3.1. Fridge verilerinin özellikleri ve türleri

ID	Özellik	Tür
1	date	Tarih
2	time	zaman
3	fridge_temperature	Sayı
4	temp_condition	Metin
5	label	Sayı
6	type	Metin

Tablo 3.2. Saldırı türleri ve dağılımı

Satır sayısı	Type
15000	normal
5000	password
2042	xss
5000	ddos
2902	ransomware
5000	injection
5000	backdoor

3.1.2.2. Garage Door (Garaj Kapısı)

Veriler, garaj kapısının durumunu ve diğer önemli değişkenleri izlemek için toplanmıştır. Tablo 3.3'te Akıllı garaj kapısına ait telemetri verilerinde yer alan özellikler ve türleri açıklanmaktadır. Bu özellikler, cihazın çalışma durumunu izlemek ve analiz etmek için temel bilgileri içermektedir.

Tablo 3.4'te yönelik saldırı türleri ve bu türlerin veri setindeki dağılımını göstermektedir.

Tablo 3.3. Garage Door verilerinin özellikleri ve türleri

ID	Özellik	Tür
1	date	Tarih
2	time	zaman
3	door_state	Boolean
4	sphone_signal	Boolean
5	label	Sayı
6	type	Metin

Tablo 3.4. Saldırı türleri ve dağılımı

Satır sayısı	Type
5000	backdoor
5000	ddos
2902	ransomware
5000	injection
15000	normal
1156	xss
5000	password
529	scanning

3.1.2.3. GPS Tracker (Konum İzleyici)

Bu bölümde, GPS konum izleyicisine ait IoT telemetri verilerinin yapısı ve saldırı türlerine göre dağılımı ele alınmıştır. GPS konum izleyicisi, cihazın bulunduğu konumu izlemek ve bu bilgileri analiz etmek için kullanılan kritik bir IoT bileşenidir. Tablo 3.5'te bu cihazdan elde edilen verilerin özelliklerini ve Tablo 3.6'da veri setindeki saldırı türlerinin dağılımını detaylı bir şekilde göstermektedir.

Tablo 3.5. GPS Tracker verilerinin özellikleri ve türleri

ID	Özellik	Tür
1	date	Tarih
2	time	zaman
3	latitude	Sayı
4	longitude	Sayı
5	label	Sayı
6	type	Metin

Tablo 3.6. Saldırı türleri ve dağılımı

Satır sayısı	Type
15000	normal
5000	password
550	scanning
577	xss
5000	ddos
2833	ransomware
5000	injection
5000	backdoor

3.1.2.4. Motion Light (Hareket Sensörlü Işık)

Aşağıda, bu cihazdan toplanan veri özellikleri ve veri setindeki saldırı türlerinin dağılımı detaylandırılmıştır. Tablo 3.7'de hareket sensörlü ışık verilerinde kullanılan özelliklerin listesi ve türleri verilmiştir.

Bu özellikler, cihazın algıladığı hareket durumunu ve ışık kontrolüyle ilgili bilgileri içermektedir. Tablo 3.8'de farklı saldırı kategorilerindeki kayıt sayıları belirtilmiş olup, bu veriler cihazın maruz kaldığı tehditleri anlamak için temel teşkil etmektedir.

Tablo 3.7. Motion Light verilerinin özellikleri ve türleri

ID	Özellik	Tür
1	date	Tarih
2	time	zaman
3	motion_status	Sayı
4	light_status	Boolean
5	label	Sayı
6	type	Metin

Tablo 3.8. Saldırı türleri ve dağılımı

Satır sayısı	Type
15000	normal
5000	password
1775	scanning
449	xss
5000	ddos
2264	ransomware
5000	injection
5000	backdoor

3.1.2.5. Thermostat (Termostat)

Termostat cihazı, ortam sıcaklığını izlemek ve kontrol etmek amacıyla kullanılan IoT sensörlerinden biridir.

Aşağıda, Tablo 3.9 ve Tablo 3.10 bu cihazın veri yapısına ve saldırı türlerine dair detaylar verilmiştir.

Tablo 3.9. Thermostat verilerinin özellikleri ve türleri

ID	Özellik	Tür
1	date	Tarih
2	time	zaman
3	current_temperature	Sayı
4	thermostat_status	Boolean
5	label	Sayı
6	type	Metin

Tablo 3.10. Saldırı türleri ve dağılımı

Satır sayısı	Type
15000	normal
5000	password
61	scanning
449	xss
2264	ransomware
5000	injection
5000	backdoor

3.1.2.6. Weather (Hava Durumu Sensörü)

Aşağıda, bu sensörden toplanan verilere ait özellikler ve saldırı türlerinin detaylı dağılımı yer almaktadır. Tablo 3.11'de hava durumu sensöründen toplanan özelliklerin listesi ve türleri. Bu özellikler, ortam koşullarına ilişkin temel bilgileri ve sensör tarafından toplanan etiketlenmiş kayıtları içermektedir.

Tablo 3.12'de yönelik saldırı türlerinin veri setindeki dağılımı. Farklı saldırı türlerine ait kayıtların frekansları belirtilmiştir.

Tablo 3.11. Weather verilerinin özellikleri ve türleri

ID	Özellik	Tür
1	date	Tarih
2	time	zaman
3	temperature	Sayı
4	pressure	Sayı
5	humidity	Sayı
6	label	Sayı
7	type	Metin

Tablo 3.12. Saldırı türleri ve dağılımı

Satır sayısı	Type
15000	normal
5000	password
529	scanning
866	xss
5000	ddos
2865	ransomware
5000	injection
5000	backdoor

3.1.2.7. Modbus

Modbus, endüstriyel otomasyon sistemlerinde yaygın olarak kullanılan bir haberleşme protokolüdür. Modbus protokolünden elde edilen veri seti, farklı işlev kodlarına ve saldırı türlerine dayalı olarak analiz edilmiştir. Aşağıda, bu protokolün özellikleri ve saldırı türlerinin veri setindeki dağılımı Tablo 3.13 ve Tablo 3.14'te detaylı şekilde sunulmuştur.

Tablo 3.13. Modbus verilerinin özellikleri ve türleri

ID	Özellik	Tür
1	date	Tarih
2	time	zaman
3	FC1_Read_Input_Register	Sayı
4	FC2_Read_Discrete_Value	Sayı
5	FC3_Read_Holding_Register	Sayı
6	FC4_Read_Coil	Sayı
7	label	Sayı
8	type	Metin

Tablo 3.14. Saldırı türleri ve dağılımı

Satır sayısı	Type
5000	backdoor
5000	injection
15000	normal
577	xss
5000	password
529	scanning

3.1.3. TON_IoT Veri Kümesi Dengesizliği

TON_IoT veri kümesindeki cihazlardan elde edilen veriler, saldırı türleri arasında ciddi bir dengesizlik göstermektedir. Örneğin, Thermostat (Termostat) cihazına ait verilere baktığımızda, normal (15.000 satır) kayıtlarının, diğer türlere kıyasla oldukça fazla temsil edildiği görülmektedir. Buna karşılık, scanning (61 satır) ve xss (449 satır) gibi saldırı türleri oldukça düşük bir temsil oranına sahiptir. Bu tür bir dengesizlik, sınıflandırma modellerinin azınlık sınıflarını doğru bir şekilde öğrenmesini zorlaştırabilir ve modelin performansını olumsuz etkileyebilir. Bu noktada, SMOTE gibi veri dengeleme yöntemlerinin önemi ortaya çıkmaktadır.

Bu çalışmada, TON_IoT veri kümesindeki dengesizliği gidermek için SMOTE yöntemi kullanılacak ve dengenin sağlanmasının sınıflandırma modellerinin doğruluğuna olan etkisi detaylı bir şekilde analiz edilecektir.

3.2. Makine Öğrenimi (ML) Yöntemleri

IoT sistemlerinde makine öğrenimi (ML), cihazlara yerel zeka kazandırmada temel bir yaklaşımdır (Mahdavinejad ve diğ., 2018). Bu çalışmada, dengesiz veri kümelerinde yaygın olarak kullanılan K-En Yakın Komşu (KNN) ve Destek Vektör Makineleri (SVM) algoritmalarına ek olarak, çok katmanlı algılayıcı (MLP) ve Lojistik Regresyon (LR) gibi güncel yöntemler de değerlendirilmiştir. KNN küçük veri setlerinde esnek sınıflandırma sunarken, SVM daha karmaşık yapılarda güçlü karar sınırları çizebilmektedir. MLP ve LR ise özellikle doğrusal olmayan ilişkilerin öğrenilmesinde etkili alternatifler olarak öne çıkmaktadır. Bu dört algoritmanın performansı karşılaştırmalı şekilde analiz edilmiştir.

3.2.1. Support Vector Machine (SVM)

SVM, genellikle regresyon veya ikili sınıflandırma görevlerinde kullanılan popüler bir denetimli öğrenme tekniğidir. Bu yöntem, 1995 yılında Corinna Cortes ve Vladimir Vapnik tarafından geliştirilmiştir (Cortes ve Vapnik, 1995). Bu yöntem, sınıflandırma ve regresyon gibi problemlerde oldukça başarılı sonuçlar verir. SVM, özellikle ikili sınıflandırma (binary classification) problemlerinde tercih edilmekle birlikte, çok sınıflı

problemlere uygulanabilir hale getirilen türevleri de mevcuttur. SVM, verileri n-boyutlu bir uzayda analiz eder ve her bir veri noktasını belirli bir koordinata atar. Amaç, sınıfları birbirinden en iyi şekilde ayıran bir hiper düzlem (hyperplane) bulmaktır. Bu ayrım, veriler arasındaki mesafeyi (margin) en üst düzeye çıkararak gerçekleştirilir. Örneğin, aşağıdaki Denklem (3.1)'de düzlemi tanımlamada kullanılır:

$$w^t x + b = 0 \tag{3.1}$$

Burada w, ağırlık vektörünü, x, özellik vektörünü, ve b, önyargı değerini temsil eder. Amaç, doğrusal hiper düzlemi kullanarak verileri doğru bir şekilde sınıflandırmaktır. Eğer sınıflar doğrusal olarak ayrılamazsa, çekirdek (kernel) fonksiyonları kullanılarak veriler daha yüksek boyutlu bir uzaya dönüştürülür. En yaygın kullanılan çekirdek fonksiyonlar matematiksel ifadeleri, Denklem (3.2), (3.3) ve (3.4)'deki verilmiştir.:

Doğrusal çekirdek (Linear Kernel): $K(x_i, x_j) = x_i^\top x_j$ (3.2)

Radyal Bazlı Fonksiyon (RBF): $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$ (3.3)

Polinom çekirdek (Polynomial Kernel): $K(x_i, x_j) = (x_i^\top x_j + c)^d$ (3.4)

SVM, sınıflandırma problemlerinde diğer algoritmalara kıyasla genellikle daha yüksek doğruluk sunar. Özellikle sınıflar arasında belirgin bir ayrım olduğunda ve veriler karmaşık yapılar içerdiğinde SVM'nin başarısı artar. Bu yöntem, az sayıda veriyle çalışılan durumlarda da etkili sonuçlar verebilir.

Ayrıca, aşırı öğrenme (overfitting) riskini azaltmak için margin optimizasyonu gibi özellikleriyle dikkat çeker. Bu çalışmada SVM'nin seçilmesinin temel nedeni, IoT tabanlı veri kümelerinde dengesiz veri dağılımına rağmen sınıflandırma başarısını artırma potansiyelidir. SVM, doğru hiper düzlemi belirleyerek sınıflandırma performansını optimize eder. Çalışmada, SVM'nin özellikle SMOTE uygulanmadan önce ve sonra performansının kıyaslanması hedeflenmiştir. Böylece, veri dengesizliği gibi zorluklara karşı algoritmanın etkinliği detaylı şekilde analiz edilecektir.

3.2.2. K-Nearest Neighbor (En Yakın K-Komşu - KNN)

Bu K-Nearest Neighbor (KNN), sınıflandırma ve regresyon gibi çeşitli problem türlerinde yaygın olarak kullanılan bir algoritmadır. Algoritmanın temel prensibi, veri noktalarının giriş uzayında birbirine olan yakınlıklarına dayanır ve benzer noktaların birbirine daha yakın olduğu varsayılır. Bu sayede, en yakın k komşu, bir veri noktası için doğru bir tahmin sağlamada kritik bir rol oynar (Cover ve Hart, 1967).

Sınıflandırma problemlerinde, tahmin edilen sınıf, en yakın komşuların çoğunluk sınıfına göre belirlenirken; regresyon problemlerinde ise tahmin edilen değer, en yakın komşuların ortalaması alınarak hesaplanır. KNN yöntemi, k değeri 1 olduğunda en temel haliyle kullanılır. Bu durum, test örneğinin sınıflandırılmasında, eğitim verilerindeki en yakın komşusunun sağladığı bilginin kullanılacağı anlamına gelir. KNN algoritmasının temel prensibi Şekil 3.1'de gösterilmektedir. Bir düzlemde iki nokta x ve y arasındaki mesafe, aşağıda verilen Denklem (3.5), (3.6) veya (3.7)'deki numaralı mesafe denklemlerinden biri kullanılarak hesaplanabilir.

Öklid Mesafesi (Euclidean Distance):

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.5}$$

Manhattan Mesafesi (Manhattan Distance):

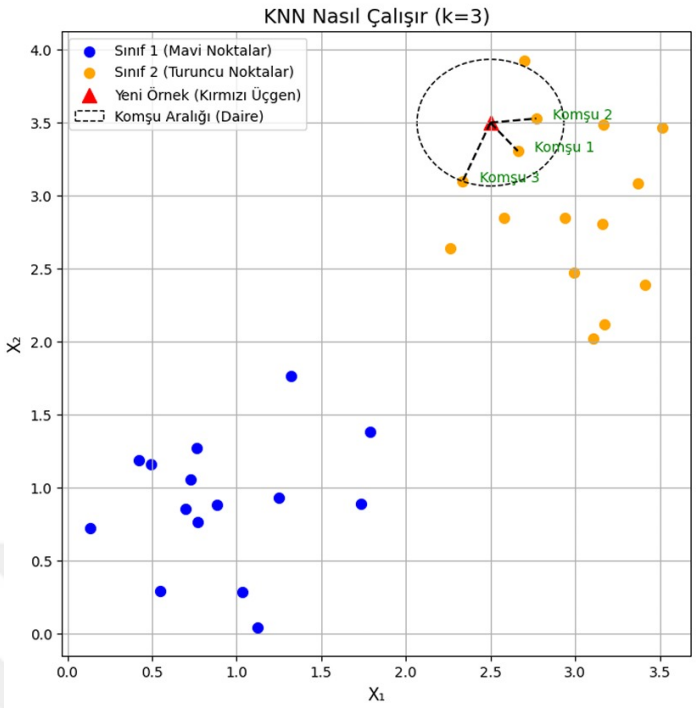
$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i| \tag{3.6}$$

Minkowski Mesafesi (Minkowski Distance):

$$d(x,y) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{3.7}$$

P güç parametresi olup mesafeyi tanımlar. p=1 olduğunda Manhattan mesafesini, p=2 olduğunda ise Öklid mesafesini verir.

KNN algoritmasının çalışma prensibini Şekil 3.1 de göstermektedir. Yeni bir örnek (kırmızı üçgen) en yakın üç komşusuna göre (k=3) sınıflandırılmaktadır. Komşular,



Şekil 3.1. KNN çalışma prensibini

çevresi siyah kesikli çizgiyle belirtilen alanda yer almaktadır. X_1 *ve* X_2 Özellikleri, iki farklı sınıfın (mavi ve turuncu) verilerini temsil etmektedir.

3.2.3. Lojistik Regresyon (LR) Algoritması

Lojistik regresyon, bir hedef değişkenin (dependent variable) olasılığını tahmin etmek için kullanılan gözetimli bir sınıflandırma algoritmasıdır. Hedef değişken ikili (dichotomous) yapıda olup yalnızca iki olası sınıfa sahiptir (örneğin 1 veya 0, başarılı/başarısız, evet/hayır). Bu nedenle bağımlı değişken ikili (binary) bir nitelik taşır ve lojistik regresyon matematiksel olarak $P(Y = 1)$ olasılığını X değişkenleri cinsinden tahmin edebilir.

Basit bir makine öğrenmesi tekniği olan lojistik regresyon, spam tespitinden diyabet öngörüsüne, malignite (kötü huylu tümör) tespitine kadar geniş bir sınıflandırma yelpazesinde uygulanabilir. Logistik regresyonun temel olasılık tahmini, Denklem (3.8)'de tanımlanmıştır:

$$\hat{y} = \frac{\exp(b + wx)}{1 + \exp(b + wx)} \tag{3.8}$$

burada:

$\hat{y}$ Tahmin edilen çıktı (bir gözlemin "pozitif sınıfa" ait olma olasılığı),

b Bias terimi,

w Girdi için katsayı (ağırlık),

x Girdi (özellik) değeri,

exp e tabanlı üstel fonksiyon.

Sigmoid (lojistik) dönüşümü Denklem (3.9)'da verilmiştir. lojistik regresyonda doğrusal bileşenden elde edilen $z = b + wx$ değerini $[0,1]$ aralığında bir olasılık değeri hâline dönüştürmek için kullanılır:

$$\sigma(z) = \frac{1}{1+\exp(-z)} \tag{3.9}$$

burada:

$\sigma(z)$ Sigmoid fonksiyonunun çıktısı (0 ile 1 arası bir olasılık),

z Fonksiyonun girdisi (genellikle $z = b + wx$).

Denklem (3.8) ile önce $z = b + wx$ doğrusal bileşeni hesaplanır, ardından bu değer $\exp(b + wx)$ şeklinde üstel kısıma girer. Sonra "$1 + \exp(b + wx)$" ifadesine bölünerek $\hat{y}$ olasılık tahmini elde edilir. Denklem (3.9) ise doğrudan z değerini sigmoid fonksiyonuna göndererek aynı sonucu verir: $\hat{y} = \sigma(z)$. Bu yaklaşımla lojistik regresyon, gözlemin "pozitif sınıfa" ait olma olasılığını güvenilir bir şekilde $[0,1]$ aralığında hesaplamış olur (Mahalaxmi ve Rekha, 2022), (Bhavitha ve Thangaraj, 2022).

3.2.4. Çok Katmanlı Algılayıcı (MLP) Algoritması

Çok Katmanlı Algılayıcı (Multilayer Perceptron, MLP) ileri beslemeli (feedforward) yapıya sahip bir yapay sinir ağıdır. Bir MLP, bir adet giriş katmanı, en az bir adet gizli katman ve bir adet çıkış katmanından oluşur. Her katmandaki her bir nöron (düğüm), bir sonraki katmandaki tüm nöronlara bağlıdır (fully connected).

Gizli katmanlar, doğrusal olmayan (nonlineer) ilişkileri modelleyebilmek için gerekli olup, çıkış katmanı ise sınıflandırma veya regresyon problemlerine uygun olacak şekilde çıktı üretir.

MLP'de her bir nöron, kendisine ait ağırlıklar (w) ve bir önyargı (bias) terimi (b) ile aldığı girdileri önce doğrusal bir bileşende toplar. Örneğin, tek bir gizli katmandaki nöron için doğrusal bileşen ve aktivasyon çıktısı, Denklem (3.10) ve (3.11)'deki verilmiştir:

$$z = b + \sum_{i=1}^{n} w_i x_i \tag{3.10}$$

şeklinde hesaplanan z değeri, daha sonra bir aktivasyon fonksiyonuna (ϕ) gönderilerek nöronun çıktısı elde edilir:

$$y = \phi(z) \tag{3.11}$$

Gizli katmanlarda yaygın olarak kullanılan aktivasyon fonksiyonlarının tanımları, Denklem (3.12), (3.13) ve (3.14)'deki verilmiştir:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{3.12}$$

$$\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \tag{3.13}$$

$$\mathrm{ReLU}(z) = \max(0, z) \tag{3.14}$$

Sigmoid ve Tanh, çıktıyı sırasıyla $(0,1)$ ve $(-1,1)$ aralıklarına sıkıştırırken, ReLU negatif değerleri sıfıra çeker ve pozitif bölgede doğrusal davranış sergiler.

Eğitim aşamasında MLP, geriye yayılım (backpropagation) algoritmasıyla parametrelerini w ve b günceller. Önce ileri besleme (forward propagation) yapılır: her katmanda aktivasyonlar hesaplanır ve çıkış katmanından tahmin $\hat{y}$ elde edilir. Daha sonra gerçek etiket y ile $\hat{y}$ arasındaki fark, uygun bir kayıp fonksiyonu — sınıflandırma için çapraz entropi (Cross-Entropy), regresyon için ortalama karesel hata (MSE) — ile ölçülür. Kayıp türevleri zincir kuralı (chain rule) ile katman katman geriye iletilir ve ağırlıklar ile biaslar gradyan inişiyle güncellenir.

Örnek olarak, tek bir çıktı nöronu için çapraz entropi kaybı ve gradyan inişine dayalı parametre güncellemeleri, Denklem (3.15),(3.16) ve (3.17)'deki tanımlanır:

$$\mathscr{L} = -\big[y\log(\hat{y}) + (1-y)\log(1-\hat{y})\big] \tag{3.15}$$

olarak tanımlandığında, her bir ağırlık ve bias güncellemesi öğrenme hızı η kullanılarak

$$w \leftarrow w - \eta \frac{\partial \mathscr{L}}{\partial w} \tag{3.16}$$

$$b \leftarrow b - \eta \frac{\partial \mathscr{L}}{\partial b} \tag{3.17}$$

şeklinde gerçekleştirilir. Bu adımlar, yeterli sayıda veri örneği üzerinden birçok yinelemeye (epoch) tabi tutularak tekrarlanır ve ağın öğrenmesi sağlanır.

MLP'nin performansı; gizli katman sayısı, katman başına nöron sayısı, aktivasyon fonksiyonu seçimi, öğrenme hızı, veri ön işleme (normalizasyon/standartizasyon) gibi hiperparametrelere bağlıdır. Bir gizli katman içeren MLP, doğrusal olarak ayrılabilir (linearly separable) problemler dışında basit doğrusal sınıflandırmaları da çözebilir; ancak verinin doğrusal olmayan yapısını öğrenmek için iki veya daha fazla gizli katman gerekebilir. Ayrıca, yeterli veri ve uygun düzenleme teknikleri (regularization: L_2 cezası, dropout vb.) kullanılmazsa ağ, karmaşık mimarilerde aşırı öğrenmeye (overfitting) yatkın hale gelebilir (Taud ve Mas, 2018).

Özetle, MLP:

- İleri beslemeli, katmanlı yapıya sahiptir.
- Gizli katmanlardaki aktivasyon fonksiyonları ile doğrusal olmayan örüntüleri öğrenebilir.
- Backpropagation ile ağırlıkları güncelleyerek veriye uyum sağlar.
- Uygun hiperparametre seçimi ve veri hazırlığı ile hem sınıflandırma hem de regresyon problemlerinde başarılı sonuçlar verir.

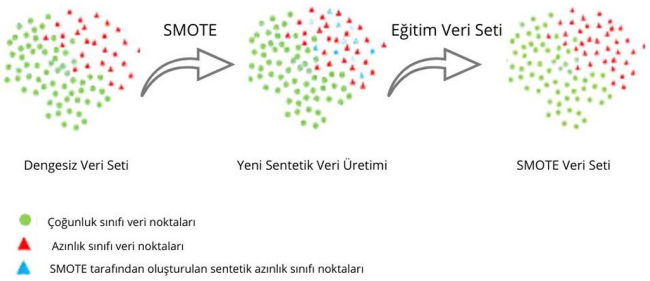
3.3. SMOTE Tekniği

Dengesiz veri kümelerindeki sorunu çözmek için SMOTE (Synthetic Minority Over Sampling Technique) yöntemi, azınlık sınıfına ait yapay örnekler oluşturur. Veri kümesini dengelemek amacıyla, azınlık sınıfındaki her örneğin en yakın komşularını bularak rastgele bir tanesini seçer ve bu komşuların özelliklerini rastgele bir şekilde birleştirerek yeni, sentetik bir örnek oluşturur. Bu sayede, azınlık sınıfının temsil oranı artırılır ve daha dengeli bir veri kümesi elde edilir (Mishra, 2017).

SMOTE algoritması şu şekilde çalışır:

- Azınlık sınıfına ait bir veri noktası (örnek: *az_obs*) rastgele seçilir.
- Seçilen *az_obs* noktasının en yakın k komşusu belirlenir.
- *az_obs* ile her bir k komşusu arasındaki fark vektörleri hesaplanır.
- k komşular arasından rastgele bir tanesi seçilir ve bu fark vektörüne $[0, 1]$ aralığında rastgele bir değer eklenerek sentetik bir örnek oluşturulur.
- Gerekli sayıda sentetik örnek oluşturulana kadar bu adımlar tekrarlanır.

Bu algoritma, azınlık sınıfının veri dengesini artırmak ve veri setini daha dengeli hâle getirmek için tasarlanmıştır. Çalışma prensibi Şekil 3.2'de gösterilmektedir.



Şekil 3.2. SMOTE yöntemi ile sentetik örneklerin oluşturulması

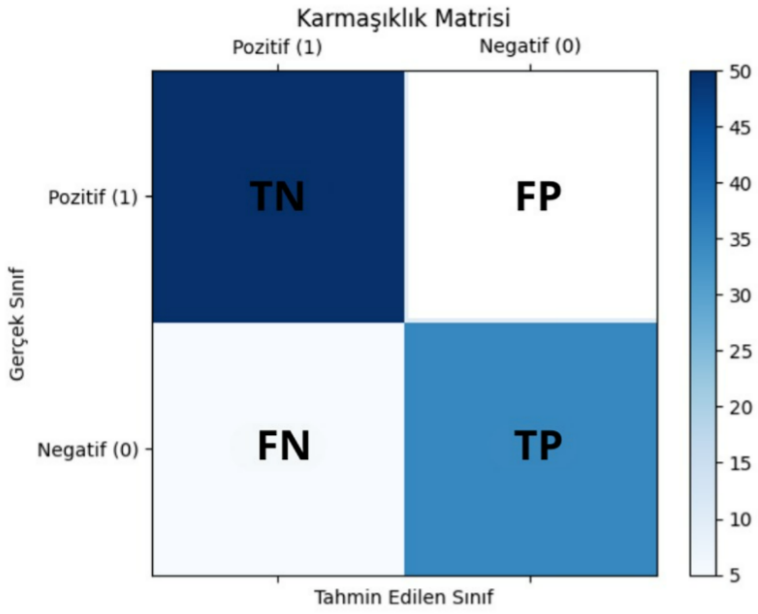
3.4. Değerlendirme Teknikleri

3.4.1. Karmaşıklık Matrisi (Confusion Matrix)

Karmaşıklık matrisi, tahmin edilen etiketlerin gerçek etiketlerle karşılaştırılmasıyla oluşturulur ve dört sonuç ortaya çıkarır: false positive (FP), false negative (FN), true positive (TP) ve true negative (TN).

- TP: Pozitif sınıfa ait olan ve doğru bir şekilde pozitif olarak tahmin edilen örneklerin sayısını ifade eder.
- TN: Negatif sınıfa ait olan ve doğru bir şekilde negatif olarak tahmin edilen örneklerin sayısını ifade eder.
- FN: Pozitif sınıfa ait olan ancak negatif olarak tahmin edilen örneklerin sayısını ifade eder.
- FP: Negatif sınıfa ait olan ancak pozitif olarak tahmin edilen örneklerin sayısıdır.

Şekil 3.3'te, iki sınıf için bir karmaşıklık matrisi örneğini göstermektedir.



Şekil 3.3. Karmaşıklık matrisi örneğini

3.4.2. Sınıflandırma Ölçütleri (Classification Measures)

Model performansı, sınıflandırma ölçütleri aracılığıyla değerlendirilir. Karmaşıklık matrisi, doğruluk (Accuracy), kesinlik (Precision), duyarlılık (Sensitivity) ve F1 skoru gibi temel metriklerin hesaplanmasında kullanılır.

Bu çalışmada, modelin performansını değerlendirmek için bu metriklere dayalı ölçümler uygulanmış olup, ilgili formüller aşağıda açıklayıcı bir şekilde Denklem (3.18), (3.19), (3.20) ve (3.21)'deki görselleştirilmiştir.

Doğruluk (Accuracy): Doğru sınıflandırılan örneklerin, tüm tahminlere oranı.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.18}$$

Hassasiyet (Precision): Pozitif tahminlerin içinde gerçekten pozitif olanların payı.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.19}$$

Geri Çağırma (Recall): Gerçek pozitiflerin ne kadarının doğru yakalandığı.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.20}$$

F1-Skoru: Precision ile Recall'un harmonik ortalaması; sınıf dengesizliğinde özellikle yararlıdır.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.21}$$

AUC (Eğri Altındaki Alan) veya ROC)

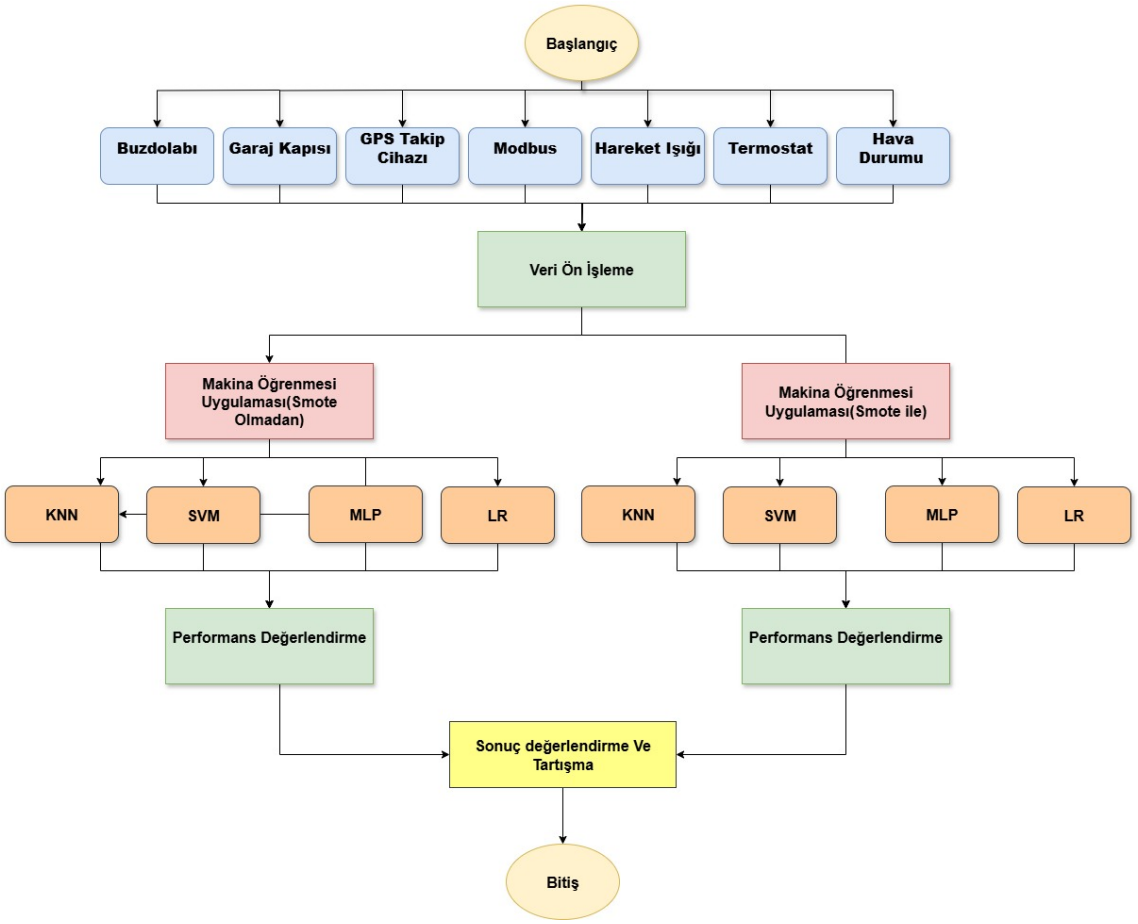
Doğru Pozitif (TP) oranı, Yanlış Pozitif (FP) oranına karşı grafiğe dökülerek ROC eğrisi oluşturulur. ROC eğrisinde X ekseninde FP oranı, Y ekseninde ise TP oranı bulunur. TP oranı, duyarlılığın bir başka terimidir ve hesaplamaları Denklem (3.22) ve (3.23)'deki, ile detaylandırılmıştır.

$$\text{TP rate} = \frac{TP}{TP+FN} \tag{3.22}$$

$$\text{FP rate} = \frac{FP}{FP+TN} \tag{3.23}$$

4. METODOLOJİ VE SONUÇLARIN DEĞERLENDİRMESİ

Bu çalışmada, TON IoT veri kümesi kullanılarak veri analizi ve makine öğrenimi modellerinin performansını değerlendirmek amacıyla aşağıdaki adımlar izlenmiştir. Bu süreçlerin genel akışı Şekil 4.1'de gösterilmektedir.



Şekil 4.1. Ton IoT veri setinde SMOTE ile makine öğrenimi ile performans analizi süreci akış diyagramı

4.1. Ton_Iot Veri Kümesi

Çalışmada, yedi farklı IoT cihazına ait veriler kullanılarak DDoS, Backdoor, Injection gibi saldırı türleri makine öğrenimi algoritmalarıyla sınıflandırılmıştır. Sınıf dengesizliğinin model performansı üzerindeki olumsuz etkisini azaltmak amacıyla SMOTE yöntemi uygulanmış ve algoritmaların SMOTE öncesi ve sonrası sonuçları karşılaştırılmıştır. Veri seti, "Normal" ve saldırı türlerine göre etiketlenmiş olup, cihazlara göre dağılım Tablo 4.1'de sunulmuştur.

Tablo 4.1. IoT CIHAZLARINA GÖRE SALDIRI TÜRLERI DAĞILIMI

Cihazlar	Normal	DDoS	Backdoor	Injection	Password	Ransomware	XSS	Scanning
Buzdolabı	15000	5000	5000	5000	5000	2902	2042	-
Garaj Kapısı	15000	5000	5000	5000	5000	2902	1156	529
GPS Takip	15000	5000	5000	5000	5000	2833	577	550
Modbus	15000	-	5000	5000	5000	-	577	529
Hareket Işığı	15000	5000	5000	5000	5000	2264	449	1775
Hava Durumu	15000	5000	5000	5000	5000	2865	866	529
Termostat	15000	-	5000	5000	5000	2264	449	61

Tabloda, özellikle XSS ve Scanning gibi düşük frekanslı saldırı türlerinde belirgin bir dengesizlik göze çarpmaktadır.

4.2. Veri Ön İşleme (Data Preprocessing)

TON_IoT veri kümesinden elde edilen alt kümeler, makine öğrenimi modellerine uygun hale getirilmesi amacıyla aşağıdaki adımlara tabi tutulmuştur:

- Veri Setinin Yüklenmesi ve Analizi:
  Veri setleri ile ilgili kaynaklardan yüklenmiş ve boyutları ile özellikleri detaylı bir şekilde incelenmiştir. Ayrıca, hedef değişken olan ‘type‘ sütunundaki sınıf dağılımları görselleştirilmiş ve dengesizlik sorununun etkileri değerlendirilmiştir.
- Kategorik Verilerin Sayısallaştırılması:
  Tarih, zaman ve durum gibi kategorik değişkenler, LabelEncoder yöntemi ile sayısal değerlere dönüştürülmüştür. Bu adım, makine öğrenimi algoritmalarının bu tür verilerle daha etkili bir şekilde çalışabilmesini sağlamak amacıyla yapılmıştır (Albasheer ve diğ., 2024).
- Bağımlı ve Bağımsız Değişkenlerin Ayrılması:
  Hedef değişken olan `type`, bağımsız değişkenlerden ayrılmıştır:
  X- Bağımsız Değişkenler: IoT cihazlarına ait özellikleri temsil eder.
  y- Bağımlı Değişken: Saldırı türlerini temsil eden değişken.
- Eğitim ve Test Kümelerine Ayrılma:
  Veri seti, modellerin daha doğru şekilde değerlendirilmesi amacıyla %70 eğitim, %15 doğrulama (validation) ve %15 test olacak şekilde üçe ayrılmıştır.

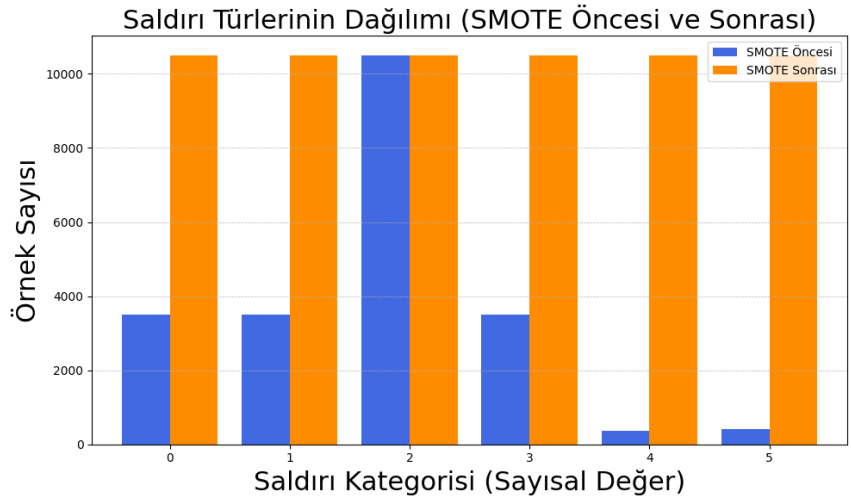
4.3. SMOTE Uygulaması

Veri kümesindeki sınıf dengesizliğini gidermek amacıyla her bir IoT cihazına ayrı ayrı SMOTE uygulanmıştır. SMOTE yalnızca azınlık sınıflara uygulanmakta olup, örnek sayısı zaten fazla olan çoğunluk sınıflara herhangi bir sentetik örnek eklenmemektedir. Bu bağlamda, her cihazda saldırı türleri arasındaki en büyük sınıf sabit tutulmuş, diğer azınlık sınıflar için sentetik örnekler üretilerek sayıları bu çoğunluk sınıfa eşitlenmiştir. Örneğin Tablo 4.2'de saldırı kategorileri ve sınıflandırmada kullanılan sayısal değerler yer almaktadır: Backdoor (0), Injection (1), Normal (2), Password (3), Scanning (4) ve XSS (5).

Şekil 4.2'de görüldüğü gibi, Modbus cihazında "Normal" sınıf 15.000 örnekle en büyük sınıfı temsil etmektedir; bu nedenle diğer sınıfların örnek sayısı, SMOTE ile bu sayıya ulaştırılmıştır.

Tablo 4.2. IoT Modbus saldırı kategorileri ve sayısal değerleri

Saldırı Adı	Sayısal Değer
Backdoor	0
Injection	1
Normal	2
Password	3
Scanning	4
XSS	5



Şekil 4.2. IoT Modbus SMOTE Öncesi ve Sonrası Kategori Dağılımı

4.4. Makine Öğrenimi Algoritmalarının Uygulanması

Deneysel aşama, K-En Yakın Komşu (KNN), Destek Vektör Makineleri (SVM), Çok Katmanlı Algılayıcı (MLP) ve Lojistik Regresyon (LR) olmak üzere dört temel makine öğrenimi sınıflandırıcısının, SMOTE uygulanmadan önce ham (dengesiz) IoT cihaz veri kümeleri üzerinde eğitilmesi ve değerlendirilmesini içermektedir. Standart makine öğrenimi uygulamalarına uygun olarak, her algoritmanın performansı, test kümesinde nihai değerlendirme yapılmadan önce ayrılmış doğrulama kümesi (validation set) kullanılarak titizlikle değerlendirilmiştir. Model etkinliği, karmaşıklık matrisi analizinden türetilen beş temel metrik üzerinden ölçülmüştür: sınıflandırma doğruluğu, kesinlik (precision), duyarlılık (recall), F1-skoru ve ROC eğrisi altındaki alan (AUC) Sayegh ve diğ., 2024. Farklı saldırı kategorileri arasındaki karşılaştırmalı performansa özellikle vurgu yapılmıştır. Bu temel değerlendirme, sınıf dengesizliği sorununu ele almadan önce her algoritmanın öncül tespit yeteneklerini ortaya koymuştur.

5. IOT CİAZLARINA GÖRE SALDIRI TESPİTİ PERFORMANS ANALİZİ

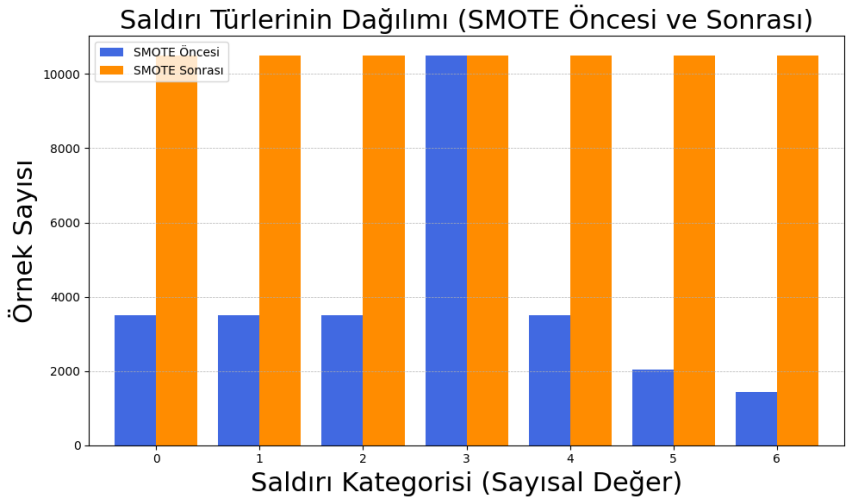
Bu bölümde, her bir IoT cihazı için KNN, SVM, Lojistik Regresyon (LR) ve Çok Katmanlı Algılayıcı (MLP) algoritmalarının SMOTE öncesi ve sonrası performans analizleri sunulmaktadır. Her cihaz için, dört farklı makine öğrenimi algoritmasının doğruluk (Accuracy), kesinlik (Precision), duyarlılık (Recall), F1 skoru ve AUC metriklerine dayanan karmaşıklık matrisleri (Confusion Matrix) karşılaştırmalı olarak değerlendirilmiştir.

5.1. Fridge (Akıllı Buzdolabı Cihazı)

Buzdolabı Cihazı için saldırı türlerinin sayısal karşılıkları Tablo 5.1'de verilmiştir. SMOTE etkisi ise Şekil 5.1'de gösterilmektedir.

Tablo 5.1. IoT AKILLI BUZDOLABI SALDIRI KATEGORILERI VE SAYISAL DEĞERLERI TABLOSU

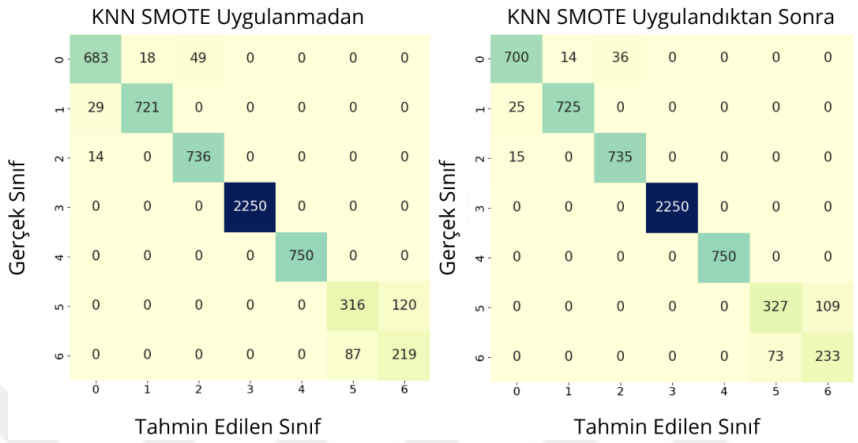
Saldırı Adı	Sayısal Değer
Backdoor	0
ddos	1
Injection	2
Normal	3
Password	4
ransomware	5
XSS	6



Şekil 5.1. IoT Akıllı Buzdolab SMOTE Öncesi ve Sonrası Kategori Dağılımı

5.1.1. KNN Algoritması

Şekil 5.2'de KNN sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



Şekil 5.2. KNN - Fridge için Confusion Matrix

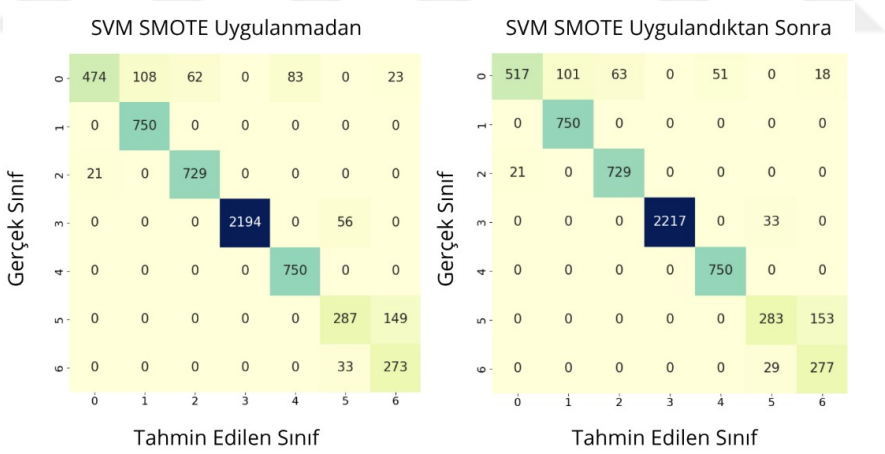
KNN algoritması ile analiz edilen akıllı buzdolabı cihazına ait karışıklık matrisi incelendiğinde, SMOTE uygulamasının özellikle temsil oranı en düşük olan sınıflarda etkili sonuçlar doğurduğu görülmektedir. En düşük temsile sahip olan XSS (6) sınıfında doğru sınıflandırma sayısı 219'dan 233'e, Ransomware (5) sınıfında ise 316'dan 327'ye yükselmiştir. Bu sınıflar için üretilen sentetik örnekler, veri dağılımındaki uçurumu kısmen kapatarak modelin bu sınıfları daha iyi öğrenmesini sağlamış, böylece ayrım gücü artırılmıştır. Özellikle bu sınıfların başlangıçta sınıf sınırlarında yeterince temsil edilememesi, SMOTE'un etkisini daha görünür kılmıştır. Backdoor (0) sınıfı ise 683'ten 700'e çıkarak görece küçük ama olumlu bir artış sergilemiştir. Bu sınıf, orta düzeyde temsile sahip olmasına rağmen sentetik örneklerden kısmen fayda sağlamıştır. Diğer taraftan, bazı sınıflarda değişim sınırlı kalmıştır. Örneğin, DDoS (1) sınıfı 721'den 725'e yükselerek hafif bir iyileşme göstermiştir. Ancak Password (4) sınıfında doğru sınıflandırma sayısı 750 olarak sabit kalmış, Injection (2) sınıfında ise 736'dan 735'e küçük bir azalma gözlemlenmiştir. Bu sonuçlar, SMOTE'un yalnızca sayısal çoğaltma değil, aynı zamanda sentetik örneklerin veri uzayındaki konumlarına ve dağılımına bağlı olarak etkili olabileceğini göstermektedir. Öte yandan, Normal (3) sınıfı için herhangi bir örnek oluşturulmamıştır. Bu sınıf, toplamda 15000 örnekle veri setindeki en baskın sınıf

olup, SMOTE işlemi sırasında referans olarak kullanılmıştır. Bu nedenle, model bu sınıfı baştan itibaren yüksek doğrulukla (2250 doğru sınıflandırma) tanıyabilmekteydi. SMOTE sonrasında da bu başarı seviyesi korunarak modelin genel istikrarı muhafaza edilmiştir.

Sonuç olarak, SMOTE uygulaması, özellikle azınlık sınıflarında KNN algoritmasının performansını artırmış; sentetik örnekler sayesinde düşük temsil edilen bazı sınıfların model tarafından daha iyi öğrenilmesi sağlanmıştır. Her ne kadar bazı sınıflarda etki sınırlı kalmış olsa da, modelin genel sınıflandırma dengesinin iyileştiği ve sınıflar arası farkların kısmen azaldığı gözlemlenmiştir. Bu durum, örnek temelli algoritmalarda veri dengesizliğinin doğrudan model başarımını etkilediğini ve SMOTE gibi tekniklerin bu etkiyi azaltmada önemli bir araç olduğunu açıkça ortaya koymaktadır.

5.1.2. SVM Algoritması

Şekil 5.3'te SVM sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



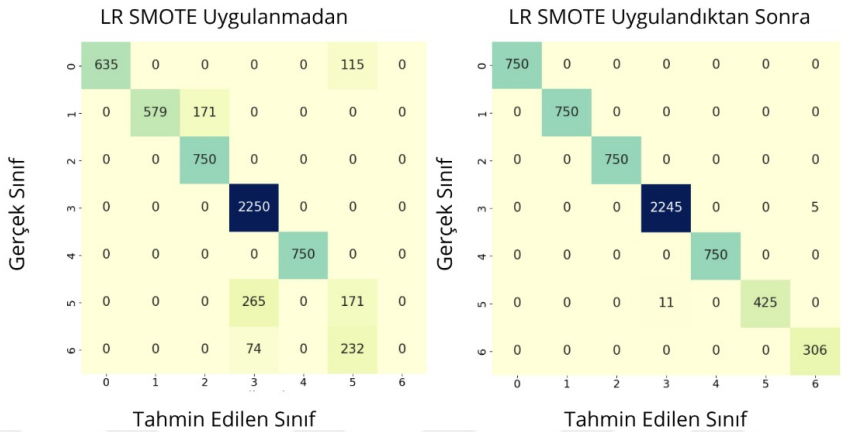
Şekil 5.3. SVM - Fridge için Confusion Matrix

SMOTE tekniğinin uygulanmasından sonra SVM algoritmasının performansı analiz edildiğinde, bu tekniğin etkisinin, KNN gibi diğer algoritmalara kıyasla görece sınırlı kaldığı görülmektedir. SMOTE, dengesiz veri setlerinde azınlık sınıflarının örnek sayılarını artırarak baskın sınıf olan "Normal" sınıfına yaklaştırmış olsa da, bu niceliksel artış tüm sınıflarda niteliksel bir iyileşmeye dönüşmemiştir. Bu sınırlı etki, SVM

algoritmasının doğasından kaynaklanmaktadır; zira SVM, sınıflar arasındaki karar sınırlarını belirlerken yalnızca az sayıda destek vektörüne dayanır. Dolayısıyla, üretilen sentetik örneklerin bu karar noktalarına doğrudan etki etmemesi durumunda, modelin performansı kayda değer biçimde değişmemektedir. Detaylı performans açısından değerlendirildiğinde, sentetik veri üretimine tabi tutulmayan "Normal" sınıfı yüksek doğruluğunu korumuş ve doğru sınıflandırılan örnek sayısında hafif bir artış kaydedilmiştir (2194'ten 2217'ye). Bu durum, baskın sınıflar karşısında SVM algoritmasının istikrarlı kaldığını göstermektedir. Öte yandan, Password (4), DDoS (1) ve Injection (2) sınıflarında doğru sınıflandırılan örnek sayısında hiçbir artış gözlenmemiş ya da kayda değer bir değişiklik yaşanmamıştır. Hatta Ransomware (5) sınıfında doğru sınıflandırma sayısı azalarak 287'den 283'e düşmüştür; bu da SMOTE'un bu sınıf üzerinde sınırlı olumsuz bir etkisinin olabileceğini göstermektedir. Bu durum, yalnızca örnek sayısını artırmanın yeterli olmadığını, üretilen örneklerin niteliği ve temsili dağılımının da büyük önem taşıdığını ortaya koymaktadır. Bunun aksine, yalnızca Backdoor (0) sınıfında anlamlı bir iyileşme gözlenmiştir; doğru sınıflandırılan örnek sayısı 474'ten 517'ye yükselmiştir. Bu durum, SMOTE'un bu sınıfta olumlu etkiler yarattığını göstermektedir. XSS (6) sınıfında ise doğru sınıflandırma sayısında yalnızca küçük bir artış (273'ten 277'ye) yaşanmış olup, bu artış model performansında anlamlı bir gelişme olarak değerlendirilemez. Bu sonuçlar, SMOTE'un etkinliğinin yalnızca örnek sayısındaki artışa değil, aynı zamanda kullanılan algoritmanın yapısal özellikleriyle ne derece uyum sağladığına da bağlı olduğunu göstermektedir. SVM gibi karar sınırlarını destek vektörlerine göre belirleyen algoritmalarda, sentetik örneklerin modelin karar mekanizmasında etkili olacak şekilde yerleşmesi gerekmektedir. Aksi takdirde, bu örnekler sayısal bir artış sağlamaktan öteye geçememektedir. Sonuç olarak, SMOTE'un modelin genel performansını önemli ölçüde artırmamakla birlikte istikrarlı bir seviyede koruduğu görülmektedir. Bu durum, SMOTE'un belirli durumlarda tamamlayıcı bir araç olarak kullanılabileceğini; ancak her algoritmanın doğasına göre etki analizi yapılarak dikkatli bir şekilde uygulanması gerektiğini ortaya koymaktadır.

5.1.3. Lojistik Regresyon (LR) Algoritması

Lojistik Regresyon için SMOTE öncesi ve sonrası karmaşıklık matrisleri Şekil 5.4'te sunulmuştur.



Şekil 5.4. LR - Fridge için Confusion Matrix

Lojistik regresyon algoritmasının SMOTE öncesi ve sonrası performansı karşılaştırıldığında, özellikle düşük temsile sahip sınıflarda anlamlı iyileşmeler gözlemlenmiştir. En az temsil edilen sınıflardan Ransomware (5) sınıfında doğru sınıflandırılan örnek sayısı 265'ten 425'e, XSS (6) sınıfında ise 232'den 306'ya yükselmiştir. Bu artışlar, sentetik örneklerin bu sınıflarda modelin ayrım kabiliyetini önemli ölçüde artırdığını göstermektedir. Daha dikkat çekici olan ise, orta düzeyde temsil edilen ancak SMOTE ile desteklenen Backdoor (0) sınıfında doğruluk oranının 635'ten 750'ye, ve DDoS (1) sınıfında ise 579'dan 750'ye yükselmiş olmasıdır. Bu durum, modelin sınıflar arası ayrımı daha dengeli bir şekilde gerçekleştirmeye başladığını göstermektedir. Buna karşılık, Injection (2) ve Password (4) sınıflarında doğru sınıflandırma sayısı 750 olarak sabit kalmıştır; bu da modelin bu sınıfları zaten yeterince iyi öğrendiğini ve SMOTE'un bu sınıflarda ek katkı sağlamadığını göstermektedir. Normal (3) sınıfında ise çok küçük bir düşüş gözlemlenmiştir (2250'den 2245'e), ancak bu değişiklik modelin genel performans istikrarını tehdit edecek düzeyde değildir.

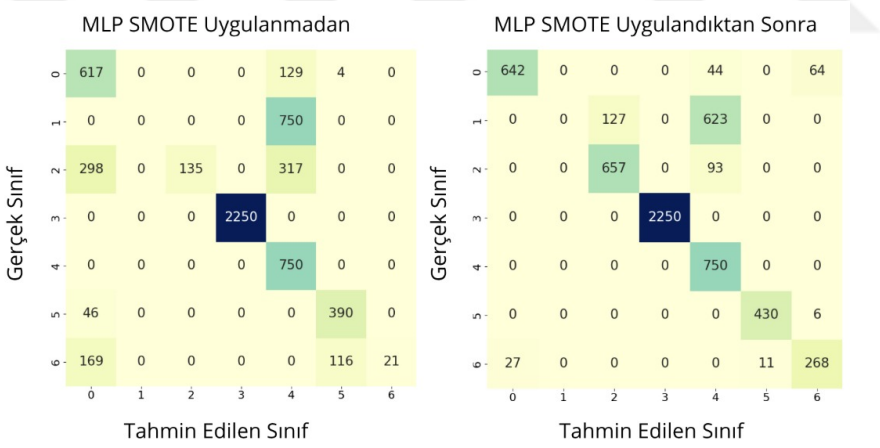
Lojistik regresyon algoritmasının SMOTE uygulamasına yüksek düzeyde tepki vermesi, doğrusallığa dayalı yapısıyla doğrudan ilişkilidir. Bu algoritma, sınıflar arasındaki ayrımı

lineer karar sınırlarıyla gerçekleştirdiği için, veri dağılımındaki dengesizlik modelin karar düzlemini ciddi şekilde etkileyebilmektedir. SMOTE ile azınlık sınıflarına eklenen sentetik örnekler, bu karar düzlemini daha dengeli bir şekilde konumlandırmış ve böylece özellikle düşük temsil edilen sınıflarda belirgin bir performans artışı sağlanmıştır. Bu durum, doğrusal sınıflandırıcıların veri dengesine olan hassasiyetini ortaya koymakta ve SMOTE'un bu tür modeller için güçlü bir dengeleme aracı olduğunu göstermektedir.

Sonuç olarak, SMOTE uygulaması, lojistik regresyon algoritması üzerinde yalnızca sayısal bir artış sağlamamış, aynı zamanda karar sınırlarının daha dengeli şekillenmesine katkı sunarak sınıflar arası öğrenme farklarını azaltmıştır. Bu da, modelin yalnızca baskın sınıflarda değil, azınlık sınıflarda da güvenilir sonuçlar üretebilmesini sağlamış ve genel sınıflandırma performansını daha dengeli hale getirmiştir.

5.1.4. MLP Algoritması

MLP'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.5'te yer almaktadır.



Şekil 5.5. MLP - Fridge için Confusion Matrix

Karışıklık matrislerinden elde edilen sonuçlar, SMOTE tekniğinin uygulanmasının ardından MLP (Çok Katmanlı Algılayıcı) algoritmasının performansında kısmi bir iyileşme sağladığını göstermektedir. Özellikle düşük temsile sahip sınıflarda belirgin bir gelişme gözlemlenmiştir. Örneğin, XSS (6) sınıfında doğru tahmin edilen örnek sayısı SMOTE öncesinde yalnızca 21 iken, uygulama sonrası 268'e yükselmiştir. Benzer

şekilde Injection (2) sınıfında bu sayı 135'ten 657'ye çıkmıştır. Bu durum, hedef sınıfın yapısal özellikleri belirgin olduğunda, algoritmanın sentetik örneklerden etkili şekilde öğrenebildiğini ortaya koymaktadır. Bununla birlikte, bazı sınıflarda hiçbir gelişme kaydedilmemiştir. Örneğin, DDoS (1) sınıfında hem SMOTE öncesi hem de sonrası doğru tahmin edilen örnek sayısı sıfırdır. Bu durum, sorunun yalnızca veri dengesizliğinden kaynaklanmadığını, aynı zamanda bu sınıfın yapısal özelliklerinin diğer sınıflar (örneğin Password veya Injection) ile örtüşmesi nedeniyle modelin ayrım yapmakta zorlandığını göstermektedir. Bu örtüşme, SMOTE'un etkinliğini sınırlayabilir. Yüksek temsile sahip sınıflar olan Normal (3) ve Password (4) için modelin doğruluk oranları sabit kalmış, herhangi bir bozulma gözlemlenmemiştir. Bu da, sentetik örneklerin modele zarar vermediğini ve MLP'nin çok katmanlı yapısı sayesinde dengeli sınıflarda istikrarlı performans sergileyebildiğini göstermektedir. Ayrıca, performans farklılıkları yalnızca SMOTE uygulamasına değil, aynı zamanda buzdolabı cihazına ait verilerin doğasına da bağlanabilir. Örneğin, XSS ve Injection gibi bazı saldırılar, belirgin davranış kalıplarına sahip olduğundan öğrenilmesi daha kolay hale gelirken; DDoS gibi saldırılar, cihaz üzerinde ayırt edici izler bırakmayabilir veya yapısal olarak yeterince farklılık göstermeyebilir. Bu analizden elde edilen bulgular, SMOTE'un bazı az temsil edilen sınıflarda model performansını artırmada etkili olduğunu, ancak bu etkinin tüm sınıflarda geçerli olmadığını göstermektedir. Özellikle DDoS gibi bazı sınıflarda performansın hiç artmaması, örnek sayısını artırmanın tek başına yeterli olmadığını ortaya koymaktadır. Buradan hareketle SMOTE'un etkinliği mutlak değil, koşulludur. Bu koşullar aşağıdaki faktörlerle ilişkilidir:

- Hedef sınıfın belirgin yapısal özelliklere sahip olması: Eğer bir saldırı türü belirgin özellikler taşıyorsa (örneğin XSS), SMOTE bu sınıfta öğrenmeyi kolaylaştırır.
- Sınıflar arası örtüşme düzeyi: Eğer bir sınıf diğer sınıflarla yüksek oranda benzerlik gösteriyorsa (örneğin DDoS), SMOTE tarafından üretilen sentetik örnekler modeli daha da şaşırtabilir.
- Modelin genelleme kapasitesi: MLP modeli belirli esneklik göstermiş olsa da, daha iyi sonuçlar için hiperparametre ayarları yapılması veya daha karmaşık modellerin tercih edilmesi gerekebilir.

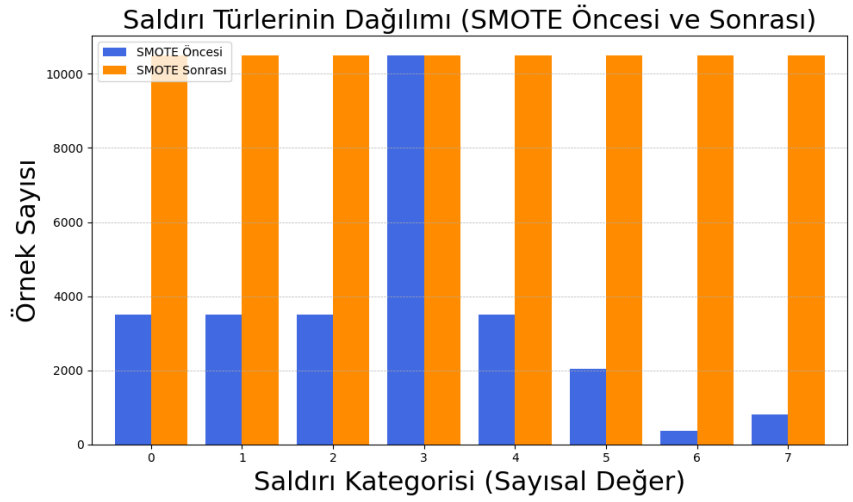
Sonuç olarak, SMOTE uygun koşullarda güçlü bir dengeleme aracı olabilir. Ancak, bu etkinin sağlanabilmesi için verinin doğasının, sınıflar arası ilişkilerin ve kullanılan modelin kapasitesinin iyi analiz edilmesi gerekir. SMOTE, yalnız başına bir çözüm değil, daha geniş ve bilinçli bir stratejinin parçası olarak değerlendirilmelidir.

5.2. Garage Door (Akıllı Garaj Kapısı Cihazı)

Akıllı Garaj Kapısı ise, saldırı türlerinin sayısal karşılıkları Tablo 5.2'de verilmiştir. SMOTE etkisi ise Şekil 5.6'da gösterilmektedir.

Tablo 5.2. IoT Garaj Kapısı saldırı kategorileri ve sayısal değerleri tablosu

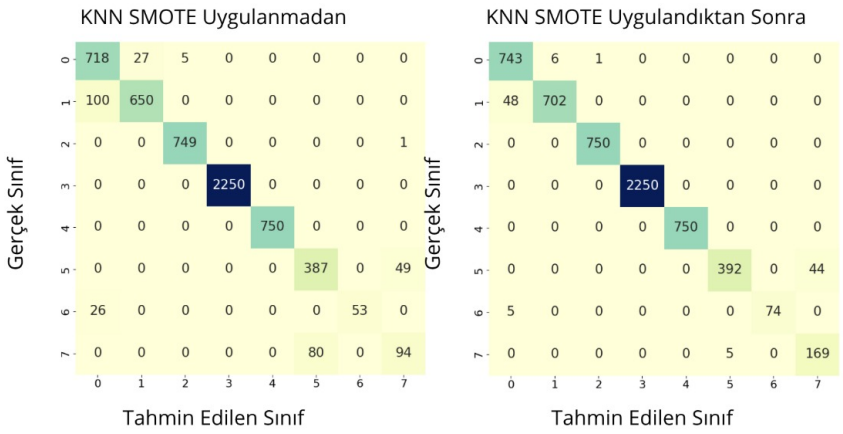
Saldırı Adı	Sayısal Değer
Backdoor	0
ddos	1
Injection	2
Normal	3
Password	4
ransomware	5
scanning	6
XSS	7



Şekil 5.6. IoT Akıllı Garaj Kapısı SMOTE Öncesi ve Sonrası Kategori Dağılımı

5.2.1. KNN Algoritması

KNN'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.7'de yer almaktadır.



Şekil 5.7. KNN - Garage Door için Confusion Matrix

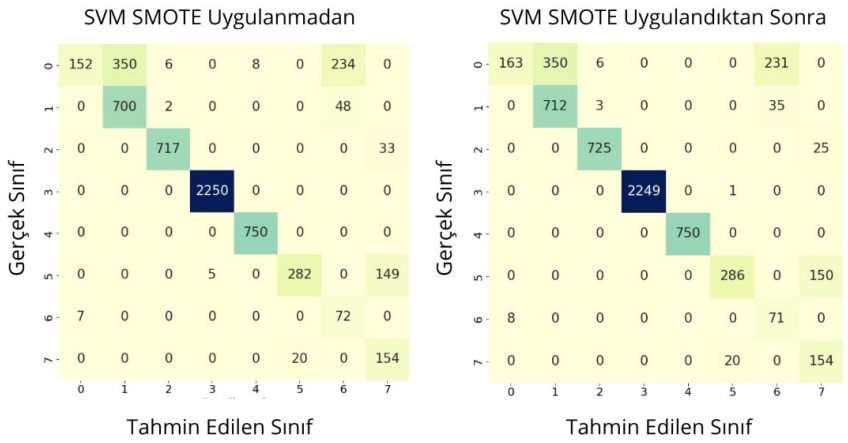
Garage Door cihazına ait veriler incelendiğinde, cihazın veri yapısının KNN algoritmasına belirli açılardan uygunluk gösterdiği, ancak sınıf dengesizliği nedeniyle bazı saldırı türlerinin düşük performans sergilediği gözlemlenmiştir.

Bu bağlamda SMOTE yöntemi uygulandığında, özellikle DDoS (1) ve XSS (7) gibi daha az temsil edilen sınıflarda önemli düzeyde doğruluk artışı sağlanmıştır. DDoS sınıfındaki doğru sınıflandırma sayısı 650'den 702'ye, XSS sınıfında ise 94'ten 169'a yükselmiştir. Bu artış, SMOTE'un sınıf sayısı az olan örnekleri başarıyla artırabildiğini ve KNN algoritmasının bu yeni örneklerden etkili şekilde öğrenebildiğini göstermektedir. Bununla birlikte, Injection (2) ve Password (4) gibi sınıflarda da hafif düzeyde iyileşmeler görülmüş, Normal (3) ve Password (4) gibi baskın sınıflarda ise modelin performansında belirgin bir düşüş yaşanmamıştır.

Bu durum, SMOTE yönteminin Garage Door cihazına ait veri yapısıyla uyumlu şekilde çalıştığını ve dengesizliği azaltırken mevcut doğruluğu olumsuz etkilemediğini ortaya koymaktadır. Sonuç olarak, Garage Door cihazının saldırı türlerine ait veri yapısı, KNN algoritmasının ve SMOTE yönteminin birlikte uygulanmasına elverişli bir yapı sergilemiş; özellikle düşük frekanslı sınıfların doğruluğunu artırmada etkili olurken, baskın sınıflarda herhangi bir performans kaybına neden olmamıştır. Bu da yöntemin sınırlı fakat hedefli bir katkı sunduğunu, zarar vermeden seçici şekilde performansı artırabileceğini göstermektedir.

5.2.2. SVM Algoritması

Şekil 5.8'de SVM sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



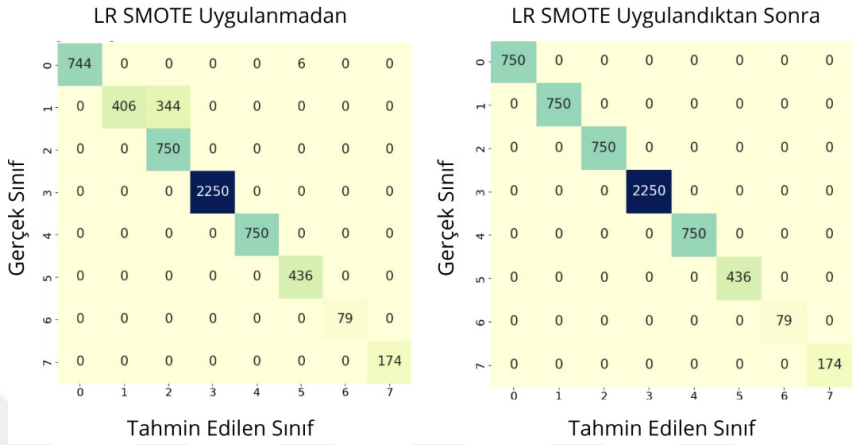
Şekil 5.8. SVM - Garage Door için Confusion Matrix

Garage Door cihazı üzerinde SVM algoritması için SMOTE uygulanmadan önceki ve sonraki karmaşıklık matrisleri karşılaştırıldığında, en belirgin iyileşmenin örnek sayısı düşük olan sınıflarda gerçekleştiği gözlemlenmiştir. Özellikle DDoS sınıfında doğru tahmin sayısı 700'den 712'ye, Injection sınıfında ise 717'den 725'e yükselmiştir; bu da SMOTE'un düşük temsile sahip sınıflarda pozitif katkı sağladığını göstermektedir. Öte yandan, önceden dengeli örnek sayılarına sahip olan Password ve XSS gibi sınıflarda hiçbir değişiklik gözlemlenmemiştir; bu durum, SMOTE'un modelin mevcut stabilitesini bozmadığını ortaya koymaktadır. Normal (2250'den 2249'a) ve Scanning (72'den 71'e) gibi bazı sınıflarda çok küçük gerilemeler meydana gelmiş olsa da, bu düşüşler genel performansı olumsuz etkileyecek düzeyde değildir.

Bu bağlamda SMOTE, sınırlı ama etkili bir iyileştirme aracı olarak değerlendirilebilir. Bu durum, SVM algoritmasının sınıf sınırlarına olan duyarlılığı ile doğrudan ilişkilidir; yani sentetik örneklerin veri uzayındaki konumlarının modele uygunluğu, SMOTE'un etkisini belirleyici bir faktördür. Sonuç olarak, nadir sınıfların performansını artırmak isteyen uygulamalarda, SMOTE'un dikkatli ve sınırlı kullanımı fayda sağlayabilir.

5.2.3. Lojistik Regresyon (LR) Algoritması

Lojistik Regresyon için SMOTE öncesi ve sonrası karmaşıklık matrisleri Şekil 5.9'da sunulmuştur.



Şekil 5.9. LR - Garage Door için Confusion Matrix

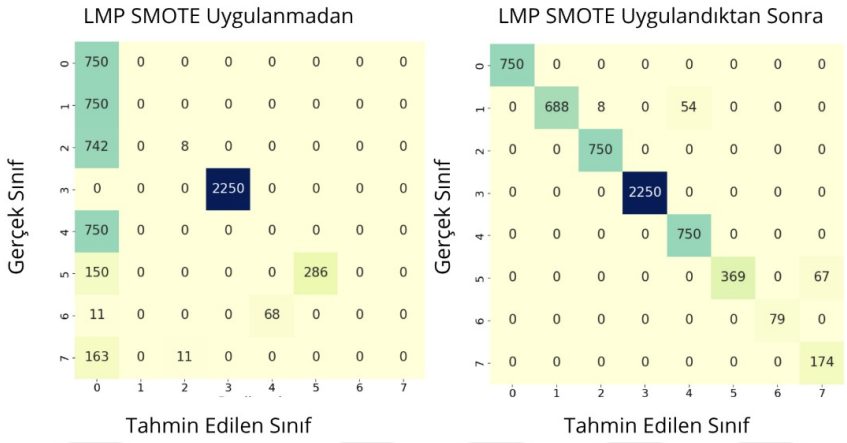
SMOTE uygulandıktan sonra, DDoS sınıfı (sınıf 1) belirgin bir iyileşme göstermiş, doğru sınıflandırma sayısı 406'ten 750'ye yükselmiştir. Bu gelişme, söz konusu saldırı türünün örüntülerinin daha net olması ve sınıflar arası ayrımın daha belirgin olmasından kaynaklanmakta olup, doğrusal sınırlara duyarlı olan LR algoritmasının bu ayrımı daha iyi öğrenmesini sağlamıştır.

Öte yandan, Backdoor sınıfı (sınıf 0) yalnızca 744'ten 750'ye çıkmış, yani çok sınırlı bir artış göstermiştir. Diğer altı sınıfta ise herhangi bir artış ya da azalma yaşanmamış, sonuçlar tamamen sabit kalmıştır. Bu durum, SMOTE'un tüm sınıflarda eşit düzeyde etkili olmadığını, sınıf sınırlarının belirginliği ve örnek dağılımının başarıyı doğrudan etkilediğini göstermektedir.

Bu bulgular, SMOTE'un mucizevi bir çözüm olmasa da modele zarar vermediğini ve belirli koşullarda fayda sağlayabileceğini ortaya koymaktadır. Genel olarak değerlendirildiğinde, LR algoritması ve Garaj Kapısı cihazı, SMOTE'a sınırlı ve seçici bir yanıt vermiştir; bu yanıt yalnızca sınıflandırılması daha kolay olan sınıflarda gözlemlenmiştir.

5.2.4. MLP Algoritması

MLP'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.10'da yer almaktadır.



Şekil 5.10. MLP - Garage Door için Confusion Matrix

SMOTE uygulanmadan önce MLP algoritması, ddos (1), injection (2), password (4), scanning (6) ve XSS (7) gibi düşük temsil edilen sınıflarda hiçbir doğru tahmin yapamamış, bu sınıfların her biri sıfır (0) doğru sınıflandırma ile temsil edilmiştir. SMOTE sonrası, injection (2) sınıfı 8'den 750'ye, password (4) ve ddos (1) sınıfları ise 0'dan 750 ve 0'dan 688 doğru tahmine yükselmiştir. XSS (7) için 0'dan 174'e, scanning (6) için ise 0'dan 79'a çıkış gözlemlenmiştir. ransomware (5) sınıfında ise 286'dan 369'a yükselerek orta düzeyde bir iyileşme sağlanmıştır. Buna karşın, backdoor (0) ve normal (3) gibi dengeli sınıflarda doğruluk oranları 750 ve 2250 olarak sabit kalmış, yani SMOTE bu sınıflarda herhangi bir bozulmaya neden olmamıştır.

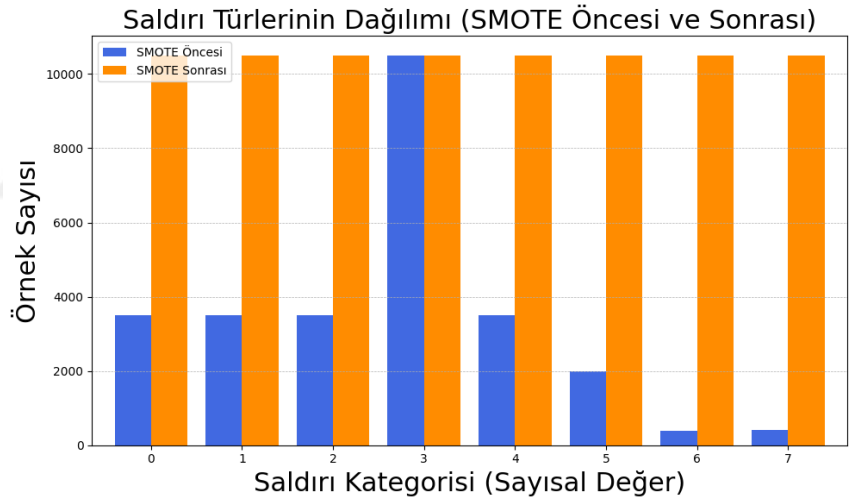
Bu sonuçlar, SMOTE'un yalnızca eksik sınıfları iyileştirdiğini, mevcut dengeleri bozmadığını göstermektedir. MLP'nin çok katmanlı ve doğrusal olmayan yapısı, üretilen sentetik örnekleri etkili biçimde öğrenmesini sağlamış ve bu nedenle SMOTE, MLP ile birlikte güçlü sonuçlar vermiştir. Ayrıca verisetinin ciddi dengesizlik içermesi, SMOTE'un katkısını daha belirgin hâle getirmiştir. Tüm bu bulgular, SMOTE'un özellikle MLP gibi derin öğrenmeye yatkın modellerle, çok sınıflı ve dengesiz veri kümelerinde güvenli ve etkili bir dengeleme yöntemi olduğunu ortaya koymaktadır.

5.3. GPS Tracker (Konum Takip Cihazı)

Saldırı türlerinin sayısal karşılıkları Tablo 5.3'te verilmiştir. SMOTE etkisi ise Şekil 5.11'de gösterilmektedir.

Tablo 5.3. IoT Konum Takip Cihazı saldırı kategorileri ve sayısal değerleri tablosu

Saldırı Adı	Sayısal Değer
Backdoor	0
ddos	1
Injection	2
Normal	3
Password	4
ransomware	5
scanning	6
XSS	7

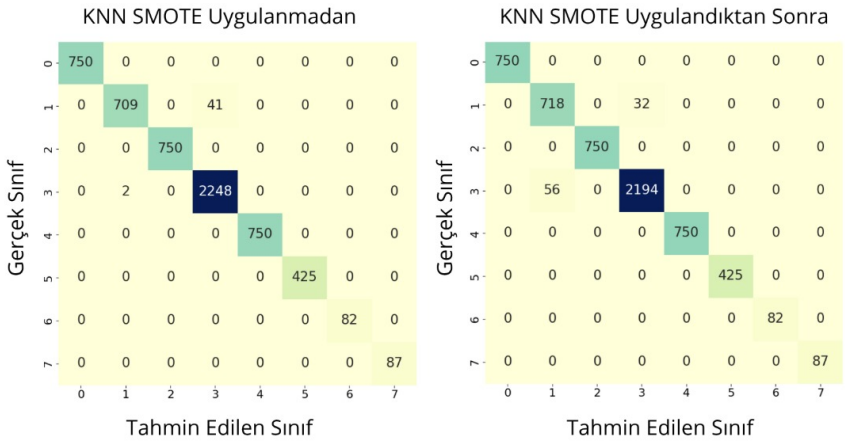


Şekil 5.11. IoT Konum Takip Cihazı SMOTE Öncesi ve Sonrası Kategori Dağılımı

5.3.1. KNN Algoritması

Şekil 5.12'de KNN sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir. SMOTE tekniği, dengesiz veri dağılımını dengelemek amacıyla uygulanmış olsa da, KNN algoritmasının GPS Tracker cihazı üzerindeki performansını kayda değer biçimde artırmamıştır. Yalnızca DDoS saldırısı (1) için doğru sınıflandırma sayısı 709'dan 718'e çıkmış; diğer sınıflarda ya değişiklik

olmamış ya da düşüş yaşanmıştır.



Şekil 5.12. KNN - GPS Tracker için Confusion Matrix

Örneğin, normal trafik (3) için doğru tahmin sayısı 2248'den 2194'e gerilemiştir. Bu sonuçlar, SMOTE tarafından üretilen sentetik örneklerin, KNN'in hassas olduğu özellik uzayında gerçek veri dağılımını yansıtamamasına bağlanabilir.

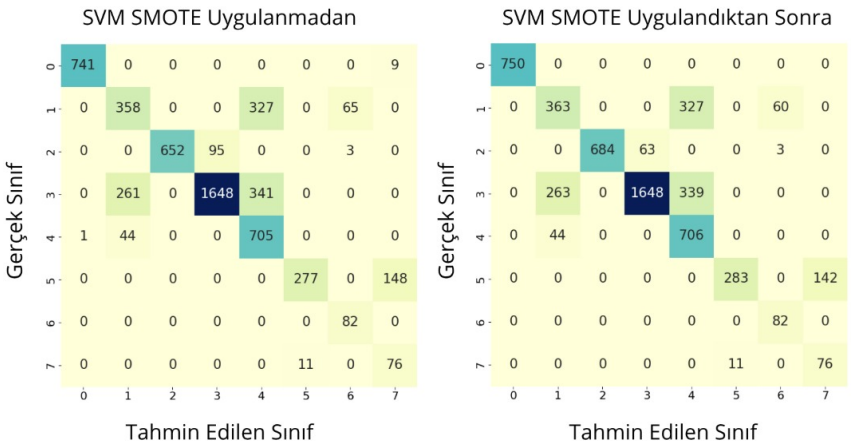
Ayrıca, sınıflar arası örtüşmenin yüksek olması ve çıkarılan özelliklerin ayrıştırıcı gücünün zayıf kalması, KNN'nin performansını sınırlandırmıştır. KNN'nin öğrenme yapısı olmadan sadece örnek benzerliğine dayanması, örnek sayısındaki artışın tek başına yeterli olmamasına neden olmuştur. Buna karşın, aynı yöntem farklı veri yapısına sahip Garage Door cihazında daha başarılı sonuçlar üretmiştir.

Bu durum, SMOTE ve KNN performansının kullanılan cihazın veri özelliklerine ve sınıf ayrımına doğrudan bağlı olduğunu göstermektedir.

5.3.2. SVM Algoritması

SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.13'te yer almaktadır.

Bu artış, sentetik örneklerin modelin bu sınıflardaki öğrenme kapasitesine katkı sunduğunu göstermektedir. Ayrıca Normal (3), Scanning (6) ve XSS (7) sınıflarında herhangi bir bozulma yaşanmamış, sonuçlar sabit kalmıştır.

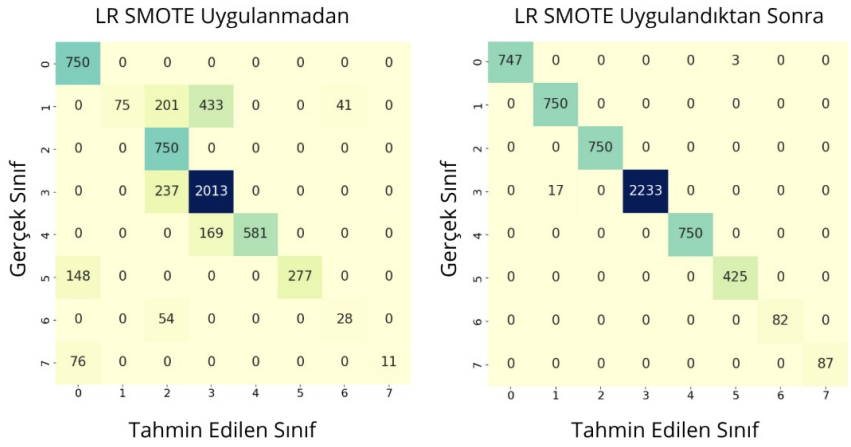


Şekil 5.13. SVM - GPS Tracker için Confusion Matrix

Bu durum, SMOTE'un genel performansa zarar vermediğini; aksine, bazı sınıflarda sınırlı fakat anlamlı kazanımlar sağlayabildiğini ortaya koymaktadır. Özellikle dengesiz veri setlerinde, bu tür yöntemlerin dikkatli şekilde uygulanması, belirli sınıflarda istikrarı koruyarak genel dengeyi iyileştirmeye katkı sunabilir.

5.3.3. Lojistik Regresyon (LR) Algoritması

Lojistik Regresyon için SMOTE öncesi ve sonrası karmaşıklık matrisleri Şekil 5.14'te sunulmuştur.



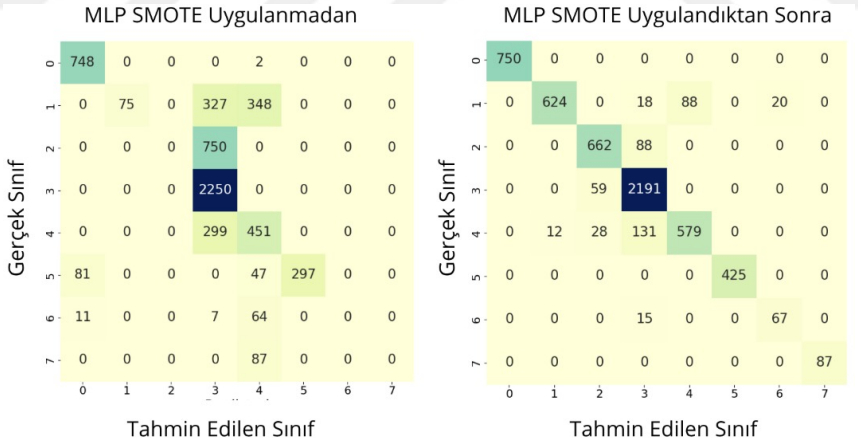
Şekil 5.14. LR - GPS Tracker için Confusion Matrix

Logistic Regression (LR) modeli, SMOTE uygulandıktan sonra çoğu saldırı sınıfında önemli performans artışları göstermiştir. Özellikle ddos (1) sınıfındaki doğru tahmin sayısı 75'ten 750'ye yükselerek dikkate değer bir gelişme sergilemiştir. Benzer şekilde

Password (4) sınıfı 581'den 750'ye, Normal (3) sınıfı ise 2013'ten 2233'e çıkmıştır. Ransomware (5), Scanning (6) ve XSS (7) sınıflarında da sırasıyla 277'den 425'e, 28'den 82'ye ve 11'den 87'ye yükselen tahmin başarıları gözlemlenmiştir. Injection (2) sınıfı sabit kalarak 750'de kalmış; bu da bu sınıfın önceden dengeli olduğunu göstermektedir. Sadece Backdoor (0) sınıfında 750'den 747'ye hafif bir düşüş yaşanmış, ancak bu fark ihmal edilebilir düzeydedir. Bu sonuçlar, SMOTE'un düşük temsil edilen sınıflarda öğrenmeyi artırdığını, dengeli sınıflarda ise performansı koruduğunu göstermektedir. Dolayısıyla, veri setinin yalnızca SMOTE ile uyumlu olmadığı, aynı zamanda kullanılan sınıflandırma algoritması olan Logistic Regression'ın da bu veri yapısıyla etkili biçimde çalıştığı görülmektedir. LR algoritmasının doğrusal yapısı, SMOTE ile üretilen sentetik örnekleri başarılı şekilde işleyerek modelin genel doğruluğuna katkı sağlamıştır. Bu da sınıflandırma algoritması ile veri yapısının SMOTE tekniğiyle uyumlu ve dengeli bir bütün oluşturduğunu ortaya koymaktadır.

5.3.4. MLP Algoritması

MLP'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.15'te yer almaktadır.



Şekil 5.15. MLP - GPS Tracker için Confusion Matrix

İlgili karışıklık matrisine göre, MLP algoritmasının SMOTE öncesi ve sonrası sınıf bazlı performansında dikkat çekici gelişmeler yaşanmıştır. Injection (2), Scanning (6) ve XSS (7) sınıflarında doğruluk sıfırken, SMOTE sonrası sırasıyla 662, 67 ve 87 doğru tahmine ulaşılmıştır. ddos (1) sınıfı 75'ten 624'e, Password (4) ve Ransomware (5) sınıfları ise belirgin oranlarda artış göstermiştir. Backdoor (0) sınıfında 2 birimlik küçük bir artış

gözlemlenirken, yalnızca Normal (3) sınıfında 59 birimlik düşüş olmuştur. Bu kayıp, genel kazanımlarla kıyaslandığında önemsizdir. Normal sınıfına SMOTE uygulanmamış, yalnızca diğer sınıfların örnek sayısı bu sınıfa eşitlenmiştir.

Bu durum, modelin eğitim sürecinde odağının değişmesine yol açmış; model, artık daha fazla temsil edilen azınlık sınıfları ayırt etmeye çalışırken, bazı normal örnekleri bu sınıflarla karıştırmaya başlamıştır. Yapılan analizler sonucunda, SMOTE tekniğinin MLP algoritmasıyla uygulandığında düşük temsil edilen sınıflarda belirgin bir performans artışı sağladığı gözlemlenmiştir.

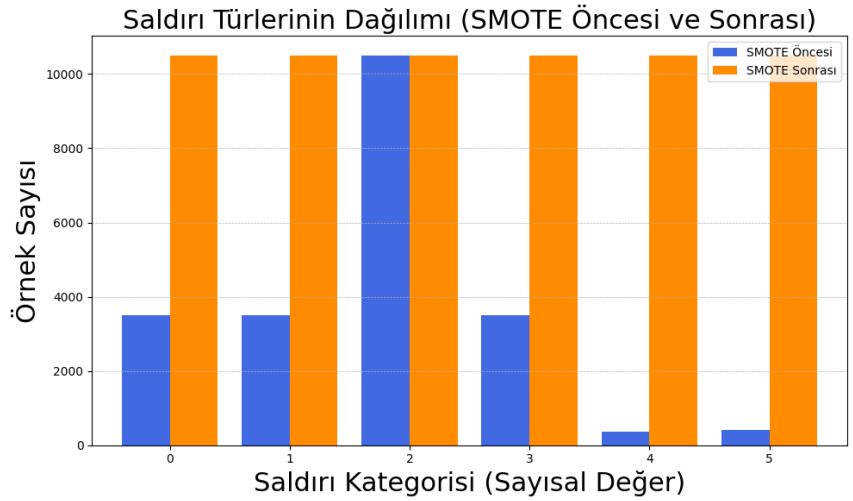
Injection, Scanning ve XSS gibi daha önce hiç doğru tahmin yapılamayan sınıflarda sıfırdan yüksek doğruluklara ulaşılması, bu tekniğin sınıflar arası dengesizliği gidermede ne kadar etkili olduğunu ortaya koymuştur. Bununla birlikte, Normal (3) gibi örnek sayısı zaten yeterli olan sınıflarda hafif performans düşüşleri yaşanmıştır. Bu durum, SMOTE'un yalnızca örnek sayısını değil, modelin öğrenme sürecindeki dikkat dağılımını da değiştirdiğini göstermektedir. Model, artık azınlık sınıfları daha iyi ayırt etmeye çalışırken, bazı baskın sınıflarda karışıklık yaşayabilmektedir. Ancak genel değerlendirmeye bakıldığında, elde edilen kazanımlar kayıplardan çok daha fazladır. Bu da, SMOTE'un dengesiz veri setlerinde dikkatli ve kontrollü bir şekilde uygulandığında sınıflandırma başarımını artıran etkili bir yöntem olduğunu göstermektedir.

5.4. Modbus (Endüstriyel Otomasyon Cihazı)

Saldırı türlerinin sayısal karşılıkları Tablo 5.4'te verilmiştir. SMOTE etkisi ise Şekil 5.16'da gösterilmektedir.

Tablo 5.4. IoT Modbus saldırı kategorileri ve sayısal değerleri

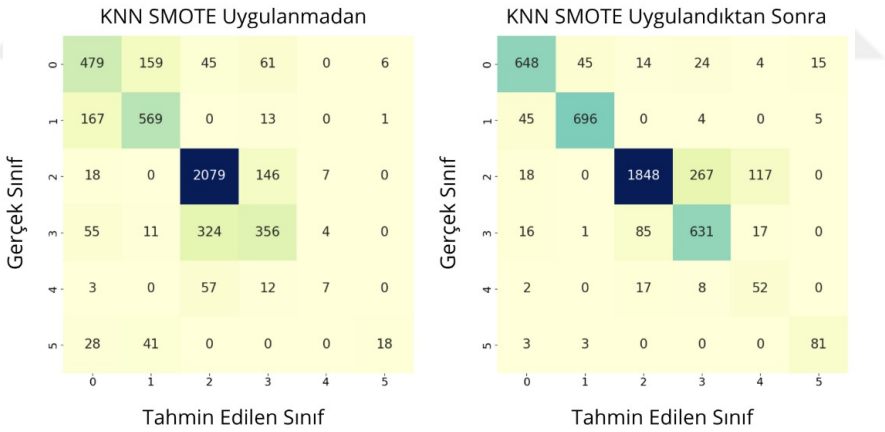
Saldırı Adı	Sayısal Değer
Backdoor	0
Injection	1
Normal	2
Password	3
Scanning	4
XSS	5



Şekil 5.16. IoT Modbus SMOTE Öncesi ve Sonrası Kategori Dağılımı

5.4.1. KNN Algoritması

Şekil 5.17'de KNN sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



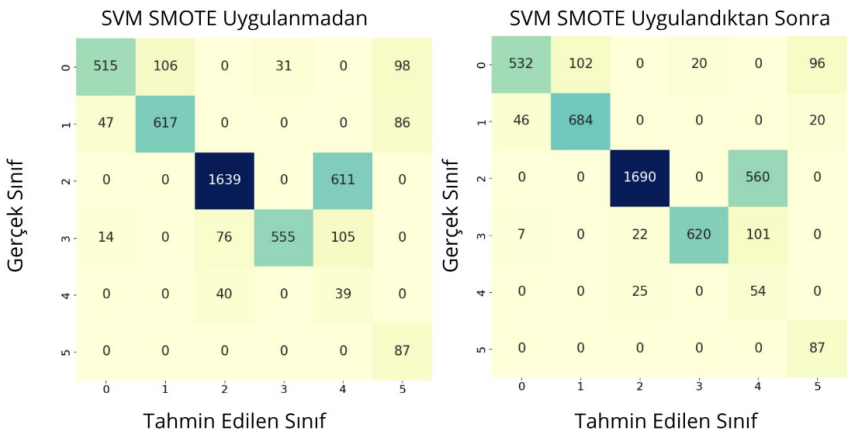
Şekil 5.17. KNN - Modbus için SMOTE Öncesi ve Sonrası Kategori Dağılımı

SMOTE tekniğinin uygulanması, KNN modelinin sınıflar arası denge açısından performansında belirgin bir etki göstermiştir. "Backdoor" (0) sınıfında doğru sınıflandırma sayısı 479'dan 648'e, "Injection" (1) sınıfında 569'dan 696'ya, "Password" (3) sınıfında 356'dan 631'e, "Scanning" (4) sınıfında 7'den 52'ye ve "XSS" (5) sınıfında 18'den 81'e yükselmiştir. Bu artışlar, özellikle nadir görülen sınıflarda önemli bir iyileşmeyi yansıtmaktadır. Ancak, "Normal" (2) sınıfında doğru tahmin sayısı 2079'dan

1848'e düşerek dikkat çekici bir azalma gözlemlenmiştir. Bu sınıfın SMOTE öncesinde yüksek doğruluk göstermesinin temel nedeni, veri kümesinde en fazla örneğe sahip olmasıdır; bu da KNN algoritmasının komşuluk temelli yapısı gereği modeli bu sınıfa doğru yönlendirmiştir. "Normal" (2) sınıfında gözlemlenen düşüşün temel nedeni, SMOTE algoritmasının bu sınıfa yeni örnekler eklememiş olmasıdır; çünkü bu sınıf zaten en baskın sınıftı. Buna karşılık, az temsil edilen diğer sınıflara sentetik örnekler eklenmiş ve böylece sınıflar arası denge sağlanmıştır. Bu durum, KNN modelinin önceden sahip olduğu baskın sınıfa yönelik önyargısını azaltarak tahminlerin daha dengeli dağılmasına yol açmıştır. "Normal" sınıfındaki doğruluk oranındaki düşüş, göreceli olarak yüksek kalmaya devam ettiği için kabul edilebilir düzeydedir. Bu düşüş, genel model adaletinin ve sınıflar arası performans dengesinin artmasıyla telafi edilmiştir. SMOTE'un KNN algoritmasıyla uyumu da dikkate değerdir. SMOTE, yeni örnekleri aynı sınıfın özellik uzayında oluşturarak KNN'in örnekler arasındaki mesafeye dayalı sınıflandırma mantığıyla tutarlılık göstermektedir. Dolayısıyla, bu bağlamda SMOTE'un uygulanması rastlantısal değil, aksine az temsil edilen sınıfların model tarafından daha iyi öğrenilmesini sağlayan, etkili ve yapısal olarak uygun bir strateji olarak değerlendirilmelidir.

5.4.2. SVM Algoritması

Şekil 5.18'de SVM sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



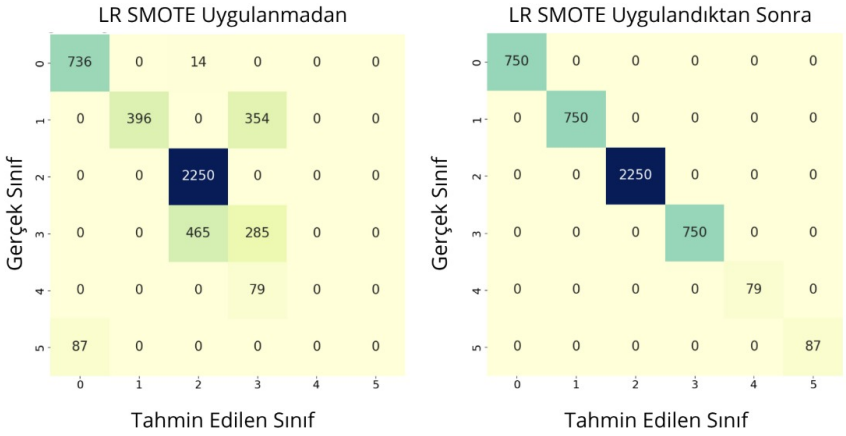
Şekil 5.18. KSVM - Modbus için SMOTE Öncesi ve Sonrası Kategori Dağılımı

SMOTE tekniğinin SVM algoritmasıyla birlikte uygulanması, modelin genel performansında dengeli ve istikrarlı bir iyileşme sağlamıştır. Özellikle "Injection" (1) sınıfında doğru tahmin sayısı 617'den 684'e, "Password" (3) sınıfında 555'ten 620'ye, "Backdoor" (0) sınıfında 515'ten 532'ye ve "Normal" (2) sınıfında 1639'dan 1690'a yükselmiştir. "Scanning" (4) sınıfında ise 39'dan 54'e çıkış gözlemlenmiş, "XSS" (5) sınıfındaki doğruluk ise 87 seviyesinde sabit kalmıştır. Dikkat çekici şekilde, veri kümesinde baskın olan "Normal" (2) sınıfının performansı KNN'de olduğu gibi azalmamış, aksine belirgin olmayan ancak olumlu bir artış göstermiştir. Bu durum, örnek yoğunluğuna değil, sınıflar arası optimal ayrım sınırlarının oluşturulmasına dayanan SVM'in yapısal avantajı ile açıklanabilir.

Her ne kadar SMOTE'un SVM üzerindeki etkisi, KNN ile karşılaştırıldığında daha sınırlı düzeyde kalmışsa da, özellikle az temsil edilen sınıflarda modelin duyarlılığını artırmış ve genel tahmin dengesini iyileştirmiştir. Bu da SMOTE'un, örnek dağılımındaki dengesizlikleri hafifleterek modelin daha tutarlı sonuçlar üretmesine katkı sağladığını göstermektedir. Bu bağlamda, SMOTE'un model davranışında gözle görülür yapısal bozulmalara yol açmadan olumlu etkiler yarattığı ve sınıflar arası adaleti desteklediği söylenebilir.

5.4.3. LR Algoritması

Lojistik Regresyon için SMOTE öncesi ve sonrası karmaşıklık matrisleri Şekil 5.19'da sunulmuştur.



Şekil 5.19. LR - Modbus için SMOTE Öncesi ve Sonrası Kategori Dağılımı

Lojistik Regresyon (LR) algoritmasının, SMOTE uygulanmadan önce ve sonra Modbus cihazı üzerindeki sınıflandırma performansı incelendiğinde, özellikle dengesiz sınıflarda olmak üzere tüm kategorilerde kayda değer ve etkili bir iyileşme sağladığı görülmüştür. Örneğin, Backdoor (0) saldırı türünde doğru sınıflandırılan örnek sayısı 736'dan 750'ye yükselmiş; Injection (1) sınıfında ise bu sayı 396'dan 750'ye çıkarak tam doğruluk elde edilmiştir. Ayrıca, düşük temsile sahip sınıflardan biri olan Password (3) için de doğru sınıflandırma sayısı 285'ten 750'ye ulaşmış ve bu da ciddi bir gelişmeyi göstermektedir.

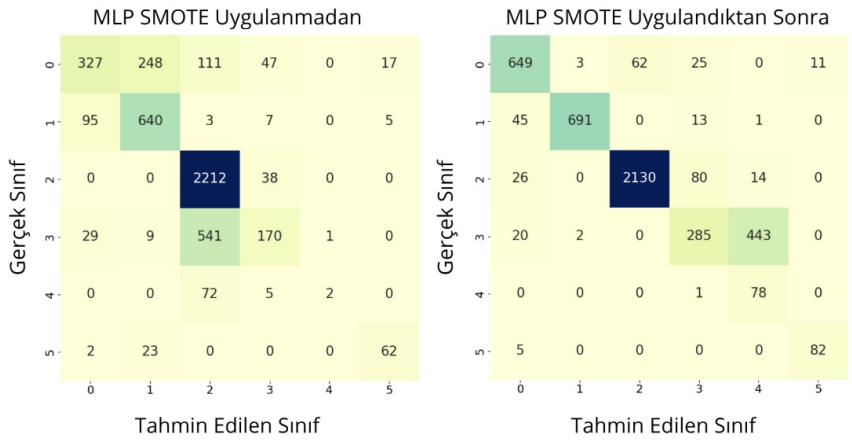
Daha da çarpıcı olan ise, SMOTE uygulanmadan önce hiçbir örneği doğru şekilde sınıflandırılamayan Scanning (4) ve XSS (5) sınıflarında (sırasıyla 0/79 ve 0/87), SMOTE uygulandıktan sonra tüm örneklerin %100 doğrulukla doğru şekilde tahmin edilmesidir. Veri setinde en fazla örneğe sahip olan Normal (2) sınıfı ise hem SMOTE öncesinde hem de sonrasında mükemmel sınıflandırma performansını korumuş (2250/2250), bu da SMOTE'un yoğun örnek içeren dengeli sınıflarda modelin performansını olumsuz etkilemediğini ortaya koymuştur. Bu önemli gelişmeler, doğrusal karar sınırlarına dayanan LR algoritmasının sınıf dengesizliğine olan duyarlılığıyla açıklanabilir.

SMOTE tekniğiyle elde edilen dengeli veri dağılımı sayesinde, model daha net ve doğru karar sınırları çizebilmiş, özellikle azınlıkta kalan sınıfları daha isabetli şekilde ayırt edebilmiştir. Buna karşın, SVM ve KNN gibi algoritmalar bu ölçüde bir iyileşme sergileyememiştir. Bunun nedeni, bu algoritmaların doğrusal olmayan sınıflandırma yapıları veya mesafeye dayalı karar mekanizmaları nedeniyle, sentetik örneklerden LR kadar etkin şekilde faydalanamaması olabilir.

Sonuç olarak, LR algoritmasının SMOTE sonrası gösterdiği bu dikkat çekici performans artışı, yalnızca SMOTE'un başarısını değil, aynı zamanda modelin yapısıyla verinin dengelenmiş dağılımı arasındaki uyumun gücünü de göstermektedir. Aynı düzeyde bir uyum, diğer algoritmalar için elde edilememiştir.

5.4.4. MLP Algoritması

MLP'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.20'de yer almaktadır.



Şekil 5.20. MLP - Modbus için SMOTE Öncesi ve Sonrası Kategori Dağılımı

Modbus cihazına ait veriler üzerinde uygulanan MLP (Multilayer Perceptron) algoritmasının, SMOTE tekniği öncesi ve sonrası performansını değerlendirdiğimizde, SMOTE'un özellikle az örnek içeren sınıflarda belirgin bir iyileşme sağladığı açıkça görülmektedir. Özellikle, Scanning (4) gibi sadece iki doğru sınıflamaya sahip olan bir sınıfın, SMOTE sonrası 78 doğru sınıflamaya ulaşması, modelin bu nadir örüntüleri tanıma kapasitesinin ciddi oranda arttığını göstermektedir. Benzer şekilde, Backdoor (0), Password (3) ve XSS (5) gibi az temsil edilen sınıflarda da doğru sınıflama sayılarında dikkate değer artışlar gözlemlenmiştir. Örneğin, Password (3) sınıfındaki doğru sınıflamalar 170'ten 285'e yükselmiş ve bu yaklaşık %67 oranında bir artışı temsil etmektedir. Bu tür bir artış rastlantısal dalgalanmalarla açıklanamayacak düzeydedir ve SMOTE'un sağladığı sentetik örneklerin modelin bu sınıfları daha iyi öğrenmesini sağladığına işaret etmektedir. Öte yandan, Normal (2) sınıfında doğru sınıflama sayısı 2212'den 2130'a gerilemiştir. Bu düşüş, sentetik örneklerin dağılımı değiştirmesi sonucu modelin bazı durumlarda normal ve saldırı örneklerini ayırt etme performansının kısmen etkilenmiş olabileceğini göstermektedir. Ancak bu düşüş, genel iyileşmenin yanında oldukça sınırlı kalmakta ve kabul edilebilir bir dengeleme bedeli olarak değerlendirilebilir. Bu bağlamda, SMOTE tekniği, MLP algoritması ile birlikte kullanıldığında, özellikle Modbus cihazına ait dengesiz veri kümelerinde anlamlı bir katkı sağlamıştır. Temsili düşük sınıflarda başarıyı artırmış ve modelin genel genelleme yetisini güçlendirmiştir. Ancak, bu analiz sadece Modbus cihazına özeldir ve farklı cihazlara genellenmeden önce ek doğrulama çalışmaları yapılması gerekmektedir. Bu

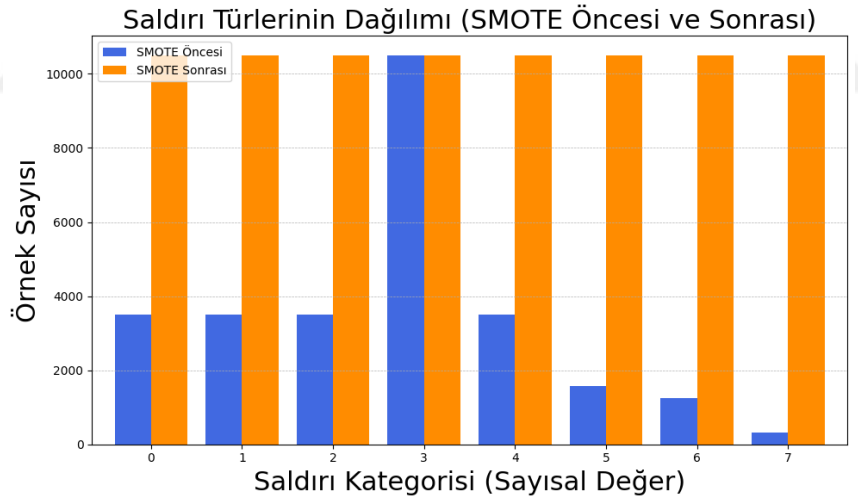
durum, SMOTE'un sınıf dengesizliği gibi problemler karşısında dikkatli ve kontrollü bir şekilde kullanılmasının önemini bir kez daha ortaya koymaktadır.

5.5. Motion Light (Hareket Işığı Cihazı)

Saldırı türlerinin sayısal karşılıkları Tablo 5.5'te verilmiştir. SMOTE etkisi ise Şekil 5.21'de gösterilmektedir.

Tablo 5.5. IoT Hareket Sensörlü Isık saldırı kategorileri ve sayısal değerleri tablosu

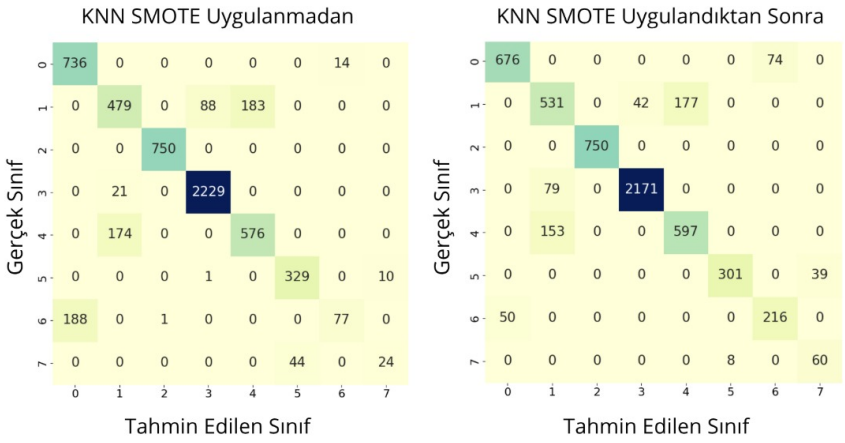
Saldırı Adı	Sayısal Değer
Backdoor	0
ddos	1
Injection	2
Normal	3
Password	4
ransomware	5
scanning	6
XSS	7



Şekil 5.21. IoT Hareket Işığı SMOTE Öncesi ve Sonrası Kategori Dağılımı

5.5.1. KNN Algoritması

KNN'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.22'de yer almaktadır.



Şekil 5.22. KNN - Motion Light için Confusion Matrix

Motion Light cihazına ait veriler üzerinde uygulanan KNN (K-Nearest Neighbors) algoritmasının, SMOTE tekniği öncesi ve sonrası sınıflandırma performansı karşılaştırıldığında, SMOTE'un özellikle nadir sınıflar üzerindeki etkisinin son derece belirgin olduğu görülmektedir. Özellikle, yalnızca 77 doğru sınıflamaya sahip olan Scanning (6) sınıfı, SMOTE sonrası 216 doğru sınıflamaya ulaşarak yaklaşık %180,5 oranında bir artış göstermiştir. Benzer şekilde, XSS (7) sınıfında da doğru sınıflama sayısı 24'ten 60'a çıkarak %150 oranında bir iyileşme elde edilmiştir. Bu dramatik artışlar, SMOTE'un düşük temsil edilen örnekleri yapay olarak çoğaltarak modelin bu örüntüleri öğrenmesini kayda değer şekilde geliştirdiğini ortaya koymaktadır.

Daha az belirgin olmakla birlikte, DDoS (1) ve Password (4) gibi sınıflarda da pozitif yönde değişim gözlenmiştir. DDoS sınıfında doğru sınıflama sayısı 479'dan 531'e yükselerek %10,85 oranında artmış, Password sınıfında ise %3,65 oranında daha fazla doğru tahmin elde edilmiştir. Injection (2) sınıfında ise SMOTE uygulaması sonrasında herhangi bir değişim gözlemlenmemiş, doğru sınıflama sayısı sabit kalmıştır (750).

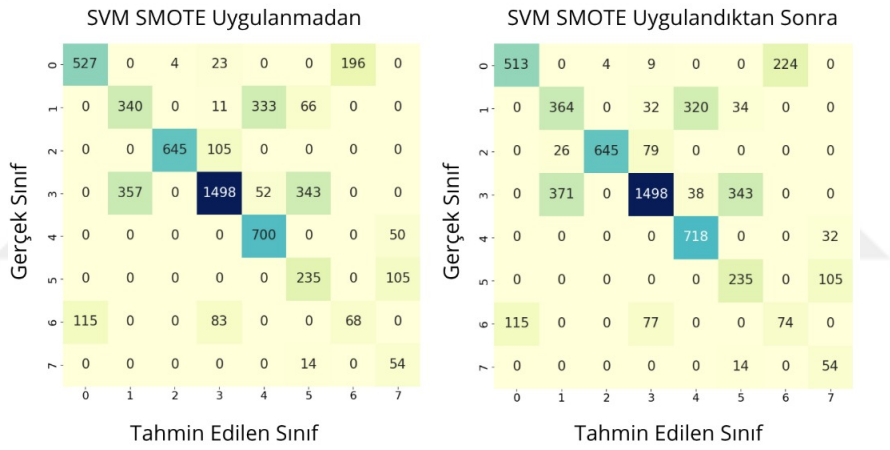
Öte yandan, bazı büyük örnekli sınıflarda sınırlı düzeyde performans kayıpları meydana gelmiştir. Normal (3) sınıfındaki doğru sınıflama sayısı 2229'dan 2171'e düşerek yaklaşık %2,6 oranında bir azalma göstermiştir. Benzer şekilde, Backdoor (0) sınıfında %8,15'lik, Ransomware (5) sınıfında ise %8,51'lik düşüşler yaşanmıştır. Bu düşüşler, SMOTE'un veri dağılımını yeniden şekillendirmesiyle modelin bazı normal örnekleri saldırı olarak yanlış etiketlemesine yol açabileceğini göstermektedir. Ancak bu düşüşlerin oran olarak

sınırlı kalması ve özellikle nadir sınıflarda sağlanan yüksek oranlı iyileşmeler dikkate alındığında, genel model performansının olumlu yönde etkilendiği söylenebilir.

Sonuç olarak, SMOTE tekniği, Motion Light cihazına özel olarak, KNN algoritması ile birlikte kullanıldığında sınıf dengesizliği problemini azaltmış ve nadir örneklerdeki model başarımını kayda değer şekilde iyileştirmiştir. Ancak her cihaz ve model için aynı etkinin garantili olmadığı unutulmamalı; bu nedenle genelleme yapılmadan önce her duruma özel değerlendirme yapılması gereklidir.

5.5.2. SVM Algoritması

Şekil 5.23'te SVM sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



Şekil 5.23. SVM - Motion Light için Confusion Matrix

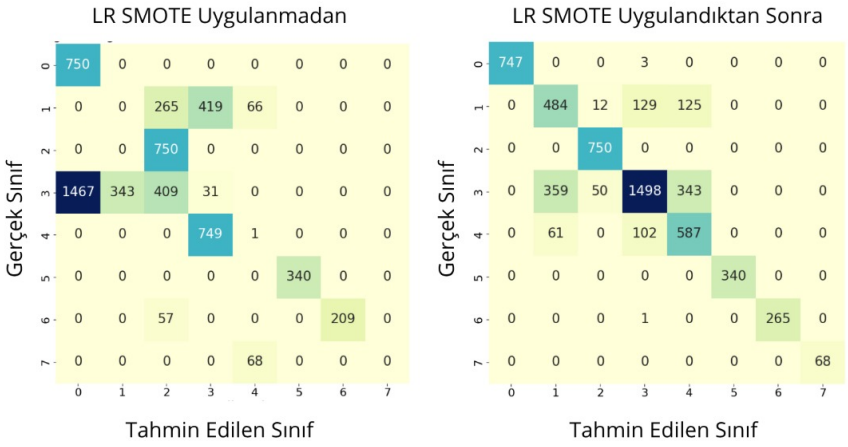
Motion Light cihazına ait veriler üzerinde uygulanan SVM (Support Vector Machine) algoritmasının, SMOTE öncesi ve sonrası sınıflandırma performansı incelendiğinde, elde edilen sonuçların genel olarak sınırlı düzeyde bir değişime işaret ettiği görülmektedir. SMOTE uygulaması sonrasında bazı sınıflarda sayısal artışlar gözlemlenmiş olsa da, bu artışlar çoğunlukla düşük oranlarda gerçekleşmiş ve modelin genel başarımını anlamlı düzeyde etkilememiştir. Örneğin, DDoS (1) saldırı kategorisinde doğru sınıflama sayısı 340'tan 364'e çıkarak %7,06 oranında bir iyileşme sağlamış, Password (4) sınıfında ise %2,57'lik bir artış kaydedilmiştir (700 $\rightarrow$ 718). Benzer şekilde, Scanning (6) sınıfında %8,82'lik bir iyileşme gözlemlenmiştir. Ancak bu artışlar sınırlı düzeyde kalmış ve

modelin genel doğruluğuna etkisi marjinal olmuştur. Injection (2), Normal (3), Ransomware (5) ve XSS (7) gibi sınıflarda doğru sınıflama sayısı sabit kalmış, yani SMOTE uygulamasının bu kategorilerde etkisiz olduğu görülmüştür. Öte yandan yalnızca Backdoor (0) sınıfında %2,66'lık hafif bir düşüş yaşanmış (527 → 513), bu da SMOTE'un model performansına zarar vermediğini ancak belirgin bir katkı da sunmadığını ortaya koymuştur. Bununla birlikte, genel değerlendirme yapıldığında, SMOTE tekniğinin bu çalışma kapsamında özellikle Motion Light cihazına ait ve belirli bir zaman aralığını kapsayan veri kümesi üzerinde sınırlı bir etki yarattığı sonucuna varılmaktadır. Elde edilen bulgular, yöntemin model performansını anlamlı ölçüde artırmadığı gibi, dikkate değer bir olumsuzluğa da yol açmadığını göstermektedir.

Bu durum, SMOTE'un her koşulda etkili olacağına dair genellemelerin geçerliliğini sorgulamakta ve yöntemin her cihaz ve veri yapısı özelinde dikkatle test edilerek uygulanması gerektiğini ortaya koymaktadır. Nitekim diğer cihazlar üzerinde ve farklı zamansal örneklemlerle gerçekleştirilen analizlerde, SMOTE uygulamasının çok daha belirgin ve anlamlı iyileştirmeler sağladığı gözlemlenmiştir. Bu bağlamda, Motion Light verileri üzerinde yapılan bu uygulamanın katkısı sınırlı düzeyde kalmış ve yöntemin bu özgül senaryodaki etkisi marjinal olarak değerlendirilmiştir.

5.5.3. LR Algoritması

Lojistik Regresyon için SMOTE öncesi ve sonrası karmaşıklık matrisleri Şekil 5.24'te sunulmuştur.



Şekil 5.24. LR - Motion Light için Confusion Matrix

Önceki çalışmalarda kullanılan SVM Algoritması ile elde edilen sonuçlar, sınıf dengesizliğini gidermede sınırlı bir etki göstermiştir. Ancak Motion Light (Hareket Işığı Cihazı) veri seti üzerinde gerçekleştirilen bu çalışmada, SMOTE algoritmasının belirgin ve etkili bir katkı sağladığı gözlemlenmiştir. Özellikle sınıf 1'in örnek sayısı 0'dan 484'e çıkarılmış; bu sayede daha önce tamamen göz ardı edilen bir sınıf, veri setinde anlamlı bir şekilde temsil edilmiştir.

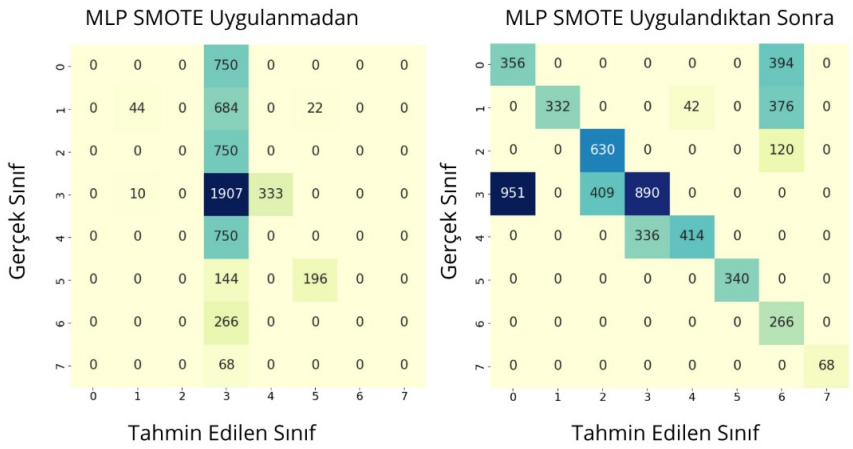
Benzer şekilde, sınıf 3'ün örnek sayısı 31'den 1498'e çıkarılarak %4700'ün üzerinde bir artış sağlanmış ve sınıf 4 için bu oran 1'den 587'ye ulaşmıştır. Bu, algoritmanın nadir sınıflara duyarlı ve kontrollü bir şekilde sentetik örnekler üretebildiğini göstermektedir. Ayrıca, sınıf 6'daki artışın 209'dan 265'e yükselmesi dengeli bir iyileşmeyi yansıtırken, başlangıçta hiç temsil edilmeyen sınıf 7 için 68 yeni örnek oluşturulması, SMOTE'un az temsil edilen sınıfları etkin biçimde dikkate aldığını göstermektedir.

Diğer yandan, yeterli temsile sahip olan sınıf 2 (750 örnek) ve sınıf 5 (340 örnek) sabit bırakılmış, bu da algoritmanın aşırı üretimden kaçındığını ve veri dengesini bozmadığını göstermektedir. Sınıf 0'da gözlemlenen 750'den 747'ye küçük bir azalma ise muhtemelen ön işleme adımlarından kaynaklanmaktadır ve SMOTE'un doğrudan bir etkisi değildir.

Sonuç olarak, Motion Light veri setinde uygulanan bu deney, SMOTE algoritmasının sınıf dengesizliğini düzeltmede etkili bir araç olduğunu ortaya koymaktadır. Bu iyileşmenin temelinde, SMOTE'un güçlü yapısının yanı sıra Logistic Regression (LR) algoritması ile olan uyumu yatmaktadır. Bu birliktelik, veri dengesini sağlarken modelin başarımını da artırmakta; bu bağlamda, SMOTE ve LR algoritmalarının birlikte kullanımı, nadir sınıfların temsili açısından önemli avantajlar sunmaktadır.

5.5.4. MLP Algoritması

MLP'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.25'te yer almaktadır.



Şekil 5.25. MLP - Motion Light için Confusion Matrix

Motion Light (Hareket Işığı Cihazı) veri seti üzerinde SMOTE tekniğinin uygulanmasından önce ve sonra MLP algoritması ile elde edilen sonuçlar, sınıf dengesizliğinin model performansı üzerindeki doğrudan etkisini açıkça ortaya koymuştur. Başlangıç durumunda, Normal (Sınıf 3) sınıfı 1907 örnekle aşırı temsil edilirken, Backdoor (Sınıf 0), Injection (Sınıf 2), Password (Sınıf 4), Scanning (Sınıf 6) ve XSS (Sınıf 7) gibi sınıfların hiçbir örneği bulunmamaktaydı (0 örnek).

Bu yapısal dengesizlik, modelin büyük oranda baskın sınıfa yönelmesine neden olmuş; dolayısıyla nadir saldırı türleri neredeyse tamamen göz ardı edilmiştir. Bu durum, dengesiz veri kümeleriyle çalışırken MLP gibi sinir ağı tabanlı modellerin temel bir zafiyetini gözler önüne sermektedir.

SMOTE uygulamasının ardından, temsil edilmeyen sınıflar için sentetik örnekler oluşturulmuş ve veri setine entegre edilmiştir: Backdoor (Sınıf 0) için 356, Injection (Sınıf 2) için 630, Password (Sınıf 4) için 414, Scanning (Sınıf 6) için 266 ve XSS (Sınıf 7) için 68 örnek üretilmiştir. Ayrıca, düşük temsile sahip olan DDoS (Sınıf 1) ve Ransomware (Sınıf 5) sınıflarında da önemli artışlar görülmüştür (sırasıyla 44'ten 332'ye ve 196'dan 340'a).

Bu yeniden dengeleme süreci, MLP algoritmasının yalnızca baskın sınıfa değil, diğer sınıflara da öğrenme kapasitesini artırmış ve modelin ayırt edici gücünü gözle görülür şekilde yükseltmiştir.

Ancak bu sonuçlar değerlendirilirken, MLP algoritmasının doğası da dikkate alınmalıdır. Çok katmanlı algılayıcılar (MLP), verideki dağılıma karşı oldukça hassastır ve içsel olarak sınıf dengesizliğini telafi etme mekanizmalarına sahip değildir. Bu nedenle, SMOTE'un sağladığı gelişme yalnızca tekniğin gücünü değil, aynı zamanda modelin nadir sınıfları öğrenmedeki sınırlılığını da ortaya koymaktadır.

Bununla birlikte, SMOTE'un oluşturduğu örneklerin gerçek veri dağılımlarını tam olarak yansıtmayabileceği unutulmamalıdır.

Özellikle siber güvenlik gibi hassas alanlarda, yapay olarak üretilen verilerin gerçek sistem davranışlarını ne ölçüde temsil ettiği dikkatle değerlendirilmelidir. Dolayısıyla, elde edilen gelişme bağlamsal olarak yorumlanmalı ve modelin performansı yalnızca bu sayısal iyileşmeye indirgenmemelidir.

Sonuç olarak, MLP algoritması dengeli bir ortamda önemli performans kazanımları elde edebilse de, orijinal verinin dengesiz yapısında ciddi sınırlamalar göstermiştir.

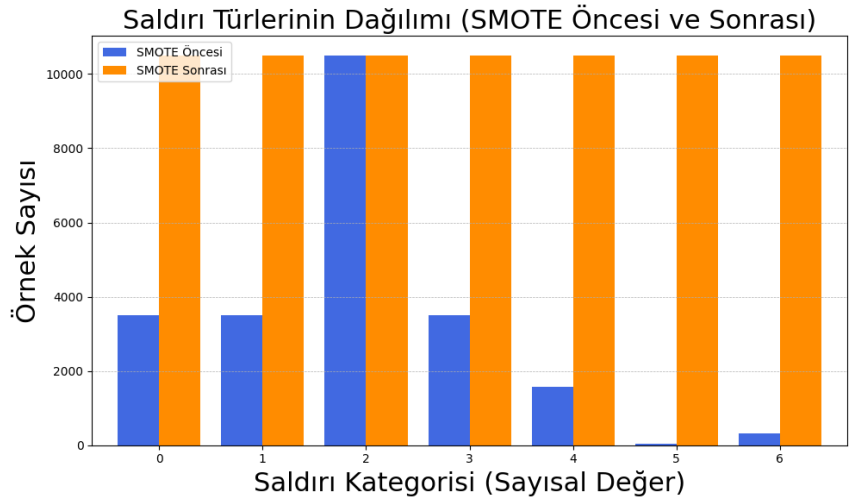
Bu durum, SMOTE'un yalnızca bir dengeleme aracı değil, aynı zamanda modelin kırılganlıklarını açığa çıkaran bir analiz aracı olarak da değer taşıdığını göstermektedir. İlerleyen çalışmalarda, daha dirençli modellerin tercih edilmesi ya da MLP'nin ağırlıklandırma ve örnekleme gibi tekniklerle desteklenmesi önerilmektedir.

5.6. Thermostat (Akıllı Termostat Cihazı)

Saldırı türlerinin sayısal karşılıkları Tablo 5.6'da verilmiştir. SMOTE etkisi ise Şekil 5.26'da gösterilmektedir.

Tablo 5.6. IoT Termostat saldırı kategorileri ve sayısal değerleri tablosu

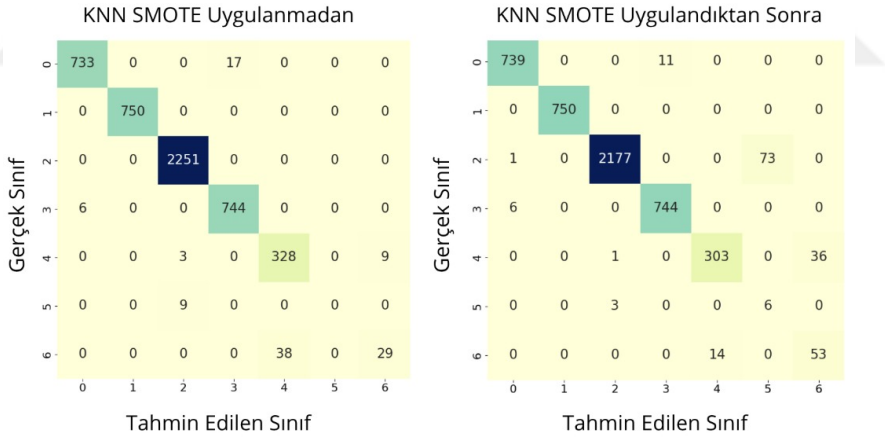
Saldırı Adı	**Sayısal Değer**
Backdoor	0
Injection	1
Normal	2
Password	3
ransomware	4
scanning	5
XSS	6



Şekil 5.26. IoT Akıllı Termostat Cihazı SMOTE Öncesi ve Sonrası Kategori Dağılımı

5.6.1. KNN Algoritması

Şekil 5.27'de KNN sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



Şekil 5.27. KNN - Thermostat için Confusion Matrix

KNN algoritmasının Thermostat (Akıllı Termostat Cihazı) veri setindeki performansı, SMOTE uygulanmadan önce ve sonra elde edilen doğru sınıflandırma sayıları incelenerek değerlendirilmiştir. Sonuçlar, SMOTE'un özellikle düşük temsile sahip sınıflarda dikkate değer iyileştirmeler sağladığını ortaya koymaktadır. En belirgin artış, XSS (Sınıf 6) kategorisinde gerçekleşmiş olup doğru tahmin sayısı 29'dan 53'e çıkarak +24 (%82,8) gibi dikkat çekici bir yükseliş göstermiştir. Benzer şekilde, scanning (Sınıf

5) sınıfında SMOTE öncesinde hiç doğru tahmin bulunmazken, sonrasında mevcut 6 örneğin tamamı doğru sınıflandırılmıştır. Bu durum, SMOTE'un nadir görülen saldırı türlerinin karakteristik özelliklerini modele etkili bir şekilde kazandırdığını göstermektedir.

Orta düzeyde bir iyileşme Backdoor (Sınıf 0) kategorisinde gözlenmiş, doğru tahmin sayısı 733'ten 739'a çıkarak +6 (%0,8) artış sağlamıştır. Password (Sınıf 3) sınıfında ise performans sabit kalmış, doğru tahmin sayısı 744'te değişmemiştir. Bu sabitlik, SMOTE'un mevcut güçlü sınıflar üzerindeki başarımı koruduğunu, onları olumsuz etkilemediğini ortaya koymaktadır.

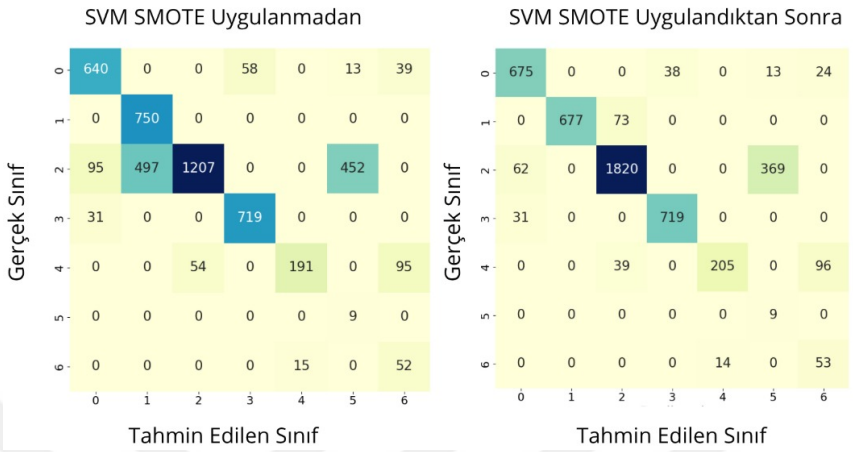
Buna karşın, Normal (Sınıf 2) sınıfında doğru tahmin sayısı 2251'den 2177'ye gerileyerek -74 (%-3,3) düşüş yaşanmıştır. ransomware (Sınıf 4) sınıfında da benzer şekilde 328'den 303'e düşüş görülmüştür (-25, %-7,6). Bu gerilemelerin muhtemel nedeni, SMOTE sonrası veri dağılımındaki dengelenmenin, modelin bazı baskın sınıflara olan aşırı uyumunu azaltarak karar sınırlarını yeniden şekillendirmesidir; bu durum, nadir sınıfların lehine olacak şekilde kaynakların yeniden dağılmasına yol açmıştır. Injection (Sınıf 1) sınıfında ise performans tamamen sabit kalmış (750 doğru tahmin), bu da yöntemin bu sınıfa etkisiz ancak zararsız olduğunu göstermektedir.

Genel olarak, KNN modeli SMOTE tekniği ile desteklendiğinde, düşük temsilli saldırı türlerinde anlamlı performans artışları sağlarken, bazı baskın sınıflarda sınırlı düşüşler yaşamıştır. Bununla birlikte, elde edilen sonuçlar, SMOTE'un KNN üzerinde sınıf çeşitliliğini öğrenme yeteneğini artırarak genel sınıflandırma dengesini iyileştirdiğini, böylece modelin veri seti içindeki tüm sınıflara karşı daha kapsayıcı bir öğrenme yaklaşımı sergilemesini sağladığını ortaya koymaktadır.

Ancak, bu iyileşme seviyesi beklenen ölçekte olmamış; SMOTE'un katkısı özellikle az temsil edilen sınıflarda belirginleşirken, genel performans artışı tüm sınıflar için homojen bir şekilde yansımamıştır. Bu durum, yöntemin faydalı olsa da daha yüksek ve dengeli kazanımlar için ek optimizasyon veya farklı stratejilerle desteklenmesi gerektiğini düşündürmektedir.

5.6.2. SVM Algoritması

Şekil 5.28'de SVM sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



Şekil 5.28. SVM - Thermostat için Confusion Matrix

SVM algoritmasının Thermostat (Akıllı Termostat Cihazı) veri setindeki performansı, SMOTE uygulanmadan önce ve sonra elde edilen doğru sınıflandırma sayıları üzerinden değerlendirilmiştir. Sonuçlar, SMOTE'un belirli sınıflarda kayda değer iyileşmeler sağladığını ortaya koymuştur. En yüksek artış, Normal (Sınıf 2) sınıfında görülmüş olup doğru tahmin sayısı 1207'den 1820'ye çıkarak +613 (%50,8) gibi belirgin bir yükseliş göstermiştir. Bunu, Backdoor (Sınıf 0) sınıfındaki +35'lik artış (640'tan 675'e, %5,5) ve ransomware (Sınıf 4) sınıfındaki +14'lük artış (191'den 205'e, %7,3) takip etmiştir. scanning (Sınıf 5) sınıfında ise SMOTE öncesinde hiç doğru tahmin bulunmazken, sonrasında 9 örneğin tamamı doğru sınıflandırılmıştır. XSS (Sınıf 6) sınıfında ise yalnızca +1 (%1,9) gibi sınırlı bir iyileşme gerçekleşmiştir.

Password (Sınıf 3) sınıfında doğru tahmin sayısı sabit kalmış, herhangi bir gelişme gözlenmemiştir. Bu durum, SMOTE'un bazı sınıflarda belirgin fayda sağlarken diğerlerinde etkisinin sınırlı kaldığını göstermektedir. Ancak, bu sınıfta performansın değişmemesi, en azından SMOTE'un mevcut öğrenme düzeyini olumsuz etkilemediğini ve modelin bu sınıftaki başarısını koruduğunu da ortaya koymaktadır.

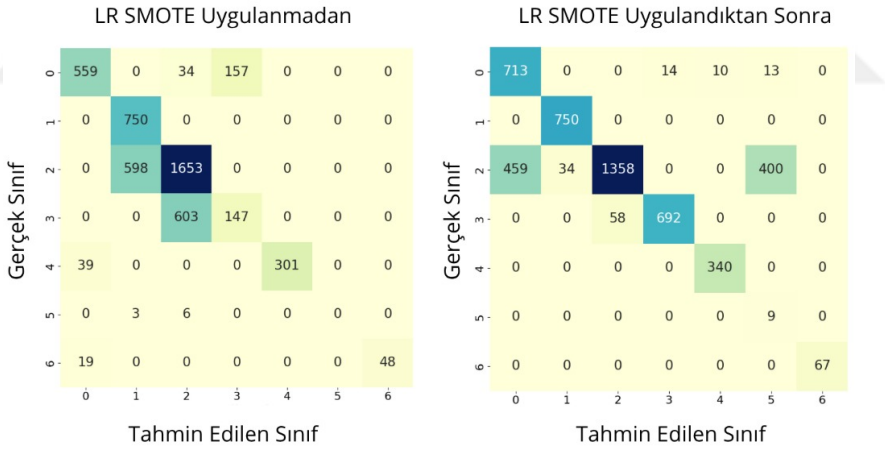
Buna karşın, tek düşüş Injection (Sınıf 1) sınıfında gözlenmiş olup doğru tahmin sayısı

750'den 677'ye gerileyerek -73 (%-9,7) azalma yaşamıştır. Bu gerileme, SMOTE sonrası veri dağılımındaki değişimin modelin bu sınıf üzerindeki ayrıştırma gücünü olumsuz etkileyebileceğini düşündürmektedir.

Genel olarak, SMOTE'un SVM için sınıf dengesizliğini azaltmada ve özellikle nadir görülen sınıfların öğrenilmesinde önemli katkılar sunduğu görülmektedir. Ancak elde edilen kazanımlar, tüm sınıflarda homojen değildir; bazı sınıflarda iyileşme sağlanırken, bazılarında sabit performans veya gerileme gözlenmiştir. Bununla birlikte, SVM algoritmasının bu veri setinde SMOTE tekniği ile uyum sağlayarak genel performansta dikkate değer bir iyileşme göstermesi, küçük çaplı düşüşlere rağmen yöntemin bütünsel açıdan faydalı olduğunu ortaya koymaktadır.

5.6.3. LR Algoritması

Lojistik Regresyon için SMOTE öncesi ve sonrası karmaşıklık matrisleri Şekil 5.29'da sunulmuştur.



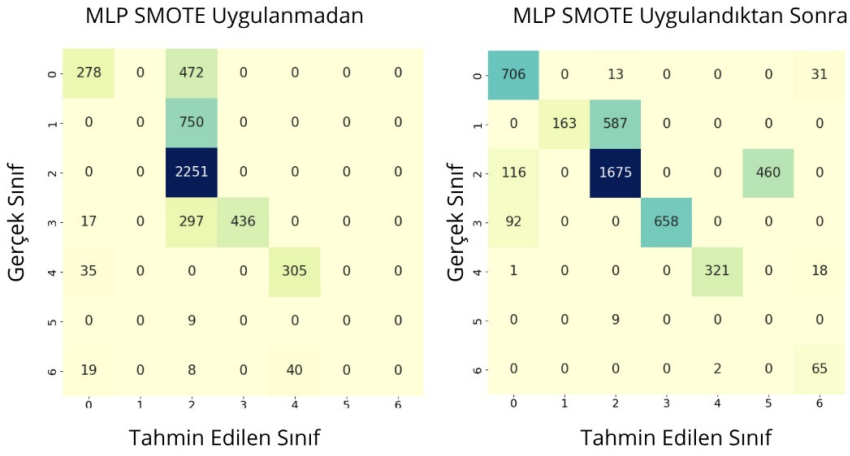
Şekil 5.29. LR - Thermostat için Confusion Matrix

LR algoritmasının Thermostat (Akıllı Termostat Cihazı) veri seti üzerinde SMOTE uygulanmadan önce ve sonra elde edilen doğru sınıflandırma sayıları incelendiğinde, düşük temsile sahip sınıflarda belirgin iyileşmeler gözlenmiştir. Özellikle scanning (Sınıf 5) sınıfında başlangıçta hiç doğru tahmin bulunmazken, SMOTE sonrasında tüm 9 örnek doğru sınıflandırılmış ve %100 doğruluk oranına ulaşılmıştır. Benzer şekilde, XSS (Sınıf 6) sınıfı 48'den 67'ye çıkarak %39,6'lık bir artış göstermiştir.

Password (Sınıf 3) sınıfında ise 147'den 692'ye çıkarak %370,7 gibi dikkat çekici bir iyileşme sağlanmıştır. Backdoor (Sınıf 0) sınıfı da 559'dan 713'e yükselerek %27,5'lik kayda değer bir artış göstermiştir. Bu iyileşmeler, SMOTE'un düşük temsilli veya dengesiz sınıflarda modelin öğrenme kapasitesini belirgin şekilde artırdığını ortaya koymaktadır. Buna karşılık, Normal (Sınıf 2) sınıfında doğru tahminler 1653'ten 1358'e düşerek -17,8%'lik bir azalma yaşamıştır. Bu durum, SMOTE öncesinde modelin ciddi sınıf dengesizliği nedeniyle bu sınıfa yönelik yüksek bir yanlılık geliştirmiş olabileceğini düşündürmektedir. Dengesiz veri yapısı, diğer sınıflara ait örneklerin yanlış biçimde Normal olarak sınıflandırılmasına yol açmış olabilir. SMOTE sonrasında veri dengesi sağlanınca model, diğer sınıfların özelliklerini de öğrenerek Normal sınıfına yönelik aşırı eğilimini azaltmış ve bu sınıfı yalnızca gerçekten ona ait örneklerle sınırlandırmıştır. Injection (Sınıf 1) sınıfında ise doğru tahmin sayısı 750'de sabit kalmıştır. Bu durum, diğer sınıflardaki performans artışlarının bu sınıfın doğruluk oranını olumsuz etkilemediğini ve SMOTE'un buradaki öğrenmeyi koruduğunu göstermektedir. Genel olarak, LR algoritması SMOTE ile yüksek uyum göstermiş, özellikle düşük temsilli sınıflarda önemli performans kazanımları sağlamıştır. Küçük çaplı düşüşlere rağmen, SMOTE'un bu veri setinde LR ile birlikte uygulanmasının modelin genel ayırt etme gücünü artıran etkili bir strateji olduğu sonucuna varılmaktadır.

5.6.4. MLP Algoritması

MLP'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.30'da yer almaktadır.



Şekil 5.30. MLP - Thermostat için Confusion Matrix

MLP algoritmasının Thermostat (Akıllı Termostat Cihazı) veri setindeki performansı, SMOTE uygulanmadan önce ve sonra elde edilen doğru sınıflandırma sayıları üzerinden değerlendirilmiştir. Sonuçlar, özellikle düşük temsile sahip sınıflarda dikkate değer iyileştirmeler olduğunu ortaya koymaktadır. En yüksek artış, Backdoor (Sınıf 0) kategorisinde gerçekleşmiş olup doğru tahmin sayısı 278'den 706'ya çıkarak +428 (%153,9) oranında kayda değer bir yükseliş göstermiştir. Benzer şekilde, Injection (Sınıf 1) sınıfında SMOTE öncesinde hiç doğru tahmin bulunmazken, uygulama sonrasında 163 örnek doğru sınıflandırılmıştır. XSS (Sınıf 6) sınıfında da benzer bir durum gözlenmiş, sıfırdan 65 doğru tahmine ulaşılmıştır. Password (Sınıf 3) kategorisinde ise doğru tahmin sayısı 436'dan 658'e çıkarak +222 (%50,9) oranında anlamlı bir iyileşme kaydedilmiştir.

Orta düzeyde bir artış ransomware (Sınıf 4) sınıfında görülmüş, doğru tahmin sayısı 305'ten 321'e çıkarak +16 (%5,2) artış sağlamıştır. scanning (Sınıf 5) kategorisinde ise SMOTE öncesi ve sonrası doğru tahmin sayısı değişmemiş, her iki durumda da 0 olarak kalmıştır. Bu durum, her ne kadar SMOTE ile bu sınıfa ait sentetik örnekler, en büyük sınıfla eşit sayıda olacak şekilde üretilmiş olsa da, modelin bu sınıfı ayırt etme başarısının artmadığını göstermektedir. Muhtemel neden, Scanning sınıfının özelliklerinin diğer saldırı türlerine oldukça benzer olması veya orijinal veri setinde yer alan sınırlı çeşitliliğin, üretilen sentetik verilerin de sınırlı temsil gücüne sahip olmasına yol açmasıdır. Buna karşın, Normal (Sınıf 2) kategorisinde belirgin bir düşüş gözlenmiş, doğru tahmin sayısı 2251'den 1675'e gerileyerek -576 (%-25,6) oranında azalmıştır. Ancak bu düşüş, olumsuz bir etki olarak yorumlanmamalıdır. SMOTE öncesinde Normal sınıfı, veri setinde baskın olduğu için model bu sınıfa aşırı uyum sağlamış ve pek çok örneği yanlış biçimde Normal olarak sınıflandırmıştır. SMOTE sonrasında veri dengesi sağlanınca bu yapay üstünlük ortadan kalkmış, model farklı sınıfların özelliklerini daha iyi öğrenerek gerçekçi bir sınıflandırma yapmaya başlamıştır. Genel olarak, MLP modeli SMOTE ile desteklendiğinde, düşük temsilli sınıfların öğrenilmesinde kayda değer ve güçlü ilerlemeler sağlamış, baskın sınıflardaki yapay başarıları dengeleyerek modelin genel ayırt edicilik gücünü artırmıştır. Bununla birlikte, Scanning (Sınıf 5) kategorisinde hiçbir gelişme gözlenmemiş olması, sentetik örnek sayısının tek başına yeterli

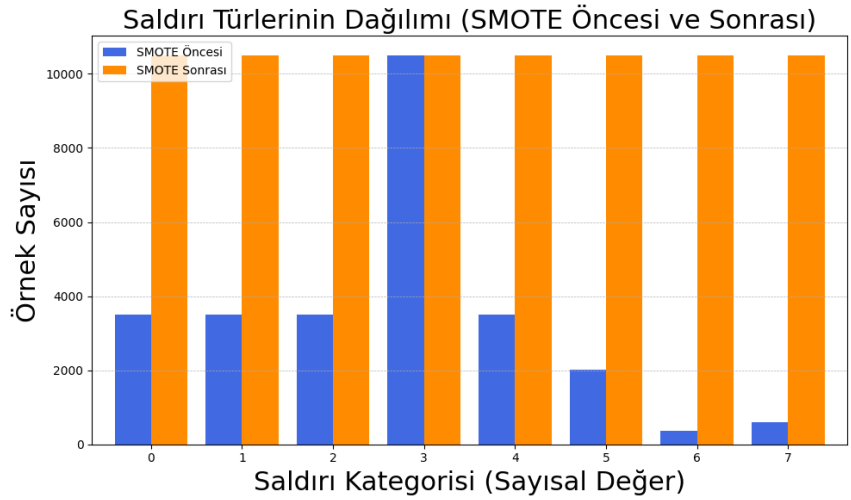
olmadığını ve sınıflar arası ayrım gücüne sahip özelliklerin önemini göstermektedir. Normal (Sınıf 2) kategorisindeki düşüş ise olumsuz bir durum olarak değil, SMOTE sonrası veri dengesinin sağlanmasıyla yapay yüksek başarının ortadan kalkmasının doğal bir sonucu olarak değerlendirilmelidir. Elde edilen sonuçlar, SMOTE'un MLP ile birlikte kullanıldığında etkili ve performans artırıcı bir strateji olduğunu, özellikle nadir görülen saldırı türlerinin doğru sınıflandırılmasında modelin yetkinliğini önemli ölçüde geliştirdiğini ortaya koymaktadır.

5.7. Weather (Hava Durumu Sensör Cihazı)

Saldırı türlerinin sayısal karşılıkları Tablo 5.7'de verilmiştir. SMOTE etkisi ise Şekil 5.31'de gösterilmektedir.

Tablo 5.7. IoT Hava Durumu Sensörü saldırı kategorileri ve sayısal değerleri tablosu

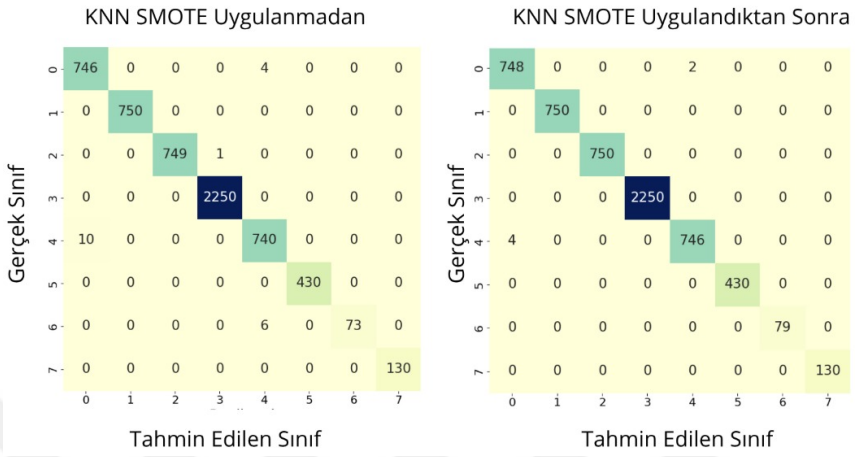
Saldırı Adı	Sayısal Değer
Backdoor	0
ddos	1
Injection	2
Normal	3
Password	4
ransomware	5
scanning	6
XSS	7



Şekil 5.31. IoT Hava Durumu Sensör Cihazı SMOTE Öncesi ve Sonrası Kategori Dağılımı

5.7.1. KNN Algoritması

Şekil 5.32'de KNN sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



Şekil 5.32. KNN - Weather için Confusion Matrix

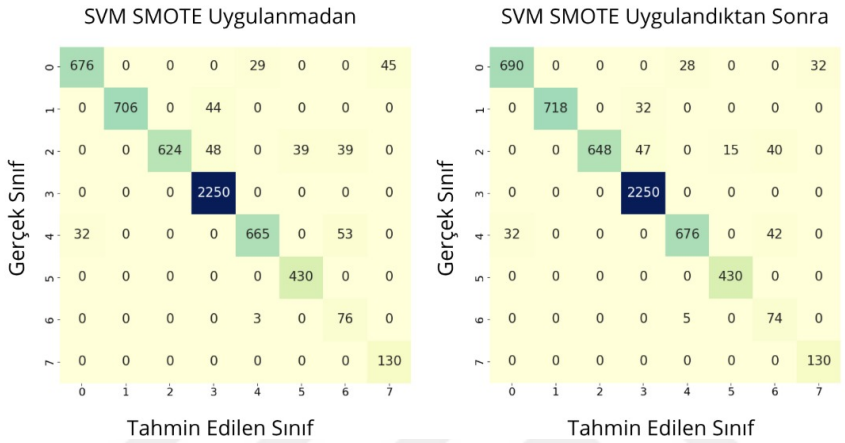
Şekil 5.32'de SMOTE uygulanmadan önce model genel olarak yüksek doğruluk sağlamış, özellikle Normal (Sınıf 3, 2250) ve DDoS (Sınıf 1, 750) kategorilerinde tam başarıya ulaşılmıştır. Benzer şekilde Injection (Sınıf 2, 749/750) ve Password (Sınıf 4, 740/750) sınıflarında da oldukça yüksek doğruluk gözlenmiştir. Ancak azınlık sınıflarda belirgin eksiklikler vardır: Scanning (Sınıf 6) sınıfında sadece 73 doğru sınıflandırma yapılabilmiş. Ayrıca Backdoor (Sınıf 0) kategorisinde 746 doğruya karşılık 4 yanlış sınıflandırma yapılmıştır. SMOTE uygulandıktan sonra, azınlık sınıflarda belirgin iyileşmeler sağlanmıştır. Scanning (Sınıf 6) sınıfında doğru tahmin sayısı 73'ten 79'a yükselmiş, Backdoor (Sınıf 0) sınıfında ise 746'dan 748'e çıkarak hata sayısı azalmıştır. Password (Sınıf 4) sınıfında 740'tan 746'ya yükselme kaydedilmiştir. Bunun yanında, Injection (Sınıf 2) sınıfında 749'dan 750'ye ulaşarak tam doğruluk elde edilmiştir. Normal (Sınıf 3, 2250) ve DDoS (Sınıf 1, 750) sınıfları ise zaten maksimum başarıya sahip olduklarından herhangi bir değişim olmamıştır.

Genel olarak değerlendirildiğinde, SMOTE'un KNN üzerindeki katkısı özellikle Scanning (Sınıf 6), Password (Sınıf 4) ve kısmen Backdoor (Sınıf 0) gibi düşük temsile sahip sınıflarda performansı artırmış; zaten yüksek doğrulukla sınıflandırılan Normal ve DDoS gibi sınıflarda ise modeli olumsuz etkilemeden mevcut seviyeyi korumuştur. Bu

sonuçlar, SMOTE'un KNN için azınlık sınıflarda dengeli öğrenme sağladığını ve veri setindeki dengesizlik sorununu hafifletmede etkili olduğunu göstermektedir.

5.7.2. SVM Algoritması

Şekil 5.33'te SVM sınıflandırıcısının SMOTE uygulanmadan önce ve sonra elde edilen karmaşıklık matrisleri gösterilmektedir.



Şekil 5.33. SVM - Weather için Confusion Matrix

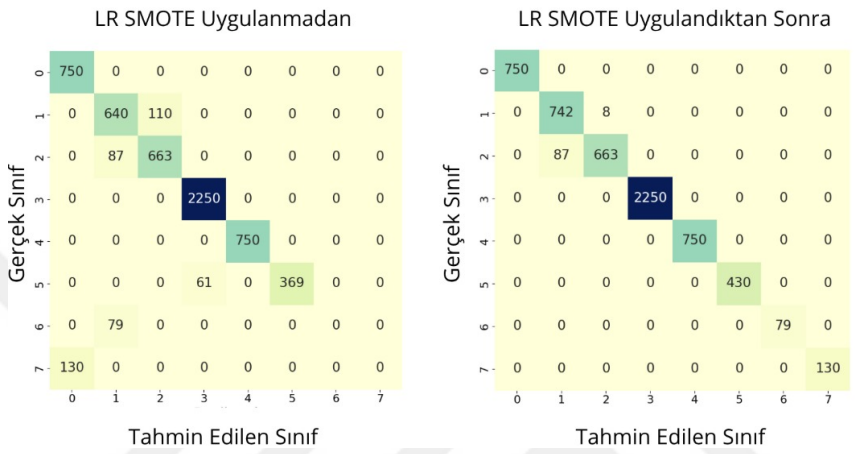
Karışıklık matrisi sonuçlarına göre, SVM algoritması SMOTE uygulanmadan önce özellikle azınlık sınıflarda bazı hatalar göstermiştir. Backdoor (Sınıf 0), DDoS (Sınıf 1), Injection (Sınıf 2) ve Password (Sınıf 4) sınıflarında hatalı sınıflandırmalar göze çarpmaktadır. Buna karşılık Normal (Sınıf 3), Ransomware (Sınıf 5) ve XSS (Sınıf 7) sınıflarında model zaten yüksek başarı göstermiştir. SMOTE uygulandıktan sonra çoğu sınıfta doğru sınıflandırılan örnek sayısında küçük artışlar elde edilmiştir. Özellikle Backdoor sınıfında 676'dan 690'a, DDoS sınıfında 706'dan 718'e, Injection sınıfında 624'ten 648'e ve Password sınıfında 665'ten 676'ya doğru sınıflandırmalar artmıştır. Bununla birlikte Normal (2250) ve Ransomware (430) sınıflarında sonuçlar değişmemiştir. Scanning (Sınıf 6) sınıfında ise doğru tahmin sayısı 76'dan 74'e gerileyerek çok küçük bir düşüş yaşanmıştır.

Genel Değerlendirmede SVM algoritması ile birlikte uygulanan SMOTE tekniği, azınlık sınıflarda %1 ila %4 arasında değişen düzeyde iyileştirmeler sağlamış, ancak bu katkılar sınırlı kalmıştır. Çoğunluk sınıflarında ise herhangi bir olumsuz etki yaratmamış, sadece Scanning (Sınıf 6)de çok küçük bir azalma olmuştur. Genel olarak SMOTE'un SVM

üzerindeki etkisi marjinal olup, modelin genel başarımında kayda değer bir değişiklik yaratmamıştır

5.7.3. Lojistik Regresyon (LR) Algoritması

Lojistik Regresyon için SMOTE öncesi ve sonrası karmaşıklık matrisleri Şekil 5.34'te sunulmuştur.



Şekil 5.34. LR - Weather için Confusion Matrix

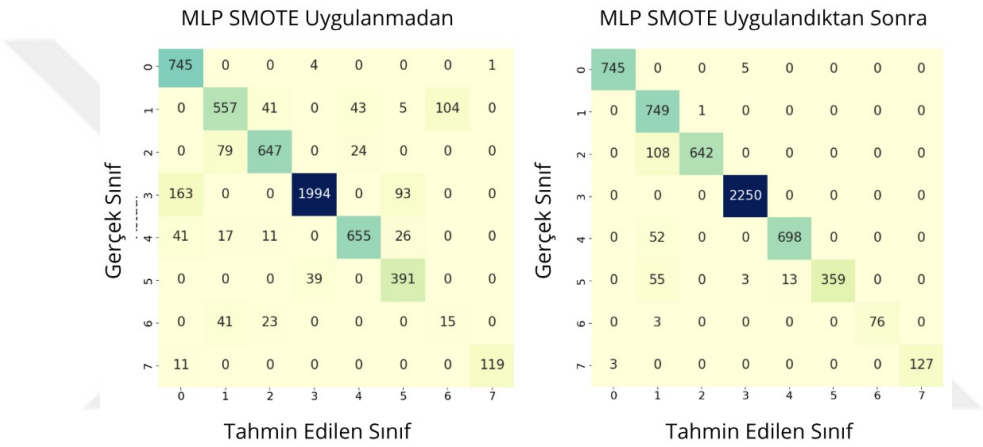
Karışıklık matrisi sonuçları incelendiğinde, LR algoritması SMOTE uygulanmadan önce genel olarak yüksek bir doğruluk oranı sergilemiş, ancak bazı sınıflarda dikkate değer hatalar gözlenmiştir. Özellikle DDoS (Sınıf 1) kategorisinde 640 doğru sınıflandırmaya karşılık 110 yanlış sınıflandırma yapılmış, benzer şekilde Injection (Sınıf 2) sınıfında da 87 örnek yanlış tahmin edilmiştir. Ayrıca Ransomware (Sınıf 5) için 369 doğru tahmine karşılık 61 hata meydana gelmiştir. Bunun yanında Scanning (Sınıf 6) sınıfında 79 örnek yanlış sınıflandırılmış, XSS (Sınıf 7) kategorisinde ise 130 hata görülmüştür.

SMOTE uygulandıktan sonra özellikle azınlık sınıflarda dikkate değer iyileşmeler elde edilmiştir. DDoS (Sınıf 1) sınıfında doğru sınıflandırma sayısı 640'tan 742'ye yükselmiş ve hatalar neredeyse tamamen ortadan kalkmıştır. Ransomware (Sınıf 5) sınıfında doğru sınıflandırma 369'dan 430'a çıkarak belirgin bir iyileşme sağlamıştır. Benzer şekilde Scanning (Sınıf 6) sınıfında 79 olan yanlış sınıflandırmalar tamamen düzeltilmiş, tüm örnekler doğru sınıflandırılmıştır. Diğer sınıflarda (Normal, Password, XSS vb.) ise sonuçlar sabit kalmış ya da küçük değişimler göstermiştir.

Bu sonuçlar, SMOTE'un özellikle dengesiz dağılıma sahip sınıflarda önemli bir katkı sunduğunu, hataları azaltarak doğruluk oranını anlamlı ölçüde artırdığını ortaya koymaktadır. Bununla birlikte, başlangıçta zaten yüksek doğrulukla sınıflandırılan sınıflarda ek bir etki yaratmamış, mevcut performansı korumuştur. Dolayısıyla, LR algoritması üzerinde SMOTE'un etkisi seçici ve hedefe yönelik olmuş, modelin genel başarımına pozitif katkı sağlamıştır.

5.7.4. MLP Algoritması

MLP'ye ait SMOTE öncesi–sonrası karmaşıklık matrisleri Şekil 5.35'ta yer almaktadır.



Şekil 5.35. MLP - Weather için Confusion Matrix

Karışıklık matrisi sonuçları incelendiğinde, MLP algoritmasının SMOTE uygulanmadan önce bazı sınıflarda belirgin hatalar yaptığı görülmektedir. Özellikle Normal (Sınıf 3) kategorisinde doğru sınıflandırılan 1994 örneğe karşılık 256 örnek yanlış sınıflandırılmış, bu da modelin bu sınıfta zorlandığını göstermektedir. Benzer şekilde, DDoS (Sınıf 1) kategorisinde 557 doğru sınıflandırmaya karşılık 204 hata meydana gelmiştir. Password (Sınıf 4) ve Ransomware (Sınıf 5) sınıflarında da hatalar kaydedilmiş, ayrıca XSS (Sınıf 7) sınıfında 11 yanlış tahmin yapılmıştır. Bu durum, modelin özellikle azınlık sınıflarda karışıklık yaşadığını ortaya koymaktadır.

SMOTE uygulandıktan sonra sonuçlarda dikkat çekici iyileşmeler gözlenmiştir. Normal (Sınıf 3) kategorisinde doğru sınıflandırma sayısı 2250'ye yükselmiş ve tüm hatalar ortadan kaldırılmıştır. DDoS (Sınıf 1) kategorisinde doğru sınıflandırmalar 557'den

749’a çıkarak hataların büyük çoğunluğu düzeltilmiştir. Password (Sınıf 4) sınıfında 655 olan doğru sınıflandırma 698’e yükselmiş, hatalar ciddi ölçüde azalmıştır. Ayrıca, Scanning (Sınıf 6) sınıfında da hatalar tamamen ortadan kalkmış ve sınıf neredeyse kusursuz şekilde tahmin edilmiştir. Buna karşın, Ransomware (Sınıf 5) sınıfında küçük bir düşüş yaşanmış, 391 doğru sınıflandırma 359’a gerilemiştir.

Sonuçlar, SMOTE’un özellikle Normal, DDoS ve Password gibi sınıflarda hataları büyük ölçüde azalttığını ve modelin genel doğruluk oranını artırdığını ortaya koymaktadır. Bununla birlikte, Ransomware sınıfında görülen küçük gerileme, dengeleme yöntemlerinin bazı sınıflarda performans kaybına yol açabileceğini göstermektedir.

6. GENEL DEĞERLENDİRME VE SONUÇLAR

Bu çalışmada, farklı IoT cihazlarına ait veri setleri üzerinde sınıflandırma algoritmalarının performansı, SMOTE yönteminin uygulanmasından önce ve sonra olmak üzere karşılaştırmalı olarak değerlendirilmiştir. Çalışma kapsamında doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1-skoru gibi temel performans metrikleri kullanılmıştır. KNN algoritmasının test verisi üzerindeki sınıflandırma sonuçları Tablo 6.1'de sunulmuş olup, SMOTE uygulamasının farklı cihazlar üzerindeki etkisi detaylı şekilde analiz edilmiştir.

Benzer şekilde, SVM algoritmasına ilişkin performans değerlendirmesi Tablo 6.2'de verilmiş, SMOTE öncesi ve sonrasında elde edilen metrikler karşılaştırılmıştır. Lojistik Regresyon algoritmasının test sonuçları Tablo 6.3'te özetlenirken, MLP algoritmasının performansı ise Tablo 6.4'te SMOTE uygulanmadan önceki ve sonraki değerler temelinde sunulmuştur. Her bir algoritma için farklı IoT cihazları üzerindeki etkiler ayrıntılı biçimde incelenmiştir.

6.1. KNN Algoritması Performansı:

KNN'nin SMOTE uygulanmadan önceki ve sonraki sonuçları Tablo 6.1'de karşılaştırılmıştır.

Tablo 6.1. KNN test kümesi performansı (SMOTE öncesi ve sonrası)

IoT Cihazı	SMOTE	Doğruluk	Kesinlik	Duyarlılık	F1-skoru
Buzdolabı	Yok	0.947	0.90	0.90	0.90
	Var	0.955	0.91	0.91	0.91
Garaj Kapısı	Yok	0.952	0.91	0.87	0.88
	Var	0.982	0.96	0.97	0.96
GPS Takip Cihazı	Yok	0.993	1.00	0.99	1.00
	Var	0.985	0.99	0.99	0.99
Modbus	Yok	0.752	0.65	0.52	0.54
	Var	0.848	0.75	0.84	0.78
Hareket Işığı	Yok	0.878	0.83	0.75	0.76
	Var	0.895	0.84	0.87	0.85
Termostat	Yok	0.983	0.80	0.77	0.78
	Var	0.971	0.80	0.90	0.81
Hava Durumu	Yok	0.996	1.00	0.99	0.99
	Var	0.999	1.00	1.00	1.00

6.2. SVM Algoritması Performansı:

SMOTE öncesi ve sonrası SVM performansı, cihaz bazında Tablo 6.2'de verilmiştir.

Tablo 6.2. SVM test kümesi performansı (SMOTE öncesi ve sonrası)

IoT Cihazı	SMOTE	Doğruluk	Kesinlik	Duyarlılık	F1-skoru
Buzdolabı	Yok	0.911	0.86	0.88	0.86
	Var	0.922	0.88	0.89	0.87
Garaj Kapısı	Yok	0.855	0.77	0.82	0.72
	Var	0.860	0.78	0.82	0.73
GPS Takip Cihazı	Yok	0.777	0.73	0.82	0.73
	Var	0.786	0.74	0.83	0.75
Modbus	Yok	0.740	0.67	0.75	0.64
	Var	0.786	0.70	0.81	0.69
Hareket Işığı	Yok	0.687	0.59	0.67	0.60
	Var	0.692	0.59	0.68	0.61
Termostat	Yok	0.726	0.65	0.81	0.62
	Var	0.846	0.72	0.85	0.70
Hava Durumu	Yok	0.944	0.87	0.94	0.89
	Var	0.954	0.89	0.95	0.91

6.3. Logistic Regression Algoritması Performansı:

Logistic Regression Algoritması'nin SMOTE uygulanmadan önceki ve sonraki sonuçları Tablo 6.3'te karşılaştırılmıştır.

Tablo 6.3. Logistic Regression test kümesi performansı (SMOTE öncesi ve sonrası)

IoT Cihazı	SMOTE	Doğruluk	Kesinlik	Duyarlılık	F1-skoru
Buzdolabı	Yok	0.857	0.72	0.72	0.71
	Var	0.997	1.00	1.00	1.00
Garaj Kapısı	Yok	0.941	0.96	0.94	0.94
	Var	1.000	1.00	1.00	1.00
GPS Takip Cihazı	Yok	0.767	0.82	0.61	0.61
	Var	0.997	1.00	1.00	1.00
Modbus	Yok	0.786	0.52	0.48	0.49
	Var	1.000	1.00	1.00	1.00
Hareket Işığı	Yok	0.351	0.36	0.48	0.38
	Var	0.800	0.86	0.89	0.87
Termostat	Yok	0.703	0.67	0.61	0.62
	Var	0.799	0.79	0.93	0.78
Hava Durumu	Yok	0.921	0.68	0.70	0.69
	Var	0.984	0.99	0.98	0.98

6.4. MLP Algoritması Performansı:

SMOTE öncesi ve sonrası MLP performansı, cihaz bazında Tablo 6.4'te verilmiştir.

Tablo 6.4. MLP test kümesi performansı (SMOTE öncesi ve sonrası)

IoT Cihazı	SMOTE	Doğruluk	Kesinlik	Duyarlılık	F1-skoru
Buzdolabı	Yok	0.695	0.67	0.57	0.50
	Var	0.834	0.72	0.80	0.75
Garaj Kapısı	Yok	0.555	0.33	0.33	0.27
	Var	0.978	0.96	0.97	0.96
GPS Takip Cihazı	Yok	0.654	0.50	0.42	0.40
	Var	0.921	0.93	0.91	0.92
Modbus	Yok	0.731	0.70	0.54	0.55
	Var	0.839	0.76	0.84	0.74
Hareket Işığı	Yok	0.362	0.26	0.19	0.16
	Var	0.556	0.72	0.71	0.65
Termostat	Yok	0.665	0.47	0.41	0.41
	Var	0.730	0.72	0.67	0.65
Hava Durumu	Yok	0.870	0.78	0.80	0.78
	Var	0.959	0.97	0.94	0.95

Deneysel çalışmaların IoT cihazlarına ait veri kümeleri üzerinde yürütülmesiyle elde edilen bulgular, SMOTE tekniğinin etkisinin kullanılan algoritmaya ve veri setinin özgün sınıf dağılımına bağlı olarak önemli ölçüde değiştiğini ortaya koymaktadır. KNN ve SVM gibi, SMOTE uygulanmadan önce pek çok cihazda yüksek doğruluk sergileyen algoritmalarda gözlenen iyileşme sınırlı kalmış, gelişmeler çoğunlukla düşük temsil edilen sınıflarda küçük artışlarla sınırlı olmuştur. Bu durum, söz konusu algoritmaların orijinal veri dağılımı üzerinde sınıflar arası ayrımı zaten etkili bir biçimde yapabilmesi ve dolayısıyla yeniden dengeleme işleminden elde edilebilecek ek faydanın düşük olması ile açıklanabilir.

Buna karşılık, SMOTE uygulamasından önce nadir sınıfları tanımakta zorlanan Logistic Regression ve MLP algoritmalarında dikkate değer sıçramalar gözlemlenmiştir. Logistic Regression örneğinde, bazı cihazlarda başlangıçta orta düzeyde olan performans, yeniden dengeleme sonrası nadir sınıfların temsili güçlendirilerek neredeyse kusursuz sonuçlar elde edilecek şekilde iyileşmiştir. MLP ise SMOTE öncesinde, bazı cihazlarda çoğu sınıfta tek bir doğru tahmin dahi yapamayacak kadar düşük performans

sergilemiştir. Bu belirgin başlangıç zayıflığı, SMOTE'un etkinliğini sergilemesi için geniş bir alan sağlamış; üretilen sentetik örnekler sayesinde ağın öğrenme sürecinde temsil uzayı yeniden şekillenmiş ve doğru tahmin sayıları sıfıra yakın değerlerden yüzlerce doğru sınıflamaya ulaşmıştır.

Bu bulgular, SMOTE'un en yüksek verimliliğini sınıf temsili dengesizliği belirgin olan senaryolarda gösterdiğini, buna karşın zaten dengeli ve yüksek doğruluklu modellerde etkisinin daha sınırlı olduğunu ortaya koymaktadır. Ayrıca, veri ön işleme aşamasında uygulanacak yeniden dengeleme tekniklerinin, algoritmanın yapısal özellikleri (doğrusal/doğrusal olmayan, yerel/küresel) ve veri setinin özellikleri (dengeli/dengesiz, düşük/yüksek boyutlu) ile uyumlu seçilmesinin, en yüksek performansın elde edilmesi açısından kritik önem taşıdığı görülmektedir.

6.5. Gelecek Çalışmalar

Bu çalışmanın bulguları, sınıf dengesizliğinin giderilme biçimi, model seçimi ve değerlendirme protokolünün sonuçlar üzerinde belirleyici olduğunu göstermektedir. Gelecek çalışmalarda şu yönler ele alınabilir:

- Yeniden örnekleme stratejileri: SMOTE'a ek olarak ADASYN, Borderline-SMOTE, SMOTE-NC, SMOTEENN ve Tomek links gibi yöntemlerin, ayrıca cost-sensitive öğrenme ve sınıf ağırlıklarının sistematik karşılaştırılması.
- Zaman serisi ve özellik mühendisliği: Pencereleme, istatistiksel/temel frekans özellikleri ve alan bilgisi odaklı özniteliklerle birlikte 1D-CNN, LSTM/GRU ve zaman serisi transformer yaklaşımlarının denenmesi.
- Modelleme ve ayarlama: Topluluk yöntemleri (Random Forest, XGBoost, stacking) ile otomatik hiperparametre optimizasyonu (ör. Bayesian arama), iç içe çapraz doğrulama ve *threshold* ayarlaması.
- Genellenebilirlik ve aktarım: Farklı kurum/cihaz tipleri ve dağılımlarda çapraz veri seti testleri; *domain adaptation* ve transfer learning ile cihazlar arası genelleme.
- Açıklanabilirlik ve kalibrasyon: SHAP/LIME ile öznitelik katkılarının analizi; güven puanı kalibrasyonu (Brier score, ECE) ve reject option kullanımı. .

KAYNAKLAR

Adi, E., Anwar, A., Baig, Z., Zeadally, S. (2020). Machine Learning and Data Analytics for the IoT. *Neural Computing and Applications*, *32*(20), 16205–16233. DOI: 10.1007/s00521-020-04874-y

Ait Mouha, R. A. R. (2021). Internet of Things (IoT). *Journal of Data Analysis and Information Processing*, *9*(2), 77–101. DOI: 10.4236/jdaip.2021.92006

Albasheer, F. O., Haibatti, R. R., Agarwal, M., Nam, S. Y. (2024). A Novel IDS Based on Jaya Optimizer and SMOTE-ENN for Cyberattacks Detection. *IEEE Access*, *12* 101506–101527. DOI: 10.1109/ACCESS.2024.3431534

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A. (2020). TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access*, *8* 165130–165150. DOI: 10.1109/ACCESS.2020.3022862

Bhavitha, K. V., Thangaraj, S. J. J. (2022). Novel Detection of Accurate Spam Content using Logistic Regression Algorithm Compared with Gaussian Algorithm. *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, Dubai, United Arab Emirates, 16–17 February 2022.

Blagus, R., Lusa, L. (2013). SMOTE for High-Dimensional Class-Imbalanced Data. *BMC Bioinformatics*, *14*(1), 106. DOI: 10.1186/1471-2105-14-106

Calp, M. H., Bütüner, R. (2024). Detecting the Cyber Attacks on IoT-Based Network Devices Using Machine Learning Algorithms. *Politeknik Dergisi*, *27*(5), 1971–1989. DOI: 10.2339/politeknik.1340515

Cortes, C., Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. DOI: 10.1007/BF00994018

Cover, T., Hart, P. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. DOI: 10.1109/TIT.1967.1053964

Dablain, D., Krawczyk, B., Chawla, N. V. (2022). DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(9), 6390–6404. DOI: 10.1109/TNNLS.2022.3157312

Doghramachi, D. F., Ameen, S. Y. (2023). Internet of Things (IoT) Security Enhancement Using XGBoost Machine Learning Techniques. *Computers, Materials & Continua*, *77*(1), 717–732. DOI: 10.32604/cmc.2023.041186

Eid, A. M., Soudan, B., Nassif, A. B., Injadat, M. (2024). Comparative Study of ML Models for IIoT Intrusion Detection: Impact of Data Preprocessing and Balancing. *Neural Computing and Applications*, *36*(13), 6955–6972. DOI: 10.1007/s00521-024-09439-x

Eren, K. K., Küçük, K. (2023). Improving Intrusion Detection Systems for IoT Devices using Automated Feature Generation based on ToN$_{IoT dataset}$. *2023 8th International Conference on Computer Science and Engineering (UBMK)*, Burdur, Turkiye, 13–15 September 2023.

Farea, A. H., Alhazmi, O. H., Kucuk, K. (2024). Advanced Optimized Anomaly Detection System for IoT Cyberattacks Using Artificial Intelligence. *Computers, Materials & Continua*, *78*(2), 1525–1545. DOI: 10.32604/cmc.2024.046135

Gad, A. R., Nashat, A. A., Barkat, T. M. (2021). Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. *IEEE Access*, *9* 142206–142217. DOI: 10.1109/ACCESS.2021.3120626

Gokhale, P., Bhat, O., Bhat, S. (2018). Introduction to IOT. *International Advanced Research Journal in Science, Engineering and Technology*, *5*(1), 41–44. DOI: 10.17148/IARJSET.2018.517

Hussain, F., Hussain, R., Hassan, S. A., Hossain, E. (2020). Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Communications Surveys & Tutorials*, *22*(3), 1686–1721. DOI: 10.1109/COMST.2020.2986444

Mahalaxmi, K. V. K., Rekha, K. S. (2022). Comparison of Logistic Regression and Artificial Neural Network for Modelling Credit Card Data Set with the Identification of Precise Fraudulent. *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, Dubai, United Arab Emirates, 16–17 February 2022.

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., Sheth, A. P. (2018). Machine Learning for Internet of Things Data Analysis: A Survey. *Digital Communications and Networks*, *4*(3), 161–175. DOI: 10.1016/j.dcan.2018.08.002

Mazhar, T., Talpur, D. B., Shloul, T. A., Ghadi, Y. Y., Haq, I., Ullah, I., Ouahada, K., Hamam, H. (2023). Analysis of IoT Security Challenges and Its Solutions Using Artificial Intelligence. *Brain Sciences*, *13*(4), 683. DOI: 10.3390/brainsci13040683

Mishra, S. (2017). Handling Imbalanced Data: SMOTE vs. Random Undersampling. *International Research Journal of Engineering and Technology*, *4*(8), 317–320.

Mohammed, A. J., Muhammed Hassan, M., Hussein Kadir, D. (2020). Improving Classification Performance for a Novel Imbalanced Medical Dataset Using SMOTE Method. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(3), 3161–3172. DOI: 10.30534/ijatcse/2020/104932020

Mohindru, G., Mondal, K., Banka, H. (2021). Different Hybrid Machine Intelligence Techniques for Handling IoT-Based Imbalanced Data. *CAAI Transactions on Intelligence Technology*, *6*(4), 405–416. DOI: 10.1049/cit2.12032

Moustafa, N. (2021). A New Distributed Architecture for Evaluating AI-Based Security Systems at the Edge: Network TON_IoT Datasets. *Sustainable Cities and Society*, *72* 102994. DOI: 10.1016/j.scs.2021.102994

Popoola, S. I., Adebisi, B., Ande, R., Hammoudeh, M., Anoh, K., Atayero, A. A. (2021). SMOTE-DRNN: A Deep Learning Algorithm for Botnet Detection in the Internet-of-Things Networks. *Sensors*, *21*(9), 2985. DOI: 10.3390/s21092985

Pradipta, G. A., Wardoyo, R., Musdholifah, A., Sanjaya, I. N. H., Ismail, M. (2021). SMOTE for Handling Imbalanced Data Problem : A Review. *2021 Sixth International Conference on Informatics and Computing (ICIC)*, Jakarta, Indonesia, 03–04 November 2021.

Sayegh, H. R., Dong, W., Al-Madani, A. M. (2024). Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data. *Applied Sciences*, *14*(2), 479. DOI: 10.3390/app14020479

Soe, Y. N., Santosa, P. I., Hartanto, R. (2019). DDoS Attack Detection Based on Simple ANN with SMOTE for IoT Environment. *2019 Fourth International Conference on Informatics and Computing (ICIC)*, Semarang, Indonesia, 23–24 October 2019.

Taud, H., Mas, J.-F. (2018). Multilayer Perceptron (MLP). *Geomatic Approaches for Modeling Land Change Scenarios*. Ed. Camacho Olmedo, M. T., Paegelow, M., Mas, J.-F., Escobar, F. Cham, Switzerland: Springer, ss. 451–455.

Verma, A., Ranga, V. (2020). Machine Learning Based Intrusion Detection Systems for IoT Applications. *Wireless Personal Communications*, *111*(4), 2287–2310. DOI: 10.1007/s11277-019-06986-8

Zhang, P., Jia, Y., Shang, Y. (2022). Research and Application of XGBoost in Imbalanced Data. *International Journal of Distributed Sensor Networks*, *18*(6), 15501329221106935. DOI: 10.1177/15501329221106935

KİŞİSEL YAYIN VE ESERLER

Kalifati, E. ve diğ. (2025a). Performance Analysis of SMOTE-Enhanced Machine Learning Approaches on IoT-Based TON IoT Dataset. *33. Sinyal İşleme ve İletişim Uygulamaları Konferansı*, İstanbul, Türkiye, 25–28 Haziran 2025.

Kalifati, E. ve diğ. (2025b). A Comparative Study of SMOTE-Enhanced Machine Learning Models for IoT Device Classification on the TON IoT Dataset. *13th IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Chisinau, Moldova, 23–26 Haziran 2025.

ÖZGEÇMİŞ

Kendisi lise öğrenimini Mersin'de tamamladı. 2018–2022 yılları arasında Fırat Üniversitesi Yazılım Mühendisliği Bölümü'nden mezun oldu. Lisans sürecinde çeşitli teknoloji firmalarında staj yaparak yazılım geliştirme, proje yönetimi ve ekip çalışması alanlarında deneyim kazandı. MSY Yazılım'da beş ay website Developer olarak görev yaparak Laravel tabanlı web projelerinde web sayfaları geliştirdi. 2023 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Yazılım Mühendisliği yüksek lisans programına kabul edildi. Tez çalışmasında TON IoT veri kümesindeki sınıf dengesizliği sorununa SMOTE yaklaşımıyla çözüm aramakta ve makine öğrenimi modellerinin performansını karşılaştırmalı olarak incelemektedir. Akademik hedefi, doktora eğitimine devam ederek üniversitede araştırmacı olarak çalışmak ve bilimsel üretime katkı sunmaktır.