Avrupa Bilim ve Teknoloji Dergisi
Sayı 31 (Ek Sayı 1), S. 725-734, Aralık 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 31 (Supp. 1), pp. 725-734, December 2021

Research Article

Saldırı Tespit Sistemlerinde Topluluk Öğrenme Yöntemlerinin Kıyaslanması

İlya Kuş[1*], Sinem Bozkurt Keser[2], Esra Nergis Yolaçan[3]

[1*] Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendsiliği Bölümü, Eskişehir, Türkiye, (ORCID: 0000-0001-5850-949X), ilyakus97@gmail.com

[2] Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendsiliği Bölümü, Eskişehir, Türkiye, (ORCID: 0000-0002-8013-6922), sbozkurt@ogu.edu.tr

[3] Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendsiliği Bölümü, Eskişehir, Türkiye, (ORCID: 0000-0002-0008-1037), yolacan@ogu.edu.tr

(İlk Geliş Tarihi: 16 Temmuz 2021 ve Kabul Tarihi: 10 Aralık 2021)

(**DOI:** 10.31590/ejosat.971875)

ATIF/REFERENCE: Kuş, İ., Bozkurt Keser, S. & Yolaçan, E. N. (2021). Saldırı Tespit Sistemlerinde Topluluk Öğrenme Yöntemlerinin Kıyaslanması. *Avrupa Bilim ve Teknoloji Dergisi*, (31), 725-734.

Öz

Günümüz bilgi çağında teknolojinin gelişmesi, çeşitli güvenlik açıklarının oluşmasına neden olmuştur. Bu durum kişilere, şirketlere ve devletlere yapılan siber saldırıların da artmasına yol açmıştır. Yapılan saldırıların ve güvenlik açıklarının önlenmesinde ise çeşitli yöntemler, teknikler ve komutlar geliştirilmiştir. Güvenliğin sağlanması için geliştirilen bu yapılar, kullanıcıların kişisel verilerini koruma altına almak ile yükümlüdür. Fakat saldırganlar kullandıkları saldırı yöntemleri ile güvenlik açığını yakaladıkları an ilgili ağa saldırmakta ve ağın işlevselliğini etkileyerek performansını düşürmektedir. Bu nedenle, sistemlerin güvenlik altına alınması ve yapılan saldırıların tespiti için Saldırı Tespit Sistemleri geliştirilmiştir. Saldırı Tespit Sistemleri'nde makine öğrenmesi algoritmalarının kullanımı artmaktadır. Bu çalışmada, topluluk öğrenme algoritmalarından Rasgele Orman (Random Forest), CatBoost (*Category Boosting*), XGBoost (*eXtreme Gradient Boosting*) ve LightGBM (*Light Gradient Boosting Machine*) Saldırı Tespit Sistemleri'nde anomali tespitinde yaygın kullanılan NSL- KDD ve UNSW-NB_{15} veri kümeleri üzerinde tanıtılmış ve kıyaslanmıştır. Doğruluk (*accuracy*), hassasiyet (*precision*), geri çağırma (*recall*), f-ölçütü (*f-measure*) ve eğri altında kalan performans metrikleri kullanılarak algoritmaların performansları hesaplanmıştır. Gerçekleştirilen deneylerde, her iki veri kümesi içinde en iyi performans değerleri Rasgele Orman algoritması ile elde edilmiştir.

Anahtar Kelimeler: Saldırı Tespit Sistemleri, Topluluk Öğrenme Algoritmaları, Rasgele Orman, XGBoost, LightGBM, CatBoost

Comparison of Ensemble Learning Methods in Intrusion Detection Systems

Abstract

The development of technology in today's information age has led to the formation of various security vulnerabilities. This situation has led to an increase in cyber attacks against individuals, companies and states. Various methods, techniques and commands have been developed to prevent attacks and security vulnerabilities. These structures developed to ensure security are obliged to protect the personal data of users. However, as soon as the attackers detect the security vulnerability with the attack methods what they use, they attack the relevant network and affect the functionality of the network, reducing its performance. Therefore, Intrusion Detection Systems have been developed to secure systems and detect attacks. The use of machine learning algorithms in Intrusion Detection Systems is increasing. In this study, ensemble learning algorithms, Random Forest, CatBoost, XGBoost and LightGBM Intrusion Detection Systems are introduced and compared on NSL-KDD and UNSW-NB_{15} datasets, which are widely used in anomaly detection. The performances of the algorithms were calculated using the accuracy, precision, recall, f-measure and area under-curve performance metrics. In the experiments carried out, the best performance values in both datasets were obtained with the Random Forest algorithm.

Keywords: Intrusion Detection Systems, Ensemble Learning Algorithms, Random Forest, XGBoost, LightGBM, CatBoost

[*] Sorumlu Yazar: ilyakus97@gmail.com

1. Giriş

Teknolojinin her geçen gün gelişmesiyle birlikte bilgi ve bilgisayar güvenliği, güvenlik açıklarından dolayı bazı tehlikeler ile karşı karşıya gelmektedir. Kötü kullanıcıların ve saldırganların yaptıkları ataklar sonucu ağ üzerindeki sistemler izinsiz girişlere maruz kalmaktadır. İzinsiz girişlerin önüne geçebilmek için kullanılan Saldırı Tespit Sistemleri, antivirüsler, web filtreleme çözümleri ve güçlü tanılama yöntemleri gibi çeşitli yöntemler bulunmaktadır. Saldırı tespit yöntemleri hem sistem içinden hem de sistem dışından oluşabilecek saldırılarının saptanması için önerilmektedir. Güvenlikte oluşan açıklar yüzünden gerçekleşen saldırıların engellenebilmesi için çeşitli çalışmalar yapılmaktadır. Bu çalışmalarda izinsiz girişlerin tespit edilmesi ve anomalilerin analiz edilmesi için birçok yöntem önerilmiştir (Kasongo ve Sun, 2020).

Son yıllarda, Saldırı Tespit Sistemleri'nde makine öğrenmesi algoritmalarının kullanımı artmaktadır. Bu çalışmada, ağ akış verileri üzerinden topluluk öğrenme algoritmalarının kullanılmasıyla anomalilerin tespitinin yapılması amaçlanmıştır. Saldırı Tespit Sistemleri'nde sıkça kullanılan veri kümelerinden NSL- KDD ve UNSW-NB_{15} kullanılmıştır. Kaggle platformundan elde edilen bu veri kümeleri çeşitli saldırı tiplerinden oluşmaktadır. Saldırı Tespit Sistemleri'nde en yaygın kullanılan bu veri kümeleri üzerinde popüler topluluk öğrenme algoritmalarının karşılaştırmalı analizi yapılmaktadır. Rasgele Arama yöntemi kullanılarak algoritmalar için en iyi hiper-parametreler belirlenmektedir. Gerçekleştirilen deneyler ile algoritmalar performans metrikleri üzerinden analiz edilerek her iki veri kümesi içinde en iyi algoritma tespit edilmektedir.

Çalışmanın takip eden bölümünde literatürde Saldırı Tespit Sistemleri alanında yapılan çalışmalar verilmektedir. Üçüncü bölümde çalışmada kullanılan veri kümeleri, makine öğrenimi algoritmaları ve performans metrikleriyle birlikte hiper-parametre optimizasyonu anlatılmaktadır. Deney sonuçları ve karşılaştırmalı analizler dördüncü bölümde verilmektedir. Çalışma, sonuçlar ve gelecek çalışmalar ile sonlandırılmaktadır.

2. Literatür Taraması

Son on yılda NSL-KDD ve UNSW-NB_{15} veri kümeleri kullanılarak çok sayıda araştırma yapılmıştır. Bu araştırmalarda kullanılan makine öğrenme algoritmalarının ve çeşitli tekniklerin uygulamaları mevcuttur (Tablo1). Saldırı Tespit Sistemleri için geliştirilen bu uygulamalar ağ veya sistem üzerinde gerçekleşebilecek olan saldırıların tespiti için tasarlanmıştır. Ambusaidi ve diğerleri tarafından gerçekleştirilen çalışmada sınıflandırma için en uygun özniteliği analitik olarak seçen karşılıklı bilgi tabanlı bir algoritmanın önerilmesi bu uygulamalara örnektir (Ambusaidi vd., 2016). Bu algoritma, doğrusal ve doğrusal olmayan bağımlı veri özniteliklerini işleyerek ağ saldırı tespitini gerçekleştirmektedir. Önerilen algoritma ile seçilen öznitelikler kullanılarak En Küçük Kare Destek Vektör Makinesi Tabanlı Saldırı Tespit Sistemi (LSSVM-IDS, *Least Square Support Vector Machine Based Intrusion Detection System*) oluşturulmuştur. Oluşturulan sistemin performansı için NSL-KDD, KDD Cup99 ve Kyoto 2006+ veri kümeleri kullanılmıştır. Bamakan ve diğerleri tarafından yapılan çalışmada ise Çok Ölçütlü Doğrusal Programlama (MCLP, *Multiple Criteria Linear Programming*) için hiper-parametre optimizasyonu ve öznitelik seçimi sistemin performansını iyileştirmek için uygulanmaktadır (Bamakan vd., 2016). Bunun için yeni bir uyarlanabilir Zamanla Değişen Kaos Parçacık Sürüsü Optimizasyonu (TVCPSO, *Time Varying Chaos Particle Swarm Optimization*) kullanılarak bir saldırı tespit sistemi önerilmiştir. Önerilen yöntemin performansını ölçmek için KDDCup99 ve NSL-KDD veri setleri kullanılmıştır. Ashfaq ve diğerleri tarafından gerçekleştirilen çalışmada ise Saldırı Tespit Sistemleri'nde kullanılan sınıflandırıcıların performansını iyileştirmek amaçlanmıştır (Ashfaq vd., 2016). Denetimli makine öğrenme algoritmaları ile desteklenen etiketsiz örnekler kullanılarak bulanık tabanlı ve yarı denetimli öğrenme yaklaşımı önerilmiştir. Yarı denetimli öğrenme denetimli ve denetimsiz makine öğrenme tekniklerinin bir birleşimidir. Etiketlenmiş örneklerle birlikte etiketlenmemiş örnekleri ele alan sınıflandırıcı ile daha yüksek performans değerleri elde edilmiştir. Aburomman ve Reaz tarafından yapılan çalışmada ise saldırı tespitine en uygun Destek Vektör Makinesi (*SVM, Support Vector Machine)* modelini bulmak amaçlanmıştır (Aburomman ve Reaz, 2017). Karşılaştırılan yöntemler sırasıyla Kalanlara Karşı-SVM (*OAR-SVM, One Against Rest SVM*), Yönlendirilmiş Asiklik Grafik-SVM (*DAG- SVM, Directed Acyclic Graph SVM*), Uyarlanabilir Yönlendirilmiş Asiklik Grafik-SVM (*ADAG-SVM*, *Adaptive Directed Acyclic Graph SVM*) ve Hata Düzeltme Kodu Çıktısı-SVM (*ECOC-SVM, Error Correcting Output Code SVM*)'dir. Sınıflandırma performansını iyileştirmek için Kalanlara Karşı Ağırlıklı-SVM (*WOAR-SVM, Weighted One Against Rest SVM)*'ye dayalı yeni bir yaklaşım önerilmiştir. Önerilen bu model ile meta sezgisel olarak oluşturulmuş ağırlıklar kullanarak sistemdeki anomali durumları tespit edilmiştir. Bostanı ve Sheikhan tarafından gerçekleştirilen çalışmada, Saldırı Tespit Sistemleri'nde kullanılan ve makine öğrenme algoritmalarının bazı sınırlamalarının üstesinden gelebilen grafik tabanlı bir makine öğrenme algoritması olan Optimum-Yol Ormanı (OPF, *Optimum Path Forest*) önerilmiştir. Önerilen algoritmanın performansının iyileştirilmesi için Çok Amaçlı Optimal Hibrit Güç Akışı (*MOPF, Multi-objective Optimal Hybrid Power Flow*) algoritması kullanılarak izinsiz giriş tespitinde yeni bir yaklaşım önerilmiştir (Bostani ve Sheikhan, 2017). NSL-KDD veri kümesi üzerinde önerilen modelin performansını değerlendirmek için Gelişmiş OPF+Bölümleme (*AOPF+P, Advanced OPF+Partitioning*), Gelişmiş OPF+Budama (*AOPF+Pr, Advanced OPF+Pruning*) ve Optimum Yol Ormanının Değiştirilmiş Versiyonu (*MOPF, modified version of optimum-path forest*) ile OPF'nin karşılaştırılması yapılmış ve önerilen model ile daha yüksek performans değerlerinin elde edildiği tespit edilmiştir. Primartha ve Taha tarafından yapılan çalışmada ise Rasgele Orman algoritması, üç farklı veri kümesi (NSL-KDD, UNSW-NB_{15} ve GPRS) üzerinde analiz edilmiştir (Primartha ve Tama, 2017). Tama ve Rhee tarafından yapılan çalışmada ise anomali tespitinde Gradyan Arttırılmış Makine (GBM, *Gradient Boosted Machine*) algoritması, Rasgele Orman, Derin Sinir Ağı (DNN, *Deep Neural Network*), SVM ve Regresyon Ağacı sınıflandırıcıları ile karşılaştırılmıştır (Tama ve Rhee, 2019). Deneylerde, NSL-KDD, UNSW-NB_{15} ve GPRS veri kümeleri üzerinde on kat çapraz doğrulama kullanılarak algoritmalar analiz edilmiştir. En yüksek performans değerlerine GBM algoritması (%93.64 doğruluk) ile ulaşılmıştır. Kamarudin ve diğerleri tarafından bilinen ve bilinmeyen web saldırıların tespit edilmesi için LogitBoost algoritması önerilmiştir (Kamarudin vd., 2017). HTTP protokolü üzerinden çalıştırılan web saldırılarını tanıyan ve topluluk tabanlı bir sınıflandırma yaklaşımı kullanan bir saldırı tespit sistemi oluşturulmuştur.

Tablo 1. Literatürde İncelenen Çalışmaların Karşılaştırılması

Çalışma	Veri Setleri	Yöntem	Platform	Performans Ölçütleri	Değerlendirme Sonuçları
Ambusaidi vd. (2016)	KDD Cup 99, NSL-KDD ve Kyoto 2006+	LSSVM-IDS, HFSA (önerilen özellik seçim algoritması)	-	ACC, DR, FPR, PRE, REC ve FSC	NSL-KDD: ACC: %99.91, FPR: %0.28, DR: %98.76 KDD Cup: ACC: %99.79, FPR: %0.13, DR: %99.46 Kyoto 2006+: ACC: %99.77, FPR: %0.13, DR: %99.64
Bamakan vd. (2016)	KDDcup99 ve NSL-KDD	MCLP / SVM sınıflandırıcısı	Matlab 2013	ACC, High DT, Low FPR	Öznitelik seçimi olmadan: TVCPSO-MCLP, ACC: %94.69, DT: %95.19, FPR: %4.81, TVCPSO–SVM ACC: %95.75 DT: %95.49 FPR: %3.29 Öznitelik seçimi olduğunda: TVCPSO-MCLP ACC: %96.88 DT: %97.23 FPR: %2.41, TVCPSO–SVM ACC: %97.84 DT: %97.03 FPR: %0.87
Ashfaq vd. (2017)	NSL-KDD	MLP, J48, Naive Bayes, Rasgele Orman, SVM ve önerilen algoritma	-	Farklı başlatma aralıklarının genel performans üzerindeki etkisinin testi	KDDTest+ ve KDDTest –21 veri setleri sırasıyla %82.41 ve %67.02 şeklinde başarı elde etmişken ikinci deneyde sırasıyla %84.12 ve %68.82 şeklinde başarı elde edilmiştir.
Aburomman vd. (2017)	NSL-KDD	OAR-SVM, DAG-SVM, ADAG-SVM, ECOC-SVM ve WOAR-SVM	Matlab 2015b	ACC, FSC	WOAR-SVM: ACC: % 80.65, OAR-SVM: ACC: %77.06, OAO-SVM: ACC: %77.53, DAG- SVM: ACC: %68.78, ADAG-SVM: ACC: %64.59 ECOC-SVM: ACC: %76,15
Bostanı ve Sheikhan (2017)	NSL-KDD	Önerilen MOPF algoritması	Matlab 2014a	ACC, DT, FAR ve Örnek Başına Maliyet	OPF: ACC: %76.88, AOPF: ACC: %90.53, AOPF+ P: ACC: %91.09, AOPF + Pr: ACC: %90.27, MOPF: ACC: %91.74
Taha ve Rhee (2017)	NSL-KDD, UNSW-NB_{15} ve GPRS	GBM, Rasgele Orman, DNN, SVM ve CART	-	ACC, Specificity, Sensitivity, FPR ve AUC	KDDTrain+: ACC: %99.85, FPR: %0.27, KDDTest+: ACC: %91.82, FPR: %4.19, KDDTest-21: ACC: %86.51, FPR: %2.65, UNSW-NB_{15}: Tenfold: ACC: %95.08, FPR: %2.97. Hold-out: ACC: %91.31, FPR: %8.60. WEP/WPA: Tenfold cross validation: ACC: % 82.6, FPR: %20.7. Hold-out: ACC: %70.8, FPR: %27.4. WPA2: Tenfold: ACC: %92.4, FPR: %2.77. Hold-out: ACC: %85.4, FPR: %0.27
Kamarudin vd. (2017)	NSL-KDD ve UNSW-NB_{15}	NB, SVM, MLP, DT, Rasgele Orman, Adaboost+Rasgele Orman ve Logitboost	MySQL, Weka	ACC, DT ve FAR	Sırasıyla ACC, DR, FAR değerleri: NB: %53.61, %42.73, %19.18, SVM: %87.41, %0.11, %32.55, MLP: %64.86, %53.43, %6.50, J48: %89.68, %88.23, %6.68, RF: %90.11, %89.32, %7.89, Adaboost+RF: %90.27, %89.71, %8.30, Logitboost: %90.33, % 89.75, %8.22
Primartha ve Taha (2017)	NSL-KDD, GPRS ve UNSW-NB_{15}	Rasgele Orman, Decision Tree	-	ACC ve FAR	UNSW-NB_{15}: ACC: %95.5 FAR: %7.22, NSL- KDD: ACC: %99.57 FAR: %7.22, GPRS: ACC: %91.8, FAR: %6.35
Divekar vd. (2018)	UNSW-NB_{15}, NSL-KDD, KDD Cup 99	Parametre Optimizasyonu, Rasgele Orman, NB, DT, NN, K-means, SVM	Python	FSC, Boş Hata Oranı	F değerlerinin çıktısı gösterilmemiştir. Sadece saldırıların f- ölçütü ile sayısı belirtilmiştir.
Yang vd. (2019)	NSL-KDD ve UNSW-NB_{15}	ICVAE-DNN (önerilen model)	Tensorflow	PRE, DR, FPR, REC, ACC, FSC.	KDDTest +: ACC: %85.97, DR: %77.43, FPR : %2.74, KDDTest-21: ACC: %75.43, DR: %72.86, FPR: %12.96, UNSW- NB_{15}: ACC: %89.08, DR: %95.68, FPR: %9.01
Chu vd. (2019)	NSL-KDD	SVM, Naive Bayes, Decision Tree, MLP	Weka	Tanıma Hızı, Hesaplama Hızı	SVM- rate- speed: %97.38, 0.83 sec, NBC-rate-speed: %90.00, 0.67 sec, DT-rate-speed: %59.00, 0.19 sec, MLP-rate-speed: %97.74, 0.19 sec.

Yang vd. (2019)	NSL-KDD ve UNSW-NB_{15}	MDPCA-DBN (önerilen model)	-	ACC, DR, PRE, REC, FSC, FPR	KDDTest+: ACC: %82.08 DR: %70.51 FPR: %2.62 KDDTest-21: ACC: %66.18, DR: %61.57, FPR: %13.06 UNSW- NB_{15}: ACC: %90.21, DR: %96.22, FPR: %17.15
Tama vd. (2019)	NSL-KDD ve UNSW-NB_{15}	SVM, NB, DT, MLP, Rasgele Orman, Logistic Regression, NN	Weka	ACC, FPR, Sensitivity, PRE	KDD: SVM: ACC: %81.58, (bagging)J48: ACC: %84.25, RF: ACC: 80.67, DT: ACC: %81.05, Naive bayes: ACC: %76.50. UNSW-NB_{15}: DT: ACC: %81.42, LR: ACC: 83.13, Naive Bayes: ACC: %82.07, Neural Network: ACC: %81.34.
Sethi vd. (2020)	NSL-KDD, UNSW-NB_{15} ve AWID	DRL, Rasgele Orman, KNN, ADB, GNB ve QDA	Python	ACC, FPR	NSL-KDD: ACC: %81, FPR: %2.6 UNSW- NB_{15}: ACC: %85.09, FPR: %3.3 AWID: ACC: %96.02, FPR: %0.3
Wu vd. (2020)	NSL-KDD ve UNSW-NB_{15}	Parametre Optimizasyonu, Adaboost, SVM, CNN, MLP, Rasgele Orman, LuNet, HAST-IDS, LSTM	Python	ACC, DR, FAR	NSL-KDD: ACC: %98.70, DR: %98.92, FAR: %0.80, UNSW-NB_{15}: ACC: %97.42, DR: %85.76, FAR: %2.37 PELİCAN: ACC: %97.75, DR: %86.64, FAR: %1.30
Aleesa vd. (2021)	UNSW-NB_{15}	Parametre Optimizasyonu, ANN, DNN, Rasgele Orman, NB, LR, DT, EM	-	İkili sistemde doğruluk değerleri, çoklu sınıfta doğruluk değerleri.	ANN: binary classification ACC: %99.26, multi-class classification ACC: %97.89, DNN binay classification ACC: %99.22, multi-class classification ACC: %99.59, RNN-LSTM: binary classification ACC: %85.42, multi-class classification ACC: %85.38

Not: ACC = Accuracy, DR = Detection Rate, FPR = False Positive Rate, PRE = Precision, REC = Recall, FSC = F-Score, FAR = False Alarm Rate, AUC = Area Under the ROC Curve

Önerilen sistemin performans değerlendirilmesinde ise NSL-KDD ve UNSW-NB_{15} veri kümeleri kullanılmıştır. Naive Bayes Sınıflandırıcı (NBC, *Naive Bayes Classifiers*), SVM, Rasgele Orman, ADB (AdaBoost), Karar Ağacı (DT, *Decision Tree*) ve Çok Katmanlı Algılayıcı (MLP, *Multilayer Perceptron*) algoritmaları karşılaştırma analizlerinde kullanılmıştır. Sethi ve diğerleri tarafından gerçekleştirilen çalışmada, ağa dağıtılmış çok sayıda bağımsız derin öğrenme yöntemini kullanan ve bağlama duyarlı olan bir Saldırı Tespit Sistemi önerilmiştir (Sethi vd., 2020). Çalışmada NSL-KDD, UNSW-NB_{15} ve AWID veri kümeleri kullanılmıştır. Önerilen Saldırı Tespit Sistemi'nde Rasgele Orman, ADB, GNB (*Gaussian Naive Bayes*), KNN (*K-Nearest Neighbors*) ve QDA (*Quadratic Discriminant Analysis*) sınıflandırıcıları kullanılmıştır. Chu ve diğerleri tarafından SVM kullanılarak gelişmiş kalıcı saldırıların tespiti ve sınıflandırılması üzerine bir çalışma gerçekleştirilmiştir (Chu vd., 2019). Gelişmiş Kalıcı Tehdit (APT, *Advanced Persistent Threat*) karmaşık ve hedefli bir saldırı türüdür. Saldırganlar hedefine saldırmadan önce bilgi toplamak için stratejik bir planlama yapmakta ve bazı özel yöntemler kullanmaktadır. Yapılan bu APT saldırılarının erken tespit edilmesini için önerilen sistemin performansı, NSL-KDD veri kümesi üzerinde analiz edilmiştir. SVM, NBC, DT ve MLP sistemde kullanılan algoritmalardır. Yang ve diğerleri tarafından gerçekleştirilen çalışmada ise Değiştirilmiş Yoğunluk Tepe Kümeleme Algoritması (MDPCA, *modified density peak clustering algorithm*) ve DBNs (deep belief networks) kullanılarak bulanık bir toplama yaklaşımı önerilmiştir (Yang vd., 2017). NSL-KDD ve UNSW-NB_{15} veri kümeleri kullanılarak gerçekleştirilen deneylerde elde edilen sonuçlara göre önerilen model ile diğer iyi bilinen sınıflandırma yöntemlerinden daha yüksek performans elde edilmiştir. Tama ve diğerleri tarafından yapılan çalışmada, NSL-KDD ve UNSW-NB_{15} veri kümeleri için en iyi özniteliklerin seçiminde partikül sürüsü optimizasyonu, karınca koloni algoritması ve genetik algoritma içeren hibrit bir öznitelik seçim tekniği kullanılmıştır (Tama ve Rhee, 2019). Azaltılmış Hata Budama Ağacı (REPT, *Reduced Error Pruning Tree*) sınıflandırıcısı kullanılarak anormali tespiti yapılmıştır. Yang ve diğerleri tarafından gerçekleştirilen çalışmada ise İyileştirilmiş Koşullu Değişken Otomatik Kodlayıcı (ICVAE, *İmproved Conditional Variational AutoEncoder*) ve DNN birleşimi yeni bir saldırı tespit sistemi geliştirilmiştir (Yang vd., 2019). ICVAE-DNN modelinin performans değerlendirilmesinde NSL-KDD ve UNSW-NB_{15} veri kümeleri kullanılmıştır. Önerilen model, eğitim verilerini dengelemek ve çeşitliliğini arttırmak için izinsiz girişlere göre yeni saldırılar oluşturmaktadır. Böylece dengesiz saldırılar tespit edilmektedir. KNN, Çok Terimli Naive Bayes (*MultinominalNB, Multinominal Naive Bayes),* Rasgele Orman, SVM, DNN ve DBN algoritmaları ile karşılaştırıldığında önerilen model ile daha yüksek performans değerleri elde edilmiştir. Wu ve diğerleri tarafından gerçekleştirilen çalışmada, Pelican adında özel olarak tasarlanmış bloklar üzerine inşa edilen derin bir sinir ağı NSL-KDD ve UNSW-NB_{15} veri kümeleri üzerindeki anomalilerin tespiti için önerilmiştir (Wu ve Moustafa, 2020). Aleesa ve diğerleri tarafından yapılan çalışmada izinsiz girişlerin tespit edilmesi için ANN, RNN ve DNN'ye dayalı derin öğrenme modelleri önerilmiştir. Çalışmada kullanılan UNSW-NB_{15} veri kümesi ikili ve çoklu sınıflandırma şeklinde ayrı ayrı olarak ele alınmıştır. En yüksek performans çıktısı ikili sınıflandırmada (%99.26) doğruluk değeri ile ANN yöntemi elde etmiştir. Çoklu sınıflandırmada ise (%99.59) doğruluk değeri ile DNN yöntemi ile elde edilmiştir (Aleesa vd., 2021). Divekar ve diğerleri tarafından yapılan çalışmada, UNSW-NB_{15}, KDD Cup

99 ve NSL-KDD veri kümeleri üzerinde SVM, Rasgele Orman, Sinir Ağı, NB ve K-means algoritmaları karşılaştırılmıştır. KDD Cup 99 ve NSL-KDD veri kümelerinde bulunan U2R ile R2L saldırıları sınıflandırıcıların etkinlikleri ile engellenmiştir. Bu şekilde olası güvenlik riskleri ortaya çıkarılmıştır. Sentetik Azınlık Örneklem Arttırma Yöntemi (*SMOTE, Synthetic Minority Over-Sampling Technique*) yönteminin uygulamasından önce ve sonra olmak üzere iki deney şeklinde ele alınmıştır. SMOTE uygulaması sonrası en yüksek çıktı NSL-KDD veri kümesinde SVM, KDD Cup 99 veri kümesinde NB ve SVM ile elde edilmiştir. Son olarak UNSW-NB_{15} veri kümesinde ise en yüksek performans değerlerine Rasgele Orman ve SVM algoritmaları ile ulaşılmıştır (Divekar vd., 2018). Baykan ve Khorram (2021) yaptıkları çalışmada ise internet ağı üzerindeki saldırı tespitinin gerçekleştirimi sonucu performansların analizi için NSL-KDD veri kümesini kullanmıştır. KNN, Destek Vektör Makineleri (DVM, *Support Vector Machine*) ve Rasgele Orman algoritmaları üzerindeki performans analizleri yapılan çalışmada Parçacık Sürü Optimizasyonu (PSO, *Partical Swarm Optimization*) ve Yapay Arı Kolonisi (YAK, *Artifical Bee Colony*) teknikleriyle de optimizasyon işlemi gerçekleştirilmiştir. Diğer bir çalışmada ise Altunay ve Albayrak (2021) saldırı tespiti için evrişimli sinir ağını CSE-CIC-IDS2018 veri kümesinde gerçekleştirmiştir. SMOTE tekniğinin kullanıldığı çalışmada saldırıların tespitinin başarımı en düşük %98.70, en yüksek %99.10 doğruluk değerleri elde edilmiştir.

Literatürde saldırı tespiti için yapılan çalışmalarda kullanılan birçok veri kümesi mevcuttur. Bu çalışmada, Tablo 1 ile özetlenen literatürdeki çalışmalarda kullanıldığı üzere NSL-KDD ve UNSW-NB_{15} veri kümeleri kullanılmıştır. Uygulanan önişlem adımları, seçilen makine öğrenmesi algoritmaları ve algoritmaların en uygun hiper-parametrelerinin tespit edilmesinde kullanılan rasgele arama yöntemi ile literatür ile elde edilen performans değerleri ile kıyaslandığında önerilen yöntemin üstünlüğü ve etkinliği kanıtlanmaktadır. NSL-KDD veri kümesi KDD Cup 99 veri kümesindeki dezavantajlar ele alınarak oluşturulan bir veri kümesidir. UNSW-NB_{15} veri kümesi ise KDD 99 veri kümesindeki elverişli olmayan özelliklerin çıkarılması ile geliştirilen bir veri kümesidir. Yapılan deneylerde iki veri kümesi ele alınarak karşılaştırmalar yapılmıştır. Karşılaştırılmaların yapıldığı ilk deneyde kullanılan topluluk öğrenme yöntemleri Rasgele Orman, CatBoost, XGBoost ve Light GBM'dir. Topluluk öğrenme yöntemleri temel olarak bagging (*torbalama*) ve boosting (*yükseltme*) olarak ikiye ayrılmaktadır. Bagging yönteminde veri kümesi içerisinden yerine konacak şekilde tekrar tekrar örnekler çekilerek yeni ağaçlar oluşturulmaktadır. Ardından oluşturulan ağaçlar ile bir topluluk ortaya çıkmaktadır. Gerçekleştirilen çalışmada kullanılan Rasgele Orman algoritması bu topluluk öğrenme yöntemleri arasındadır. Boosting yöntemi ise veri kümesine farklı ağırlıklar vererek oluşan topluluktan varsayımlar yapmaktadır. Kullanılan XGBoost, CatBoost ve LightGBM algoritmaları da bu topluluk öğrenme yönteminin içerisinde yer almaktadır. Çalışmada yapılan ikinci deneyde ise kullanılan topluluk öğrenme yöntemlerine Rastgele Arama yöntemi ile hiper-parametre optimizasyonu yapılmıştır. Son olarak performans metrikleri sonucunda elde edilen iyileştirmeler incelenerek karşılaştırmalar yapılmıştır.

3. Materyal ve Yöntemler

Gerçekleştirilen çalışmada saldırıların tespiti için topluluk öğrenme yöntemleri kullanılarak analizler yapılmıştır. Kullanılan hiper-parametre optimizasyon yöntemi ile yapılan deneyler sonucu elde edilen performans çıktılarının karşılaştırılması yapılmıştır. Bu bölümde, NSL-KDD ve UNSW-NB_{15} veri kümeleri ve kullanılan topluluk öğrenme yöntemleri anlatılmıştır. Ardından ele alınan hiper-parametre optimizasyonu için uygulan yöntem ve performans metrikleri verilmiştir.

3.1. Veri Kümeleri

Çalışmada NSL-KDD ve UNSW-NB_{15} veri kümeleri kullanılmıştır. NSL-KDD veri kümesi ağ üzerinde simülasyonu yapılan çeşitli izinsiz girişleri içerir. Veri kümesinde 41 tane öznitelik bulunmaktadır. Bu özniteliklerin üç tanesi nitel diğer otuz sekiz tanesi de nicel özniteliklerdir. Ayrıca bu veri kümesi üç ayrı başlık altında incelenmektedir. Bu başlıklar; içerik özellikleri, sunucu tabanlı trafik özellikleri ve zamana bağlı trafik özellikleridir. İçerik özellikleri TCP bağlantısı ile oluşturulan özniteliklerdir. Sunucu tabanlı özellikler ise etki alanı(domain) bilgisi ile sağlanmaktadır. Aynı sunucu ve servis özelliklerini kullanan özniteliklerde zamana bağlı trafik özelliklerini oluşturmaktadır. Veri kümesinde bilgisayara yapılan atak türleri ise DoS, U2R, R2L ve Probing'dir. DoS saldırıları binlerce IP adresinin kullanılmasıyla saniyeler içerisinde birçok veri gönderimi sonucu yapılan siber saldırılardan oluşmaktadır. U2R, kullanıcıların hesaplarının yönetici hesaplarındaki yetkinlikler ile sistemlere erişebilme haklarının sağlanması için yapılan saldırılardır. R2L saldırıları uzak makinelerden yetkisiz bir şekilde girişin yapıldığı ve kullanılan araçlar yardımıyla sistemde bir açık olduğu anda sisteme gönderilen paketler ile yapılan saldırılardır. Probing saldırı türünde ise saldırgan sistem hakkında bilgi toplayarak sisteminin zayıflıklarını bulmaktadır. Ardından bu zayıflıklardan yola çıkarak çeşitli araçlar ile saldırısını gerçekleştirmektedir.

UNSW-NB_{15} veri kümesi ise Avustralya Siber Güvenlik Merkezi'nin Cyber Range Laboratuvarındaki IXIA PerfectStorm aracı tarafından oluşturulmuştur. Bu veri kümesi 2007 yılındaki bir DDos saldırısından yaklaşık bir saatlik anonim trafik izlerini içermektedir (Yavanoglu ve Aydos, 2017). Veri kümesinde dokuz ayrı saldırı türü (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode ve Worms) ve 49 öznitelik bulunmaktadır. Bu özniteliklerin çıkarılmasında ise Argus, Bro-IDS araçları kullanılmış ve 12 model geliştirilmiştir. Öznitellikler akış özellikleri, temel özellikler, içerik özellikleri, zaman özellikleri ve oluşturulan ek özellikler olmak üzere beş gruba ayrılmıştır.

3.2. Öznitelik Seçimi

NSL-KDD veri kümesi 41 adet öznitelik ve UNSW-NB_{15} veri kümesi ise 49 adet öznitelik içermektedir. Bu özniteliklerin önem dereceleri farklıdır. Saldırı tespit sistemleri için NSL-KDD ve UNSW-NB_{15} gibi büyük boyutlu veri kümelerinde verimli bir şekilde sonuç alabilmek için bütün özniteliklerin kullanılması gerekli değildir. Ayrıca önem derecesi düşük ve gereksiz öznitelikler işlem süresi artıracak, performansı olumsuz etkileyecek ve verimliliği düşürecektir. Öznitelik seçimi ile hesaplama maliyetinin düşürüldüğü bir önişlem adımıdır. Veri kümelerinden öznitelik seçimi için boyu indirgeme ve öznitelik çıkarma için birçok yöntem bulunmaktadır. Bu çalışmada önem derecesi düşük olan özniteliklerin seçimi için Rasgele Orman algoritmasından yararlanılmıştır. Rasgele Orman algoritması, özniteliklerin önem derecesini ağaç temelli stratejilerin düğümün Gini safsızlığını (Gini impurity) ne kadar iyi arttırdıklarına göre

sıralamaktadır. Belirli bir düğümün altındaki ağaçları indirgeyerek en önemli özniteliklerin bir alt kümesini oluşturur. Ağaç düğümlerinin tüm ağaçtaki karmaşıklığı azaltılmasıyla bir özniteliğin önemini belirten yeni bir ağaç oluşturulur. Bu çalışmada da işlem adımları aşağıdaki yazıldığı gibi gerçekleştirilmiştir;

1. Öncelikle kullanılan veri kümeleri tanımlanarak veri kümeleri eğitim ve test kümelerine bölünmüştür.
2. Sklearn nesnesi kullanılarak veri kümesindeki tüm özelliklerin ortalama önem değerinden büyük olan öznitelikler seçilmiştir.
3. Eğitim aşamasından sonra her özellik için bir puan otomatik hesaplanarak tüm önem derecelerinin toplamı bire eşit olacak şekilde ölçeklendirilmiştir.
4. Elde edilen önem derecelerine göre kullanılacak olan öznitelikler belirlenerek bir alt küme oluşturulmuştur.
5. Oluşturulan yeni alt küme ile sınıflandırıcı tekrar eğitilerek deneyler yapılmıştır.

Sonuç olarak NSL-KDD veri kümesinde önem derecesine göre kullanılan öznitelik sayısı 15 iken UNSW-NB_{15} veri kümesinde önem derecesine göre kullanılan öznitelik sayısı 23'tür.

3.3. Kullanılan Topluluk Öğrenme Algoritmaları

Rasgele Orman: Bir karar ağacı sınıflandırma yöntemi olan Rasgele Orman sınıflandırması birden fazla karar ağacı kullanarak daha tutarlı sonuçların elde edildiği bir topluluk öğrenme modelidir. Hem regresyon hem de sınıflandırma problemlerinde kullanılan bu yöntem hiper-parametre kestirimi olmadan da iyi sonuçlar veren bir sınıflandırıcıdır (Şimsek, 2018). Eğitim sırasından birden fazla karar ağacı oluşturan bu yöntem kestirim aşamasında bu ağaçların sonuçlarından yola çıkarak girdi olan verinin çoğunluk oyu aracılığı ile karar vermesini sağlamaktadır (Kalaycı, 2018). Bu sınıflandırıcı aşırı uyum sorununu çözdüğü için başarılı sonuçlar vermektedir. Rasgele Orman algoritması 'n' tane karar ağacı ile oluşturulan modellerden oluşmaktadır. Bu algoritma iki aşamada işlemleri gerçekleştirmektedir. Birinci aşamada rasgele orman modeli oluşturulmaktadır. İkinci aşamada ise oluşturulan modelden sınıflandırıcı yardımı ile tahminler yapılmaktadır.

CatBoost: CatBoost veri kümesindeki kategorik öznitelikleri hızlı bir şekilde işleyen Gradyan Arttırma Karar Ağacı (*GBDT, Gradyan Boosted Decision Tree*) algoritmasıdır. Büyük veri kümelerini kullanan derin öğrenme modellerinin aksine bu algoritma az veriyle de başarılı sonuçlar elde etmektedir. Performanslı, kategorik verileri otomatik olarak işleyen ve kullanımı kolay olan bir algoritmadır. Geleneksel GBDT algoritmalarından farklı olarak önişleme süresi yerine eğitim süresi boyunca kategorik öznitelikleri ele almaktadır. Ayrıca Gradyan arttırmada kategorik özniteliklerin kullanılmasında çeşitli yöntemler kullanılsa da bu yöntemler tahminlerde kaymalara neden olmaktadır. Bu nedenle tahminleri iyileştirmek ve aşırı uyum sorununu çözmek için CatBoost algoritması önerilmiştir (Muratlar, 2020).

$$x_{\sigma_p}k = \frac{\sum_{j-1}^{p-1}\left[x_{\sigma_j}\mathrm{k}=x_{\sigma_p}\mathrm{k}\right] Y_{\sigma_J+a.P}}{\sum_{j-1}^{p-1}\left[x_{\sigma_j}\mathrm{k}=x_{\sigma_p}\mathrm{k}\right]+a} \quad k \in (1,d) \qquad (1)$$

Eşt. 1'de $D = \{(X_i, Y_i)\}$ belirli bir n kümesini temsil etmektedir. $X_i = (x_i, 1, \ldots, x_{i,} d)$ ise d özniteliklerinin bir vektörüdür. $\sigma = (\sigma_1, \ldots, \sigma_s)$, veri kümesinin s rastgele permütasyonlarının sayısıdır ve a, önceki P değerinin ağırlığıdır. Ayrıca CatBoost kategorik özelliği x_{σ_p}k değeri yukarıdaki formül ile elde edilir.

XGBoost: XGBoost, karar ağacı tabanlı, hızlı ve performanslı bir algoritmadır. XGBoost algoritması karar ağaçlarını oluştururken paralel çalıştırma yaptığı için işlemler hızlı bir şekilde sonuç vermektedir. Algoritma öncelikle *max_depth* ile ağacın derinliğini belirlemektedir. Ağaç aşağı yönde fazla derin ise geriye doğru budama yaparak işlemlerine devam etmektedir. XGBoost algoritmasında max_depth, min_child_weight ve gamma hiper-parametrelerini optimize edilmesi karmaşıklığı düşürmektedir. Sub_sample ve comsample_bytree hiper-parametrelerini optimize ederek oluşturulan modelin rasgeleliği arttırmaktadır. Bu şekilde modelin veriyi ezberlemesini engelleyerek aşırı öğrenme sorununa çözüm bulmaktadır. En önemli özelliklerinden bir tanesi ise verileri ağaçlara ayırırken doğru noktayı ayırmak için veri kümesindeki gözlem noktalarını ağırlıklarına göre kullanıyor olmasıdır (Muratlar, 2020). Veri kümesindeki eksik değer eğilimlerini de tespit eden ve genel anlamda minimum gereksinimle başarılı sonuçlar üreterek tahminler yapan bir algoritmadır.

LighGBM: LightGBM algoritması ise histogram tabanlıdır. CatBoost algoritmasına benzer şekilde öznitelik adlarının girdi olarak alarak kategorik öznitelikleri işlemektedir. LightGBM algoritmasında karar ağaçlarının eğitim süresi, yapılan hesaplama ve dolayısıyla bölünme sayısı ile doğru orantılıdır. Bu özelliği sayesinde hem eğitim süresi kısa olmakta hem de kaynak kullanımı düşmektedir. Bu algoritma karar ağacı algoritmalarına dayandığı için, ağacı en uygun olacak şekilde yaprak bilgisine bölerken, diğer artırma algoritmaları ağacı yaprak bazında değil, derinlik bazında veya seviye bazında bölmektedir. Bu nedenle, LightGBM algoritması aynı yaprak üzerinde büyüdüğünden yaprak-bazlı algoritma ve seviye-bazlı algoritmadan daha fazla kaybı azaltmaktadır. Bu şekilde mevcut artırma algoritmalarından herhangi biri tarafından elde edilebilecek çok daha iyi doğruluk değerlerine ulaşılmaktadır (Khandelval, 2017). Ayrıca büyük boyutlu verileri işleyebilen ve yüksek düzeyde optimize edilmiş bir karar ağacı öğrenme algoritmasıdır.

3.4. Hiper-Parametre Optimizasyonu

Parametre optimizasyonu genel anlamda belirli makine öğrenimi algoritmalarını büyük boyutlu probleme doğrudan uygulamadan önce problemi daha küçük problem parçacıklara ayırmaktır. Ardından farklı parametre kombinasyonları uygulayarak amaç fonksiyonu sonucunda en iyi çıktıyı üreten parametreyi elde etmektir. Parametre optimizasyonu genelde hiper-parametre optimizasyonu olarak karşımıza çıkmaktadır. Parametrelerden farlı olarak hiper-parametre değerlerinin eğitim aşamasında öğrenilmesi mümkün değildir. Modelleme kısmına gelmeden önce veri bilimci tarafından belirlenmektedir. Hiper-parametre optimizasyonu ise makine öğrenimi algoritmalarında en iyi sonucu veren hiper-parametreleri bulma işlemlerinden oluşmaktadır. Bu işlemler oldukça zaman alıcı ve uğraştırıcı işlemlerdir. Izgara araması, rasgele arama, bayes optimizasyonu, gradyan tabanlı optimizasyon, evrimsel optimizasyon ve erken durdurmaya dayalı gibi seçim işlemini gerçekleştiren çeşitli yöntemler kullanılmaktadır (Hiper-parametre Optimizasyonu, 2021). Gerçekleştirilen çalışmada ise bu yöntemlerden biri olan

rastgele arama yöntemi kullanılmıştır. Rasgele arama yöntemi, tüm kombinasyonların kapsamlı numaralandırılmasını rastgele değiştiren bir yöntemdir. Rasgele arama algoritması, ızgara aramasını temel alan bir yöntem olsa da optimum seviyeye yakın bir performans elde etmesinden dolayı ızgara araması yönteminden daha başarılı sonuçlar elde edildiği kanıtlanmıştır (Altun and Talu, 2021). Rasgele arama algoritmasında, hiper-parametre optimizasyonu sonucunda başarılı çıktılar elde edebilmek için çok sayıda ağaç ile daha az varyansla optimum bir model oluşturulur. Ayrıca her bölmede dikkate alınacak özelliklerin sayısı, çapraz doğrulama yöntemi ile elde edilir. Bu yöntem ile bir dizi değerin rastgele kombinasyonları denenmektedir. Bu optimizasyon yönteminde en önemli hiper-parametreler 'n_estimators', 'max_depth', 'max_features', 'bootstrap', 'min_samples_split' ve 'min_sample_leaf' hiper-parametreleridir. n_estimators ağaç sayısını temsil etmektedir. Sklearn'de varsayılan değer 100'dür. Genellikle veri boyutu ile ilişkili bir parametredir. max_depth, bir ağaç için izin verilen maksimum seviyedir. Ayarlanması yapılmaz ise ağaç saflığa ulaşana kadar bölünmeye devam edecektir. max_features, bir düğümde bölme işlemi için kullanılan maksimum özellik sayısını temsil etmektedir. Bootstrap parametresi ise "*true*" olarak belirlenmelidir. Aksi takdirde karar ağaçları oluşturulurken bootstrap örnekleri yerine her karar ağacı için tüm veriler kullanılır. min_samples_split parametresi ise bir düğümü bölmek için gereken minimum sayısıdır. min_sample_leaf parametresi de karar ağacının bir düğümündeki minimum veri noktasının sayısını ayarlamaktadır. Hiper-parametrelerin uygun değerlerinin ayarlaması sonucunda optimizasyon için deneyler yapılarak sonuçlar değerlendirilmelidir. Bu tekniğin düşük boyutlu verilerde iyi sonuçlar vermesinin nedeni ise doğru kümenin bulunması için geçen sürenin ve yineleme sayısının az olmasındandır (Senapati, 2018).

3.5. Performans Metrikleri

Çalışmada kullanılan sınıflandırıcıların performansını analiz ederken göz önünde bulundurulan metrikler Doğruluk (ACC), Kesinlik (PRE), Duyarlılık (REC), F1 değeri (FSC) ve İşlem Karakteristik (ROC, Receiver Operating Characteristic)'dir.

Doğruluk: Sınıflandırma modellerinin performans ölçümlerinden biri olan doğruluk değeri doğru tahmin edilen verilerin tüm tahmin değerlerine oranını göstermektedir. Eşt. 2'de gösterilen formül ile elde edilmektedir (Öğündür, 2019).

$$\text{Doğruluk} = \frac{TP+FP}{TP+TN+FP+FN} \quad (2)$$

Duyarlılık: Doğru sınıflandırılan pozitif olan verilerin, toplam pozitif olan verilere oranına denir. Bu metrik kullanılarak oluşturulan modelin verilerinden, pozitif sınıf etiketlerinin bulmasındaki oranı belirlenmektedir. Eşt. 3'te gösterilen formül ile hesaplanmaktadır.

$$\text{Duyarlılık} = \frac{TP}{TP+FN} \quad (3)$$

Kesinlik: Duyarlılıktan farklı olarak bu ölçüt ile doğru olan sınıflandırması yapılan pozitif olan örneklerin, toplam pozitif tahmin edilen örneklere oranı ölçülmektedir. Eşt. 4 ile verilmektedir.

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (4)$$

F-ölçütü: Duyarlılık ve kesinlik performans ölçütlerinin harmonik bir ortalaması olan F değeri bu özelliği sayesinde iki farklı performans ölçütünü kendi bünyesinde değerlendirmektedir. Bu metrik tek bir karşılaştırma ölçüsü vermektedir (Eşt. 5).

$$\text{F-ölçütü} = \frac{2*Duyarlılık*Kesinlik}{Duyarlılık+Kesinlik} \quad (5)$$

ROC: Makine öğrenimi algoritmalarının performans ölçümlerinde sıklıkla kullanılan bir ölçüm değeridir. Oluşturulan modelin tahmininde ne derecede başarılı olduğunu göstermektedir. ROC bir olasılık eğrisidir ve x ekseninde FPR, y ekseninde ise TPR değerleri bulunmaktadır. ROC eğrisinde her bir nokta ise belirli bir karar eşiğine karşılık olan duyarlılık çiftini temsil etmektedir (Aksu, 2020).

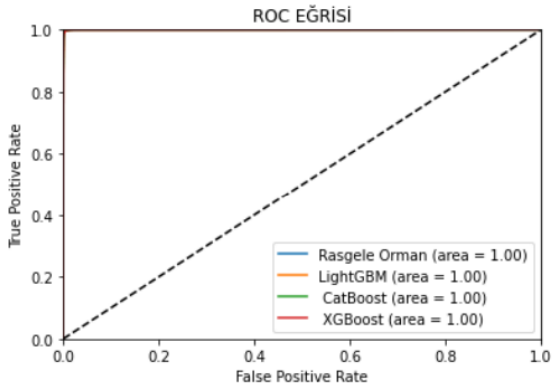
4. Deneysel Sonuçlar ve Tartışma

Bu çalışmada, gerçek zamanlı veri kümelerinden yola çıkarak normal ve anormal olma durumuna göre saldırının olduğu ya da olmadığı durumlar tahmin edilmektedir. Yapılan çalışma Python programla dilinde gerçekleştirilmiştir. Deneyler Windows 10 işletim sisteminde, Python'un 3.8.2 sürümünde yapılmıştır. Gerçekleştirilen tüm deneyler, 8 GB RAM'e sahip olan Intel(R) Core (TM) i7-5500U CPU @ 2.40GHz bir bilgisayarda yapılmıştır. Gerçekleştirilen çalışmada saldırıların tespiti için Rasgele Orman, CatBoost, XGBoost ve LightGBM algoritmalarının NSL-KDD ve UNSW-NB_{15} veri kümelerinde gerçekleştirimi sonucu elde edilen deney sonuçları ele alınmıştır. Algoritmalara uygulanan rasglele arama yöntemi ile hiper-parametreler optimize edilmiştir. Öncelikle Tablo 2'de NSL-KDD veri kümesinin ilk deney sonucundaki elde edilen ölçümler gösterilmiştir.

Tablo 2. NSL-KDD Veri Kümesi Test Sonuçları

	PRE	**REC**	**FSC**	**AUC**	**ACC**
Rasgele Orman	%99.99	%99.99	%99.99	%99.98	%99.99
CatBoost	%99.99	%99.99	%99.99	%99.98	%99.99
XGBoost	%99.99	%99.99	%99.99	%99.97	%99.99
Light GBM	%99.99	%99.99	%99.99	%99.98	%99.99

Performans metriklerinin sonucuna göre NSL-KDD veri kümesinde dört algoritma ile (%99.99) değerinde doğruluk skoru elde edilmiştir. Bu veri kümesinde algoritmalar başarılı performanslar göstermiştir. Şekil 1'de ise NSL-KDD veri kümesinin test sonuçlarını içeren ROC eğrisi gösterilmiştir. Bu şekilde elde edilen AUC değeri ise Rasgele Orman algoritmasında ve diğer üç algoritmada da 1.00 şeklinde elde edilmiştir.



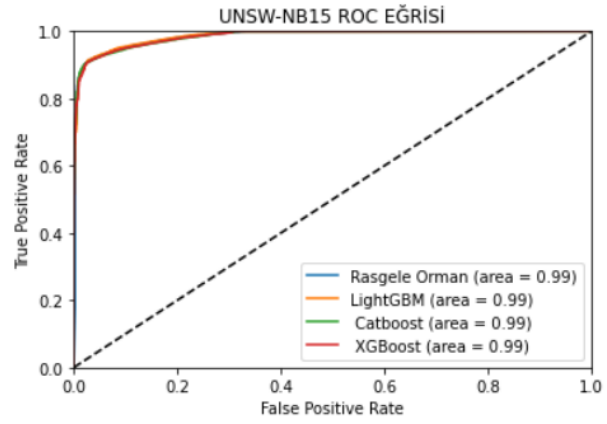
Şekil 1. NSL-KDD ROC Eğrisi

Tablo 3'te ise UNSW-NB$_{15}$ veri kümesinin performans metriklerinin sonuçları gösterilmiştir. Buradaki veri kümesinde en yüksek performans değeri Rasgele Orman algoritması (%93.62) ile elde edilirken en düşük performans değerini LightGBM algoritması (%93.29) ile elde edilmiştir.

Tablo 3. *UNSW-NB$_{15}$ Veri Kümesi Test Sonuçları*

	PRE	REC	FSC	AUC	ACC
Rasgele Orman	%93.63	%99.63	%99.54	%98.43	%93.62
CatBoost	%93.56	%99.63	%96.50	%98.55	%93.55
XGBoost	%93.47	%99.60	%96.44	%98.53	%93.43
LightGBM	%93.26	%99.66	%96.36	%98.67	%93.29

Şekil 2'de ise UNSW-NB$_{15}$ veri kümesinin test sonuçlarını içeren ROC eğrisi gösterilmiştir. Bu şekilde elde edilen AUC değerleri ise en yüksek (%98.67) ile LightGBM algoritması ile elde edilmiştir.



Şekil 2. UNSW-NB$_{15}$ ROC Eğrisi

Hiper-parametre optimizasyonundan önce en yüksek doğruluk değeri, NSL-KDD veri kümesinde Rasgele Orman sınıflandırıcısı (%99.99) ile UNSW-NB$_{15}$ veri kümesinde ise Rasgele Orman, XGBoost ve LightGBM sınıflandırıcısı ile elde edilmiştir. Hiper-parametre optimizasyonu sonucunda algoritmaların doğruluk değerlerinde artış olmuştur.

Optimizasyon sonucu en yüksek doğruluk değeri NSL-KDD veri kümesinde XGBoost sınıflandırıcısıyla, UNW-NB$_{15}$ veri kümesinde ise Rasgele Orman sınıflandırıcısı ile elde edilmiştir.

Rasgele arama yöntemi ile optimize edilen hiper-parametreler NSL-KDD veri kümesi için Tablo 4 ile gösterilmiştir. Bu optimizasyonu sonucu NSL-KDD veri kümesinde Rasgele Orman algoritması ile en iyi hiper-parametreler ele alınarak (%99.99) doğruluk değeri elde edilmiştir. CatBoost algoritmasında ise *depth*=9, *iterations*=38 ve *learning_rate*=2.7 hiper-parametrelerine karşılık %99.36 değerinde doğruluk skoru elde edilmiştir. LighGBM algoritmasında ise *scale_post_weight*=12 değeri ile (%99.92) doğruluk değeri elde edilmiştir. XGBoost algoritmasında da hiper-parametre optimizasyonu sonucu (%99.99) doğruluk değerine ulaşılmıştır.

Tablo 4. NSL-KDD Veri Kümesi Topluluk Öğrenme Algoritmaları için Hiper-parametreler

Algoritma	Hiper-parametre	Değer
Rasgele Orman	n_estimators	10
	max_depth	10
	depth	9
CatBoost	iterations	38
	learning_rate	2.70
LightGBM	scale_post_weight	12
	min_child_samples	100
XGBoost	max_depth	10
	learning_rate	0.1
	min_child_weight	10

Tablo 5 ile rasgele arama yöntemi ile optimize edilen hiper-parametrelerin UNSW-NB$_{15}$ veri kümesi için değerleri gösterilmiştir. UNSW-NB$_{15}$ veri kümesinde Rasgele Orman algoritmasına rasgele arama yönteminin uygulanması sonucunda (%96.86) değerinde doğruluk değeri elde edilmiştir. CatBoost ve LightGBM algoritmalarında bu değerler sırasıyla (%98.53) ve (%98.63) olarak elde edilmiştir.

Tablo 5. UNSW-NB$_{15}$ Veri Kümesi Topluluk Öğrenme Algoritmaları için Hiper-parametreler

Algoritma	Hiper-parametre	Değer
Rasgele Orman	n_estimators	10
	max_depth	10
CatBoost	depth	8
	iterations	38
	learning_rate	2.70
LightGBM	scale_post_weight	6
	min_child_samples	100
XGBoost	max_depth	10
	learning_rate	0.2
	min_child_weight	10

Tablo 6. *NSL-KDD ve UNSW-NB$_{15}$ Veri Kümelerinin Hiper-Parametre Optimizasyonu Sonucu Doğruluk Değerleri*

Veri Kümesi	Topluluk Öğrenme Algoritmaları			
	Rasgele Orman	CatBoost	XGBoost	LightGBM
NSL-KDD	%99.99	%99.36	%99.99	%99.92
UNSW-NB_{15}	%96.86	%98.53	%98.63	%98.54

Kullanılan her iki veri kümesinde elde edilen doğruluk değerleri Tablo 6 ile gösterilmiştir. Rasgele arama yöntemi ile hiper-parametrelerin optimizasyonu sonucu elde edilen performans sonuçları incelendiğinde kullanılan sınıflandırıcılar arasında çok farklılık olmadığı görülmüştür. Bu doğrultuda Rasgele Orman, XGBoost, CatBoost ve LighGBM sınıflandırıcılarının her iki veri kümesinde de normal ve anormal saldırıların tespit edilmesi işlemlerinde başarılı olduğu görülmektedir.

Literatürde NSL-KDD ve UNSW-NB_{15} veri kümesini kullanarak çok sayıda çalışma yapılmıştır. Bu çalışmada önerilen yöntem ile elde edilen performans ölçütlerinin değerleri literatürde önerilen yöntemler ile elde edilen performans değerleri ile kıyaslandığında önerilen yöntemin etkinliği ve üstünlüğü kanıtlanmaktadır. Bu çalışmada, veri boyutunun indirgenmesi sonucunda yüksek performans değerleri elde edilmiştir. Çalışmada kullanılan veri kümelerinden biri olan NSL-KDD veri kümesi için en iyi performans değerleri Hakim vd. (2019) tarafından Naive Bayes algoritması ile elde edilen doğruluk değeri (%89.40) iken, bu çalışmada öznitelik seçim işlemlerinin uygulanmasından sonra %97.00 doğruluk değerine ulaşılmıştır. Diğer bir çalışmada, hem NSL-KDD hem de UNSW-NB_{15} veri kümelerini kullanan Kamarudin vd. (2017) web saldırılarını tespit etmek için çeşitli deneyler yaptı. Deneyler sonucunda NSL-KDD veri kümesinde Logitboost ve Rasgele Orman algoritmalarının birleşimi sonucunda %90.33 doğruluk değeri elde edilmiştir. UNSW-NB_{15} veri kümesi kullanılarak yapılan deney sonucunda da aynı şekilde bu iki algoritmanın birleştirilmesi ile %99.45 doğruluk değeri elde edilmiştir. Literatürde bulunan diğer mevcut çalışmalarda aynı bu doğrultudaki teknikleri kullanarak yüksek çıktılar elde etmeyi amaçlamıştır. Gerçekleştirilen bu çalışmada da öznitelik seçiminden sonra en yüksek skor her iki veri kümesinde de Rasgele Orman algoritması ile elde edilmiştir. Bu açıdan yapılan çalışmanın diğer yayınlara benzer şekilde her iki veri setinde de yüksek sonuçlar elde ettiği görülmüştür.

5. Sonuç

Saldırı tespit sistemleri, ağlara ve sistemlere olan saldırıların tespit edilmesi için kullanılan sistemlerdir. Bir saldırı tespit edildiğinde veya anormal davranışların tespitinde bu sistemler uyarıyı yönlendiriciye iletmektedir. Ağ üzerindeki saldırıların analizi için elde edilen veriler en önemli bileşenler olarak kabul edilmektedir. Saldırı tespitinin ele alındığı çalışmalarda tutarlı sonuçların elde edilmesi için verilerin seçimi ve kullanımı da oldukça önemlidir. Yapılan çalışmalarda amaç ağda gerçekleşen aktivitelerin izlenmesi ve trafiğin kontrol edilerek güvenliğin sağlanmasıdır. Bu çalışmada saldırı tespit sisteminin performansını ölçmek için NSL-KDD ve UNSW-NB_{15} veri kümeleri kullanılarak ağ üzerinde gerçekleşen saldırıların tespit edilmesi amaçlanmıştır. İkili sınıflandırma ayarları dikkate alınarak topluluk öğrenme algoritmalarının performansı ölçülerek çeşitli deneyler yapılmıştır. Ayrıca literatürde bu veri kümelerini kullanarak yapılan çalışmaların çeşitli sınıflandırıcılar kullanması sonucunda elde edilen performans sonuçları incelenmiştir. Mevcut olan çalışmalara benzer doğrultuda algoritmaların analiz edilmesinde doğruluk, hassasiyet, f-ölçütü ve ROC eğrisi dikkate alınmıştır. Yapılan deneyler sonucunda elde edilen performanslar arasında en başarılı sınıflandırıcının her iki veri kümesinde de Rasgele Orman algoritması olduğu görülmüştür. Deneysel sonuçlarda tutarlı sonuçlar elde edilmesi için öznitelik seçimi yapılarak test verileri üzerindeki doğruluğu arttırmak amaçlanmıştır. NSL-KDD veri kümesinin ikili sınıflandırmasında önem derecesine göre 15 öznitelik kullanılırken UNSW-NB_{15} veri kümesinde önem derecesine göre 23 öznitelik kullanılmıştır. Öznitelik seçiminden sonra önem derecesine göre ele alınan test verileri ile literatürdeki çalışmalar ile kıyaslandığında önerilen yöntem ile daha yüksek performans değerleri elde edilmiştir.

Gelecekteki çalışmalarda ise farklı öznitelik seçim ve sınıflandırıcılarının anomali tespitindeki başarıları analiz edilecektir. Derin öğrenme modellerinin dahil edilmesiyle önerilecek bir Saldırı Tespit Sistemi'nin başarımı literatürdeki diğer sistemler ile kıyaslanacaktır. Hiper-parametre optimizasyonunda metasezgisel yöntemlere başvurularak algoritmaların performans analizlerinin yapılacaktır. Aşırı öğrenme probleminin çözülmesinde kullanılan birçok yöntem mevcuttur. Örneklem sayısı ile veri boyutunu büyütmek, L1 ve L2 düzenleme (regularization) yöntemi ile modelin karmaşıklığının azaltmak ya da L1 veya L2 düzenlemesinde olduğu gibi, aşırı karmaşık bir model daha fazla uyum sağladığından katmanları kaldırarak modelin karmaşıklığını doğrudan azaltabilmek bu yöntemlerin bazılarıdır. Gelecek çalışmalarda bahsedilen bu tekniklerin kullanılması ve farklı veri kümelerinin topluluk öğrenme yöntemleri ile bir araya getirilmesi ile daha iyi sonuçlar elde edilmesi planlanmaktadır.

Kaynakça

Aburomman, A. A., & Reaz, M. B. I. (2017). A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems. *Information Sciences*, *414*, 225-246.

Aksu Ç., (2020). ROC eğrisi nedir? Erişim adresi: https://www.datasciencearth.com/roc-egrisi-nedir/

Aleesa, A., Younis, M., Mohammed, A. A., & Sahar, N. (2021). Deep-Intrusion Detection System With Enhanced UNSW-NB15 Dataset Based On Deep Learning Techniques. *Journal of Engineering Science and Technology*, *16*(1), 711-727.

Altunay, H. C., & Albayrak, Z. (2021). Network Intrusion Detection Approach Based on Convolutional Neural Network. *Avrupa Bilim ve Teknoloji Dergisi*, (26), 22-29.

Altun, S., & Talu, M. F. Derin Sinir Ağları için Hiperparametre Metodlarının ve Kitlerinin İncelenmesi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, *12*(2), 187-199.

Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE transactions on computers*, *65*(10), 2986-2998.

Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, *378*, 484-497.

Bamakan, S. M. H., Wang, H., Yingjie, T., & Shi, Y. (2016). An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. *Neurocomputing*, *199*, 90-102.

Baykan, N. A., & Khorram, T. (2021). Network Intrusion Detection using Optimized Machine Learning Algorithms. *Avrupa Bilim ve Teknoloji Dergisi*, (25), 463-474.

Bostani, H., & Sheikhan, M. (2017). Modification of supervised OPF-based intrusion detection systems using unsupervised learning and social network concept. *Pattern Recognition*, *62*, 56-72.

Chu, W. L., Lin, C. J., & Chang, K. N. (2019). Detection and classification of advanced persistent threats and attacks using the support vector machine. *Applied Sciences*, *9*(21), 4579.

Divekar, A., Parekh, M., Savla, V., Mishra, R., & Shirole, M. (2018, October). Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* (pp. 1-8). IEEE.

Hakim, L., & Fatma, R. (2019, October). Influence analysis of feature selection to network intrusion detection system performance using nsl-kdd dataset. In *2019 International conference on computer science, information technology, and electrical engineering (ICOMITEE)* (pp. 217-220). IEEE.

Hiperparameter Optimization. (2021). Wikipedia. Erişim: https://en.wikipedia.org/wiki/Hyperparameter_optimization

Kalaycı, T. E. (2018). Kimlik hırsızı web sitelerinin sınıflandırılması için makine öğrenmesi yöntemlerinin karşılaştırılması. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, *24*(5), 870-878.

Kamarudin, M. H., Maple, C., Watson, T., & Safa, N. S. (2017). A logitboost-based algorithm for detecting known and unknown web attacks. *IEEE Access*, *5*, 26190-26200.

Kasongo, S. M., & Sun, Y. (2020). Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB_{15} Dataset. Journal of Big Data, 7(1), 1-20.

Khandelval P., (2017). Which algorithm takes the crown: Light GBM vs XGBoost? Erişim adresi: https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

Muratlar E. R., (2020). CatBoost nedir? Diğer Boosting Algoritmalarından Farkı Nedir? Erişim adresi: https://www.veribilimiokulu.com/catboost-nedir-diger-boosting-algoritmalarindan-farki-nelerdir/

Muratlar E. R., (2020). XGBoost Nasıl Çalışır? Neden İyi Performans Gösterir? Erişim adresi: https://www.veribilimiokulu.com/xgboost-nasil-calisir/

Öğündür G., (2019). Doğruluk (Accuracy) , Kesinlik(Precision) , Duyarlılık(Recall) ya da F1 Score ?. Erişim adresi: https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy- kesinlik-precision-duyarl%C4%B1l%C4%B1k-recall-ya-da-f1-score-300c925feb38

Primartha, R., & Tama, B. A. (2017, November). Anomaly detection using random forest: A performance revisited. In *2017 International conference on data and software engineering (ICoDSE)* (pp. 1-6). IEEE.

Senapati D., (2018). Grid Search vs. Random Search. Erişim adresi: https://medium.com/@senapati.dipak97/grid-search-vs-random-search-d34c92946318

Sethi, K., Rupesh, E. S., Kumar, R., Bera, P., & Madhav, Y. V. (2020). A context-aware robust intrusion detection system: a reinforcement learning-based approach. *International Journal of Information Security*, *19*(6), 657-678.

Simsek, H. K., (2018). Makine Öğrenmesi Dersleri 5a: Random Forest (Sınıflandırma). Erişim adresi: https://medium.com/data-science-tr/makine-%C3%B6%C4%9Frenmesi-dersleri-5- bagging-ve-random-forest-2f803cf21e07

Tama, B. A., Comuzzi, M., & Rhee, K. H. (2019). TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. *IEEE Access*, *7*, 94497-94507.

Tama, B. A., & Rhee, K. H. (2019). An in-depth experimental study of anomaly detection using gradient boosted machine. *Neural Computing and Applications*, *31*(4), 955-965.

Yang, Y., Zheng, K., Wu, C., Niu, X., & Yang, Y. (2019). Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. *Applied Sciences*, *9*(2), 238.

Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2019). Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. *Sensors*, *19*(11), 2528.

Yavanoglu, O., & Aydos, M. (2017, December). A review on cyber security datasets for machine learning algorithms. In *2017 IEEE international conference on big data (big data)* (pp. 2186-2193). IEEE.

Wu, P., Guo, H., & Moustafa, N. (2020, June). Pelican: A deep residual network for network intrusion detection. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 55-62). IEEE.