EMBER 2024 Üzerinde 5-Fold ve Temporal Değerlendirme ile Malware Sınıflandırma: Hibrit SAE–Top-K–LightGBM Modeli

[1]Sultan Tazefidan, [2]Heya Meylem ve [3]Gülay Çiçek
[1,2] Yazılım Mühendisliği Departmanı, Mühendislik-Mimarlık Fakültesi
İstanbul Beykent Üniversitesi, Sarıyer, İstanbul, Türkiye
[1]sultantazefidan.1@gmail.com [2]heyameylem96@gmail.com [3]gulaycicek@beykent.edu.tr

I. Literatür Çalışması

Bu bölümde, zararlı yazılımların (malware) sınıflandırılmasına yönelik gerçekleştirilen güncel literatür çalışmaları kapsamlı bir biçimde incelenmiştir. Çalışmalar, yöntemsel yaklaşımlarına göre makine öğrenimi, derin öğrenme ve hibrit modeller olmak üzere üç ana grupta sınıflandırılmıştır. Her grup için ilgili araştırmalar özet tablolar halinde sunulmuş, kullanılan yöntemler, veri setleri ve elde edilen performans sonuçları detaylı biçimde analiz edilmiştir. Bölümün sonunda ise, literatürdeki mevcut eğilimler ve boşluklar değerlendirilerek, bu çalışmanın özgün katkı ve farklılaştırıcı yönleri tartışılmıştır.

A. *Makine Öğrenmesi Tabanlı Malware Sınıflandırma Çalışmaları*

Bu alt başlıkta, makine öğrenmesi tabanlı olarak gerçekleştirilen malware analiz çalışmalarına yer verilmiş olup, Tablo I'de bu çalışmalara ait özet bilgiler sunulmuştur.

Kincl ve ekibi (2025) [1], android uygulama tabanlı zararlı yazılım (malware) verilerini kullanarak knowledge graph (KG) embedding + makine öğrenmesi ML temelli bir analiz hattı (pipeline) önermiştir. Çalışmada, yaygın olarak kullanılan CICMalDroid 2020 veri seti kullanılmıştır. Veri setinde başlangıçta 17.000'den fazla android uygulaması bulunmakta olup, özellik çıkarımı sonrasında bu sayı 13.077 örneğe düşürülmüştür. Ardından, 850 benign (zararsız) ve 850 malware olmak üzere dengeli bir şekilde toplam 1.700 örnek üzerinde deneyler gerçekleştirilmiştir.

Araştırmada SVM, RF, KNN, MLP, LR, GNB, DT, AB, GB, GP, RG ve PA olmak üzere 12 farklı makine öğrenmesi sınıflandırıcısı değerlendirilmiş; veri %80 eğitim (1500 örnek) ve %20 test (200 örnek) olarak ayrılmıştır. Sonuçlara göre, KG tabanlı temsillerde doğruluk oranı %85,5–%96,0, PCA temsillerinde %77,0–%94,9 ve BOW temsillerinde %89,7–%98,4 aralığında elde edilmiştir. KG tabanlı yaklaşımlar, PCA'ya kıyasla daha yüksek doğruluk sağlamış ve bazı durumlarda BOW yöntemine yakın ya da üstün sonuçlar üretmiştir.

Çalışma, Android platformu, ikili (binary) sınıflandırma ve ML tabanlı KG-embedding yaklaşımı kapsamında değerlendirilebilir. Bununla birlikte, yalnızca statik analiz kullanılmıştır; bu nedenle görülmemiş (out-of-sample) zararlı yazılımların tespitinde sınırlılıklar bulunmaktadır. Gerçek zamanlı veya dinamik analiz henüz uygulanmamıştır. Gelecek çalışmalarda, dinamik analiz ve çoklu kenar (multi-edge) yapıların bilgi grafına entegrasyonu, yeni malware ailelerinin otomatik öğrenilmesi ve modelin inductive (öğrenme sonrası güncellenebilir) hale getirilmesi planlanmaktadır.

Vasan ve ekibi (2025) [2], farklı platformlarda (Windows, Android ve IoT) kötü amaçlı yazılımların tespiti için AEF (Adaptive Embedded Framework) adını verdikleri makine öğrenmesi temelli bir model önermiştir. Çalışmada, Canadian Institute for Cybersecurity (CIC) tarafından sağlanan CICMalDroid-2020 (Android), MalMem-Obfuscated (Windows, bellek tabanlı), Dumpware10 (Windows, bellek imajı) ve IoMT-2024 (IoT cihazları) veri setleri kullanılmıştır. Veri setlerinin örneklem büyüklükleri sırasıyla CICMalDroid-2020: 17.341, CIC-MalMem-2022: 58.596, IoMT-2024 ve Dumpware10: 4.292 örnektir.

Önerilen AEF modeli, gömülü özellik seçimi (embedded feature selection) ve PCA tabanlı dönüşümler içermekte olup, hem klasik makine öğrenmesi sınıflayıcıları (LR, RF, KNN, XGB) hem de derin öğrenme modelleri (1D-CNN, Bi-LSTM) ile karşılaştırılmıştır. Deneysel sonuçlara göre doğruluk oranları; CICMalDroid-2020 veri setinde %54,99–%96,46, Dumpware10 veri setinde %92,73–%97,44, MalMem-Obfuscated veri setinde %98,90–%99,98 ve IoMT-2024 veri setinde %89,80–%98,39 aralığında elde edilmiştir. Özellikle Windows tabanlı veri setlerinde (Dumpware10, MalMem) AEF modeli en yüksek doğruluğu sağlamış; Android veri setinde (CICMalDroid) ise çok sınıflı yapıya rağmen yüksek başarı korunmuştur.

Çalışmanın sınırlılığı olarak, kullanılan modellerin karmaşık yapısı ve yorumlanabilirliğinin kısıtlı olması vurgulanmıştır. Ayrıca, bazı veri setlerinde özellik önem sıralamalarının ve açıklanabilirliğin tam olarak analiz edilmediği belirtilmiş; derin mimarilerin henüz tam optimize edilmediği ifade edilmiştir. Gelecek çalışmalarda, önerilen AEF modelinin yorumlanabilirliğini artırmaya yönelik geliştirmeler yapılması ve daha derin sinir ağı mimarilerinin AEF yapısına entegre edilerek karmaşık ve yeni zararlı yazılım türlerinin tespitinde performansın artırılması hedeflenmektedir.

Güngör ve ekibi (2023) [3], Malimg görüntüleştirilmiş PE veri setini kullanarak 25 zararlı yazılım ailesi üzerinde sınıflandırma gerçekleştirmiştir. Çalışmada, Android APK dosyaları gri tonlamalı görüntülere dönüştürülmüş, bu görüntülerden elde edilen öznitelikler analiz edilerek makine öğrenmesi yöntemleri (LR, LDA, KNN, CART, RF, NB, SVM) ile sınıflandırılmıştır. Sonuçlar K-kat çapraz doğrulama yöntemi kullanılarak değerlendirilmiştir.

TABLE I: Makine Öğrenmesi Tabanlı Literatür Taraması Sonuçları

Yazarlar (Yıl)	Örneklem Büyüklüğü	Dil (belirtilmişse)	Platform	Yöntem	Sonuçlar	Eksiklikler	Gelecek Çalışmalar
Kincl ve ekibi (2025) [1]	CICMalDroid 2020, 1700 Örnek (Benign:850, Malware:850)	Python (dolaylı olarak belirtilmiş)	Android (Docker)	KG gömme (TransE, DistMult, ComplEx, HoLE, PCA, BOW) + 12 ML sınıflandırıcı	HoLE:%97.4, PCA:%94.9, TransE:%94.0	modelin genelleştirilememesi, Sadece statik analiz, yönsüz kenarlar	Dinamik analiz ve yönlü grafik yapıları eklenecek, model dış bilgi tabanlarıyla entegre edilerek genelleştirilebilir hale getirilecektir.
Vasan ve ekibi (2025) [2]	CICMalDroid-2020: 17 341 örnek; CIC-MalMem-2022: 58.596 örnek; IoMT-2024 ve Dumpware10:4292 örnek	-	Windows ve Android (çoklu platform)	LR, AEF — gömülü özellik seçimi, stacking (yığma), RF, KNN, XGBoost	LR:%54.99, AEF:%96.46 (CICMalDroid veri seti için)	Modelin yorumlanabilirliği sınırlı; derin mimariler henüz tam optimize edilmemiş; bazı veri setlerinde özellik katkı analizi yapılmamış.	Yorumlanabilirliğin artırılması, özellik katkı analizi ve görselleştirme araçlarının entegrasyonu; derin ağlarla genişletme planı.
Güngör ve ekibi (2023) [3]	Malimg, 50.000 örnek (25 sınıf, her biri 2000 örnek)	Python (sklearn kütüphanesi)	Android (APK)	SVM, RF, LR, LDA, KNN, CART, NB,	SVM:%68.94, RF:%97.44, LR:%84.22	Veri çeşitliliği sınırlı; derin öğrenme yontemleri uygulanmamış; model karmaşıklığı düşük	veri seti geliştirilmesi ve derin öğrenme yöntemlerinin de çalışmaya dahil edilmesi.
Islam ve ekibi (2023) [4]	CCCS-CIC-AndMal-2020, 400.000 örnek (benign:200K; malware:200K)	Python	Android (dinamik)	Ensemble ML, MV	Ensemble ML:%95 MV:%93.4	Modelin bazı sınıflarda (ör. Riskware–Adware) karışıklık göstermesi ve yanlış sınıflandırma oranlarının bulunması.	Önerilen modelin farklı ve yeni veri setleri üzerinde test edilerek etkinliğinin değerlendirilmesi planlanmaktadır.
Rawat ve ekibi (2022) [5]	Emotet dinamik ağ trafiği veri seti, 483 782 flow (Emotet:2.143, non-Emotet:481 639)	-	Windows	RF, MLP, NB, SMO, LR	RF:%99.9726 kesinlik ve %92,3 TPR	Model mevcut saldırı varyantlarıyla sınırlı ve yeni vektörlere duyarsız olabilir.	modelin yeniden eğitilmesi ve optimize edilmesi planlanıyor.
Zada ve ekibi (2024) [6]	MMCC (yaklaşık 10.000 örnek)	Python	Windows	KNN, NB, SGDC, DT	KNN:%93.3, NB:%99.54	Veri kalitesi, dengesizlik, öznitelik seçimi hataları, model genellenebilirliği eksikliği	Hibrit yaklaşımlar, ileri özellik mühendisliği, gerçek zamanlı analiz, düşük kaynakta çalışan modeller, siber güvenlik uzmanlarıyla işbirliğinin gerçekleştirilmesi.
Bakshi ve ekibi (2025) [7]	PE_Header, PE_Section, DLLs_Imported (yaklaşık 29.000 örnek)	Pyton	Windows	KNN, RF, DT	KNN:%90.01, RF:%98.13	derin öğrenme yöntemleri kullanılmamıştır ve genelleme sınırlıdır	derin öğrenme yöntemlerinin entegre edilmesi ve doğruluğu arttıremak için ek hibrit modeller önerilmektedir.

Not: Tablo, makine öğrenmesi tabanlı malware tespit çalışmalarının temel özelliklerini, yöntemlerini, performans karşılaştırmalarını ve sınırlılıklarını özetlemektedir.

Elde edilen bulgulara göre, en düşük doğruluk oranı %68,94 (SVM), en yüksek doğruluk oranı ise %97,44 (RF) olarak raporlanmıştır.

Çalışmada yalnızca Malimg veri setinin kullanılması, modelin karmaşıklığının düşük olması ve derin öğrenme yöntemlerinin uygulanmaması başlıca eksiklikler olarak belirtilmiştir. Gelecek çalışmalarda, veri setinin kapsamının genişletilmesi ve derin öğrenme tabanlı yöntemlerin kullanılmasıyla modelin doğruluk oranının ve genelleme yeteneğinin artırılmasının hedeflendiği ifade edilmiştir.

Islam ve ekibi (2023) [4], CCCS-CIC-AndMal-2020 veri seti üzerinde Android tabanlı kötü amaçlı yazılımların dinamik özelliklerinden yararlanarak ağırlıklı oylama (Weighted Voting) tabanlı bir topluluk makine öğrenmesi modeli önermiştir. Çalışmada Random Forest (RF), K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), Decision Tree (DT), Support Vector Machine (SVM) ve Logistic Regression (LR) olmak üzere altı farklı makine öğrenmesi algoritması bir araya getirilmiştir. Modelin alt katmanında RF ve KNN yer almakta, ikinci katmanda ise bu iki modelin çıktıları üzerinden ağırlıklı oylama yöntemiyle nihai tahmin gerçekleştirilmektedir.

Boyut indirgeme aşamasında PCA kullanılarak 141 öznitelik 45 bileşene düşürülmüş ve modelin hesaplama karmaşıklığı azaltılmıştır. Bireysel modeller arasında en düşük doğruluk %57,7 (LR), en yüksek doğruluk ise %90'ın üzerinde (RF ve KNN) olarak raporlanmıştır. Önerilen Weighted Voting Ensemble modeli, %95 doğruluk ile hem bireysel modelleri hem de geleneksel Majority Voting (MV) yöntemini (%93,4) geride bırakmıştır.

Çalışmada, modelin yüksek doğruluk elde etmesine rağmen bazı sınıflarda yanlış pozitif ve yanlış negatif örneklerin gözlendiği belirtilmiş; bu durumun modelin genelleme kabiliyeti ve sınıflar arası ayrım gücü açısından bir sınırlılık oluşturduğu ifade edilmiştir. Gelecek çalışmalarda, önerilen yöntemin farklı ve güncel veri setleri üzerinde test edilerek genellenebilirliğinin ve etkinliğinin değerlendirilmesi planlanmaktadır.

Rawat ve ekibi (2022) [5] tarafından gerçekleştirilen çalışmada, Windows işletim sistemlerini hedef alan Emotet trojanının oluşturduğu ağ trafiği dinamik analiz yoluyla toplanmış ve bu veriler kullanılarak ikili (binary) sınıflandırma yapılmıştır. Araştırmacılar, Emotet örneklerini sürüm 2'den sürüm 187'ye kadar on aylık bir süre boyunca dinamik olarak çalıştırarak özel bir Emotet dinamik ağ trafiği veri seti oluşturmuşlardır. Veri seti toplam 483.782 ağ akışından (2.143 Emotet akışı ve 481.639 non-Emotet akışı) oluşmaktadır. Sınıflandırma aşamasında RF, MLP, NB, SMO ve LR gibi makine öğrenmesi yöntemleri değerlendirilmiştir. Analiz sonuçlarına göre RF sınıflandırıcısı %99,9726 kesinlik ve %92,3 gerçek pozitif oranı (TPR) elde etmiştir. Çalışmada, zararlı yazılım davranışlarının ve enfeksiyon vektörlerinin zamanla değişebileceği; dolayısıyla geliştirilen modellerin gelecekte güncelliğini yitirebileceği bir sınırlılık olarak belirtilmiştir. Gelecek çalışmalarda ise yeni saldırı yöntemlerinin takibi, farklı trojan ailelerinin incelenmesi ve zararlı yazılım sınıflarındaki özellik/vektör değişimlerinin izlenmesi hedeflenmektedir.

Zada ve ekibi (2024) [6], tarafından gerçekleştirilen çalışmada, Windows sistemlerine yönelik kötü amaçlı yazılım saldırılarını tespit etmek amacıyla Microsoft Malware Classification Challenge (MMCC) veri seti kullanılmıştır. Veri setinde yaklaşık 10.000 kötü amaçlı yazılım örneği yer almakta olup, veriler %70 eğitim ve %30 test oranında bölünmüştür. Çalışmada binary sınıflandırma amacıyla KNN, NB, SGDC, DT, modelleri kullanılmıştır.

Ön işleme aşamasında, veri boyutunu azaltmak ve bellek kullanımını optimize etmek için bellek optimizasyonu uygulanmış; ardından model karmaşıklığını azaltmak ve overfitting riskini düşürmek amacıyla Principal Component Analysis (PCA) yöntemi kullanılmıştır.

Çalışmada iki aşamalı deneysel analiz gerçekleştirilmiştir. İlk aşamada, modeller varsayılan parametrelerle test edilmiş ve doğruluk oranları %53–59 aralığında kalmıştır. Bu sonuçlar, hiperparametre optimizasyonu yapılmadan modellerin temel performansını yansıtmaktadır. İkinci aşamada ise Grid Search ve Cross-validation yöntemleriyle parametre optimizasyonu yapılmış; performans önemli ölçüde artmıştır. Elde edilen sonuçlara göre en yüksek doğruluk %99.54 ile NB modelinden, en düşük doğruluk ise %99.3 ile KNN modelinden elde edilmiştir.

Çalışmanın sınırlılıkları arasında, eğitim verisinin kalitesi ve çeşitliliğinin modellerin performansını doğrudan etkilemesi, modelin genellenebilirliğinin sınırlı olabileceği ve kontrollü test koşullarında elde edilen sonuçların gerçek sistemlerde düşebileceği belirtilmiştir. Ayrıca, yanlış seçilen özellik çıkarımı ve ön işleme tekniklerinin modele gürültü veya önyargı ekleyebileceği, bu durumun doğruluk oranlarını olumsuz etkileyebileceği vurgulanmıştır. Sınırlı donanım kaynaklarına sahip sistemlerde bu modellerin uygulanabilirliğinin de kısıtlı olabileceği ifade edilmiştir.

Gelecek çalışmalar kapsamında, yeni tespit yaklaşımlarının geliştirilmesi, gelişmiş özellik mühendisliği, hibrit öğrenme yöntemlerinin entegrasyonu ve gerçek zamanlı analiz sistemlerinin uygulanması hedeflenmektedir.

Bakshi ve ekibi (2025) [7], tarafından gerçekleştirilen çalışmada, Windows tabanlı sistemlere zarar veren kötü amaçlı yazılımların sınıflandırılması amacıyla Figshare platformunda yer alan ve yaklaşık 29.000 örnek içeren üç farklı veri kümesi (PE_Header, PE_Section ve DLLs_Imported) kullanılmıştır. Her bir veri kümesi; SHA256 değerleri, etiket bilgileri (örneğin "Benign", "RAT", "Downloader", "Spyware" vb.) ve DLL/section başlıklarını içermektedir. Veri dengesizliğini gidermek amacıyla, veri seti %50 benign – %50 malware oranında dengelenmiştir.

Sınıflandırma sürecinde, Machine Learning-Based Malware Detection and Analysis Mechanism (MLBM-DAM) olarak adlandırılan çerçevede üç makine öğrenmesi modeli — KNN, RF ve DT — karşılaştırılmıştır. Deneysel sonuçlara göre en yüksek doğruluk oranı %98.13 ve en yüksek kesinlik (precision) değeri %98.21 ile RF modelinden elde edilmiştir. En düşük başarı ise %90.01 doğruluk ve %88.29 kesinlik oranlarıyla KNN modelinde gözlemlenmiştir.

Çalışmanın sınırlılıkları arasında, yalnızca makine öğrenmesi tabanlı modellerin değerlendirilmesi ve derin öğrenme yöntemlerinin entegrasyonunun eksikliği yer almaktadır. Ayrıca modelin genellenebilirlik kapasitesinin sınırlı olduğu ifade edilmiştir.

Gelecek çalışmalar kapsamında, derin öğrenme tabanlı yaklaşımların MLBM-DAM sistemine entegre edilmesi, mevcut yöntemin doğruluğunun artırılması amacıyla sinir ağı tabanlı hibrit modellerin geliştirilmesi ve kötü amaçlı yazılım tespit performansının artırılması hedeflenmektedir.

B. *Derin Öğrenme Tabanlı Malware Sınıflandırma Çalışmaları*

Bu alt başlıkta, derin öğrenme tabanlı malware sınıflandırmasına yönelik gerçekleştirilen literatür çalışmaları özetlenmiştir. İlgili çalışmaların karşılaştırmalı sonuçları Tablo II'de sunulmuştur.

Gupta ve ekibi (2025) [8] binary sınıflandırma problemine odaklanılmıştır. Araştırmada, Windows PE tabanlı zararlı (malware) ve benign dosyaların API çağrı dizileri kullanılmıştır. Veri kümesi, GitHub üzerinden elde edilen 2.570 örnekten ve yazarlar tarafından oluşturulan, 1.793 örnekten oluşan doğrulama veri setinden meydana gelmiştir.

Çalışmada, LSTM tabanlı bir RNN modeli ile üç farklı Çok Katmanlı Algılayıcı (MLP) mimarisi — ReLU, Tanh ve önerilen MLP — karşılaştırılmıştır. Windows API çağrıları embedding vektörlerine dönüştürülerek, veri seti %80 eğitim ve %20 test oranında ayrılmıştır.

Deneysel sonuçlara göre, en düşük doğruluk ve duyarlılık (recall) değerleri RNN modelinde sırasıyla %55.408 ve %40.77 olarak elde edilmiştir. Buna karşın, en yüksek doğruluk ve duyarlılık oranları önerilen MLP modelinde sırasıyla %89.309 ve %84.549 olarak hesaplanmıştır. Ayrıca, önerilen model nihai değerlendirme aşamasında %93.33 doğruluk oranına ulaşmıştır.

Çalışmada kullanılan veri setinin örnek sayısının az olması, modelin genellenebilirliğini sınırlamıştır. Bununla birlikte, k-fold çapraz doğrulama yöntemi uygulanmamış, ayrıca yüksek işlem gücü gereksinimleri nedeniyle epoch sayısı arttıkça modelin varsayılan davranışlar sergilediği gözlemlenmiştir.

Gelecek çalışmalarda, Transformer ve LSTM tabanlı hibrit modellerin kullanılması, daha büyük ve dengeli veri setleriyle yeniden eğitim yapılması ve modellerin gerçek zamanlı sandbox ortamlarına entegre edilmesi hedeflenmektedir.

Taheri ve ekibi (2024) [10] tarafından gerçekleştirilen çalışmada, Android tabanlı kötü amaçlı yazılımların (malware) tespiti için diferansiyel gizlilik (Differential Privacy, DP) ilkelerine dayalı DP-NC (Differential Privacy–Noise Cancellation) adlı bir derin öğrenme tabanlı savunma mekanizması önerilmiştir. Çalışma, Drebin (115.080 örnek), Genome (1.200 örnek) ve Contagio (13.670 örnek) veri setleri üzerinde, benign–malware ikili sınıflandırma odağında yürütülmüştür.

Araştırmada iki tür saldırı senaryosu ele alınmıştır: DP-NI (Data Poisoning with Noise Injection): Diferansiyel gizlilik ilkeleri doğrultusunda veriye gürültü ekleyerek modelin hatalı örnekleri öğrenmesini sağlayan saldırı türüdür. ve GDP (Gradient Differential Privacy): Modelin gradyan bilgilerini manipüle ederek gizlilik ihlali ve sınıflandırma hatalarına neden olan saldırı türüdür. Bu saldırılara karşı geliştirilen DP-NC savunma mekanizması, DNN tabanlı sınıflandırma modellerini DP-NI ve GDP saldırılarına karşı korumayı hedeflemektedir.

Deneysel sonuçlara göre: Drebin veri setinde, saldırı bulunmayan durumda doğruluk oranı %94.63, GDP saldırısı altında ise %27.27 olarak elde edilmiştir. Contagio veri setinde, en yüksek doğruluk saldırısız durumda %94.21, saldırı altında ise %34.89 olarak gözlemlenmiştir. Genome veri setinde ise, saldırısız durumda %94.63, saldırı altında %44.8 doğruluk oranına ulaşılmıştır. Çalışmanın başlıca eksiklikleri arasında, yüksek hesaplama maliyeti ve uzun eğitim süresi yer almaktadır. Ayrıca model, karmaşık saldırı senaryolarında sınırlı genellenebilirlik göstermiştir. Gelecek çalışmalar kapsamında, DP-NC savunma mekanizmasının hiperparametre ayarlamalarının ve mimari optimizasyonlarının yapılması, gerçek zamanlı Android malware tespitinde uygulanabilirliğin artırılması ve Transformer ile attention tabanlı modellerin diferansiyel gizlilikle birleştirilmesi planlanmaktadır.

Ghahramani ve ekibi (2024) [11] tarafından gerçekleştirilen çalışmada, Windows tabanlı kötü amaçlı yazılımların (malware) sınıflandırılması amacıyla UCI Malware veri seti kullanılmıştır. Veri seti, 100.000'den fazla zararlı yazılım örneği ile 486 statik özelliği (API, File System, Directory, Misc.) içermektedir.

Çalışmada Clustering, Similarity-based, Probabilistic, Lattice-based ve Deep Learning (Deep Image) yaklaşımları arasında karşılaştırmalı analiz yapılmıştır. Elde edilen sonuçlara göre, derin öğrenme yöntemi diğer yaklaşımların (%3.63–%7.31 doğruluk oranı) üzerinde performans göstererek %45.1 doğruluk ve %11.1 hata oranına ulaşmıştır. Probabilistic yaklaşım, doğruluk–performans dengesi açısından ikinci sırada yer almıştır.

Buna karşın, Clustering tabanlı yöntem yüksek hata oranı (%41.59) ve yüksek hesaplama maliyeti göstermiştir. Ayrıca modelin düşük doğruluk oranı, veri çeşitliliğinin sınırlı olmasından olumsuz etkilenmiştir. Gelecek çalışmalar kapsamında, modelin daha geniş veri kümeleriyle test edilmesi, Deep Image mimarisinin optimize edilmesi ve gerçek zamanlı tespit için hafif (lightweight) derin öğrenme modellerinin geliştirilmesi planlanmaktadır.

Çatak ve ekibi (2020) [13] tarafından gerçekleştirilen çalışmada, API çağrı dizilerine dayalı LSTM tabanlı bir derin öğrenme modeli kullanılarak Windows tabanlı malware hem binary hem de çok sınıflı (multi-class) sınıflandırması yapılmıştır. Çalışmada kullanılan veri kümesi, toplam 7.107 zararlı yazılım örneği içeren Windows PE API Sequence veri setidir. Bu veri seti, Adware, Backdoor, Downloader, Dropper, Spyware, Trojan, Virus ve Worm olmak üzere 8 farklı malware sınıfını kapsamaktadır.

Veri kümesi %80 eğitim ve %20 test olacak şekilde ikiye ayrılmıştır. Deneysel analizler Python 3.6.5 ortamında gerçekleştirilmiş olup, ana model olarak LSTM kullanılmıştır. Ayrıca karşılaştırma amacıyla KNN, DT, RF, RBF-SVM ve Sigmoid-SVM modelleri de uygulanmıştır.

İkili sınıflandırmada doğruluk oranı %83.5 ile %98.5 arasında değişirken, çok sınıflı sınıflandırmada ortalama F1 skoru %39–%47 aralığında elde edilmiştir. LSTM modeli, çok sınıflı sınıflandırmada %50 kesinlik ile en istikrarlı sonuçlardan birini göstermiştir. Diğer modellerden KNN %35, RF %46, SVM %78 ve DT %40 precision değeri elde etmiştir.

Bu çalışma, Windows PE tabanlı, hem binary hem de multi-class sınıflandırma yapan, derin öğrenme tabanlı bir yaklaşım olarak literatürde öne çıkmaktadır. Ancak, VirusTotal servisinin etiketleme doğruluğundaki sınırlılıklar bazı malware türlerinde yanlış sınıflandırmalara neden olmuş, dolayısıyla modelin genellenebilirliği kısıtlı kalmıştır. Gelecek çalışmalarda, LSTM modellerinin metamorphic malware gibi daha karmaşık zararlı yazılım türlerine uygulanması, sandbox ortamında çalışmayı tespit eden malware'lere karşı yeni yöntemlerin geliştirilmesi ve TF-IDF dışında sıralı veri sınıflandırıcılarının (RNN, GRU vb.) da değerlendirilmesi önerilmektedir.

Aryal ve ekibi (2025) [9] tarafından yapılan çalışmada, Windows PE dosyaları temel alınarak 6000 malware ve 1000 benign örnekten oluşan veri setiyle ikili (binary) malware sınıflandırması

TABLE II: Derin Öğrenme Tabanlı Literatür Taraması Sonuçları

Yazarlar (Yıl)	Örneklem Büyüklüğü	Dil (belirtilmişse)	Platform	Yöntem	Sonuçlar	Eksiklikler	Gelecek Çalışmalar
Gupta ve ekibi (2025) [8]	Birincil veri seti:2570 örnek, ikincil veri seti:1793 örnek	Python	Windows	RNN, üç farklı MLP (Relu, Tanh, MLP)	RNN:%55.408, MLP:%89.309	genelleme sınırlı ve Model, fonksiyon çağrılarının sıralı etkisini tam yakalayamamıştır.	Daha nüyük eri setinin kullanılması, sandbox ortamında entegrasyon ve Transformer ve LSTM temelli hibrit modellerin kullanılması önerilmiştir.
Aryal ve ekibi (2025) [9]	Windows PE dosyaları 7000 örnek (malware:6000, benign:1000)	Python	Windows	MalConv, MalConv2	MalConv:%92.77, MalConv2:%94.58	Zaman tabanlı (temporal) analiz yapılmamış ve model kıyaslaması sınırlıdır.	gerçek dünya black-box detektörlere karşı test edilip genelleştirilmesi hedeflenmektedir.
Taheri ve ekibi (2024) [10]	Drebin: 115.080 örnek, Genome:1200 örnek, Contagio:13.670 örnek	Python 3.10.12 (Google Colab ortamında)	Android	Dört katmanlı DNN modeli	DNN Doğruluk:%27.27, duyarlılık:%8.66 (Drebin veri seti için)	Hesaplama gücü ve eğitim süresi yüksek, düşük genellenebilirlik	mimari optimiazonların yapılması, Gerçek zamanlı Android malware tespitinin gerçekleştirilmesi önerilmektedir.
Ghahramani ve ekibi (2024) [11]	UCI Malware (100.000'den fazla örnek)	-	Windows	Deep Image–DNN	Deep Image – DNN:%45.1	Derin öğrenme modelinin hesaplama maliyeti yüksek ve model doğruluğu sınırlıdır.	Deep Image mimarisinin optimize edilmesi ve Daha geniş ve çeşitli veri kümelerinde test edilmesi planlanmaktadır.
Divakarla ve ekibi (2022) [12]	EMBER veri seti (%75 eğitim, %25 test.)	-	Windows	DNN	DNN:%96.76	temporal veya gerçek zamanlı saldırı senaryosu içermemektedir.	GAN tabanlı yaklaşımın gerçek dünya malware tespit sistemlerine uygulanabilirliğinin araştırılması önerilmektedir.
Çatak ve ekibi (2020) [13]	Windows PE API çağrı dizileri (7.107 örnek)	Python 3.6.5	Windows	LSTM ve ek olarak KNN, DT, RF, SVM	LSTM doğruluk:%98.5, F1:%47	VirusTotal etiketleme sistemi tam doğru değildir ve genellenebilirlik sınrılıdır.	Görsel tabanlı sınıflandırma tekniklerinin kullanılması ve LSTM modellerinin karmaşık türlerde de uygulanması planlanmaktadır.

Not: Tablo, derin öğrenme tabanlı malware tespit çalışmalarında kullanılan veri setlerini, model mimarilerini, başarı oranlarını ve sınırlılıkları özetlemektedir.

yapılmıştır. Çalışmada, CNN tabanlı derin öğrenme modelleri olan MalConv ve MalConv2 kullanılarak kötü amaçlı yazılımların tespiti gerçekleştirilmiştir. Eğitim aşamasında farklı veri bölme oranları denenmiş, 70:20:10 split konfigürasyonunda MalConv modeli %92.77, MalConv2 modeli %94.58 doğruluk oranına ulaşmıştır.

Alternatif olarak kullanılan 80:10:10 split konfigürasyonunda ise bu doğruluk oranları MalConv için %94 ve MalConv2 için %98 olarak rapor edilmiştir. Araştırmada ayrıca, intra-section code cave injection temelli yeni bir adversarial saldırı yöntemi geliştirilmiş ve bu saldırıların derin öğrenme modelleri üzerindeki etkisi analiz edilmiştir. Bu saldırılar sonucunda modellerin tespitten kaçınma oranları, yani evasion rate değerleri, MalConv için %92.31, MalConv2 için %97.93 olarak ölçülmüştür.

Elde edilen bu sonuçlar, söz konusu modellerin binary sınıflandırma görevinde adversarial örnekler karşısında ciddi zafiyetler barındırdığını göstermektedir. Bilimsel açıdan değerlendirildiğinde, çalışma önemli bir başarı ortaya koymaktadır; çünkü geliştirilen intra-section adversarial evasion yöntemi, mevcut literatürdeki tekniklerden çok daha yüksek kaçış başarısı sağlamıştır. Ancak siber güvenlik perspektifinden bakıldığında, bu durum olumsuz bir tabloyu ortaya koymaktadır; zira bu denli yüksek tespitten kaçınma oranları, derin öğrenme tabanlı malware dedektörlerinin pratik kullanımda yeterince güvenilir olmadığını ve adversarial saldırılara karşı savunmasız olabileceğini göstermektedir. Çalışmada zaman (temporal) tabanlı analiz yapılmamış ve model karşılaştırmaları sınırlı kalmıştır; farklı mimarilerin ve zamanlı veri bölünmelerinin etkisi incelenmemiştir. Önerilen yöntemin gerçek dünya black-box detektörlere karşı test edilerek genelleştirilmesi; temporal deneylerin yapılması, farklı model mimarilerinin karşılaştırılması ve runtime/dinamik senaryoların değerlendirilmesi hedeflenmektedir.

Divakarla ve ekibi (2022) [12] tarafından gerçekelştirilen çalışmada, statik analiz yöntemi kullanılarak JSON formatlı EMBER veri seti üzerinde ikili (binary) malware sınıflandırması yapılmıştır. Veri seti, 200.000 örnekten oluşmakta olup bunların 100.000'i benign, 100.000'i malicious olarak etiketlenmiştir. Veriler, %75'i eğitim ve %25'i test olacak şekilde ikiye ayrılmıştır. Çalışmada, GAN (Generative Adversarial Network) yöntemiyle geliştirilmiş DNN tabanlı hibrit model kullanılmıştır. Veriler vektörleştirilmiş ve one-hot encoded biçimde modele aktarılmıştır. Deneysel sonuçlara göre, temel DNN modeli %96.76 doğruluk oranı göstermiştir. Buna karşılık, GAN tabanlı geliştirilmiş hibrit modelin doğruluğu %97.42'ye yükselmiştir. Ayrıca sonuçlar, veri miktarı arttıkça model doğruluk oranının da yükseldiğini ortaya koymuştur.

Çalışma, yalnızca statik özelliklere (vektörize edilmiş veri) dayanmakta olup, dinamik analiz, temporal veya gerçek zamanlı saldırı senaryoları içermemektedir. Gelecek çalışmalarda, modelin daha geniş veri kümelerinde test edilmesi, dinamik analizle birleştirilmesi ve GAN tabanlı yaklaşımın gerçek dünya malware tespit sistemlerine uygulanabilirliğinin araştırılması planlanmaktadır.

C. Hibrit Yöntemlerle Yapılan Malware Sınıflandırma Çalışmaları

Bu bölümde, hibrit model entegrasyonu temelinde gerçekleştirilen malware sınıflandırma çalışmalarına yer verilmiş olup, bu çalışmalara ilişkin özet bilgiler Tablo III'te sunulmuştur.

Awwal ve ekibi (2025) [14] yaptıkları çalışmada, EMBER veri setinden türetilmiş üç farklı alt küme (2017–2018) kullanılarak Windows PE dosyaları üzerinde ikili (binary) sınıflandırma yapılmıştır. Çalışmada, Autoencoder + 1D-CNN + BiLSTM mimarilerini birleştiren ve BVR-SFO-AEDL optimizasyonu ile güçlendirilmiş hibrit bir derin öğrenme modeli önerilmiştir. Deneysel sonuçlara göre, 1.veri seti için doğruluk oranı %89.73–%96.46, 2.veri seti için %90.00–%96.53 ve 3.veri seti için %90.00–%96.73 aralığında elde edilmiştir. Benzer şekilde, F1-score değerleri 1.veri seti %93.29–%97.75, 2.veri seti %93.08–%97.65 ve 3.veri seti %91.97–%97.41 arasında raporlanmıştır. Bu bulgular, önerilen hibrit BVR-SFO-AE-DL modelinin tüm veri kümelerinde yüksek başarı gösterdiğini ortaya koymaktadır.

Model yüksek doğruluk oranlarına ulaşmasına rağmen overfitting riski taşımaktadır. Ayrıca çalışma, gerçek zamanlı uygulama ve farklı platform (ör. Android) testleri içermemektedir. Gelecek çalışmalarda modelin gerçek zamanlı dağıtık sistemlere entegrasyonu ve yapay zekâ tabanlı adaptif koruma sistemleriyle bütünleştirilmesi planlanmaktadır.

Rodrigo ve ekibi (2021) [17] tarafından gerçekleştiirlen çalışmada, Omnidroid veri setinden faydalanılarak 22.636 statik ve 2.210 dinamik özelliği içeren veriyle, %70 eğitim, %15 doğrulama ve %15 test oranları kullanılarak Android uygulamaları için binary malware sınıflandırması yapılmıştır. Önerilen hibrit modelin performans sonuçlarına göre; dinamik model %81.1 doğruluk ve %83.4 kesinlik, statik model %92.9 doğruluk ve %91.1 kesinlik, hibrit model ise %91.1 doğruluk ve %91.0 kesinlik oranına ulaşmıştır. Hibrit model, yalnızca statik veya dinamik yaklaşımlara göre daha yüksek genel performans göstermiştir.

Çalışmanın eksiklikleri arasında, iOS veya diğer mobil platformlarda test edilmemesi ve gerçek zamanlı analiz yeteneğinin sınırlı olması yer almaktadır. Gelecek çalışmalarda, veri setinin güncellenmesi, yeni özellik çıkarım araçlarının geliştirilmesi, modelin iOS platformuna genelleştirilmesi ve sürekli öğrenen sistemlerin tasarlanması hedeflenmektedir.

Feng ve ekibi (2025) [15] tarafından yapılan çalışmada, CICAndMal2017 (2100 örnek; 1700 benign, 426 malware) ve CICMalDroid2020 (1884 örnek; 981 benign, 903 malware) veri setleri kullanılarak Android uygulamaları üzerinde, Windows 10 ortamında hem binary hem de çok sınıflı (multi-class) malware sınıflandırması gerçekleştirilmiştir. Çalışmada önerilen hibrit model HGDetector, Network Behavior Function Call Graph (Ağ Davranış Fonksiyon Çağrı Grafiği) ve Network Traffic Node Interaction Graph (Ağ Trafiği Düğüm Etkileşim Grafiği) bileşenlerini birleştirerek hibrit özellik çıkarımı yapmaktadır. Ayrıca LR, SVM, RF ve MLP gibi farklı sınıflandırıcılar kullanılmıştır. Hibrit özellik çıkarımı aşamasında statik + trafik tabanlı (graph embedding + network flow analizi) yaklaşımı uygulanmıştır. Deneysel sonuçlara göre, CICMalDroid2020 veri setinde doğruluk oranı %93.2, CICAndMal2017 veri setinde ise %96.7 olarak ölçülmüştür. Hibrit özelliklerin kullanımı, tekil (sadece statik veya sadece trafik tabanlı) yaklaşımlara kıyasla daha yüksek performans sağlamıştır.

Çalışmanın eksiklikleri arasında, modelin yalnızca Android ortamında test edilmiş olması, gerçek zamanlı trafik analizinin sınırlı kalması ve büyük ölçekli ağ verilerinde doğrulama yapılmaması yer almaktadır. Gelecekteki çalışmalarda, modelin diğer mobil platformlara (örneğin iOS) uyarlanması ve daha derin öğrenme tabanlı yapılarla birleştirilmesi hedeflenmektedir.

TABLE III: Hibrit Yöntem Literatür Taraması Sonuçları

Yazarlar (Yıl)	Örneklem Büyüklüğü	Dil (belirtilmişse)	Platform	Yöntem	Sonuçlar	Eksiklikler	Gelecek Çalışmalar
Awwal ve ekibi (2025) [14]	EMBER veri setinden türetilmiş, 3 farklı alt küme (2017-2018)	-	Windows	Hibrit model (Autoencoder + 1D-CNN + BiLSTM)	Doğruluk: %96.47	overfitting riski taşımakta ve gerçek zamanlı uygulama yapılmamıştır.	Modelin gerçek zamanlı dağıtık sistemlere entegrasyonu planlanmaktadır.
Feng ve ekibi (2025) [15]	CICAndMal2017: 2100 örnek, CICMal-Droid2020: 1884 örnek	Python 3.9	Android	Hibrit model: HGDetector	CICMalDroid2020 veri seti için doğruluk:%93.2	Büyük ölçekli gerçek ağ verilerinde doğrulama yapılmamıştır.	Modelin diğer mobil platformlara (ör. iOS) uyarlanması ve daha derin öğrenme yapılarıyla birleştirilmesi hedeflenmekte-dir.
Taher ve ekibi (2023) [16]	Drebin, CICAnd-Mal2017 vs. (4890 örnek)	-	Android	Hibrit model: DroidDetectMW	statik ikili sınıflandırma doğruluk:%96.9	Yalnızca ağ trafiği veya sistem çağrıları gibi tek tip dinamik özelliklerin yeterli olmadığı belirtilmiştir.	temporal analizlerin eklenmesi önerilmektedir.
Rodrigo ve ekibi (2021) [17]	Omnidroid veri seti (31.931 Android uygulaması; 22.636 statik + 2.210 dinamik özellik)	Python 3.7.6	Android	Hibrit model (Statik + Dinamik + Hibrit Fully Connected Neural Network)	Hibrit model doğruluk: %91.1, Dinamik model:%81.1	iOS veya diğer mobil platformlar test edilmemiştir; gerçek zamanlı analiz sınırlıdır.	Modelin iOS'a genelleştirilmesi ve otomatik etiketleme ile sürekli öğrenen sistemlerin tasarlanması planlanmaktadır.

Not: Tablo, hibrit model tabanlı malware tespit çalışmalarında kullanılan veri setlerini, yöntemleri, elde edilen başarı oranlarını ve model sınırlılıklarını özetlemektedir.

Taher ve ekibi (2023) [16] tarafından gerçekleştirilen çalışmada, Drebin, CICAndMal2017, APKMirror ve VirusShare veri setleri kullanılarak binary ve multi-class (çoklu sınıf) malware sınıflandırması gerçekleştirilmiştir. Toplam 4.890 örneklemden oluşan veri kümesinde 1.910 malware (zararlı yazılım), 2.980 benign (zararsız uygulama) ve 13 farklı malware sınıfı bulunmaktadır. Android uygulamaları üzerinde hem statik analiz (Apktool) hem de dinamik analiz (CuckooDroid sandbox) yöntemleri uygulanmıştır. Çalışmada önerilen hibrit mimari, statik ve dinamik özellikleri birleştirerek ANN tabanlı bir model oluşturmuştur. ANN'in hiperparametre optimizasyonu, Enhanced Harris Hawks Optimization (EHHO) algoritması ile gerçekleştirilmiş ve performansı klasik makine öğrenmesi modelleriyle karşılaştırılmıştır. Elde edilen sonuçlara göre, multi-class sınıflandırmada doğruluk oranı %82.7, binary sınıflandırmada ise sırasıyla %96.7 (statik) ve %89 (dinamik) olarak raporlanmıştır.

Çalışmanın sınırlılıkları arasında, kod gizleme (obfuscation) ve şifreleme kullanılan malware türlerinde düşük tespit başarımı ile yalnızca Android platformunda test edilmesi yer almaktadır. Gelecek çalışmalarda, modelin gerçek zamanlı tespit sistemleriyle bütünleştirilmesi ve temporal analiz yöntemleriyle desteklenmesi planlanmaktadır.

D. Yöntemlerin Karşılaştırmalı Değerlendirilmesi

Yöntemlerin Performans Karşılaştırılması: Makine, derin ve hibrit öğrenme yöntemlerinin sonuçları incelendiğinde; makine öğrenmesi ve hibrit temelli yaklaşımların, derin öğrenme yöntemlerine kıyasla genellikle daha yüksek ve istikrarlı başarılar sergilediği görülmektedir. Makine öğrenmesi tabanlı modellerde doğruluk oranları literatür genelinde %54–98 aralığında değişirken, hibrit modellerde bu oran %81–96, derin öğrenme tabanlı modellerde ise %27–98 aralığında ölçülmüştür. Özellikle CNN + BiLSTM ve Autoencoder tabanlı hibrit entegrasyonlarda, doğruluk oranlarının belirgin biçimde arttığı tespit edilmiştir. Bu durum, statik ve dinamik özelliklerin bir arada kullanılmasının, tekil özellik türlerine göre daha yüksek ve dengeli performans sağladığını göstermektedir. Bununla birlikte, Random Forest (RF) ve K-Nearest Neighbors (KNN) gibi klasik makine öğrenmesi mimarilerinin birçok çalışmada yüksek doğruluk oranı sunduğu; ancak karmaşık ve çok katmanlı saldırı tiplerinde derin öğrenme tabanlı modellerin üstün performans gösterdiği gözlemlenmiştir.

Veri Setlerinin Etkisi: Literatür incelemesindeki çalışmalar incelendiğinde, kullanılan veri setlerinin boyutu, dengesi ve türünün, modellerin başarı ve doğruluk oranlarında kritik bir rol oynadığı görülmektedir. Örneğin, CICMalDroid2020, CICAndMal2017 ve EMBER gibi büyük ve dengeli veri setleriyle eğitilen modeller, küçük veya dengesiz veri setleri (örneğin Emotet veya Drebin) üzerinde eğitilen modellere kıyasla daha kararlı sonuçlar vermiştir. Ayrıca, Windows tabanlı veri setlerinde genellikle statik analiz yöntemleri daha yüksek doğruluk sağlamışken, Android tabanlı veri setlerinde dinamik veya hibrit

yaklaşımların daha etkili olduğu tespit edilmiştir. Neredeyse tüm veri setlerinde belirli düzeyde sınıf dengesizliği bulunduğundan, çalışmalarda veri dengeleme işlemleri uygulanmış ve bu, model başarımına olumlu katkı sağlamıştır. Özellikle SMOTE ve AEF gibi özellik seçimi ve dengeleme teknikleriyle elde edilen sonuçlarda doğruluk oranlarının belirgin biçimde iyileştiği rapor edilmiştir.

Uygulama Alanlarına Göre Dağılım: Literatürde incelenen çalışmaların büyük çoğunluğu siber güvenlik ve zararlı yazılım tespiti alanına odaklanmış olsa da, kullanılan platformlar ve analiz yaklaşımları arasında farklılıklar gözlemlenmiştir. Android ortamında, genellikle dinamik analiz veya ağ trafiği özelliklerini kullanan modeller (örneğin HGDetector) ağırlıktadır. Buna karşılık, Windows ortamında ise statik analiz tabanlı, dosya içeriklerine ve PE yapısına dayalı özellik çıkarımı yöntemleri yaygın olarak kullanılmaktadır. Bununla birlikte, çoklu platform (örneğin Taher ve ekibi, 2023) üzerinde yapılan bazı çalışmalarda, Android ve Windows verilerinin aynı modelde değerlendirilerek platformdan bağımsız hibrit tespit sistemleri geliştirildiği görülmektedir. Genel olarak değerlendirildiğinde, Android ortamı için geliştirilen modeller dinamik tehditleri daha etkin bir şekilde yakalayabilirken, Windows tabanlı modeller genellikle daha yüksek doğruluk oranları elde etmiştir.

Ortak Sınırlılıklar ve Boşluklar: Literatürdeki çoğu çalışmada benzer sınırlılıklar tespit edilmiştir. İlk olarak, temporal analizlerin sınırlı olması öne çıkmaktadır. Modellerin geçmiş tehditlerden öğrenerek gelecekteki saldırı türlerini öngörmeye yönelik sınıflandırma yeteneği genellikle incelenmemiş veya çalışmalara dahil edilmemiştir. İkinci olarak, karşılaştırmaların sınırlı sayıda model üzerinde yapılması ve bu nedenle değerlendirmelerin istatistiksel olarak yeterli çeşitlilik göstermemesi dikkat çekmektedir. Üçüncü olarak, birçok çalışmada k-fold çapraz doğrulama yönteminin kullanılmaması, elde edilen sonuçların genellenebilirliğini sınırlamaktadır. Dördüncü olarak, gerçek zamanlı analiz eksikliği önemli bir boşluk olarak öne çıkmaktadır. Çoğu model yalnızca offline veri üzerinde test edilmiş, canlı sistemlerde doğrulama yapılmamıştır. Bunun yanında, çalışmaların büyük kısmı yalnızca tek bir platformda (çoğunlukla Android) test edilmiştir; bu durum da modellerin platformlar arası genellenebilirliğini azaltmaktadır. Son olarak, derin öğrenme tabanlı modellerin model karmaşıklığı, pratik uygulamalarda kullanılabilirliği sınırlayan temel faktörler arasında yer almaktadır.

Grafiklerle Destekleme: Şekil 1'de, Güngör ve ekibi çalışmasında LR , Zada ve ekibi KNN, Awwal ve ekibi Autoencoder + 1D-CNN + BiLSTM hibrit modeli, Taher ve ekibi DroidDetectMW hibrit çoklu sınıflandırma modeli, Feng ve ekibi HGDetector (CICMalDroid2020 veri seti), Aryal ve ekibi MalConv, ve Gupta ve ekibi RNN kullanılarak elde edilen sonuçlar gösterilmiştir. Grafik, farklı yaklaşımların doğruluk performanslarını karşılaştırmalı olarak sunmakta ve hibrit modellerin çoğunlukla en yüksek başarıyı gösterdiğini ortaya koymaktadır.

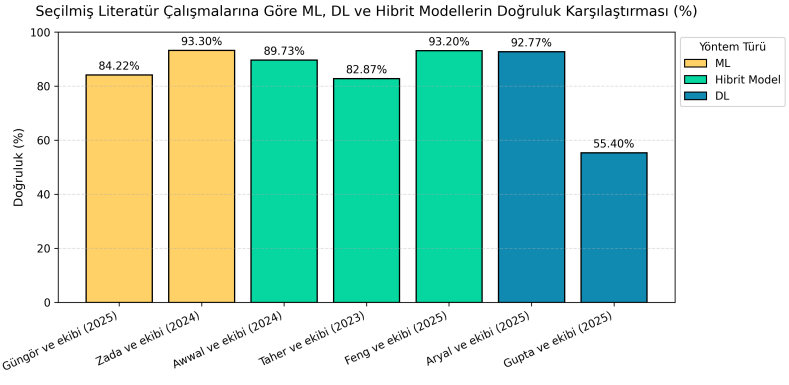

Fig. 1: Seçilmiş literatür çalışmalarında kullanılan ML, DL ve hibrit model yaklaşımlarına ait doğruluk oranlarının karşılaştırması.

Şekil 2'de gösterilen grafik, literatürde yer alan seçilmiş çalışmalar temelinde ML, DL ve hibrit modellerin ortalama doğruluk değerlerini göstermektedir. Görüldüğü üzere, hibrit modeller (%88.6) ve makine öğrenmesi tabanlı yöntemler (%88.8) birbirine yakın yüksek performans sergilerken, derin öğrenme tabanlı yaklaşımların ortalama doğruluğu (%74.1) daha düşük kalmıştır. Bu sonuç, hibrit ve makine öğrenmesi tabanlı yaklaşımların literatürde genellikle daha kararlı ve yüksek doğruluk oranları sağladığını; derin öğrenme yöntemlerinin ise model karmaşıklığına rağmen bazı veri setlerinde düşük performans gösterebildiğini ortaya koymaktadır.

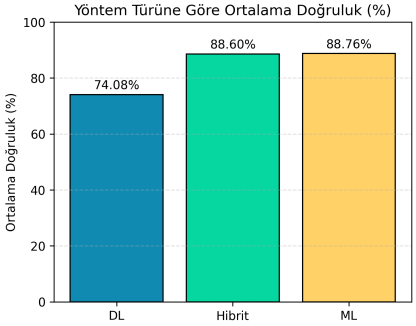

Fig. 2: Yöntem türüne göre ortalama doğruluk oranlarının karşılaştırması.

E. Literatürün Genel Değerlendirilmesi ve Bu Çalışmanın Özgün Katkısı

Kapsamlı şekilde gerçekleştirilen literatür incelemeleri ve analizler sonucunda, son yıllarda kötü amaçlı yazılım tespitinde önemli ilerlemeler kaydedildiği görülmektedir. Ancak mevcut çalışmaların çoğunda bazı ortak sınırlılıklar öne çıkmaktadır.

Örneğin, birçok çalışmanın genellikle tek bir platform (çoğunlukla Android) üzerinde gerçekleştirilmesi, veri seti dengesizliğinin ihmal edilmesi, dinamik analiz yaklaşımlarının ağırlıklı olarak tercih edilmesi ve model genellenebilirliğini göstermek için çapraz doğrulama (K-Fold) yöntemlerinin yeterince kullanılmaması bu sınırlılıkların başında gelmektedir. Ayrıca, derin öğrenme tabanlı çalışmalar yüksek doğruluk oranları elde etse de, hesaplama maliyeti ve aşırı uyum (overfitting) problemleri model kararlılığını sınırlandırmıştır.

Bu çalışmada ise literatürdeki söz konusu eksiklikler sistematik bir biçimde ele alınarak, güncel hiçbir çalışmada kullanılmamış

EMBER 2024 veri seti üzerinde kapsamlı bir bütünleşik analiz gerçekleştirilmektedir. EMBER 2024, 3.2 milyondan fazla örnek ve altı farklı platformu (Win32, Win64, .NET, APK, ELF, PDF) içeren geniş ölçekli, güncel bir veri setidir.Bu yönüyle, çalışma, literatürde kullanılan mevcut veri setleri ve önceki EMBER sürümlerine (2017/2018) kıyasla çok daha geniş kapsamlı, güncel ve çok platformlu bir veri yapısına dayanmaktadır. Çalışmada zararlı yazılımların ikili sınıflandırması (benign–malware) için 6 makine öğrenmesi, 4 derin öğrenme ve 1 hibrit model uygulanacaktır. Modellerin performansı ana deneylerde Stratified K-Fold (katmanlı) yöntemiyle değerlendirilecek; ek deneyde, temporal eğitim–test ayrımı yapılarak modellerin güncel saldırı örneklerine karşı zamansal öğrenme yeteneği incelenecektir. Ayrıca, zaman serisi drift analizi kapsamında hafta bazlı doğruluk oranları hesaplanarak performans değişiminin zaman içindeki eğilimi değerlendirilecektir. Analiz aşamasında modern boyut indirgeme ve veri dengeleme teknikleriyle dengesiz veri problemi azaltılacak, model kararlılığı ve genellenebilirliği artırılacaktır. Sonuç olarak, bu çalışma yalnızca yüksek doğruluk oranlarına ulaşmakla kalmayıp; ölçeklenebilir, genellenebilir ve güncel tehditlere uyarlanabilir bir malware tespit çerçevesi sunarak literatüre bilimsel açıdan özgün katkı sağlamaktadır.

Literatür incelemesinden elde dilen bulgular doğrultusunda, bu çalışmada uygulanacak yöntemler ve kullanılcak veri seti metot bölümünde sunulmuştur.

KAYNAKLAR

[1] Jan Kincl, Tome Eftimov, Adam Viktorin, Roman Šenkeřík, and Tanja Pavleska. Comprehensive benchmarking of knowledge graph embeddings methods for android malware detection. *Expert Systems with Applications*, 288:127888, 2025.

[2] Danish Vasan, Junaid Akram, Mohammad Hammoudeh, and Adel F. Ahmed. An advanced ensemble framework for defending against obfuscated windows, android, and iot malware. *Applied Soft Computing*, 173:112908, 2025.

[3] Aslıhan Güngör, İbrahim Dogru, Necaattin Barışçı, and Sinan Toklu. Görüntü tabanlı özelliklerden ve makine öğrenmesi yöntemlerinden faydalanılarak kötücül yazılım tespiti. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 38(3):1781–1792, 2023.

[4] Rejwana Islam, Moinul Islam Sayed, Sajal Saha, Mohammad Jamal Hossain, and Md Abdul Masud. Android malware classification using optimum feature selection and ensemble machine learning. *Internet of Things and Cyber-Physical Systems*, 3:100–111, 2023.

[5] Romil Rawat, Sanjaya Kumar Sarangi, Yagya Nath Rimal, P. William, Snehil Dahima, Sonali Gupta, and K. Sakthidasan Sankaran. Malware threat affecting financial organization analysis using machine learning approach. *International Journal of Information Technology and Web Engineering*, 17(1), 2022.

[6] Islam Zada, Mohammed Naif Alatawi, Syed Muhammad Saqlain, Abdullah Alshahrani, Adel Alshamran, Kanwal Imran, and Hessa Alfraihi. Fine-tuning cyber security defenses: Evaluating supervised machine learning classifiers for windows malware detection. *Computers, Materials and Continua*, 80(2):2917–2939, 2024.

[7] Rabia Bakshi, Sakshi Lingwal, Kumud Chandra Bhatt, Siddhant Thapliyal, Mohammad Wazid, and Devesh Pratap Singh. A robust machine learning-based mechanism for detection and analysis of malware attacks. *Procedia Computer Science*, 259:193–201, 2025. Sixth International Conference on Futuristic Trends in Networks and Computing Technologies (FTNCT06), held in Uttarakhand, India.

[8] Umesh Gupta, Shubham Kandpal, Hayam Alamro, Mashael M. Asiri, Meshari H. Alanazi, Ali M. Al-Sharafi, and Shaymaa Sorour. Efficient malware detection using nlp and deep learning model. *Alexandria Engineering Journal*, 124:550–564, 2025.

[9] Kshitiz Aryal, Maanak Gupta, Mahmoud Abdelsalam, and Moustafa Saleh. Intra-section code cave injection for adversarial evasion attacks on windows pe malware file. 159:104690.

[10] Rahim Taheri, Mohammad Shojafar, Farzad Arabikhan, and Alexander Gegov. Unveiling vulnerabilities in deep learning-based malware detection: Differential privacy driven adversarial attacks. *Computers Security*, 146:104035, 2024.

[11] Meysam Ghahramani, Rahim Taheri, Mohammad Shojafar, Reza Javidan, and Shaohua Wan. Deep image: A precious image based deep learning method for online malware detection in iot environment. *Internet of Things*, 27:101300, 2024.

[12] Usha Divakarla, K. Hemant Kumar Reddy, and K. Chandrasekaran. A novel approach towards windows malware detection system using deep neural networks. 215:148–157. 4th International Conference on Innovative Data Communication Technology and Application.

[13] Ferhat Ozgur Catak, Ahmet Faruk Yazı, Ogerta Elezaj, and Javed Ahmed. Deep learning-based sequential model for malware analysis using windows exe api calls. *PeerJ Computer Science*, 6:e285, 2020.

[14] Payal Awwal and Smita Naval. Development of heuristic adapted serial-based deep learning for efficient adversarial malware detection framework in windows. 326:114032.

[15] Jiayin Feng, Limin Shen, Zhen Chen, Yu Lei, and Hui Li. Hgdetector: A hybrid android malware detection method using network traffic and function call graph. 114:30–45.

[16] Fatma Taher, Omar AlFandi, Mousa Al-kfairy, Hussam Al Hamadi, and Saed Alrabaee. Droiddetectmw: A hybrid intelligent model for android malware detection. 13(13).

[17] Corentin Rodrigo, Samuel Pierre, Ronald Beaubrun, and Franjieh El Khoury. Brainshield: A hybrid machine learning-based malware detection model for android devices. 10(23).