T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM TANIMLI AĞLAR ÜZERİNDE MAKİNE ÖĞRENİMİ KULLANARAK OPTİMAL ÖZELLİK ÇİFTLERİNİ BELİRLEME VE ANOMALİ TESPİTİ

DOKTORA TEZİ

Erman ÖZER

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Dr. Öğretim Üyesi Murat İSKEFİYELİ**

Ocak 2022

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM TANIMLI AĞLAR ÜZERİNDE MAKİNE ÖĞRENİMİ KULLANARAK OPTİMAL ÖZELLİK ÇİFTLERİNİ BELİRLEME VE ANOMALİ TESPİTİ

DOKTORA TEZİ

Erman ÖZER

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez 25 / 01 /2022 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

Jüri Başkanı **Üye** **Üye**

Üye **Üye**

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Erman ÖZER

TEŞEKKÜR

Doktora eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Dr. Öğretim Üyesi Murat İSKEFİYELİ'ye teşekkürlerimi sunarım.

Çalışmamda bana yön gösteren, destek ve emeklerini esirgemeyen saygıdeğer jüri hocalarıma teşekkür ederim.

Bu zorlu süreçte her zaman destekçim olan sevgili eşim Dilek KIRCA ÖZER'e ve hayatım boyunca yanımda olan aileme teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

DDoS	: Dağıtık Hizmet Engelleme
DoS	: Hizmet Reddi
DT	: Decision Tree
FP	: False Positive
FN	: False Negative
GP	: Gaussian Process
IoT	: Nesnelerin İnterneti
KNN	: K en yakın komşu
LTE	: Uzun Süreli Evrim
NFC	: Yakın Alan İletişimi
NIDS	: Ağ saldırı tespit sistemleri
MÖ	: Makine Öğrenmesi
QDA	: Quadratic Discriminant Analysis
RBF	: Radial Basis Function
RF	: Random Forest
RFID	: Radyo Frekansı Tanımlama
SVM	: Support Vector Machine
TN	: True Negative
TP	: True Positive
YTA	: Yazılım Tanımlı Ağ
YSA	: Yapay Sinir Ağı

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Yazılım Tanımlı Ağ, Siber Güvenlik, Bot-Iot Veriseti, Makine Öğrenmesi

Günümüz internet teknolojilerinde siber saldırılar en büyük sorunlarındandır. Son yıllarda bilişim teknolojileri büyük gelişme gösterirken, geleneksel ağ prensipleri hemen hemen hiç değişmemiştir. Bu doğrultuda, çalışmada Yazılım Tanımlı Ağ kullanılarak merkezi kontrol ve programlanması sayesinde ağ trafiğinin etkin bir şekilde izlenmesi sağlanmıştır.

Tam özniteliklerle eğitilmiş herhangi bir makine öğrenimi tespit sistemi, verimsiz ve ağır saldırı tespit sistemlerine dönüşür. Bu nedenle, verimsiz ve ağır saldırı tespit sistemleri yerine hafif, doğru ve yüksek performanslı izinsiz giriş tespit sistemleri çok önemlidir. Bu amaçla en popüler 10 adet makine öğrenme algoritması ve BoT-IoT (2018) veri seti seçilmiştir.

Bu çalışmada, bu veri kümesinin geliştiricileri tarafından önerilen en iyi on iki öznitelik kullanılmıştır. Benzer şekilde, 66 öznitelik çiftinden her bir öznitelik çifti aracılığıyla 10 makine öğrenme algoritması eğitilerek 660 öznitelik çifti tabanlı hafif saldırı tespit sistemi geliştirilmiştir. Ayrıca, 12 en iyi öznitelik ile eğitilen 10 saldırı tespit sistemi ve 66 öznitelik çifti ile eğitilen 660 saldırı tespit sistemi, makine öğrenmesi algoritmik gruplarına göre birbirleriyle karşılaştırılmıştır. Öznitelik çiftleriyle elde edilen sonuçlar %95'in üzerinde iken ayrıca birim maliyet açısından %20-%30 arasında kazanç sağlamaktadır.

IDENTIFYING OPTIMAL FEATURE PAIRS AND DETECTING ANOMALIES USING MACHINE LEARNING ON SOFTWARE-DEFINED NETWORKS

SUMMARY

Keywords: Software-Defined Networks, Cyber Security, Bot-Iot Dataset, Machine Learning

One of the biggest problems of today's internet technologies is cyber attacks. Although information technologies have made great progress in recent years, traditional networking principles have hardly changed. In our study, efficient monitoring of network traffic was achieved thanks to its centralized control and programmability using Software-Defined Network.

Any fully-featured machine learning detection system turns into inefficient and heavy intrusion detection systems. Therefore, light, accurate and high-performance intrusion detection systems are very important instead of inefficient and heavy intrusion detection systems. For this purpose, the 10 most popular machine learning algorithms and BoT-IoT (2018) dataset were selected.

The twelve best features recommended by the developers of this dataset are used in this study. Similarly, 660 feature-pair-based lightweight intrusion detection systems were developed by training the 10 machine learning algorithms via each feature pair out of the 66 feature pairs. Moreover, the 10 intrusion detection systems trained with 12 best features and the 660 intrusion detection systems trained via 66 feature pairs were compared to each other based on the machine learning algorithmic groups. While the results obtained with feature pairs are over 95%, it also provides 20%-30% savings in terms of unit cost.

BÖLÜM 1. GİRİŞ

Bilgisayar ve internet teknolojilerinde yaşanan gelişmeler, şirketlerin ve devletlerin güvenlik açıklarına neden olmaktadır. Son yıllarda sosyal, mobil, büyük veri, nesnelerin interneti vb. alanlarda bilgi ve iletişim teknolojileri büyük gelişmeler gösterirken, ağ prensipleri hemen hemen hiç değişmemiştir. Mevcut piyasa gereksinimlerini geleneksel ağ yapısıyla karşılamak oldukça zorlaşmıştır. Geleneksel ağ mimarisinin son yıllarda değişmeden kaldığı ve oldukça hantal olduğu görülmektedir [1].

Geleneksel ağ, donanım ve yazılım parçaların kısıtlılığına neden olmaktadır. Bu durum, ek durumlara ihtiyaç duyulduğunda, yeni donanım alımında ve yeni lisans satın alımında yüksek maliyetlere neden olmaktadır. Neticede yeni iş durumları için kullanışlı değildir.

Bilgi ve iletişim teknolojileri, yüksek bant genişliği, erişilebilirlik, yüksek bağlantı hızı, dinamik yönetim taleplerinden dolayı geleneksel ağlarda karmaşıklığı ve yönetilebilirlik sorunlarını beraberinde getirmiştir. Bu sorunları önlemek amacıyla devletler ve şirketler Yazılım Tanımlı Ağ(YTA) kullanmaya başlamıştır.

YTA, ağ kontrol düzleminin veri düzleminden fiziksel olarak ayrılmasını ve bir kontrol düzleminin çeşitli aygıtları denetlemesini sağlamaktadır. YTA, güvenli, esnek ve güvenilir bir ağ sistemi sağlamak için daha büyük bir potansiyel sunduğu için geleneksel ağların yerini almıştır. YTA, kontrolcü vasıtasıyla ağı merkezi olarak kontrol etme, değiştirme, yönetme imkanı sunmaktadır. YTA, geleneksel ağ yapısında bir arada bulunan kontrol ile veri düzlemini birbirinden ayırarak bilgisayar ağlarına yeni bir anlayış getirmektedir. İlaveten, YTA, kurulum süreleri boyunca ve gereksinimlerdeki değişikliklere bağlı olarak daha sonraki aşamalarda

programlanabilmektedir. Esneklik ve sanallaştırma yoluyla yeni iş durumlarına yardımcı olmaktadır.

YTA ve geleneksel ağ iletişimi arasındaki en belirgin fark, YTA'nın yazılım tabanlı olması, geleneksel ağın ise genelde donanım tabanlı olmasıdır. Yazılım tabanlı olduğu için YTA daha esnektir ve kullanıcılarına kaynakları yönetme konusunda daha fazla kontrol ve kolaylık sağlar. Google, Yahoo, HP, Cisco gibi üst düzey kurumlar sistemini anomalilere karşı korumak ve güvenliğini sağlamak için oldukça fazla para harcamaktadır. Ağ kapasitesinin artması için geleneksel bir ağın yeni donanıma ihtiyacı vardır. Bu durum şirketleri maliyet azaltmak amacıyla YTA teknolojisine yöneltmektedir. Aynı zamanda YTA, günümüz uygulamalarına yüksek bant genişliğini sunmaktadır.

Son on yılda, Nesnelerin İnterneti alanı, bilgisayarlar, farklı türlerde akıllı araçlar, geleneksel veya akıllı sensörler, sağlık, eğitim, enerji ve ulaşım gibi akıllı uygulamalar, 4G ve 5G internet erişim cihazları dahil olmak üzere 20 milyardan fazla cihazla büyük ölçüde genişlemektedir. IoT cihazlarının hızla artması ve yaygınlaşması nedeniyle bu sayının 2030 yılına kadar 50 milyardan fazla sayıya ulaşması beklenmektedir [2]. Bu kadar çok sayıda birbirine bağlı cihaz, sürekli olarak büyük miktarda veri (Büyük Veri) alıp vermekte ve iletmektedir. Bu durum sistemleri çeşitli siber saldırı türlerinin hedefi haline getirmektedir.

IoT sistemleri, araştırmacılar ve mühendislik geliştiricileri tarafından uluslararası olarak tanınan tek bir standart mimariye sahip değildir. IoT'nin temel mimarisi üç katmandan [3-5] oluşurken, diğer araştırmacılar dört ve beş katmanlı [6], [7] mimariler önermektedir. Standart bir mimariye sahip olmamak doğal olarak güvenlik ve gizlilik sorunlarına neden olmaktadır. Çünkü akıllı ortamlar, evrensel bir standart dili paylaşmayan çeşitli teknoloji şirketlerinden birkaç farklı sensör, farklı donanım araçları veya yazılım uygulamaları dahil olmak üzere farklı türde IoT sistemlerinden oluşmaktadır [8,9].

Ayrıca IoT, tüm mimarilerinde tamamen internet bağlantısına bağlıdır. Radyo Frekansı Tanımlama (RFID) [10], Yakın Alan İletişimi (NFC) [11], Bluetooth [12], Wi-Fi [13] ve Uzun Süreli Evrim (LTE) [14] gibi iletişim teknolojilerini kullandığı için internetteki hizmet saldırıları, kimlik doğrulama sorunları, Hizmet Reddi (DoS) ve Dağıtılmış DOS (DDoS) [15] gibi siber saldırıların en büyük hedefi haline getirmiştir. Bu tür siber tehditlere sahip siber saldırganların bir numaralı hedefi olmak; araştırmacılar, IoT üreticileri ve hatta IoT kullanıcıları için etkili zor durum yaratmaktadır. Ek olarak, IoT ağları güç verimliliği sorunlarından da muzdariptir.

Bu zorluğun üstesinden gelmek için, ağa izinsiz girişi algılama, kötü amaçlı yazılım tespiti ve ağ adli sistemleri gibi ciddi ve gerçekçi güvenlik ve soruşturma önlemlerinin etkili bir şekilde geliştirilmesi ve uygulanması gerekmektedir. Bu nedenle, ağ saldırılarını verimli bir şekilde tespit edebilen ve karşı önlemleri alabilen sağlam ve yüksek performanslı yapay zeka algoritmalarına ihtiyaç vardır. Bu tür algoritmaları geliştirmek için, iyi yapılandırılmış ve temsili veri kümeleri eğitim ve sistemlerin güvenilirliğini doğrulamak için çok önemlidir [16]. Bu alanda çok çeşitli çalışmalar yapılmış olsa da IoT, siber uzay çok geniş olduğu ve siber uzaydaki saldırılar rastgele ve öngörülemez olduğu için daha fazla araştırılması ve geliştirilmesi gereken bir konu haline dönüşmüştür. Buna ek olarak, metodolojiler, uygulama ve veri setleri açısından yeterli araştırma eksiklikleri ve boşlukları bulunur. Bu eksiklikleri gidermek ve boşlukları doldurmak için, yapay zeka algoritmalarını, gerçekçi veri kümelerini ve anomali tespiti hakkında kapsamlı araştırma yapmak çok önemlidir [17]. Bu nedenle, veri madenciliği, bulanık teknikler, klasik sinir ağları, genetik algoritmalar, nöro-genetik algoritmalar, parçacık sürüsü zekası, kaba kümeler, istatistiksel öğrenme ve klasik makine öğrenme algoritmalarına dayanan geleneksel ağır IDS, IoT ağı için uygun değildir [8,9].

Algoritmaların doğasına bağlı olarak, her makine modeli bir veri kümesindeki farklı öznitelikler içermektedir. Örneğin, doğrusal eğilim doğasına sahip öznitelikler, doğrusal regresyon, ridge regresyonu veya doğrusal destek vektör makineleri gibi doğrusal yöntemlerde yüksek etkilere sahipken, doğrusal olmayan algoritmalar verilerdeki daha karmaşık bağlantılardan yararlanmaktadır. Bu nedenle, farklı

öznitelikler veya öznitelik çiftleri araştırılmalı ve çeşitli tekniklerle uygulanmalı, bu modellerin doğruluğu ve performansı üzerinde hangi özniteliklerin daha önemli, daha büyük bir etkiye sahip olduğu keşfedilmelidir.

Buna ek olarak, IoT cihazları sürekli olarak veri üretmektedir. Bu büyük miktardaki veriye büyük veri denir ve gün geçtikçe gigabaytlar ,terabaytlar olarak artar. Bu nedenle, büyük veri kümelerinin boyutu ve boyutluluğu astronomiktir. Bu nedenle, bir veri kümesinin hangi özniteliklerinin (hangi sütunların) çok önemli olduğunu belirlemek ve tahmin etmek, en etkili parametreleri analiz etmemizi ve bunlara odaklanmamızı sağlayarak değerli zaman ve kaynaklardan tasarruf etmemizi sağlar.

Bir ağ anomalisi ağın güvenliği için etkileri olan zararlı herhangi bir trafiktir. Gün geçtikte yeni tip saldırılar ve zararlı verilerin sızdırılması durumları ortaya çıkmaktadır. Anomali tespiti bunların önlenmesi için özel önem taşır. Anomali Tespiti, en basit anlamıyla belirli bir kümede olağandışı bir nokta veya durum bulma tekniğidir. Bu beklenmedik durumlar aslında bir verinin beklenen davranışlarına uymayan niteliktedir veya kalıptadır. Anomali tespiti hatalı cihazların neden olduğu olağan davranışları tespit etmeyi amaçlar. Bu nedenle bilgisayar uygulamalarında büyük öneme sahiptir. Anomali tespitindeki en büyük zorluk büyük ölçekli veri setleri ile nasıl başa çıkılacağıdır [18-20].

Ağlarda anomali tespit ise ağ trafiğinde oluşan olağan dışı durumlar olarak ifade edilir. Kredi kartı dolandırıcılığı, internet hizmetine erişememe sorunu ve bilgisayara ait gizli bilgilerin başka bilgisayara aktarılması anomali sorunlarından yalnızca birkaçıdır [21,22].

Ağ saldırı tespit sistemleri(NIDS-Network Instrusion Detection Systems), siber güvenlikte kötü ve şüpheli etkinliklere karşı savunmada oldukça önemlidir. NIDS, ağ sistemlerinde tehdit ve saldırılara karşı korunmada önemli bir yer kaplamaktadır. NIDS ağ trafiğindeki anomaliyi tespit etmekte oldukça fazla zorluklarla ve sorunlarla karşı karşıya gelmektedir. Saldırıların çeşitliliğinin sürekli artması ve geleneksel yöntemleri kullanma sebebiyle yüksek değerde yanlış pozitif alarmlar üretmektedir.

Trafik davranışlarını izleyerek analiz etmek, kritik bilgileri toplamasına engel olmak ve bilgisayar sistemlerine erişim hakkı olmadığı halde erişim sağlamasını engellemek Malware'nin (zararlı yazılımın) temel görevlerindendir [23-25].

Bu araştırmanın diğer çalışmalardan farkı; YTA üzerinde gerçekçi veriler ve nesnelerin internet verileri kullanılarak anomali tespit edilmesidir. Veriseti olarak gerçekçi bir ağ ortamında elde edilen veriler kullanılmıştır. Çalışmada farklı özniteliklerin sonuçlar üzerindeki etkilerini gözlemleyerek en uygun öznitelik değerlerini bulmak amaçlanmıştır. Elde edilen sonuçlar modelimizin çok iyi performans gösterdiği ve yüksek doğrulukla neticelendiği tespit edilmiştir.

Ayrıca, makine veya derin öğrenme tabanlı anomali algılama sistemleri veya ağ adli sistemleri genellikle, düşük doğruluk sonuçlarına yol açan veya bir sistemin performansı üzerinde olumsuz etkileri olan rastgele öznitelikler veya tam öznitelikler aracılığıyla veri kümeleriyle eğitilir. Bu zorlukların üstesinden gelmek için, IoT ağ sistemlerinde izinsiz giriş veya anomali algılama sistemlerinin doğruluğunun ve performansının iyileştirilmesini kolaylaştıran veri kümelerinin en etkili ve en uygun öznitelik çiftlerini belirlemeye yardımcı olabilecek yeni bir yaklaşım önerdik. Bu amaçla, Yeni Güney Galler Üniversitesi Canberra, Avustralya'daki Mühendislik ve Bilgi Teknolojileri Okulu, UNSW Canberra Siber Merkezi tarafından üretilen en yeni ve gerçekçi Bot-IoT verisetini ve en popüler 10 makine öğrenimi algoritmasını seçtik.

Veri kümesinin en iyi 12 özelliğinden 66 benzersiz öznitelik çifti oluşturduk. Seçilen makine öğrenimi algoritmaları, oluşturulan benzersiz öznitelik çiftleri aracılığıyla eğitilmiştir. Ardından, en iyi ve en yüksek doğrulukta performans gösteren öznitelik çiftleri, en etkili ve en uygun eğitilebilir girdi verileri olarak belirlendi. Ayrıca, Bot-IoT (2018) veri seti kullanılarak geliştirilecek çevrimiçi ve çevrimdışı IoT ağ saldırı tespit sistemleri için doğruluk ve performans açısından hangi makine öğrenimi algoritmalarının en uygun ve optimum olduğu keşfedilmiştir.

Çalışmanın ikinci kısmında literatürde yapılmış çalışmalardan bahsedilmiştir. Üçüncü bölümünde yazılım tanımlı ağa değinilmiştir. Dördüncü bölümünde anomaliye ve anomali tespitinde kullanılan algoritmalardan bahsedilmiştir. Beşinci kısımda test ortamına ve kullandığımız veriseti hakkında bilgi ve verisetinin diğer veri setlerinden farkı, altıncı kısımda sınıflandırma ölçütleri, metot ve elde edilen sayısal sonuçlar gelecekte yapılması önerilen çalışmalar belirtilerek çalışma tamamlanmıştır.

BÖLÜM 2. LİTERATÜR TARAMASI

Son zamanlarda internet teknolojilerinde yaşanan gelişmeler ağlar ve sistemler üzerinde DoS [26], DDoS [27], malware [28], zero-day [29] attacks gibi birçok farklı türden saldırılara karşı zaafiyet gösteren güvenlik açıklarına neden olmaktadır. 1970'lerde "Creeper Worm" olarak bilinen ilk bilgisayar virüsü ve 1980'lerde önyükleme sektörü virüsü "Elk Cloner" yaratıldığından beri siber saldırganlar ile güvenlik geliştiricileri veya araştırmacılar arasındaki savaş devam ediyor. Örneğin, Symantec'e göre, akıllı telefon kötü amaçlı yazılım tehditleri 2017'de %54 artarken, IoT cihazlarına yapılan saldırılar %600 artmıştır [30].

Ayrıca sosyal kullanıcıların sürekli artan taleplerini karşılamak için multimedya analitiği, yüksek işleme, sürekli veri toplama, büyük bant genişliği ve hesaplama açısından daha az karmaşık kodlama teknikleri gerektirir [31]. Fakat geleneksel ağlarla bu talepleri karşılamak oldukça zordur. Hedeflenen kullanıcılara yüksek kalitede deneyim (QoE) sağlamak için yeni nesil ağ teknolojilerini dahil etmek bir trend haline geldi. Bununla birlikte, sosyal ağların karmaşıklığı ve açıklığı nedeniyle, yüksek sermaye ve bakım maliyetleriyle sonuçlandığından, yeni teknolojileri dağıtmadan önce doğasında bulunan güvenlik ve gizlilik endişelerini gidermek zorunlu hale gelmektedir. Sonuç olarak, iletişim ağının sosyal multimedya ağlarının ortaya çıkardığı zorlukları karşılamak için özelleştirilmesi gerekiyor. Bu iletişim ağını yönetmenin ve kontrol etmenin bir yolu, Yazılım Tanımlı Ağ (YTA) [32] kullanımı olmuştur. Kritik bir etkinleştirme teknolojisi olan YTA, ölçeklenebilirlik, gizlilik, hata toleransı, dağıtılmış yönlendirme kontrolü, artımlı dağıtım, ağ programlanabilirliği, çalışma zamanı, güvenlik ilkeleri ve prosedürleri gibi çeşitli öznitelikleri sağlamak için ağ kontrol düzlemini iletme düzleminden ayırır. Literatürde, ölçeklenebilir ve esnek iletişim sağlamak için YTA denetleyicisinin üzerinde farklı anormallik algılama şemalarının kullanıldığı çeşitli yaklaşımlar

önerilmektedir. Örneğin, He ve ark. [33], standart kullanıcı profillerine yapılan saldırıları belirlemek için YTA tabanlı anomaliyi algılama modeli tasarladı. Büyük ölçekli, yüksek boyutlu ve etiketlenmemiş ağ verilerini işlemek için örnekleme uyarlamalı yoğunluk tepe tabanlı kümeleme algoritması ve denetimsiz küme tabanlı öznitelik seçim mekanizması olmak üzere iki algoritma önerildi. Benzer şekilde, Peng ve ark. [34], YTA akışlarının sınıflandırılması için K-en yakın komşu algoritmasını kullanarak YTA tabanlı bir akış algılama yöntemi sunmuştur. [35]'da, Ha ve ark. YTA tabanlı trafiğin üzerinde yeni bir strateji sundu. Tüm paketleri analiz etmek yerine, kötü niyetli akışın yakalama hatası oranını en aza indirmek için sadece şüpheli trafikler kontrol edildi. Tüm bu yöntemlerde YTA'nın umut verici sonuçlar verdiği kanıtlanmıştır.

Diğer bir yandan, ağ trafiği üzerinde yapılan saldırılar, ağ güvenliğiyle ilgilenen araştırmacıların ilgisini çekmektedir. Araştırmacılar saldırı türlerini saptamak ve önlemek amacıyla Saldırı Tespit Sistemleri(IDS) gibi yaklaşımlar geliştirmişdir.

Klasik bir IDS; ağ trafiğini şüpheli davranışlar için izleyen ve bu tür davranışlar keşfedildiğinde uyarı veren bir sistemdir. Geleneksel IDS yaklaşımları daha önceden karşılaşmadığı saldırı paketlerini tespit etme veya saldırı türünü sınıflandırmada yetersiz kalmaktadır [36].

Bu durum, acilen çözülmesi gereken yöntem arayışını tetiklemiştir. Neticede, son dönemlerde yaygın olarak kullanılan makine öğrenmesi ve derin öğrenme gibi yapay zeka teknikleri ile sağlıklı saldırı saptayıcı sistemler gelişine katkı sağlamıştır.

Makine Öğrenmesi metotları ile ağ davranışını analiz ederek ağ paketlerinin normal veya anomali olup olmadığını saptanmaktadır. Alt bölümde son zamanlarda yazılan makaleler incelenerek; IDS saldırı tespit sistemlerinde kullanılan bu modeller açıklanmıştır.

Firewall, erişim kontrol mekanizmaları ve şifreleme gibi geleneksel saldırı tespit ve önleme tekniklerinin, ağları ve sistemleri, DoS DdoS (hizmet reddi gibi) gittikçe

karmaşıklaşan saldırılara karşı tamamen koruma konusunda bazı sınırlamaları vardır. Ayrıca, bu tekniklere dayalı olarak inşa edilen sistemlerin çoğu, yüksek yanlış pozitif ve yanlış negatif sonuçları vermektedir. Daha da fazlası sürekli değişen davranışlara karşı uyum sağlama eksikliği görülmektedir [37]. Bununla birlikte, son on yılda, izinsiz giriş tespiti sorununa tespit oranlarını ve uyarlanabilirliğini geliştirme umuduyla çeşitli Makine Öğrenimi (MÖ) teknikleri uygulanmıştır. Bu teknikler genellikle saldırı bilgisi tabanlarını güncel ve kapsamlı tutmak için kullanılır.

Makine öğrenmesi bir iş akışında mevcut verilerin toplanmasını, verilerin temizlenmesini, hazırlanmasını ve modellerin oluşturulmasını sağlayan bir süreçtir. Makine öğrenmesi yaklaşımları; denetimli öğrenme, denetimsiz öğrenme ve yarı denetimli öğrenme gibi üç farklı öğrenme modelinden oluşmaktadır. Denetimsiz öğrenme modelinde veriler etiketlenmez iken denetimli öğrenme modelinde veriler tamamen etiketlenmektedir. Yarı denetimli öğrenme modeli ise etiketlenmiş ve etiketlenmemiş verilerin bir karışımını içermektedir [38]. Anomali tespitinde kullanılan en popüler algoritmalar şunlardır:

Bulanık mantık, modern bilgisayarların doğruluk derecelerine dayalı olarak hesaplama yöntemidir. Ayrıca, izinsiz giriş tespiti problemi, toplanan verilerdeki birçok sayısal özelliği ve çeşitli türetilmiş istatistiksel verileri içerir. Modelleri doğrudan sayısal veriler üzerine oluşturmak genellikle yüksek algılama hatalarına(false positive) neden olur [39].Aynı şekilde Gharboui ve arkadaşları uçtan uca yol hesaplamasında Sequential Hypothesis Testing algoritmasını kullanarak bilgi işlem hizmetlerinde kötü niyetli kullanımı tespit etmişlerdir. Tespit doğruluğu oldukça iyi olmasına rağmen, yanlış positif oranı yüksek çıkmıştır. Dezavantajlarından bir diğeri ise gerçek zamanlı veriler kullanılmamıştır [40]. Nguyen ve Roughan birden fazla ISP(Internet Service Provider) verileri gözlemleyerek ve Hidden Markov Modeli ile anomaliler algılanmıştır. Küçük boyutlu veriler üzerinde oldukça iyi sonuçlar vermesine rağmen büyük boyutlu verilerde performansı oldukça düşüktür [41]. Fernandes ve arkadaşları gerçek ve simüle edilmiş ağ trafiği üzerinde Principal Component Analysis algoritmasını kullanarak olağan dışı durumlar tespit edilmesine rağmen hesaplamalar oldukça

karmaşıktır [42]. Bu karmaşıklakları önlemek adına Lei çalışmasında Destek Vektör Makinesi yöntemini kullanmıştır [43]. SVM; sınıfları kategorize etmek için verilen etiketli verilerle ayrıştırma yapan denetimli makine öğrenmesi algoritmasıdır. Sınıflarına göre verileri ayırırken hiper düzlem çizer. Sınıf etiketlerinin tanımlanmasına bağlı olarak siber saldırıları ve ağın davetsiz misafirlerini tespit etmek için kullanılmaktadır. SVM, genelleme hatasını en aza indirmek için farklı sınıflar arasındaki temalar üzerinde bir üst sınır tanımlamaktadır. Lojistik regresyon da temel olarak ikili bağımlı değişkeni modellemek için lojistik bir işlev kullanan istatistiksel bir modeldir. Regresyon analizinde, lojistik regresyon bir mantıksal modelin parametrelerini tahmin etmektedir [44].

Veri madenciliğine dayalı geleneksel ve ağır çalışan yöntemler, bulanık teknikler, klasik sinir ağları, genetik algoritmalar, nöro-genetik algoritmalar, istatistiksel öğrenme ve klasik algoritmalar geleneksel anomali algılama yöntemlerinden birkaçıdır. Ancak, bu geleneksel ağır yaklaşımlar enerji açısından verimli değildir ve uygun olmayan öznitelik seçimi veya veri kümelerinin tüm özniteliklerinin kullanılması nedeniyle beklendiği kadar doğru performans göstermez. Klasik IDS yöntemleri genellikle veri kümelerinin tüm öznitelikleri aracılığıyla eğitilir veya veri kümelerinin uygun olmayan özniteliklerini (sütunları) kullanırlar. Bu koşullar, paket işlemenin zaman karmaşıklığını artırır ve daha fazla enerji tüketimine neden olur. Ancak IoT sistemleri, düşük güç tüketimli IDS sistemlerine ihtiyaç duyar, çünkü IoT sistemleri çoğunlukla güç verimliliği sorunlarından muzdariptir. Bu nedenle, veri kümelerinin tam veya uygun olmayan öznitelikleri üzerine inşa edilen ağır IDS'ler yerine hafif (enerji verimli) IDS'ye ihtiyaç vardır. Bu nedenlerden dolayı, bu yöntemlerin doğruluğu ve performansı önemli ölçüde düşmektedir [45]. Çoğu saldırı veya kötü amaçlı yazılım tespit çalışmalarında belirtildiği gibi, geleneksel IDS yaklaşımları, yüksek yanlış pozitif oranları ve hesaplama karmaşıklıkları açısından zorluklarla karşı karşıyadır. Yüksek yanlış pozitif ve yanlış negatif oranları, bu tür bir ağ sisteminin Hizmet Kalitesini (QoS) azaltır. Herhangi bir kullanıcı paketi yanlışlıkla düşürülürse, kullanıcı bir faturalama hatası yaşayacak ve kullanıcı paketi gecikecektir. Anomali tabanlı IDS'ler ayrıca, kullanıcı gizliliğini ihlal eden paket

tabanlı yöntemler gibi yasa dışı analiz yöntemlerine ilişkin zorluklarla da karşı karşıyadır [46].

Literatürde, izinsiz giriş algılama için çok sayıda veriseti vardır, ancak bunların çoğu veriseti gereksinimleri için genel FAIR [47] kavramına uymaz. FAIR konsepti, bilimsel verilerin yerine getirmesi gereken dört ilkeyi tanımlar: Findability, Accessibility, Interoperability and Reusability [48]. Bir veri kümesi, kavramı yerine getirdiğinde bir makine öğrenme algoritması eğitmek için kullanılabilir. Literatürde 10'dan az anomali algılama veriseti vardır ve bunların arasında, diğer veri kümeleri IoT'deki Botnet senaryoları hakkında herhangi bir bilgi içermediğinden, son Bot-IoT veri kümesi IoT botnet algılama için tek seçenektir. Ayrıca, önceki veri kümeleri iyi yapılandırılmış değilken, Bot-IoT (2018) veri kümesi iyi yapılandırılmıştır ve yaklaşık 40 öznitelik (sütun) [49] içerir. Bu nedenlerden dolayı bu çalışmada Bot-IoT (2018) veri kümesini seçtik. Aynı zamanda, doğru öznitelikleri veya öznitelik çiftlerini seçmek saldırı algılama sistemlerinin doğruluğunu ve performansını artırmaya yardımcı olduğundan, doğru seçilen öznitelik (sütun), doğru etiketlenmiş veri kümeleri kadar önemlidir. Ancak, Bot-IoT (2018) veri setinde, diğer veri kümelerinde olduğu gibi, bu veri kümesinin hangi özniteliklerinin veya öznitelik çiftlerinin daha önemli olduğu ve bu özniteliklerden veya öznitelik çiftlerinden hangisinin hangi makine öğrenimi algoritmasıyla daha uyumlu olduğu hakkında net bir bilgi yoktur. Bu nedenlerden dolayı, Nesnelerin İnterneti alanındaki izinsiz giriş ve anormallik algılama sistemleri için en uygun makine öğrenme algoritmaları belirlemeye yardımcı olan yeni bir sistem önerdik.

BÖLÜM 3. YAZILIM TANIMLI AĞLAR

Mobil, bulut, sosyal ağ, büyük veri, multimedya ve dijital topluma yönelik gerçekleştirilen internet teknolojilerinin bir sonucu olarak, bunların yönetimi ve yapılandırması oldukça karmaşık ve zorlu hale gelmiştir. Ayrıca, yüksek bant genişliği, genişletilebilirlik ve dinamik yönetime erişim son zamanlarda oldukça kritik öneme sahiptir. Önceden tanımlanmış geleneksel yöntemler oldukça hantal hale gelmiştir. Geleneksel ağlar; yenilikler, güvenilirlik, genişletilebilirlik, esneklik ve yönetilebilirlik ile ilgili önemli eksikliklerden muzdarip donanım merkezli ağlardır. Bu durumlar düşünüldüğünde çok sayıda ağ cihazını yönetmenin birçok hataya açık olan büyük bir zorluk olduğu tartışılabilir. Bu nedenle; YTA (Yazılım Tanımlı Ağ-Software Defined Network), kontrol düzlemini ve veri düzlemini ayırarak yüksek esneklikle ağ yönetimini basite indirgemek ve geliştirmek için tasarlanmış bir yapıdır. YTA yeni bir araştırma konusu olarak görülmesine rağmen, hem endüstriyel hem de akademik kurumlarda çok sayıda araştırmacının ilgisini çekmiştir. Google, Yahoo, HP, Cisco gibi üst düzey kurumlar YTA teknolojisini kullanmaya başlamışlardır.

İnternet ve mobil ağlar geliştiğinden ve bulut, sosyal ağ ve sanallaştırma gibi yeni teknolojiler ortaya çıktığından, daha yüksek bant genişliğine sahip ağlar için ihtiyaç duyulduğu için, daha yüksek erişilebilirlik ve dinamik yönetim kritik bir sorun haline gelmektedir.

Geleneksel ağların sorunlarını ve sınırlamalarını çözmek için, YTA olarak bilinen, ağ kontrolünün yönlendirme mekanizmasından ayrıldığı ve doğrudan programlanıp kontrol edilebildiği bir yapı önerildi. YTA, mantıksal olarak merkezileştirilmiş ve ağa yönelik küresel bir görünüme sahip olan bir denetleyici kullanır ve çeşitli basit paket iletme cihazları (YTA anahtarları) ve Open-Flow gibi arabirimler aracılığıyla

kontrol edilir ve yapılandırılır. YTA anahtarları, merkezi kontrolör tarafından kontrol edilen bir veya daha fazla iletme tablosundan oluşur. Başka bir deyişle, kontrol düzleminde kontrol edilir ve programlanırlar. Bu mekanizmayı kullanarak, yazılım geliştiriciler ağ kaynaklarını kolayca kontrol edebilirler. Ayrıca, paketler tabloları ileterek işlenir, yani merkezi denetleyicinin yönlendirme tablolarında gerçekleştirdiği politikalara göre, YTA anahtarları yönlendirici, anahtar, güvenlik duvarı vb. ile aynı şekilde çalışabilir. YTA modern ağların yönetimini basitleştirir ve daha fazla yenilik için fırsat sağlar. Sonuç olarak, araştırmacılar kendi fikirlerini test edip inceleyebilir ve sonuçları değerlendirebilirler. YTA, modern internet teknolojilerinde önemli roller oynadığı için araştırmacıların ilgisini çekmiştir. Tablo 1'de detaylı bir şekilde geleneksel ağlar ve YTA arasındaki farklar gösterilmiştir [50]:

Tablo 3.1. YTA ile geleneksel ağlar farkı

Kriter	Geleneksel Ağ	Yazılım Tanımlı Ağ
Ağ Yönetimi	Zor; çünkü değişiklikler her cihazda ayrı ayrı uygulanıyor	Kontrolör yardımıyla daha kolay
Küresel ağ görünümü	Zor	Kontrolörde merkezi görünüm
Bakım maliyeti	Daha yüksek	Daha az
Güncelleme için gereken süre	Bazen aylar sürer	Merkezi kontrolör sayesinde oldukça kolay
Kontrolör kullanımı	Alakalı değil	Önemli
Kontrolörün orijinalliği, bütünlüğü ve tutarlılığı	Önemli değil	Önemli
İletim tablolarının ve ağ durumunun bütünlüğü ve tutarlılığı	Önemli	Önemli
Denetleyicinin kullanılabilirliği	Alakalı değil	Önemli
Kaynak kullanımı	Daha az	Yüksek

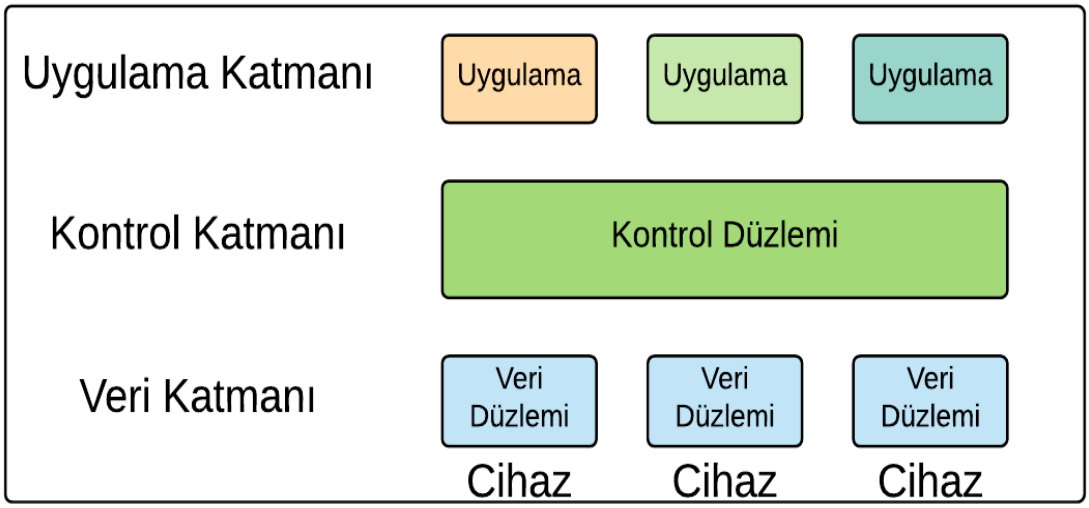
Yazılım tanımlı ağın getirdiği yenilikler sırasıyla şu şekilde sıralanabilir [51,52]:

Veri düzleminin programlaması: Ağ kontrolü direkt olarak programlanabilir çünkü yönlendirme işlevlerinden ayrılmıştır.

Kontrolün ve veri düzleminin ayırımı programlı olarak yapılandırılmış: YTA, ağ yöneticilerinin ağ kaynaklarını dinamik, otomatik YTA programları yoluyla yapılandırmalarını, yönetmelerini, güvenliğini ve optimize etmelerini sağlar; bu programlar, tescilli yazılımlara bağımlı olmadığı için kendileri yazabilirler.

Açık standartlar- Tedarikçi Tabanlı- Tarafsız: YTA, açık standartlar aracılığıyla uygulandığında, ağ tasarımını ve operasyonunu basitleştirir, çünkü talimatlar birden çok satıcıya özgü cihaz ve protokol yerine YTA denetleyicileri tarafından sağlanmaktadır.

Şekil 3.1.'de YTA mimarisinin genel yapısı görülmektedir:



Şekil 3.1. YTA mimarisi

YTA mimarisi veri düzlem katmanı, kontrol katmanı ve uygulama katmanı olmak üzere üç katmandan oluşur:

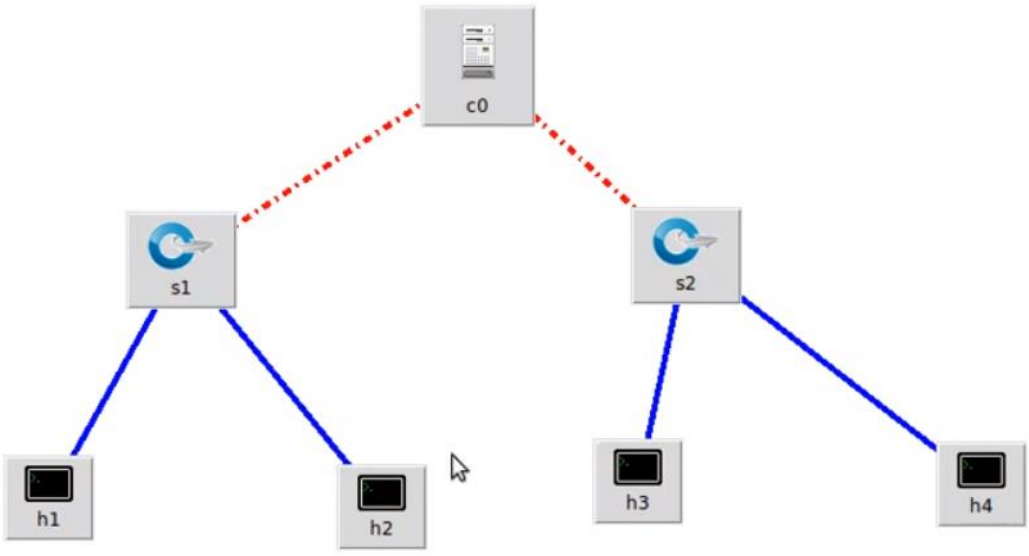
Kontrol Katmanı (Control Plane): Paketlerin bir veya daha fazla ağ cihazı tarafından nasıl iletilmesi gerektiğine karar vermek ve bu kararları yürütmek için ağ cihazlarına itmekle sorumludur. Kontrol düzleminin ana görevi, ağ topolojisine veya harici hizmet taleplerine göre yönlendirme düzleminde bulunan yönlendirme tablolarının ince ayarını yapmaktır. NOX, POX, Beacon, Maestro ve Floodlight YTA kontrolör platformlarından birkaçıdır. POX, YTA kontrol uygulamaları için açık kaynak kodlu bir platformdur. NOX ve POX platformları Python tabanlı iken; Beacon, Maestro ve Floodlight platformları Java tabanlıdır [53]. Hızlı gelişim ve prototip oluşturmayı sağlayan POX, NOX'ten daha yaygın olarak kullanıldığı için tezimizde POX platformu kullanılacaktır [54,55].

Veri Katmanı (Data Plane): İletişim cihazlarından oluşur. Kontrol düzleminden alınan talimatlara göre veri paketlerinin işlenmesinden sorumludur. Veri düzleminin eylemlerini, paketlerin iletilmesini, bırakılmasını ve değiştirilmesini içerir, ancak bunlarla sınırlı değildir. Veri kaynaklarının örneklerini, sınıflandırıcıları, ölçerler içerir. Bu düzlem kontrolör tarafından programlanır ve yönetilir.

Uygulama Katmanı (Application Plane): Ağ davranışını tanımlayan uygulamaların ve hizmetlerin bulunduğu düzlem. Yönlendirme düzleminin çalışmasını doğrudan (veya öncelikli olarak) destekleyen uygulamalar (kontrol düzlemi içindeki yönlendirme işlemleri gibi), uygulama düzleminin bir parçası olarak kabul edilmez.

YTA veri düzlem katmanı görünümü ortaya çıkaran lojik bir ağ cihazıdır. YTA kontrol düzleminde YTA yönlendirme, gönderme ve yol seçimi gibi kararları verir. Cihazların durumu hakkında bilgi verir. YTA uygulama katmanındaki YTA uygulamaları ağın ihtiyaçları ve ağın davranışları arasında bağlantıyı açıkça ve direkt olarak sağlayan programdır [56-58].

Şekil 3.2.'de örnek bir YTA topolojisi gözükmektedir:



Şekil 3.2. Örnek bir YTA topolojisi

YTA sonuç olarak, aşağıdaki maddeleri mümkün kıldığı için geleneksel ağdan ileriye doğru önemli bir adımı temsil eder:

Daha yüksek hız ve esneklikle artırılmış denetim: Birden çok donanım aygıtını manuel olarak programlamak yerine, geliştiriciler, yalnızca açık standart bir yazılım tabanlı denetleyici programlayarak bir ağ üzerinden trafik akışını kontrol edebilir. Ağ yöneticileri ayrıca, merkezi bir kontrolör aracılığıyla herhangi bir sayıda donanım cihazıyla iletişim kurmak için tek bir protokol seçebildikleri için ağ ekipmanı seçiminde daha fazla esnekliğe sahiptir.

Özelleştirilebilir ağ altyapısı: YTA ile kullanıcılar, ağ altyapısını gerçek zamanlı olarak tek bir merkezi konum üzerinden değiştirmek için ağ hizmetlerini yapılandırabilir ve sanal kaynakları tahsis edebilir. Bu, ağ yöneticilerinin ağ üzerinden veri akışını optimize etmesine ve daha fazla kullanılabilirlik gerektiren uygulamalara öncelik vermesine olanak tanır.

Güçlü güvenlik: YTA, tüm ağa görünürlük sağlayarak güvenlik tehditlerine ilişkin daha bütünsel bir görünüm sağlar. İnternete bağlanan akıllı cihazların yaygınlaşmasıyla birlikte YTA, geleneksel ağlara göre açık avantajlar sunar. Operatörler, farklı güvenlik seviyeleri gerektiren cihazlar için ayrı bölgeler oluşturabilir veya güvenliği ihlal edilmiş cihazları ağın geri kalanına bulaşmamaları için hemen karantinaya alabilir.

BÖLÜM 4. ANOMALİ TESPİTİ

Ağlarda anomali tabanlı izinsiz giriş tespit terimi, ağ trafiğinde normal davranışa uymayan durumları bulma olarak ifade edilebilir. Anomali tespiti, ağ izinsiz girişlerini belirlemek için yararlı olan önemli bir veri analizi görevidir. Anomali tespiti, kredi kartlarında dolandırıcılık tespiti, ağlarda izinsiz giriş tespiti gibi alanlarda kapsamlı uygulamalara sahiptir. Örneğin, bir bilgisayar ağındaki olağan dışı bir trafik, saldırıya uğramış bir bilgisayarın hassas verileri yetkisiz bir ana bilgisayara gönderdiği anlamına gelebilir [59].

Bilgisayar sistemlerinde çeşitli izinsiz giriş veya saldırı sınıfları vardır. Bunlardan bazıları aşağıdaki gibi sıralanmıştır [60]:

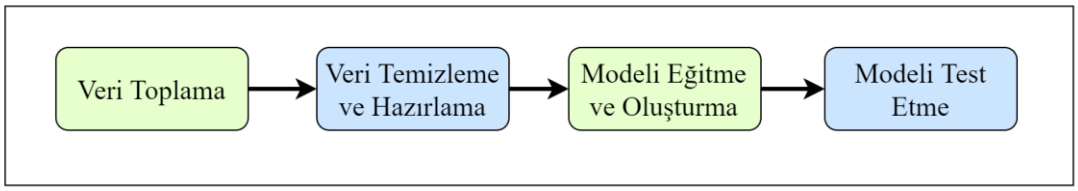
Denial of Service(DoS): normal bilgi işlem ortamını bozmayı ve hizmeti kullanılamaz hale getirmeyi hedefleyen bir ağ veya ana bilgisayarın kaynaklarına yönelik hakların kötüye kullanılmasıdır. DoS saldırısının basit bir örneği, sunucu çok sayıda bağlantı isteğiyle dolup taştığında meşru kullanıcıların bir web hizmetine erişimini reddetmesidir.

Probe: Hedeflenen bir ağ veya ana bilgisayar hakkında bilgi toplamak için ve daha resmi olarak keşif amacıyla kullanılır. Keşif saldırıları, bir ağa bağlı makinelerin türleri ve sayıları hakkında bilgi toplamanın oldukça yaygın yollarından biridir ve bir ana bilgisayara, yüklenen yazılım türlerini ve kullanılan uygulamaları belirlemek için saldırı yapılabilir. Bir Probe saldırısı, bir ana bilgisayar veya ağın güvenliğini aşmak için gerçek bir saldırının ilk adımı olarak kabul edilir.

Worm: Kullanıcı müdahalesi olmadan bilgisayar sistemlerindeki ağ hizmetleri aracılığıyla yayılan kendi kendini kopyalayan saldırı türüdür. Ağ bant genişliğini tüketerek ağa büyük zarar verebilir.

4.1. Anomali Tespit Algoritmaları

Son yıllarda makine öğrenimi, anomali tespitinde önemli bir rol oynamaya başladı. Araştırmacılar tarafından ağlarda çok sayıda anomali tabanlı saldırı tespit tekniği geliştirilmiştir. Makine öğrenimi algoritmaları, temel olarak mevcut verileri toplama, verileri temizleme ve hazırlanma, modelleri oluşturma, eğitme ve en sonda modeli test etme gibi Şekil 4.1.'deki gösterilen dört aşamadan oluşan tekrarlayan bir süreçtir. Veri toplama için, gerçek hayattaki anketlerden veya deneylerden veri toplama ve sentetik veri oluşturma dahil olmak üzere iki metot vardır. Birinci metotta, verileri gerçek hayattan gözlemleyerek veya deneyerek elde etmek mümkün ise veriler doğrudan sahadan toplanmaktadır. Veriler mevcut değilse, veriseti programlı olarak oluşturulur.



Şekil 4.1. Makine öğrenimi algoritmalarının ana iş akış diyagramı

Bulanık mantık, modern bilgisayarların doğruluk derecelerine dayalı olarak hesaplama yöntemidir. Ayrıca, izinsiz giriş tespiti problemi, toplanan verilerdeki birçok sayısal özelliği ve çeşitli türetilmiş istatistiksel verileri içerir. Modelleri doğrudan sayısal veriler üzerine oluşturmak genellikle yüksek algılama hatalarına(false positive) neden olur [61].Aynı şekilde Gharboui ve arkadaşları uçtan uca yol hesaplamasında Sequential Hypothesis Testing algoritmasını kullanarak bilgi işlem hizmetlerinde kötü niyetli kullanımı tespit etmişlerdir. Tespit doğruluğu oldukça iyi olmasına rağmen, yanlış positif oranı yüksek çıkmıştır. Dezavantajları aşağıdaki gibidir:ndan bir diğeri ise gerçek zamanlı veriler kullanılmamıştır [62].

Nguyen ve Roughan birden fazla ISP(Internet Service Provider) verileri gözlemleyerek ve Hidden Markov Modeli ile anomaliler algılanmıştır. Küçük boyutlu veriler üzerinde oldukça iyi sonuçlar vermesine rağmen büyük boyutlu verilerde performansı oldukça düşüktür [63].

Ancak performans sebebiyle bu çalışmada on popüler makine öğrenimi yaklaşımını seçtik. Bu çalışma için seçilen makine öğrenimi yöntemleri; K-Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes ve QDA. Seçilen makine öğrenimi algoritmalarına ilişkin kısa açıklamalar aşağıda verilmiştir:

4.1.1. K-nearest neighbors

K-nearest neighbors(KNN) k > 0 komşuları ve Öklid mesafesi veya diğer mesafe ölçüleri gibi benzerlik ölçülerini kullanarak giriş verilerini kategorilere ayıran basit bir sınıflandırma algoritmasıdır.

$$D = \sum_{i=0}^{k} \sqrt{(X_i^{input} - X_i^{test})^2} \tag{4.1}$$

KNN, girdi verilerini k > 0 komşular ve Öklid mesafesi gibi benzerlik ölçüleri kullanarak kategorize eden basit bir sınıflandırma algoritmasıdır. KNN yeni durum ile mevcut durumlar arasındaki benzerliği karşılaştırır ve yeni durumu mevcut kategorilere en çok benzeyen kategoriye yerleştirir. Yapılan denemelerden sonra en iyi sonuç komşu sayısı k=3 olarak alındığında elde edilmiştir. Burada D mesafe, k komşu sayısıdır, Xinput ve Xtest giriş ve test verilerinin öznitelik vektörleridir

Avantajları aşağıdaki gibidir;

- KNN oldukça sezgisel ve basittir. KNN algoritmasının anlaşılması ve uygulanması oldukça kolaydır.
- KNN'nin varsayımları yoktur.

- Sürekli gelişir. Sınıflandırıcı, yeni eğitim verilerini topladığımızda hemen adapte olur. Algoritmanın gerçek zamanlı kullanım sırasında girişteki değişikliklere hızlı yanıt vermesini sağlar.
- Çok sınıflı problem için uygulanması çok kolaydır. Sınıflandırıcı algoritmalarının çoğu ikili problemler için uygulanması kolaydır ve çoklu sınıf için uygulanması için çaba gerektirirken, KNN herhangi bir ekstra çaba harcamadan çoklu sınıfa ayarlanır.

Dezavantajları aşağıdaki gibidir;

- KNN'nin uygulanması çok kolay olabilir, ancak çok büyük verilerde hesaplama hızı düşmektedir.
- Boyutluluk: KNN, az sayıda girdi değişkeni ile iyi çalışır, ancak değişkenlerin sayısı arttıkça KNN algoritması yeni veri noktasının çıktısını tahmin etmekte zorlanır.
- KNN homojen özniteliklere ihtiyaç duymaktadır. Öklid veya Manhattan mesafeleri gibi ortak bir mesafe kullanarak KNN oluşturmaya karar verirseniz, özniteliklerdeki mutlak farklılıklar aynı ağırlıkta olduğu için özniteliklerin aynı ölçeğe sahip olması tamamen gereklidir, yani belirli bir öznitelik 1'deki mesafe, öznitelik 2 için aynı anlamına gelmelidir.
- KNN ile ilgili en büyük sorunlardan biri, yeni veri girişini sınıflandırırken dikkate alınacak optimum komşu sayısını seçmektir.

4.1.2. Linear SVM ve RBF SVM

Sınıflandırma için iki grup arasında bir sınır çizilerek grupları birbirinden ayırmak mümkündür. SVM(Support Vector Machine) bu sınırın nasıl çizileceğini belirler [64].

Linear SVM ve RBF(Radial Basis Function) SVM'nin arkasındaki fikir, girdi verilerinin doğrusal veya doğrusal olmayan hiper düzlemler kullanılarak birkaç sınıfa ayrılmasıdır. Algoritma, doğrusal bir SVM için doğrusal bir fonksiyon denklemi (2)

ve giriş verilerini ikili sınıflara ayıran bir radyal tabanlı fonksiyon (RBF) SVM denklemi için doğrusal olmayan bir fonksiyon oluşturur. Bir SVM fonksiyonunun ayırma çizgisine hiperdüzlem denir.

$$\min\frac{1}{2}||w_1||^2,\ \min\frac{1}{2}||w_2||^2,\ y_i^{(-1)} \le w_1 x_i^{(1)} + w_2 x_i^{(2)} + b,\\ y_i^{(+1)} \ge w_1 x_i^{(1)} + w_2 x_i^{(2)} + b \qquad (4.2)$$

w1 ve w2, karar verme fonksiyonunun sınırını belirleyen normal veya ağırlık vektörleridir, y (-1) ve y (+1), sınıflar için hiperdüzlem fonksiyonlarıdır (1 - normal, -1 - saldırı), x (1) ve x (2), belirli bir giriş paketinin öznitelik vektörleridir ve b biastır.

RBF SVM algoritması da doğrusal SVM yöntemine benzer, ancak doğrusal bir işlev yerine radyal tabanlı doğrusal olmayan bir işlev çekirdeği kullanır. Bu doğrusal ve RBF ikili SVM algoritmaları, ekstra büyük veri kümeleriyle (örneğin, birkaç milyon eğitim veri çifti) çalışırken verimli bir sınıflandırma ortamı sağlar. Bizim çalışmamızda veri setinde 3 milyon kayıt olduğu için bu yöntem kullanılmıştır. Çalışmamızda doğrusal SVM için düzenlileştirme parametresi C=0,001 ve RBF tabanlı SVM sınıflandırıcısı için C=1 kullanılmıştır. Normal ve saldırı sınıfı etiketi olduğundan, her iki sınıflandırıcı için de Derece=3 ve gama=2 ayarlanmıştır.

Avantajları aşağıdaki gibidir;

- SVM, doğru parametrelerle doğru çekirdeği seçmekle ilgilidir. Bu durum doğrusal olmayan verileri de kullanmayı sağlar.
- SVM, yüksek boyutlu alanlarda daha etkilidir.
- SVM, boyut sayısının örnek sayısından fazla olduğu durumlarda etkilidir.
- Veriler hakkında bilgimiz olmasa bile SVM'ler oldukça iyi sonuçlar vermektedir.

Dezavantajları aşağıdaki gibidir;

- SVM, doğru çekirdeği seçmek kolay değildir.
- SVM ile yüksek boyutlu verileri eğitmek oldukça uzun zaman almaktadır.
- SVM, insanlar tarafından anlaşılması ve yorumlanması oldukça zordur.

4.1.3. Gaussian process

Gaussian Process (GP) sınıflandırıcı, regresyon ve sınıflandırma görevlerinde kullanılan denetimli bir makine öğrenme algoritmasıdır. Girdi verilerinin etiketlerini, eğitim verisetindeki gözlemleri inceleyerek tahmin eder.

GP sınıflandırıcısı, ikili lojistik regresyon ve ikili sınıflandırma görevlerinde kullanılan denetimli bir makine öğrenme algoritmasıdır. Laplace yaklaşımını kullanarak bir eğitim veri kümesindeki gözlemleri yaparak olasılıksal güven düzeyiyle girdi verilerinin etiketini tahmin eder.

$$k(X^{(input)}, X^{(test)}) = (f(X^{(input)}) - m(X^{(input)})) \times \\ \times (f(X^{(test)}) - m(X^{(test)})) \quad (4.3)$$

GP fonksiyon denkleminde (3), ortalama m(X(input)) ile ve ortak varyans ise (çekirdek) k(X(input) , X(test)) fonksiyonları ile belirtilir. Çekirdek seçim süreci ikili sınıflandırma için çok önemlidir, bu nedenle giriş verileri doğrusal olarak dağıtıldığı için bu çalışmada çekirdek = 1.0 * RBF(1.0) seçilmiştir. Diğer parametreler aynı şekilde bırakılmıştır.

Avantajları aşağıdaki gibidir:

- Tüm parametreleri tam olarak optimize edebilmektedir.
- Hesaplama yaparken oldukça hızlıdır.
- Model çalışırken esneklik sağlamaktadır. Farklı çekirdekler kullanabilir.

Dezavantajları aşağıdaki gibidir:

- Oldukça zayıf bir şekilde ölçeklendirilmektedir.

- Yüksek boyutlu alanlarda verimlilik kaybına neden olur.

4.1.4. Decision tree ve random forest

Decision Tree(DT) algoritması, ağaç yapılı kurallar üzerine kurulmuştur. Kurallar, entropi bilgi kazanma fonksiyonu [63] kullanılarak oluşturulurken; Random Forest(RF) algoritması, sınıflandırma modelini birkaç ayrı karar ağacı ile oluşturur. Doğru bir karar vermek için tek tek karar ağaçlarından tüm tahmin değerlerini bir araya getirir.

DT algoritması, ağaç yapılı kurallar üzerine kurulmuştur. Kurallar, bir entropi bilgi kazanım fonksiyonu denklemi (4) veya Gini yaklaşımı kullanılarak çıkarılırken, RF algoritmasında ise sınıflandırma modelini birkaç bireysel karar ağacıyla oluşturur. Nihai bir karar vermek için tek tek karar ağaçlarından tüm tahmin değerlerini toplar. Karar ağacı, eğitim verilerinden genelleme yapmaz, ancak sınıflandırma için parametrik olmayan denetimli bir öğrenme yöntemi kullanıldığından tüm örnekleri ezberler.

$$E(S) = \sum_{c \in C} - p(c)\ \log_2\ p(c) \tag{4.4}$$

S, mevcut veri kümesinin öznitelikleridir, C, sınıfların kümesidir, c - etiket ve p(c) sınıfların olasılığıdır.

Avantajları aşağıdaki gibidir:

- Anlaşılması ve yorumlaması oldukça basittir.
- Çok az veri hazırlığı gerektirir.
- Çoklu çıktı sorunlarını çözebilir.
- İstatistiksel testler kullanarak bir modeli doğrulamak mümkündür. Bu, modelin güvenilirliğini mümkün kılar.
- Verilerin üretildiği gerçek model tarafından varsayımları bir şekilde ihlal edilse bile iyi performans gösterir.

Dezavantajları aşağıdaki gibidir:

- Çok sayıda ağaç oluşturduğu için karmaşıklığa neden olur.
- Tahminleri ne düzgün ne de süreklidir.
- Verilerdeki küçük bir değişiklikle önemli ölçüde değişebilir.

4.1.5. Neural net

Yapay Sinir Ağı(YSA), insan beynini taklit eden bir sistemdir. İnsan beyninden esinlenerek çevresel değişikliklere göre, anormal saldırı tespit edilmiştir [65]. Aşağıdaki denklemdeki gibi girdi-çıktı katmanları, gizli katman(lar), nöronlar, aktivasyon ve karar fonksiyonlarını içerir. Bizim çalışmamızda basit, üç katmanlı bir sinir ağı seçtik. Daha fazla katman ve daha fazla nöron daha yüksek doğruluk sağladığı için daha büyük sinir ağları oluşturulabilir. Ancak bu, sistemin performansını düşürür. Aktivasyon fonksiyonu olarak ReLu kullanıldı. Bu çalışmada ağırlık optimizasyonu için Adam optimizer yaklaşımı kullanılmıştır. Öğrenme oranı 0,001 ve bozulma 0,9'du.

$$y = f\left(\sum_{i=0}^{n} w_i X_i^{giriş} + b\right) \tag{4.5}$$

burada, y tahmin edilen etikettir, wi ağırlık vektörleridir ve X(giriş) giriş verileridir (öznitelik vektörü).

Avantajları aşağıdaki gibidir:

- Girilen veriler, veritabanı yerine kendi ağlarında depolanır. Veri kaybı çalışma şeklini etkilemez.
- Eksik bilgi ile çalışabilme.
- YSA olayları öğrenir ve benzer olaylar hakkında yorum yaparak karar verir.
- Hata toleransına sahiptir.

Dezavantajları aşağıdaki gibidir:

- YSA, yapılarına uygun olarak paralel işlem gücüne sahip işlemciler gerektirir. Bu nedenle algoritmanın gerçekleştirilmesi donanıma bağımlıdır.
- YSA herhangi bir çözüm ürettiğinde, neden ve nasıl olduğuna dair bir bilgi vermez. Bu durum YSA'ya olan güveni azaltır.
- Deneyim ve deneme yanılma yoluyla uygun ağ yapısı elde edilir. YSA'nın çalışmasını belirlemek için belirli bir kural yoktur.
- Ağın çalışma süresi bilinmemektedir.

4.1.6. Ada-boost

Ada-boost sınıflandırma algoritması, rastgele ormanlar gibi topluluk algoritmasıdır. Bu algoritmanın rastgele ormandan farkı; rastgele ormanda tek tek karar ağaçlarını kullanırken; Ada-boost algoritmasında, birkaç zayıf sınıflandırıcı algoritmayı tek bir sistem altında birleştirmektir. Bu algoritma kullanılarak girdi verilerini denklem (6)'daki gibi sınıflandırılır. Bu çalışma için 50 zayıf karar ağacı sınıflandırıcı seçilmiş ve öğrenme oranı 1.0 olarak seçilmiştir. Bu işlev ile giriş verileri sınıflandırılır.

$$H(X^{giriş}) = \text{sign}(\sum_{t=1}^{T} \alpha_t \ h_t(X^{giriş})) \quad (4.6)$$

burada H(X(giriş)) karar fonksiyonudur, T bir sınıflandırıcılar kümesidir, α t sınıflandırıcıların ağırlığıdır, h t (Xgiriş) zayıf sınıflandırıcıların çıktısıdır.

Avantajları aşağıdaki gibidir:

- Uygulaması çok basittir. Parametrelerde ince ayar yapmaya daha az ihtiyaç duyulduğundan kullanımı daha kolaydır.
- Oldukça iyi bir genellemeye sahiptir.
- Sistemde zayıf sınıflandırıcılar olsa bile doğruluğu artırmaya yarar.

Dezavantajları aşağıdaki gibidir:

- Yetersiz çözüm

- Gürültülü verilere ve farklı değerlere oldukça duyarlıdır.

4.1.7. Naive bayes

Naive Bayes ağı, ilgi değişkenleri arasındaki olasılıksal ilişkileri karşılaştıran algoritmadır. Bu teknik genellikle, istatistiksel planlar ile kombinasyon halinde izinsiz giriş tespiti için kullanılmıştır. Naive Bayes algoritması, öznitelik vektörleri arasındaki varsayımlara dayanan Bayes'in olasılık teoremini kullanarak oluşturur [66].

$$p(C_k|X^{giriş}) = \frac{p(C_k)p(X^{giriş}|C_k)}{p(X^{giriş})} \quad (4.7)$$

Bu denklemde, C k sınıfların kümesidir, p (C_k) olasılık fonksiyonudur.

Avantajları aşağıdaki gibidir:

- Bu algoritma hızlı çalışır ve çok zaman kazandırabilir.
- Naive Bayes, çok sınıflı tahmin problemlerini çözmek için uygundur.
- Özniteliklerin bağımsızlığı varsayımı doğruysa, diğer geleneksel algoritmalardan daha iyi performans gösterebilir Bu durum sayesinde az eğitim verisi ile iyi sonuçlar almayı sağlamaktadır.

Dezavantajları aşağıdaki gibidir:

- Naive Bayes, kullanılan özniteliklerin birbirinden bağımsız olduğunu ve gerçek hayatta nadiren gerçekleştiğini varsayar. Bu durum algoritmanın gerçek dünyadaki kullanım durumlarında uygulanabilirliğini sınırlar.
- Bazı durumlarda tahminleri yanlış olabilir. Doğruluk kaybı olasılığına sebep verebilir.

4.1.8. Quadratic Discriminant Analysis

Normal olarak dağıtılan her bir sınıfın ölçüm varsayımlarını kullanan doğrusal olmayan bir diskriminant analiz örneğidir.

Quadratic Discriminant Analysis(QDA), ölçümlerin normal olarak dağıldığının varsayıldığı doğrusal diskriminant analizi (LDA) ile yakından ilgilidir. Bununla birlikte, LDA'dan farklı olarak, QDA'da her bir sınıfın kovaryansının aynı olduğu varsayımı yoktur.

$$P(x|y = k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{\left(-\frac{1}{2}(x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k)\right)} \quad (4.8)$$

Bu denklemde ‘d’ öznitelik sayısıdır ve bizim çalışmamızda öznitelik çifti eğitimi için ‘d’ 2 seçilmiştir, tüm öznitelik eğitimi için 12 seçilmiştir.

Avantajları aşağıdaki gibidir:

- Hızlı sınıflandırma.
- Sınıflandırma daha doğru.
- LDA’ya göre daha iyi performans gösterir.

Dezavantajları aşağıdaki gibidir:

- Boyut azaltma tekniği olarak kullanılamamasıdır.
- Gerekli parametreleri tahmin etmek için LDA’ya göre daha fazla hesaplama gerekir.

4.2. Sonuç

Sonuç olarak son on yılda makine öğrenimi geleneksel yöntemlere göre, anomali tespitinde daha iyi doğruluk sonucu verdiği tespit edilmiştir. Geliştiriciler tarafından ağlarda çok sayıda anomali tabanlı saldırı tespit ve önleme tekniği geliştirilmiştir.

Farklı saldırı senaryolarını tespit ederek ağlar üzerinden koruma sağlamak, herhangi bir geleneksel yöntemle zorlu bir iştir. Saldırı tespit sistemleri, veri alışverişi için daha güvenli bir ortam sağlamak amacıyla ağlara yapılan farklı saldırı türlerini analiz etmek ve belirlemek için ortaya çıkmıştır. Bu çalışmada en popüler on makine öğrenimi algoritmalarının(Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes ve QDA) çalışma sistemi açıklandı. Bu algoritmaların öznitelikleri, avantajları ve dezavantajları madde madde yukarıda açıklanmıştır.

BÖLÜM 5. TEST ORTAMI

5.1. Platformun Oluşturulması

Çalışmamızı yürüteceğimiz platformun oluşturulması için VirtualBox, POX, Python ve Mininet kullanılacaktır.

5.1.1. VirtualBox

YTA platformunun kurulması için VirtualBox kullanılacaktır. VirtualBox işletim sistemleri sanal olarak çalıştırabilen sanallaştırma programıdır. VirtualBox; Windows, Mac, Linux ve Solaris dahil olmak üzere tüm x86 platformlarında çalışan ücretsiz ve açık kaynaklı bir çözümdür. VirtualBox ayrıca sistemler arasında geçişi kolaylaştıracak özelliğe sahiptir [67].

5.1.2. Python

YTA kontrolde makine öğrenme algoritmasının kodlanması için Python kullanılacaktır. Python nesne yönelimli ve yüksek seviyeli bir dildir. Python, veri analizi ve istatistiksel işlemlerde kullanılan çok popüler bir dildir. Programlama dilleri makine mantığı ile insan mantığı arasında köprüdür. Bir dilin makine mantığına daha yakın olması makine üzerinde daha hızlı çalışabilmesi sonucunu doğurur. Python açık ve ücretsiz olduğu için yazılım araçlarını tasarlaması aşamasında oldukça yardımcı olacaktır [68].

5.1.3. POX

NOX, POX, Beacon, Maestro ve Floodlight YTA kontrolör platformlarından birkaçıdır. POX, YTA kontrol uygulamaları için açık kaynak olarak kullanılan bir platformdur. NOX ve POX platformları Python tabanlı iken, Beacon,Maestro ve Floodlight platformları Java tabanlıdır [69]. Hızlı gelişim ve prototip oluşturmayı sağlayan POX, NOX'ten daha yaygın olarak kullanıldığı için tezimizde POX platformu kullanılacaktır [70].

5.1.4. Mininet

YTA simülatörü için Mininet kullanılacaktır. Mininet YTA oluşturmayı, test etmeyi ve gerçekleştirmeyi sağlayan açık kaynak kodlu bir uygulamadır.

5.2. Bot-Iot Veriseti

BoT-IoT veri seti UNSW Canberra Cyber Merkezinin Cyber Range Lab'ında gerçekçi bir ağ ortamı tasarlanarak üretilmiştir. Ortam, normal ve botnet trafiğinin oluşturduğu trafiği içerir. Verisetinin kaynak dosyaları pcap şeklinde oluşturulmuştur.

Bu laboratuar ortamında çeşitli IoT sensörleri (hava durumu istasyonu, akıllı buzdolabı, hareketle etkinleştirilen ışıklar, uzaktan etkinleştirilen garaj kapısı, akıllı termostat) kullanılarak paketler elde edilmiştir

Yakalanan pcap dosyalarının boyutu 69,3 GB olup 72.000.000'den fazladır. Saldırı olarak DDoS, DoS, İşletim Sistemi Saldırıları, Servis Taraması Saldırıları ve Keylogging saldırılarını içerir.

Bot-Iot verisetinde bulunan öznitelikler ve açıklamaları aşağıdaki gibidir ve * işaretli olanlar seçilen özniteliklerdir [16]:

Tablo 5.1. Öznitelikler ve Açıklamaları

Öznitelik	Açıklama
pkSeqID	Satır Tanımlayıcısı
Stime	Başlangıç zamanı kaydı
flgs	İşlemlerde görülen akış durum bayrakları
flgs number	Bayrak özelliğinin sayısal gösterimi
proto	Ağ akışında mevcut olan işlem protokollerinin metinsel gösterimi
proto number	Proto özelliğinin sayısal gösterimi
saddr	Kaynak IP Adresi
*sport	Kaynak port numarası
daddr	Hedef IP adresi
*dport	Hedef port numarası
pkts	İşlemdeki toplam paket sayısı
bytes	İşlemdeki toplam bayt sayısı
state	İşlem durumu
*state number	Durum özelliğinin sayısal gösterimi
ltime	Bitiş zaman kaydı
*seq	Argus sıra numarası(Bu öznitelik, Argus aracı kullanılarak .pcap dosyalarından .argus formatındaki dosyalar olarak elde edilmiştir.)
dur	Toplam süre kaydı
*mean	Toplam kayıtların ortalama süresi
*stddev	Toplam kayıtların standart sapması
sum	Toplam kayıtların toplam süresi
*min	Toplanan kayıtların minimum süresi
*max	Toplanan kayıtların maksimum süresi
spkts	Kaynaktan hedefe paket sayısı
dpkts	Hedeften kaynağa paket sayısı
sbytes	Kaynaktan hedefe bayt sayısı
dbytes	Hedeften kaynağa bayt sayısı
rate	İşlemdeki saniye başına toplam paket sayısı
*srate	Kaynaktan hedefe saniyedeki paket sayısı
*drate	Hedeften kaynağa saniyedeki paket sayısı
*N_IN_Conn_P_SrcIP	Kaynak IP başına gelen bağlantı sayısı
*N_IN_Conn_P_DstIP	Hedef IP başına gelen bağlantı sayısı
attack	0:Normal 1:Saldırı
category	Trafik kategorisi
subcategory	Trafiğin alt kategorisi

Tablo 5.2.'de verisetlerinin karşılaştırılmış hali gözükmektedir [16]:

Tablo 5.2. Verisetlerinin Karşılaştırılması(E:Evet,H:Hayır)

Veriseti	Gerçekçi deneme ortamı	Gerçekçi trafik	Etiketli veri	IoT verileri	Farklı saldırı senaryoları	Tam paket yakalama	Yeni oluşturulan öznitelikler
Darpa98	E	H	E	H	E	E	H
KDD99	E	H	E	H	E	E	E
DEFCON	H	H	H	H	E	E	H
UNIBS	E	E	E	H	H	E	H
CAIDA	E	E	H	H	H	H	H
LBNL	H	E	H	H	E	H	H
ISCX	E	E	E	H	E	E	E
CICIDS	E	E	E	H	E	E	E
TUIDS	E	E	E	H	E	E	E
Bot-IoT	E	E	E	E	E	E	E

DARPA 98 veri seti, MITS Lincoln Lab tarafından izinsiz giriş tespit sistemlerini değerlendirmek için üretildi. 7 haftada üretilen veriseti, 4 GB'lık ikili veriden yapıldı ve internete bağlı küçük bir Hava Kuvvetleri ağını simüle etmiştir [71].

KDD99 veri kümesi, saldırılar ile normal bağlantılar arasında ayrım yaparak; izinsiz giriş tespit sistemlerinin değerlendirilmesi için DARPA 98 veri kümesinden üretildi. Günümüzde hala kullanılmasına rağmen mevcut saldırı durumlarını yansıtmama ve öğrenme problemi gibi sorunları vardır [72].

DEFCON-8 veri kümesi Bayrak Yakalama yarışması sırasında kaydedilen port taramalarından ve saldırılarından oluşur [73].

UNIBS veriseti, İtalya'daki Brescia Üniversitesi tarafından geliştirilmiştir. Tcpdump aracılığıyla 20 iş istasyonundaki verileri toplamıştır. Gerçekçi bir veriseti olmasına rağmen DoS saldırılarıyla sınırlı olması eksik yönlerinden birisidir [74].

CAIDA veriseti çeşitli türdeki verilerin toplanmış halidir. Payload hariç başlıktan oluşur [75]. Popüler bir veri kümesi, 4 Ağustos 2007'de gerçekleşen DDoS saldırılarından bir saatlik anonim saldırı izlerini içeren CAIDA DDoS 2007'dir.

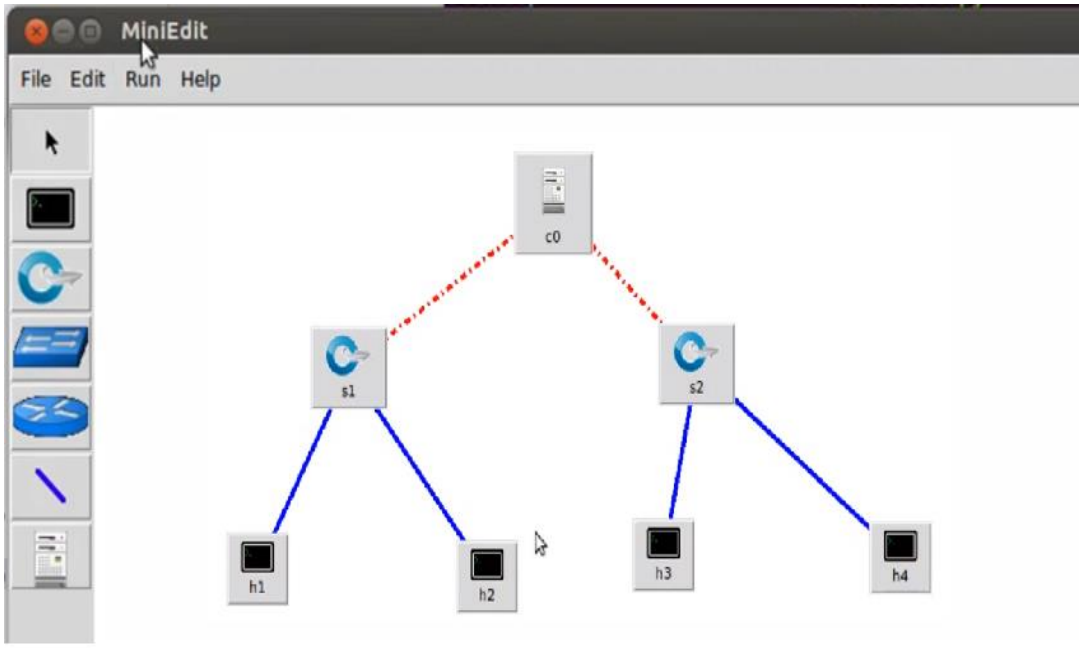
LBNL veriseti, sadece başlık verilerinden oluşmaktadır. Lawrence Berkley Ulusal Laboratuvarı'nda iki uçlu yönlendiriciden gerçek gelen, giden ve yönlendirme trafiği toplanarak geliştirilmiştir [76].

ISCX veri eti, Kanada Siber Güvenlik Enstitüsü'nde üretildi. Saldırı tespit sistemlerini değerlendirmek için birçok gerçek iz analiz edildi [77]. Son zamanlarda, aynı kurumda CICDS2017'de yeni bir veri seti oluşturuldu. CICIDS2017 çeşitli saldırı senaryolarından oluşmaktadır.

TUIDS veriseti Hindistan'daki Tezpur Üniversitesi tarafından oluşturuldu. Bu veriseti fiziksel ortamda oluşturulan DoS ve DDoS atak senaryolarından oluşmaktadır [78].

Bu çalışmada, sistemimizi eğitmek ve test etmek için Nvidia 4 GB 1050, 11 GB 2080Ti GPU'lar, CPU 9700K ve 32 gb Ram kullanıldı. Eğitim ve test sistemleri, çok çeşitli esnek makine öğrenimi ve Scikit-learn [79] veri ön işleme ve makine öğrenimi eğitimi ve testi için, Numpy [80] (matris için) gibi diğer bilimsel çerçeveler nedeniyle Python programlama dili kullanılarak geliştirildi. Pandas [81] dosyadan veri okumak, veri işlemek ve işlenmiş veriyi yazmak için ve Matplotlib [82] veri ve sonuçları görüntülemek için kullanıldı. Sınıflandırma sistemlerinin doğruluk hesaplaması için, karışıklık matrisi olarak bilinen geleneksel bir doğruluk hesaplama metriği kullanıldı.

Yazılım Tanımlı Ağ kontrolcüsü kuruldu. Topoloji çizimini yapmak için mininet programını kullandık. Mininet arayüzü aşağıdaki resimdeki gibidir:



Şekil 5.1. Mininet arayüzü

Seç Aracı: Arayüz üzerinde düğümleri hareket etmeye ve seçmeye yarar. Seçilen öğenin yapılandırma menüsünü ortaya çıkarmak için düğümün üzerinde sağ tıklanır.

Ana Bilgisayar Aracı: Arayüz üzerinde ana bilgisayarların işlevini gerçekleştirecek düğümler oluşturur. Aracın üzerine tıklayıp arayüzde herhangi bir yere bırakmak

yeterlidir. Araç seçili kaldığı sürece arayüz üzerinde her tıklamada bilgisayar eklemeye devam edilebilir.

Anahtar Aracı: Arayüz üzerinde OpenFlow anahtarları oluşturur. Bu anahtarlar aracılığıyla bilgisayar arasında anahtar görevi görür. Bağlı olduğu bilgisayarların haberleşmesini sağlar.

Netlink Aracı: Arayüz üzerinde düğümler arasında bağlantıyı sağlar. Netlink aracını seçerek ardından seçeceğimiz düğümü tıklayıp hedef düğüme sürükleyerek bağlantıyı sağlar.

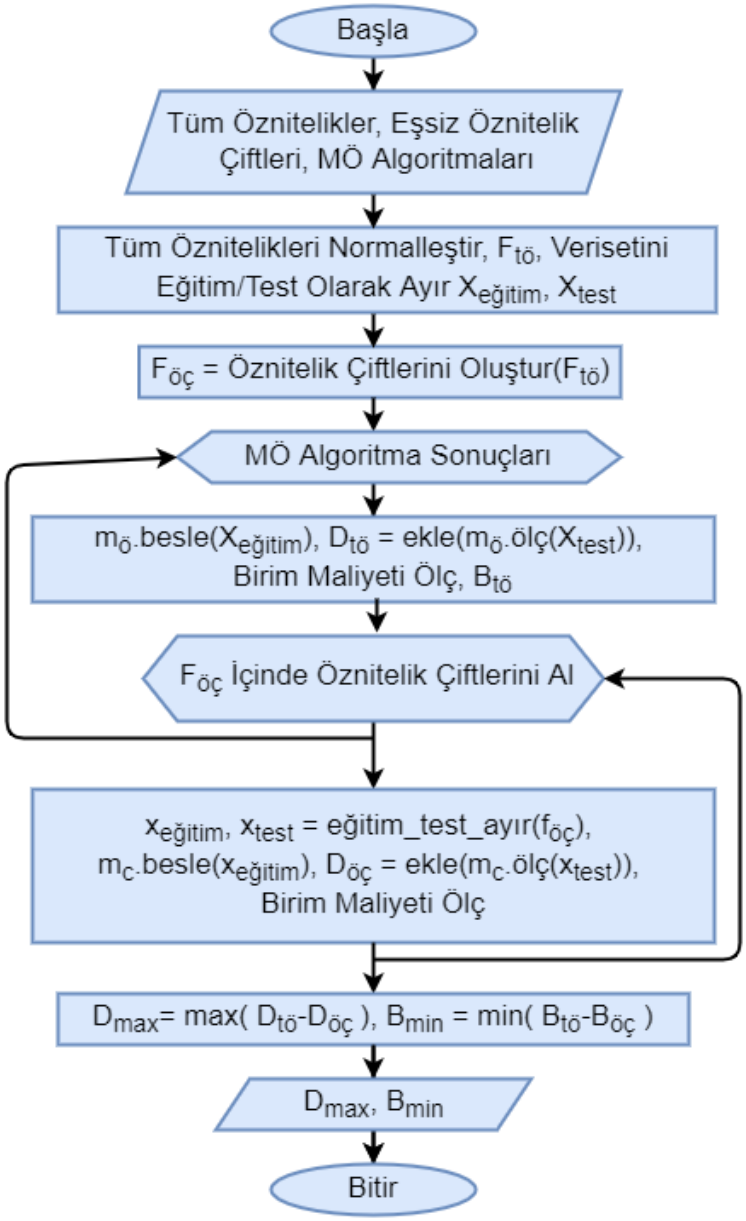
Kontrolör Aracı: Veri düzlemindeki cihazların yönetilmesini sağlar. Ağın beyni olarak kabul edilir.

5.3. Sonuç

Sonuç olarak platformda kullanılacak programların avantajları tek tek açıklanmıştır. Bu çalışmada VirtualBox, POX, Python ve Mininet kullanılacaktır. Veriseti olarak UNSW Canberra Cyber Merkezinin Cyber Range Lab'ında gerçekçi bir ağ ortamı tasarlanarak üretilen Bot-Iot veriseti kullanılacaktır. Bu verisetinin diğer verisetlerinden farkları ve kullanılacak öznitelikler detaylı bir şekilde anlatılmıştır.

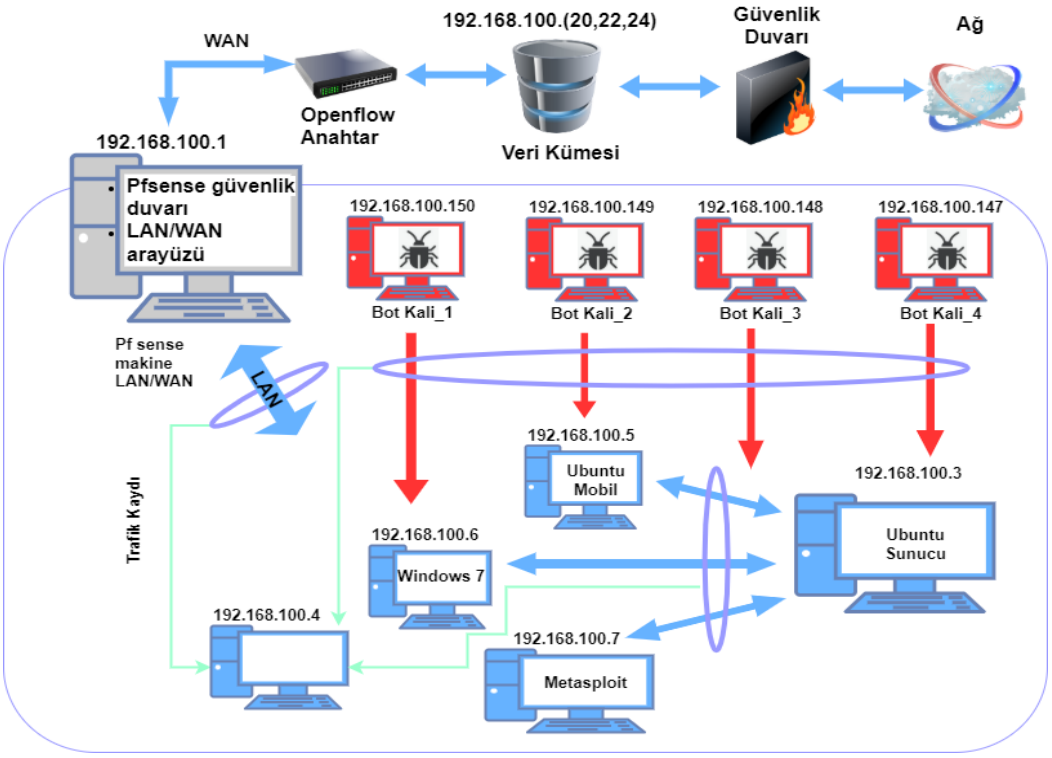
BÖLÜM 6. SİMÜLASYON SONUÇLARI

Çalışmada geliştirilen sistem Şekil 6.1.'de gösterilmiştir.



Şekil 6.1. Sistem Topolojisi

$F_{tö}$ tüm öznitelikleri ifade ederken, $F_{öç}$ birbirinden eşsiz öznitelik çiftlerini tanımlar. $D_{tö}$ tüm özniteliklerle eğitilmiş sistemin doğruluk değerlerini içerirken, $D_{öç}$ öznitelik çiftlerinin doğruluk değerlerini içerir. $B_{tö}$ tüm öznitelikler elde edilen birim maliyet sonuçları iken $B_{öç}$ öznitelik çiftleri ile elde edilen birim maliyet sonuçlarını içerir. Çalışmada oluşturulan topoloji aşağıdaki gibidir:



Şekil 6.2. Çalışmada oluşturulan topoloji

Ağımızda Şekil 6.2.'de de görüldüğü üzere 4 saldırgan Bot bilgisayarı ve 4 hedef bilgisayarımız bulunmaktadır. H1 botu (192.168.100.4) ile wireshark kullanılarak veri trafiği üretilmiştir. H2 botu (192.168.100.150) hedef olarak H6 Windows'a (192.168.100.6), H3 botu (192.168.100.149) hedef olarak H7 Mobil'e (192.168.100.5), H4 botu (192.168.100.148) hedef olarak H8 Metasploit'e (192.168.100.7) ve H5 botu (192.168.100.147) hedef olarak H9 Sunucu'ya (192.168.100.3) saldırmaktadır.

Veri Trafiği üretimi için wireshark kullanılmıştır. Wireshark, tamamen ücretsiz olup ağ bağlantısından gelen paketleri yakalayan bir uygulamadır. Wireshark, bir ağ veya

ağ protokolünü analiz etmeye yarar. Tezimizde wireshark kullanılarak Bot-Iot verisetindeki pcap dosyası kullanılarak veri trafiği üretilmiştir. Veri trafiği aşağıdaki şekil 6.3.'te gösterilmiştir.

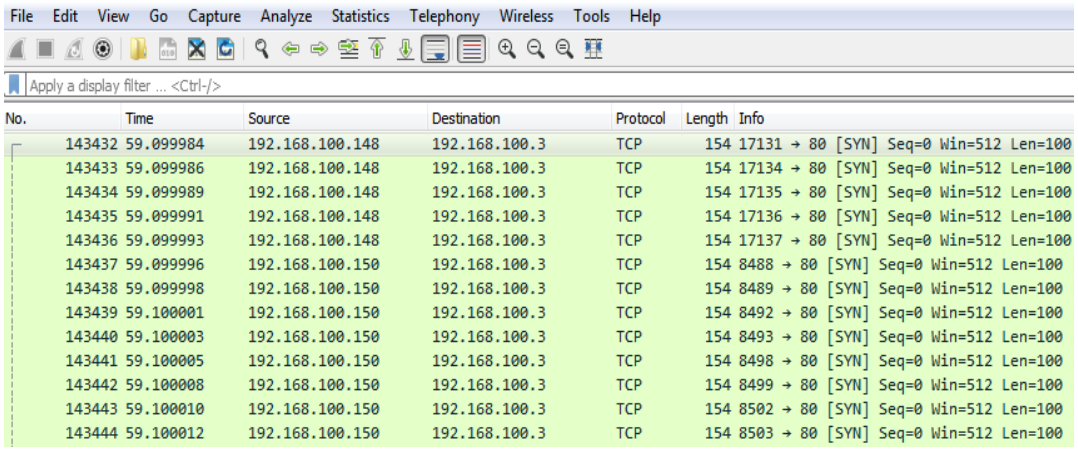
Oluşturulan veriseti çok büyük olduğu için(72.000.000 kayıt) ve 69,3 GB pcap dosyası olması nedeniyle verilerin ele alınması oldukça hantal hale getirmiştir. Üretilen trafikten 3.000.000 kayıt(eğitim+test seti) seçilmiştir. Bu yüzden elde ettiğimiz trafik toplam verisetinin yaklaşık %4'ünü oluşturmaktadır. Üretilen veriseti Python scripti ile %80 eğitim seti %20 test seti olarak ayrılmıştır.

Bir sonraki adımda verileri değiştirmeden [0,1] gibi belirli bir aralıkta ölçeklendirmek için normalizasyon uygulanmıştır. Bu adım, makine öğrenme ve derin öğrenme yöntemlerinin, hedeflerini yakınlaştırmasına ve gerçekleştirmesine yardımcı olur.

Min-max ölçeklendirme aşağıdaki formülle gerçekleştirilmiştir:

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{6.1}$$

x_i' = yeniden ölçeklendirilmiş değer
x_i = orijinal değer
x_{min} = öznitelikteki minimum değer
x_{max} = öznitelikteki maksimum değer



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
143432	59.099984	192.168.100.148	192.168.100.3	TCP	154	17131 → 80 [SYN] Seq=0 Win=512 Len=100
143433	59.099986	192.168.100.148	192.168.100.3	TCP	154	17134 → 80 [SYN] Seq=0 Win=512 Len=100
143434	59.099989	192.168.100.148	192.168.100.3	TCP	154	17135 → 80 [SYN] Seq=0 Win=512 Len=100
143435	59.099991	192.168.100.148	192.168.100.3	TCP	154	17136 → 80 [SYN] Seq=0 Win=512 Len=100
143436	59.099993	192.168.100.148	192.168.100.3	TCP	154	17137 → 80 [SYN] Seq=0 Win=512 Len=100
143437	59.099996	192.168.100.150	192.168.100.3	TCP	154	8488 → 80 [SYN] Seq=0 Win=512 Len=100
143438	59.099998	192.168.100.150	192.168.100.3	TCP	154	8489 → 80 [SYN] Seq=0 Win=512 Len=100
143439	59.100001	192.168.100.150	192.168.100.3	TCP	154	8492 → 80 [SYN] Seq=0 Win=512 Len=100
143440	59.100003	192.168.100.150	192.168.100.3	TCP	154	8493 → 80 [SYN] Seq=0 Win=512 Len=100
143441	59.100005	192.168.100.150	192.168.100.3	TCP	154	8498 → 80 [SYN] Seq=0 Win=512 Len=100
143442	59.100008	192.168.100.150	192.168.100.3	TCP	154	8499 → 80 [SYN] Seq=0 Win=512 Len=100
143443	59.100010	192.168.100.150	192.168.100.3	TCP	154	8502 → 80 [SYN] Seq=0 Win=512 Len=100
143444	59.100012	192.168.100.150	192.168.100.3	TCP	154	8503 → 80 [SYN] Seq=0 Win=512 Len=100

Şekil 6.3. Veri Trafiği (.pcap formatında)

Daha sonra veri trafiği .csv formatına dönüştürülerek kaydedilmiştir:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
pkSeqID	proto	saddr	sport	daddr	dport	seq	stddev	min	state_nun	mean	drate	srate	max	attack	category	subcategory
792371	udp	192.168.100.150	48516	192.168.100.3	80	175094	0.226784	4.100.436	4	4.457.383	0.0	0.404711	47.194.379.999.999.900	1	DoS	UDP
2056418	tcp	192.168.100.148	22267	192.168.100.3	80	143024	0.451998	3.439.257	1	3.806.172	0.225077	0.401397	44.429.300.000.000.000	1	DDoS	TCP
2795650	udp	192.168.100.149	28629	192.168.100.3	80	167033	1.931.553	0.0	4	2.731.204	0.0	0.407287	4.138.455	1	DDoS	UDP
2118009	tcp	192.168.100.148	42142	192.168.100.3	80	204615	0.428798	3.271.411	1	3.626.428	0.0	0.343654	42.297	1	DDoS	TCP
303688	tcp	192.168.100.149	1645	192.168.100.5	80	40058	2.058.381	0.0	3	1.188.407	0.0	0.1358420	4.753.628	1	DoS	TCP
420025	tcp	192.168.100.149	39733	192.168.100.5	80	156396	2.177.835	0.0	3	1.539.962	0.0	0.127912	4.619.887	1	DoS	TCP
3008812	udp	192.168.100.147	10800	192.168.100.3	80	118034	1.368.196	197.518	4	3.910.081	0.0	102.512	4.885.159	1	DDoS	UDP
1064106	udp	192.168.100.150	19625	192.168.100.3	80	184672	17.884.520.000.000.000	0.0	4	3.576.574	0.0	0.446612	44.920.800.000.000.000	1	DoS	UDP
3258414	udp	192.168.100.147	22692	192.168.100.3	80	105486	0.822443	298.003	4	3.982.845	0.0	1.003.092	4.994.536	1	DDoS	UDP
1793063	tcp	192.168.100.148	39738	192.168.100.3	80	141822	0.030759	0.143091	1	0.173851	0.103113	0.309338	0.20461	1	DDoS	TCP

Şekil 6.4. Veri Trafiği (.csv formatında)

Araştırmacılar, Korelasyon Katsayısı ve Entropi tekniklerini kullanarak en önemli on iki özelliği seçmiştir. Bizim çalışmamızda bu en önemli öznitelikleri kullanarak x ve y eksenine göre öznitelik çiftleri oluşturulmuştur. Birbirinden eşsiz 66 adet öznitelik çifti oluşturulmuştur. Her bir makine öğrenimi algoritması 66 öznitelik çiftinin her bir öznitelik çiftiyle değerlendirildi. Bu amaçla, on makine öğrenme algoritması ve bir öznitelik çifti oluşturma modülü içeren yeni bir sistem geliştirildi. Sistem, belirli sayıda öznitelikten otomatik olarak öznitelik çiftleri oluşturur ve kendisini üretilen öznitelik çiftleri ile eğitir, ardından yöntemler, oluşturulan öznitelik çiftleri ile eğitildiğinde doğruluk ve birim maliyet açısından on algoritmayı değerlendirir. Bu çalışmada en optimal ve en etkili öznitelik çiftlerini kullanarak hem sistemi hafiflettik hem de birim maliyet olarak kazanç sağladık.

6.1. Doğruluk, Precision, Recall ve Fscore Sonuçları

Şekil 6.5.'de öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde sport-dport, sport-seq, sport-stddev, sport-N_IN_Conn_P_SrcIP sonuçları elde edilmiştir. Elde edilen diğer şekiller Ek-1'de belirtilmiştir. Kırmızı atak verileri ve mavi ise normal verileri göstermektedir. Açık mavi testteki normal verileri gösterirken açık kırmızı testteki atak verileri göstermektedir. Koyu mavi eğitimdeki normal verileri gösterirken koyu kırmızı eğitimdeki atak verileri göstermektedir.

Sınıflandırma ölçütleri için kullanılan parametreler aşağıda maddeler halinde belirtilmiştir:

- True Positive(TP): Saldırıya saldırı demek.
- False Positive(FP): Saldırı olmayana saldırı demek.
- True Negative(TN):Saldırı olmayana saldırı değil demek.
- False Negative(FN): Saldırı olana saldırı değil demek.
- Accuracy(AC): Doğruluk sonucu anlamına gelmektedir. AC hesaplaması aşağıdaki denklemde gösterilmiştir.

$$AC = \frac{TP+TN}{\mathrm{TP+TN+FP+FN}} \tag{6.2}$$

- Precision: Pozitif tanımladıklarımızın gerçekten ne kadar doğru tahmin ettiğimizdir. Saldırı dediklerimizden gerçekten kaçı saldırı.

$$Precison = \frac{TP}{\mathrm{TP+FP}}$$

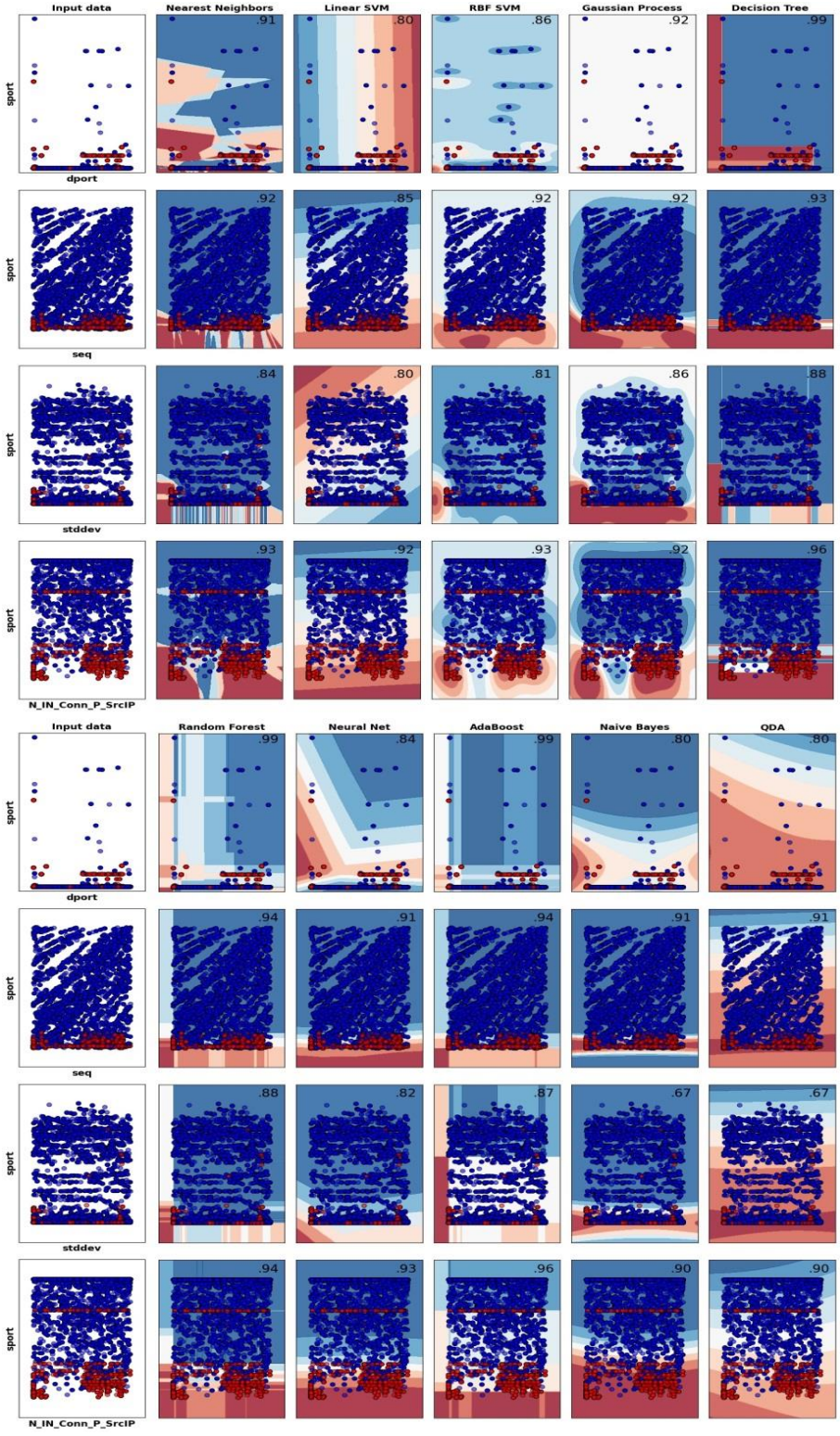
(6.3)

- Recall: Tüm pozitif sınıflardan ne kadar doğru ettiğimizdir. Saldırı olanları doğru tespit etme oranı.

$$Recall = \frac{TP}{\mathrm{TP+FN}} \tag{6.4}$$

- Fscore: Doğruluğun ölçümü: precision ile recall değerlerinin harmonik ortalaması

$$Fscore = \frac{TP+TN}{\mathrm{TP+TN+FP+FN}} \tag{6.5}$$



Şekil 6.5. Doğruluk Sonuçları(1)

Tüm öznitelik çiftleriyle elde edilen doğruluk sonuçları Tablo 6.1. ve Tablo 6.2.'de gösterilmiştir:

Tablo 6.1. Öznitelik çiftlerinin doğruluk sonuçları (ilk beş algoritma)

Öznitelik Çiftleri / Metodlar	Nearest Neighbors	Linear SVM	RBF SVM	Gaussian Process	Decision Tree
sport-dport	90,62	79,92	86,07	91,42	98,99
sport-seq	92,23	85,07	92,23	91,93	92,53
sport-stddev	84,46	79,92	81,43	85,87	88,29
sport-N_IN_Conn_P_SrcIP	92,73	91,52	92,73	92,33	95,66
sport-min_	85,77	79,92	82,04	83,55	88,4
sport-state_number	93,44	79,92	92,63	92,63	93,14
sport-mean	85,87	79,92	82,34	84,66	85,67
sport-N_IN_Conn_P_DstIP	98,49	97,68	98,49	98,39	98,79
sport-drate	82,64	80,02	83,25	83,65	84,16
sport-srate	81,74	80,12	82,44	83,25	94,25
sport-max_	86,07	79,92	82,44	83,25	84,76
dport-seq	97,17	81,23	91,83	98,69	98,79
dport-stddev	97,28	79,92	83,75	97,68	98,18
dport-N_IN_Conn_P_SrcIP	98,49	91,52	92,73	97,88	99,7
dport-min_	97,88	79,92	85,27	98,39	98,69
dport-state_number	98,49	79,92	96,37	98,49	99,19
dport-mean	98,69	79,92	85,07	99,29	99,6
dport-N_IN_Conn_P_DstIP	98,99	97,48	98,69	99,6	99,5
dport-drate	98,59	80,02	84,26	98,59	98,69
dport-srate	98,79	80,12	83,85	98,49	98,89
dport-max_	98,59	79,92	84,16	98,59	98,99
seq-stddev	95,16	92,94	94,45	94,65	96,06
seq-N_IN_Conn_P_SrcIP	99,09	94,65	98,39	98,99	99,09
seq-min_	92,73	91,73	92,23	93,04	91,62
seq-state_number	97,28	85,07	96,27	96,17	95,66
seq-mean	93,95	91,83	94,75	94,85	95,16
seq-N_IN_Conn_P_DstIP	99,39	98,79	99,5	99,6	99,7
seq-drate	92,43	80,12	90,72	90,82	91,73
seq-srate	91,62	80,12	90,72	91,02	97,98
seq-max_	94,55	92,23	94,75	94,85	95,46
stddev-N_IN_Conn_P_SrcIP	97,28	92,84	93,24	97,07	97,88
stddev-min_	89,71	79,92	81,33	87,18	88,19
stddev-state_number	98,49	79,92	95,46	98,69	97,98
stddev-mean	89,81	79,92	82,04	83,15	89,3
stddev-N_IN_Conn_P_DstIP	99,19	98,08	98,89	98,89	98,89
stddev-drate	84,46	80,02	80,32	84,86	84,76
stddev-srate	94,65	80,12	79,92	84,36	96,27
stddev-max_	90,01	79,92	81,63	83,15	88,9
N_IN_Conn_P_SrcIP-min_	97,88	92,53	96,27	96,37	98,18
N_IN_Conn_P_SrcIP-state_number	98,49	91,83	98,28	97,78	98,28
N_IN_Conn_P_SrcIP-mean	97,68	92,33	96,06	96,27	97,28
N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP	98,89	97,28	98,99	98,99	98,99
N_IN_Conn_P_SrcIP-drate	96,17	91,62	92,33	96,27	95,96
N_IN_Conn_P_SrcIP-srate	98,18	91,62	92,53	96,57	97,78
N_IN_Conn_P_SrcIP-max_	97,58	92,13	94,95	96,57	97,38
min_-state_number	92,53	79,92	92,73	92,84	92,63
min_-mean	90,21	79,92	81,94	87,08	89,3
min_-N_IN_Conn_P_DstIP	99,09	97,48	98,79	98,89	98,89
min_-drate	88,8	80,02	80,52	86,38	88,09
min_-srate	91,02	80,12	81,23	87,08	93,64
min_-max_	89,91	79,92	81,63	87,18	88,9
state_number-mean	99,29	79,92	99,29	99,6	99,19

Tablo 6.1. (Devamı)

state_number-N_IN_Conn_P_DstIP	99,09	97,48	98,69	94,85	98,59
state_number-drate	56	80,02	92,53	92,53	92,33
state_number-srate	99,29	80,12	92,73	93,34	96,17
state_number-max_	98,89	79,92	99,29	99,29	98,79
mean-N_IN_Conn_P_DstIP	98,69	97,68	99,19	99,5	99,29
mean-drate	84,86	80,02	81,13	82,74	84,36
mean-srate	90,01	80,12	81,33	82,24	96,17
mean-max_	89,91	79,92	81,43	87,49	88,4
N_IN_Conn_P_DstIP-drate	98,99	97,48	98,28	98,49	98,69
N_IN_Conn_P_DstIP-srate	99,09	97,68	98,49	98,49	99,29
N_IN_Conn_P_DstIP-max_	99,09	98,08	98,89	98,99	99,29
drate-srate	95,86	80,12	80,32	93,24	93,64
drate-max_	83,15	80,02	80,52	81,13	85,97
srate-max_	89,3	80,12	80,42	80,63	96,17

Tablo 6.2. Öznitelik çiftlerinin doğruluk sonuçları (son beş algoritma)

Öznitelik Çiftleri / Metodlar	Random Forest	Neural Net	Ada Boost	Naive Bayes	QDA
sport-dport	99,09	83,65	99,19	79,92	79,92
sport-seq	93,24	91,32	94,45	90,92	91,32
sport-stddev	88,4	79,92	86,98	67,1	67,1
sport-N_IN_Conn_P_SrcIP	93,84	92,33	95,86	90,21	90,21
sport-min_	90,01	79,92	88,09	79,92	79,92
sport-state_number	93,14	92,43	94,25	79,92	79,92
sport-mean	88,4	79,92	87,99	79,92	79,92
sport-N_IN_Conn_P_DstIP	98,89	98,49	98,89	97,48	97,48
sport-drate	84,36	79,82	84,56	80,52	80,52
sport-srate	94,35	80,12	91,62	81,63	81,03
sport-max_	86,18	79,92	87,79	80,52	80,73
dport-seq	98,89	91,52	99,6	91,32	91,12
dport-stddev	98,18	83,05	98,08	67,2	67,31
dport-N_IN_Conn_P_SrcIP	99,6	92,53	99,7	90,01	89,71
dport-min_	98,49	83,45	98,69	79,92	79,92
dport-state_number	99,09	95,06	99,6	79,92	79,92
dport-mean	99,19	83,25	99,5	79,92	79,92
dport-N_IN_Conn_P_DstIP	99,6	98,49	99,5	97,48	97,48
dport-drate	98,59	83,45	98,79	80,63	80,52
dport-srate	99,09	83,75	98,99	81,63	81,03
dport-max_	98,39	83,05	98,89	83,25	79,92
seq-stddev	96,27	94,65	95,86	94,05	94,05
seq-N_IN_Conn_P_SrcIP	99,39	94,85	99,39	95,26	95,26
seq-min_	94,45	91,73	94,05	91,42	91,32
seq-state_number	96,37	93,44	96,47	91,12	91,12
seq-mean	95,66	93,54	94,85	92,33	92,53
seq-N_IN_Conn_P_DstIP	99,7	99,39	99,4	99,5	99,5
seq-drate	90,51	91,02	91,52	80,93	80,93
seq-srate	97,38	91,02	97,78	81,84	81,84
seq-max_	96,37	93,54	96,27	93,34	93,14
stddev-N_IN_Conn_P_SrcIP	97,98	93,64	98,08	90,31	90,41
stddev-min_	88,5	79,92	88,29	75,58	75,58
stddev-state_number	98,89	91,73	98,69	66,8	66,7
stddev-mean	89,61	79,92	88,19	75,48	75,48
stddev-N_IN_Conn_P_DstIP	99,19	98,28	99,29	97,98	97,98
stddev-drate	84,96	79,82	84,96	80,93	80,93

Tablo 6.2. (Devamı)

stddev-srate	96,27	80,12	95,26	81,33	80,63
stddev-max_	89	79,92	88,5	75,48	75,48
N_IN_Conn_P_SrcIP-min_	98,18	92,73	97,68	90,01	90,11
N_IN_Conn_P_SrcIP-state_number	98,49	94,55	98,49	89,71	89,71
N_IN_Conn_P_SrcIP-mean	97,48	93,84	97,78	91,42	91,32
N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP	98,79	98,49	99,09	97,28	97,68
N_IN_Conn_P_SrcIP-drate	95,86	92,53	96,37	80,63	80,63
N_IN_Conn_P_SrcIP-srate	97,88	92,53	97,88	81,63	81,43
N_IN_Conn_P_SrcIP-max_	97,48	93,64	97,38	92,03	92,23
min_-state_number	92,63	92,33	92,94	79,92	80,02
min_-mean	89,4	79,92	88,7	74,07	75,38
min_-N_IN_Conn_P_DstIP	99,09	98,49	99,19	97,68	97,68
min_-drate	87,79	80,02	88,7	80,52	80,52
min_-srate	96,47	80,12	92,84	81,43	80,83
min_-max_	89,51	79,92	89,1	75,18	75,28
state_number-mean	99,29	98,79	99,09	79,01	86,98
state_number-N_IN_Conn_P_DstIP	98,79	98,49	99,29	97,48	97,48
state_number-drate	92,33	91,93	92,43	80,52	80,52
state_number-srate	96,67	91,93	96,47	81,63	81,03
state_number-max_	99,19	98,89	99,29	75,58	76,49
mean-N_IN_Conn_P_DstIP	99,5	98,69	99,29	97,78	97,88
mean-drate	86,07	79,82	87,29	80,63	80,63
mean-srate	96,27	80,12	95,46	81,43	80,83
mean-max_	88,5	79,92	88,09	75,28	75,38
N_IN_Conn_P_DstIP-drate	98,69	98,49	98,99	93,84	93,84
N_IN_Conn_P_DstIP-srate	99,5	98,49	99,29	91,02	90,82
N_IN_Conn_P_DstIP-max_	99,39	98,69	99,5	98,18	98,18
drate-srate	93,64	80,12	93,54	81,33	81,43
drate-max_	84,16	79,82	86,68	80,63	80,63
srate-max_	96,47	80,12	95,66	81,33	80,83

Her bir makine öğrenimi algoritması 66 öznitelik çiftinin her bir öznitelik çiftiyle değerlendirildi. Bu amaçla, on makine öğrenme algoritması ve bir öznitelik çifti oluşturma modülü içeren yeni bir sistem geliştirildi. Sistem, belirli sayıda öznitelikten otomatik olarak öznitelik çiftleri oluşturur ve kendisini üretilen öznitelik çiftleri ile eğitir, ardından yöntemler, oluşturulan öznitelik çiftleri ile eğitildiğinde doğruluk ve performans açısından on algoritmayı değerlendirir. Çalışmada en iyi 12 öznitelik kullanıldı. Sistem 66 benzersiz öznitelik çifti oluşturdu ve her öznitelik çiftini belirli bir makine öğrenimi algoritmasıyla beslendi. Ardından, her bir anomali algılama algoritmasının doğruluğunu ve performansını değerlendirdi. Sonuç olarak, birbirleriyle çok iyi dağıtıldıkları için tüm algoritmalarda çok iyi performans gösteren 37 öznitelik çifti belirlendi. Dört makine öğrenimi algoritması; Nearest

Neighbor, Decision Trees, Random Forests and Ada-Boost % 95'in üzerinde yüksek bir doğruluk elde etti.

Tüm öznitelik çiftleriyle elde edilen precision sonuçları Tablo 6.3. ve Tablo 6.4.'te gösterilmiştir:

Tablo 6.3. Öznitelik çiftlerinin precision sonuçları(ilk beş algoritma)

Öznitelik Çiftleri / Metodlar	Nearest Neighbors	Linear SVM	RBF SVM	Gaussian Process	Decision Tree
sport-dport	90,63	71	84,42	92,23	99,1
sport-seq	91,34	83,13	91,54	92,57	92,68
sport-stddev	85,43	71	78,25	86,21	88,64
sport-N_IN_Conn_P_SrcIP	91,65	91,52	93,16	92,21	95,73
sport-min_	83,91	71,13	76,28	81,12	86,43
sport-state_number	93,38	71	92,47	91,97	92,61
sport-mean	87,23	71	78,22	85,81	83,35
sport-N_IN_Conn_P_DstIP	98,3	98,73	98,51	98,6	98,4
sport-drate	81,26	71,24	78,81	80,36	79,61
sport-srate	81,19	71,48	78,1	94,17	78,71
sport-max_	85,47	71	78,94	84,6	82,39
dport-seq	96,37	74,73	91,88	98,89	98,68
dport-stddev	95,29	71	79,19	98,29	97,68
dport-N_IN_Conn_P_SrcIP	97,45	91,73	92,69	99,5	97,92
dport-min_	98,89	71	82,18	98,69	98,38
dport-state_number	98,4	71	96,23	99,2	98,5
dport-mean	97,59	71	81,45	98,99	99,29
dport-N_IN_Conn_P_DstIP	98,23	97,53	98,7	99,6	99,6
dport-drate	98,48	71,24	80,08	98,69	98,59
dport-srate	97,59	71,48	79,37	98,49	98,5
dport-max_	98,38	71	80	98,58	98,59
seq-stddev	95,37	93,27	94,53	96,12	94,71
seq-N_IN_Conn_P_SrcIP	98,9	94,64	98,41	99,3	99
seq-min_	91,7	91,79	92,36	94,54	93,18
seq-state_number	96,37	82,3	96,27	95,99	96,19
seq-mean	93,23	92,15	94,88	95,78	94,92
seq-N_IN_Conn_P_DstIP	99,19	98,8	99,5	99,6	99,6
seq-drate	93,16	71,84	90,82	90,79	91,98
seq-srate	91,56	71,48	90,82	96,1	97,99
seq-max_	95,23	92,51	94,92	96,61	95,69
stddev-N_IN_Conn_P_SrcIP	96,12	92,89	93,08	98,21	97,91
stddev-min_	89,13	71	76,66	87,66	87,09
stddev-state_number	98,15	71	95,59	99,09	98
stddev-mean	88,85	71	76,08	88,15	88,7
stddev-N_IN_Conn_P_DstIP	98,23	98,11	98,9	99,4	98,89
stddev-drate	86,17	71,24	71,95	86,07	85,91
stddev-srate	95,22	71,48	71	85,48	96,34
stddev-max_	89,46	71	74,84	80,37	88,24
N_IN_Conn_P_SrcIP-min_	98,11	92,69	96,32	96,41	98,21
N_IN_Conn_P_SrcIP - state_number	98,35	94,49	98,3	97,79	98,3
N_IN_Conn_P_SrcIP - mean	98,21	93,96	96,08	96,32	97,33

Tablo 6.3. (Devamı)

N_IN_Conn_P_SrcIP - N_IN_Conn_P_DstIP	98,79	98,51	99	99	99
N_IN_Conn_P_SrcIP - drate	96,69	92,5	92,32	96,37	96,03
N_IN_Conn_P_SrcIP - srate	98,23	92,5	92,5	96,66	97,8
N_IN_Conn_P_SrcIP - max_	98,52	94,05	94,93	96,63	97,43
min_ - state_number	92,35	91,65	92,09	92,21	92,08
min_ - mean	89,45	71	77,72	85,54	88,86
min_ - N_IN_Conn_P_DstIP	99,1	98,51	98,8	98,9	98,9
min_ - drate	88,06	71,24	72,76	84,47	87,72
min_ - srate	92,82	71,48	76,14	85,5	93,59
min_ - max_	88,54	71	77,09	85,73	88,39
state_number - mean	99,09	71	99,29	99,6	99,2
state_number - N_IN_Conn_P_DstIP	99,3	97,53	98,7	94,56	98,6
state_number - drate	91,8	71,24	91,89	91,89	91,7
state_number - srate	96,39	71,48	92,09	92,83	96,07
state_number - max_	99,29	71	99,29	99,29	98,8
mean - N_IN_Conn_P_DstIP	99,3	97,73	99,2	99,5	99,3
mean - drate	86,39	71,24	74,08	78,29	82,58
mean - srate	95,56	71,48	74,2	77,74	96,22
mean - max_	87,32	71	74,42	85,95	87,38
N_IN_Conn_P_DstIP - drate	99	97,53	98,31	98,51	98,7
N_IN_Conn_P_DstIP - srate	99,3	97,73	98,51	98,51	99,3
N_IN_Conn_P_DstIP - max_	99,5	98,11	98,9	99	99,3
drate - srate	93,53	71,48	72,3	93,27	93,62
drate - max_	85,19	71,24	72,92	74,23	83,47
srate - max_	95,76	71,48	72,36	72,98	96,21

Tablo 6.4. Öznitelik çiftlerinin precision sonuçları(son beş algoritma)

Öznitelik Çiftleri / Metodlar	Random Forest	Neural Net	Adaboost	Naive Bayes	QDA
sport-dport	99,29	80,52	99,19	71,2	71,23
sport-seq	92,89	92,58	94,5	91,13	91,49
sport-stddev	85,46	77,47	87,28	70,21	70,51
sport-N_IN_Conn_P_SrcIP	93,23	92,35	95,95	90,44	90,44
sport-min_	87,29	71,23	87,48	71	71,23
sport-state_number	93,51	91,45	93,88	71	71,43
sport-mean	85,81	78,22	87,26	71	71,53
sport-N_IN_Conn_P_DstIP	98,6	98,51	98,9	97,43	97,53
sport-drate	80,36	78,81	81,28	73,75	73,55
sport-srate	94,17	78,1	91,27	75,47	74,88
sport-max_	84,6	78,94	87,08	77,25	77,29
dport-seq	98,89	91,88	99,6	91,59	91,48
dport-stddev	98,29	79,19	98,08	70,33	70,52
dport-N_IN_Conn_P_SrcIP	99,5	92,69	99,7	90,46	90,06
dport-min_	98,69	82,18	98,68	71	71
dport-state_number	99,2	96,23	99,6	71	71
dport-mean	98,99	81,45	99,5	71	71
dport-N_IN_Conn_P_DstIP	99,6	98,7	99,5	97,13	97,53
dport-drate	98,69	80,08	98,79	73,62	73,55
dport-srate	98,49	79,37	98,99	75,27	74,88
dport-max_	98,58	80	98,89	78,39	71
seq-stddev	96,12	94,53	95,92	94,25	94,25

Tablo 6.4. (Devamı)

seq - N_IN_Conn_P_SrcIP	99,3	98,41	99,4	95,22	95,22
seq - min_	94,54	92,36	93,91	91,7	91,61
seq - state_number	95,99	96,27	96,42	91,49	91,51
seq – mean	95,78	94,88	94,92	92,55	92,74
seq - N_IN_Conn_P_DstIP	99,6	99,5	99,54	99,5	99,5
seq – drate	90,79	91,06	91,44	73,8	73,8
seq – srate	96,1	91,04	97,79	75,41	75,41
seq - max_	96,61	93,67	96,29	93,49	93,3
stddev - N_IN_Conn_P_SrcIP	98,21	93,64	98,09	90,57	90,66
stddev - min_	87,66	71	87,73	77,93	77,93
stddev - state_number	99,09	90,97	98,68	70,03	69,94
stddev – mean	88,15	71	87,51	77,83	77,83
stddev - N_IN_Conn_P_DstIP	99,4	98,5	99,3	98	98
stddev - drate	86,07	71,14	86,08	73,8	73,8
stddev - srate	96,34	71,48	95,35	75,07	74,61
stddev - max_	88,24	71	87,94	77,83	77,83
N_IN_Conn_P_SrcIP - min_	97,69	92,69	97,49	90,27	90,36
N_IN_Conn_P_SrcIP - state_number	98,5	94,49	98,25	90,09	90,09
N_IN_Conn_P_SrcIP - mean	97,8	93,96	97,28	91,63	91,54
N_IN_Conn_P_SrcIP - N_IN_Conn_P_DstIP	99,1	98,51	98,51	97,34	97,73
N_IN_Conn_P_SrcIP - drate	96,47	92,5	96,17	73,62	73,62
N_IN_Conn_P_SrcIP - srate	97,91	92,5	97,81	75,27	75,14
N_IN_Conn_P_SrcIP - max_	97,41	94,05	97,51	92,13	92,36
min_ - state_number	92,33	91,65	92,23	71	71,61
min_ - mean	88,03	71	88,34	76,07	77,73
min_ - N_IN_Conn_P_DstIP	99,2	98,51	99,12	97,73	97,73
min_ - drate	88,18	71,24	88,19	73,55	73,55
min_ - srate	92,82	71,48	92,82	74,86	74,74
min_ - max_	88,54	71	88,54	77,55	77,64
state_number – mean	99,09	71	99,09	70,55	84,72
state_number - N_IN_Conn_P_DstIP	99,3	97,53	99,3	97,53	97,53
state_number - drate	91,8	71,24	91,8	73,55	73,55
state_number - srate	96,39	71,48	96,59	75,27	74,88
state_number - max_	99,29	71	99,39	74,98	76,19
mean - N_IN_Conn_P_DstIP	99,3	97,73	99,23	97,82	97,92
mean - drate	86,39	71,24	86,49	73,62	73,62
mean - srate	95,56	71,48	95,76	74,86	74,74
mean - max_	87,32	71	87,52	77,64	77,74
N_IN_Conn_P_DstIP - drate	99	97,53	99,21	93,58	93,58
N_IN_Conn_P_DstIP - srate	99,3	97,73	99,13	90,03	89,84
N_IN_Conn_P_DstIP - max_	99,5	98,11	99,45	98,21	98,21
drate - srate	93,53	71,48	93,53	75,21	75,54
drate - max_	85,19	71,14	85,29	73,62	73,62
srate - max_	95,76	71,48	95,56	74,89	74,74

Tüm öznitelik çiftleriyle elde edilen recall sonuçları Tablo 6.5. ve Tablo 6.6.'te gösterilmiştir:

Tablo 6.5. Öznitelik çiftlerinin recall sonuçları(ilk beş algoritma)

Öznitelik Çiftleri / Metodlar	Nearest Neighbors	Linear SVM	RBF SVM	Gaussian Process	Decision Tree
sport-dport	88,93	71,00	83,21	91,41	98,90
sport-seq	93,34	81,35	93,36	91,36	92,68
sport-stddev	83,51	71,00	76,47	85,22	88,64
sport-N_IN_Conn_P_SrcIP	93,75	91,94	92,05	92,21	95,73
sport-min_	85,85	70,87	78,51	81,12	86,43
sport-state_number	92,98	71,00	91,48	91,97	92,61
sport-mean	83,70	71,00	78,22	81,03	82,95
sport-N_IN_Conn_P_DstIP	98,70	96,75	98,51	98,20	99,20
sport-drate	81,06	71,24	78,81	78,87	80,96
sport-srate	79,02	71,48	78,10	67,61	118,00
sport-max_	85,67	71,00	78,94	80,29	83,34
dport-seq	97,92	74,73	91,88	98,47	98,50
dport-stddev	99,31	71,00	79,19	97,08	98,69
dport-N_IN_Conn_P_SrcIP	99,57	91,73	92,69	96,39	101,55
dport-min_	96,89	71,00	82,18	98,07	99,00
dport-state_number	98,60	71,00	96,23	97,81	99,91
dport-mean	99,82	71,00	81,45	99,59	99,91
dport-N_IN_Conn_P_DstIP	99,78	97,53	98,70	99,60	99,40
dport-drate	98,68	71,24	80,08	98,49	98,77
dport-srate	100,02	71,48	79,37	98,51	99,30
dport-max_	98,78	71,00	80,00	98,60	99,39
seq-stddev	94,97	93,27	94,53	93,34	97,66
seq-N_IN_Conn_P_SrcIP	99,30	94,64	98,41	98,70	99,20
seq-min_	93,52	91,79	92,36	91,86	89,40
seq-state_number	98,19	82,30	96,27	96,39	95,31
seq-mean	94,64	92,15	94,88	94,08	95,74
seq-N_IN_Conn_P_DstIP	99,59	98,80	99,50	99,60	99,46
seq-drate	91,97	71,84	90,82	91,03	91,98
seq-srate	91,76	71,48	90,82	86,58	97,99
seq-max_	93,84	92,51	94,92	93,35	95,69
stddev-N_IN_Conn_P_SrcIP	98,55	92,89	93,08	96,07	97,91
stddev-min_	88,65	71,00	76,66	84,69	87,09
stddev-state_number	98,85	71,00	95,59	98,29	98,00
stddev-mean	89,23	71,00	76,08	73,97	88,70
stddev-N_IN_Conn_P_DstIP	100,19	98,11	98,90	98,41	98,89
stddev-drate	85,10	71,24	71,95	85,87	85,91
stddev-srate	94,23	71,48	71,00	85,48	96,34
stddev-max_	89,06	71,00	74,84	80,37	88,24
N_IN_Conn_P_SrcIP-min_	97,71	92,61	96,32	96,41	98,21
N_IN_Conn_P_SrcIP-state_number	98,65	89,64	98,30	97,79	98,30
N_IN_Conn_P_SrcIP-mean	97,22	90,64	96,08	96,32	97,33
N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP	99,01	96,20	99,00	99,00	99,00
N_IN_Conn_P_SrcIP-drate	95,87	91,15	92,32	96,37	96,03
N_IN_Conn_P_SrcIP-srate	98,17	91,15	92,50	96,66	97,80
N_IN_Conn_P_SrcIP-max_	96,74	90,19	94,93	96,63	97,43
min_-state_number	91,61	57,94	92,09	92,21	92,08
min_-mean	89,45	71,00	77,72	85,54	88,86
min_-N_IN_Conn_P_DstIP	99,10	96,57	98,80	98,90	98,90
min_-drate	88,06	71,24	72,76	84,47	87,72
min_-srate	88,85	71,48	76,14	85,50	93,59
min_-max_	89,81	71,00	77,09	85,73	88,39
state_number-mean	99,49	71,00	99,29	99,60	99,20

Tablo 6.5. (Devamı)

state_number-N_IN_Conn_P_DstIP	98,90	97,53	98,70	94,56	98,60
state_number-drate	43,63	71,24	91,89	91,89	91,70
state_number-srate	102,39	71,48	92,09	92,83	96,07
state_number-max_	98,49	71,00	99,29	99,29	98,80
mean-N_IN_Conn_P_DstIP	98,07	97,73	99,20	99,50	99,30
mean-drate	85,55	71,24	74,08	78,29	82,58
mean-srate	84,66	71,48	74,20	77,74	96,22
mean-max_	91,16	71,00	74,42	85,95	87,38
N_IN_Conn_P_DstIP-drate	99,00	97,53	98,31	98,51	98,70
N_IN_Conn_P_DstIP-srate	98,90	97,73	98,51	98,51	99,30
N_IN_Conn_P_DstIP-max_	98,70	98,11	98,90	99,00	99,30
drate-srate	98,50	71,48	72,30	93,27	93,62
drate-max_	83,66	71,24	72,92	74,23	83,47
srate-max_	83,20	71,48	72,36	72,98	96,21

Tablo 6.6. Öznitelik çiftlerinin recall oranları (son beş algoritma)

Öznitelik Çiftleri / Metodlar	Random Forest	Neural Net	Ada Boost	Naive Bayes	QDA
sport-dport	98,89	77,77	99,19	70,80	70,77
sport-seq	93,23	90,41	94,50	91,47	91,89
sport-stddev	88,93	76,28	87,28	70,41	70,11
sport-N_IN_Conn_P_SrcIP	94,84	92,65	95,95	90,64	90,64
sport-min_	88,90	70,77	87,48	71,00	70,77
sport-state_number	91,92	91,85	93,88	71,00	70,58
sport-mean	85,81	74,76	87,26	71,00	70,48
sport-N_IN_Conn_P_DstIP	98,60	98,51	98,90	97,63	97,53
sport-drate	80,36	64,83	81,28	73,35	73,55
sport-srate	94,17	65,89	91,27	75,07	74,88
sport-max_	84,60	73,57	87,08	77,55	77,29
dport-seq	98,89	90,59	99,60	91,77	91,48
dport-stddev	98,29	77,31	98,08	70,51	70,52
dport-N_IN_Conn_P_SrcIP	99,50	92,31	99,70	90,26	90,06
dport-min_	98,69	75,59	98,68	71,00	71,00
dport-state_number	99,20	93,51	99,60	71,00	71,00
dport-mean	98,99	75,94	99,50	71,00	71,00
dport-N_IN_Conn_P_DstIP	99,60	98,50	99,50	97,93	97,53
dport-drate	98,69	77,39	98,79	73,62	73,55
dport-srate	98,49	78,87	98,99	75,27	74,88
dport-max_	98,58	76,56	98,89	78,39	71,00
seq-stddev	96,12	95,31	95,92	94,25	94,25
seq-N_IN_Conn_P_SrcIP	99,30	91,50	99,40	95,22	95,22
seq-min_	94,54	91,37	93,91	91,70	91,61
seq-state_number	95,99	91,64	96,42	91,49	91,51
seq-mean	95,78	92,51	94,92	92,55	92,74
seq-N_IN_Conn_P_DstIP	99,60	99,30	98,92	99,50	99,50
seq-drate	90,79	91,06	91,44	73,80	73,80
seq-srate	96,10	91,04	97,79	75,41	75,41
seq-max_	96,61	93,67	96,29	93,49	93,30
stddev-N_IN_Conn_P_SrcIP	98,21	93,64	98,09	90,57	90,66
stddev-min_	87,66	71,00	87,73	77,93	77,93
stddev-state_number	99,09	90,97	98,68	70,03	69,94
stddev-mean	88,15	71,00	87,51	77,83	77,83
stddev-N_IN_Conn_P_DstIP	99,40	98,50	99,30	98,00	98,00
stddev-drate	86,07	71,14	86,08	73,80	73,80

Tablo 6.6. (Devamı)

stddev-srate	88,24	71,00	87,94	77,83	77,83
stddev-max_	98,13	92,69	97,89	90,27	90,36
N_IN_Conn_P_SrcIP-min_	98,50	94,49	98,75	90,09	90,09
N_IN_Conn_P_SrcIP-state_number	98,20	93,96	98,33	91,63	91,54
N_IN_Conn_P_SrcIP-mean	98,50	98,51	99,70	97,34	97,73
N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP	96,47	92,50	96,77	73,62	73,62
N_IN_Conn_P_SrcIP-drate	98,29	92,50	98,01	75,27	75,14
N_IN_Conn_P_SrcIP-srate	98,21	94,05	97,31	92,13	92,36
N_IN_Conn_P_SrcIP-max_	91,83	91,65	92,43	71,00	71,61
min_-state_number	89,06	71,00	87,72	76,07	77,73
min_-mean	99,20	98,51	99,28	97,73	97,73
min_-N_IN_Conn_P_DstIP	87,03	71,24	88,17	73,55	73,55
min_-drate	99,90	71,48	92,82	74,86	74,74
min_-srate	86,41	71,00	88,54	77,55	77,64
min_-max_	99,71	162,33	99,09	70,55	84,72
state_number-mean	98,11	99,51	99,30	97,53	97,53
state_number-N_IN_Conn_P_DstIP	91,60	126,54	91,80	73,55	73,55
state_number-drate	96,83	125,79	96,19	75,27	74,88
state_number-srate	98,69	162,87	99,19	74,98	76,19
state_number-max_	99,70	99,89	99,37	97,82	97,92
mean-N_IN_Conn_P_DstIP	82,21	71,04	86,29	73,62	73,62
mean-drate	97,09	71,48	95,36	74,86	74,74
mean-srate	84,45	71,00	87,12	77,64	77,74
mean-max_	98,40	99,51	98,79	93,58	93,58
N_IN_Conn_P_DstIP-drate	99,50	99,30	99,47	90,03	89,84
N_IN_Conn_P_DstIP-srate	99,30	99,30	99,55	98,21	98,21
N_IN_Conn_P_DstIP-max_	93,71	71,48	93,53	75,21	75,54
drate-srate	81,95	71,14	85,09	73,62	73,62
drate-max_	96,91	71,48	95,96	74,69	74,74
srate-max_	88,24	71,00	87,94	77,83	77,83

Tüm öznitelik çiftleriyle elde edilen fscore sonuçları Tablo 6.7. ve Tablo 6.8.'te gösterilmiştir:

Tablo 6.7. Öznitelik çiftlerinin fscore sonuçları(ilk beş algoritma)

Öznitelik Çiftleri / Metodlar	Nearest Neighbors	Linear SVM	RBF SVM	Gaussian Process	Decision Tree
sport-dport	89,77	71	83,81	91,82	99
sport-seq	92,33	82,23	92,44	91,96	92,68
sport-stddev	84,46	71	77,35	85,71	88,64
sport-N_IN_Conn_P_SrcIP	92,69	91,73	92,6	92,21	95,73
sport-min_	84,87	71	77,38	81,12	86,43
sport-state_number	93,18	71	91,97	91,97	92,61
sport-mean	85,43	71	78,22	83,35	83,15
sport-N_IN_Conn_P_DstIP	98,5	97,73	98,51	98,4	98,8
sport-drate	81,16	71,24	78,81	79,61	80,28
sport-srate	80,09	71,48	78,1	78,71	94,43
sport-max_	85,57	71	78,94	82,39	82,86
dport-seq	97,14	74,73	91,88	98,68	98,59
dport-stddev	97,26	71	79,19	97,68	98,18
dport-N_IN_Conn_P_SrcIP	98,5	91,73	92,69	97,92	99,7
dport-min_	97,88	71	82,18	98,38	98,69

Tablo 6.7. (Devamı)

dport - state_number	98,5	71	96,23	98,5	99,2
dport - mean	98,69	71	81,45	99,29	99,6
dport - N_IN_Conn_P_DstIP	99	97,53	98,7	99,6	99,5
dport - drate	98,58	71,24	80,08	98,59	98,68
dport - srate	98,79	71,48	79,37	98,5	98,9
dport - max_	98,58	71	80	98,59	98,99
seq - stddev	95,17	93,27	94,53	94,71	96,16
seq - N_IN_Conn_P_SrcIP	99,1	94,64	98,41	99	99,1
seq - min_	92,6	91,79	92,36	93,18	91,25
seq - state_number	97,27	82,3	96,27	96,19	95,75
seq – mean	93,93	92,15	94,88	94,92	95,33
seq - N_IN_Conn_P_DstIP	99,39	98,8	99,5	99,6	99,53
seq – drate	92,56	71,84	90,82	90,91	91,98
seq – srate	91,66	71,48	90,82	91,09	97,99
seq - max_	94,53	92,51	94,92	94,95	95,69
stddev - N_IN_Conn_P_SrcIP	97,32	92,89	93,08	97,13	97,91
stddev - min_	88,89	71	76,66	86,15	87,09
stddev - state_number	98,5	71	95,59	98,69	98
stddev – mean	89,04	71	76,08	80,44	88,7
stddev - N_IN_Conn_P_DstIP	99,2	98,11	98,9	98,9	98,89
stddev - drate	85,63	71,24	71,95	85,97	85,91
stddev - srate	94,72	71,48	71	85,48	96,34
stddev - max_	89,26	71	74,84	80,37	88,24
N_IN_Conn_P_SrcIP - min_	97,91	92,65	96,32	96,41	98,21
N_IN_Conn_P_SrcIP - state_number	98,5	92	98,3	97,79	98,3
N_IN_Conn_P_SrcIP - mean	97,71	92,27	96,08	96,32	97,33
N_IN_Conn_P_SrcIP - N_IN_Conn_P_DstIP	98,9	97,34	99	99	99
N_IN_Conn_P_SrcIP - drate	96,28	91,82	92,32	96,37	96,03
N_IN_Conn_P_SrcIP - srate	98,2	91,82	92,5	96,66	97,8
N_IN_Conn_P_SrcIP - max_	97,62	92,08	94,93	96,63	97,43
min_ - state_number	91,98	71	92,09	92,21	92,08
min_ - mean	89,45	71	77,72	85,54	88,86
min_ - N_IN_Conn_P_DstIP	99,1	97,53	98,8	98,9	98,9
min_ - drate	88,06	71,24	72,76	84,47	87,72
min_ - srate	90,79	71,48	76,14	85,5	93,59
min_ - max_	89,17	71	77,09	85,73	88,39
state_number – mean	99,29	71	99,29	99,6	99,2
state_number - N_IN_Conn_P_DstIP	99,1	97,53	98,7	94,56	98,6
state_number - drate	59,15	71,24	91,89	91,89	91,7
state_number - srate	99,3	71,48	92,09	92,83	96,07
state_number - max_	98,89	71	99,29	99,29	98,8
mean - N_IN_Conn_P_DstIP	98,68	97,73	99,2	99,5	99,3
mean - drate	85,97	71,24	74,08	78,29	82,58
mean - srate	89,78	71,48	74,2	77,74	96,22
mean - max_	89,2	71	74,42	85,95	87,38
N_IN_Conn_P_DstIP - drate	99	97,53	98,31	98,51	98,7
N_IN_Conn_P_DstIP - srate	99,1	97,73	98,51	98,51	99,3
N_IN_Conn_P_DstIP - max_	99,1	98,11	98,9	99	99,3
drate - srate	95,95	71,48	72,3	93,27	93,62
drate - max_	84,42	71,24	72,92	74,23	83,47
srate - max_	89,04	71,48	72,36	72,98	96,21

Tablo 6.8. Öznitelik çiftlerinin fscore sonuçları(son beş algoritma)

Öznitelik Çiftleri / Metodlar	Random Forest	Neural Net	Adaboost	Naive Bayes	QDA
sport-dport	99,09	79,12	99,19	71	71
sport-seq	93,06	91,48	94,5	91,3	91,69
sport-stddev	87,16	76,87	87,28	70,31	70,31
sport-N_IN_Conn_P_SrcIP	94,03	92,5	95,95	90,54	90,54
sport-min_	88,09	71	87,48	71	71
sport-state_number	92,71	91,65	93,88	71	71
sport-mean	85,81	76,45	87,26	71	71
sport-N_IN_Conn_P_DstIP	98,6	98,51	98,9	97,53	97,53
sport-drate	80,36	71,14	81,28	73,55	73,55
sport-srate	94,17	71,48	91,27	75,27	74,88
sport-max_	84,6	76,16	87,08	77,4	77,29
dport-seq	98,89	91,23	99,6	91,68	91,48
dport-stddev	98,29	78,24	98,08	70,42	70,52
dport-N_IN_Conn_P_SrcIP	99,5	92,5	99,7	90,36	90,06
dport-min_	98,69	78,75	98,68	71	71
dport-state_number	99,2	94,85	99,6	71	71
dport-mean	98,99	78,6	99,5	71	71
dport-N_IN_Conn_P_DstIP	99,6	98,6	99,5	97,53	97,53
dport-drate	98,69	78,71	98,79	73,62	73,55
dport-srate	98,49	79,12	98,99	75,27	74,88
dport-max_	98,58	78,24	98,89	78,39	71
seq-stddev	96,12	94,92	95,92	94,25	94,25
seq-N_IN_Conn_P_SrcIP	99,3	94,83	99,4	95,22	95,22
seq-min_	94,54	91,86	93,91	91,7	91,61
seq-state_number	95,99	93,9	96,42	91,49	91,51
seq-mean	95,78	93,68	94,92	92,55	92,74
seq-N_IN_Conn_P_DstIP	99,6	99,4	99,23	99,5	99,5
seq-drate	90,79	91,06	91,44	73,8	73,8
seq-srate	96,1	91,04	97,79	75,41	75,41
seq-max_	96,61	93,67	96,29	93,49	93,3
stddev-N_IN_Conn_P_SrcIP	98,21	93,64	98,09	90,57	90,66
stddev-min_	87,66	71	87,73	77,93	77,93
stddev-state_number	99,09	90,97	98,68	70,03	69,94
stddev-mean	88,15	71	87,51	77,83	77,83
stddev-N_IN_Conn_P_DstIP	99,4	98,5	99,3	98	98
stddev-drate	86,07	71,14	86,08	73,8	73,8
stddev-srate	96,34	71,48	95,35	75,07	74,61
stddev-max_	88,24	71	87,94	77,83	77,83
N_IN_Conn_P_SrcIP-min_	97,91	92,69	97,69	90,27	90,36
N_IN_Conn_P_SrcIP-state_number	98,5	94,49	98,5	90,09	90,09
N_IN_Conn_P_SrcIP-mean	98	93,96	97,8	91,63	91,54
N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP	98,8	98,51	99,1	97,34	97,73
N_IN_Conn_P_SrcIP-drate	96,47	92,5	96,47	73,62	73,62
N_IN_Conn_P_SrcIP-srate	98,1	92,5	97,91	75,27	75,14
N_IN_Conn_P_SrcIP-max_	97,81	94,05	97,41	92,13	92,36
min_-state_number	92,08	91,65	92,33	71	71,61
min_-mean	88,54	71	88,03	76,07	77,73
min_-N_IN_Conn_P_DstIP	99,2	98,51	99,2	97,73	97,73
min_-drate	87,6	71,24	88,18	73,55	73,55
min_-srate	96,23	71,48	92,82	74,86	74,74
min_-max_	87,46	71	88,54	77,55	77,64
state_number-mean	99,4	98,79	99,09	70,55	84,72

Tablo 6.8. (Devamı)

state_number-N_IN_Conn_P_DstIP	98,7	98,51	99,3	97,53	97,53
state_number-drate	91,7	91,16	91,8	73,55	73,55
state_number-srate	96,61	91,16	96,39	75,27	74,88
state_number-max_	98,99	98,89	99,29	74,98	76,19
mean-N_IN_Conn_P_DstIP	99,5	98,8	99,3	97,82	97,92
mean-drate	84,25	71,14	86,39	73,62	73,62
mean-srate	96,32	71,48	95,56	74,86	74,74
mean-max_	85,86	71	87,32	77,64	77,74
N_IN_Conn_P_DstIP-drate	98,7	98,51	99	93,58	93,58
N_IN_Conn_P_DstIP-srate	99,4	98,51	99,3	90,03	89,84
N_IN_Conn_P_DstIP-max_	99,4	98,7	99,5	98,21	98,21
drate-srate	93,62	71,48	93,53	75,21	75,54
drate-max_	83,54	71,14	85,19	73,62	73,62
srate-max_	96,33	71,48	95,76	74,79	74,74

6.2. Birim Maliyet

Tablo 6.9. ve Tablo 6.10.'da ise öznitelik çiftleri ile eğitildiğinde birim maliyet hesaplanmıştır.

Tablo 6.9. Öznitelik çiftlerinin birim maliyeti(ilk beş algoritma)

Öznitelik Çiftleri / Metodlar	Nearest Neighbors	Linear SVM	RBF SVM	Gaussian Process	Decision Tree
sport-dport	0,3	0,36	0,31	0,36	1,12
sport-seq	0,29	0,38	0,35	0,35	0,91
sport-stddev	0,29	0,35	0,36	0,36	0,93
sport-N_IN_Conn_P_SrcIP	0,3	0,39	0,35	0,36	0,97
sport-min_	0,33	0,35	0,31	0,34	0,94
sport-state_number	0,29	0,35	0,35	0,35	0,93
sport-mean	0,29	0,35	0,36	0,36	0,94
sport-N_IN_Conn_P_DstIP	0,29	0,36	0,35	0,36	0,93
sport-drate	0,3	0,36	0,36	0,35	0,98
sport-srate	0,3	0,35	0,36	0,34	0,91
sport-max_	0,29	0,35	0,36	0,34	0,92
dport-seq	0,29	0,35	0,39	0,37	0,91
dport-stddev	0,29	0,35	0,36	0,35	0,91
dport-N_IN_Conn_P_SrcIP	0,28	0,35	0,35	0,35	0,92
dport-min_	0,29	0,35	0,35	0,35	0,93
dport-state_number	0,31	0,35	0,35	0,35	0,97
dport-mean	0,29	0,36	0,36	0,37	0,93
dport-N_IN_Conn_P_DstIP	0,29	0,35	0,35	0,35	1
dport-drate	0,3	0,36	0,36	0,36	0,95
dport-srate	0,29	0,35	0,36	0,37	0,98
dport-max_	0,31	0,36	0,38	0,38	0,97
seq-stddev	0,29	0,35	0,35	0,35	0,95
seq-N_IN_Conn_P_SrcIP	0,29	0,35	0,35	0,36	0,96
seq-min_	0,3	0,35	0,07	0,35	0,91
seq-state_number	0,3	0,35	0,35	0,36	0,95
seq-mean	0,3	0,36	0,35	0,35	0,95
seq-N_IN_Conn_P_DstIP	0,3	0,35	0,35	0,36	0,95

Tablo 6.9. (Devamı)

seq-drate	0,3	0,36	0,36	0,35	0,94
seq-srate	0,29	0,35	0,35	0,35	0,92
seq-max_	0,31	0,37	0,38	0,35	0,95
stddev-N_IN_Conn_P_SrcIP	0,3	0,35	0,35	0,35	0,93
stddev-min_	0,29	0,35	0,36	0,35	0,91
stddev-state_number	0,29	0,35	0,35	0,35	0,91
stddev-mean	0,3	0,36	0,36	0,34	0,92
stddev-N_IN_Conn_P_DstIP	0,29	0,35	0,35	0,36	0,9
stddev-drate	0,29	0,35	0,36	0,35	0,94
stddev-srate	0,29	0,35	0,36	0,34	0,91
stddev-max_	0,29	0,35	0,36	0,36	0,94
N_IN_Conn_P_SrcIP-min_	0,33	0,36	0,35	0,35	0,92
N_IN_Conn_P_SrcIP-state_number	0,29	0,35	0,35	0,36	0,92
N_IN_Conn_P_SrcIP-mean	0,29	0,35	0,35	0,35	0,91
N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP	0,29	0,36	0,35	0,35	0,97
N_IN_Conn_P_SrcIP-drate	0,3	0,35	0,35	0,36	0,9
N_IN_Conn_P_SrcIP-srate	0,29	0,36	0,35	0,37	0,92
N_IN_Conn_P_SrcIP-max_	0,29	0,37	0,35	0,35	0,93
min_-state_number	0,3	0,36	0,35	0,37	0,96
min_-mean	0,3	0,36	0,36	0,37	0,95
min_-N_IN_Conn_P_DstIP	0,29	0,35	0,35	0,35	0,91
min_-drate	0,28	0,35	0,36	0,35	0,9
min_-srate	0,29	0,35	0,36	0,37	0,97
min_-max_	0,3	0,35	0,36	0,35	0,93
state_number-mean	0,29	0,35	0,35	0,34	0,94
state_number-N_IN_Conn_P_DstIP	0,3	0,35	0,35	0,35	0,96
state_number-drate	0,3	0,35	0,35	0,34	0,96
state_number-srate	0,29	0,35	0,35	0,36	0,94
state_number-max_	0,3	0,36	0,35	0,36	0,94
mean-N_IN_Conn_P_DstIP	0,3	0,35	0,35	0,36	0,76
mean-drate	0,29	0,35	0,31	0,35	0,92
mean-srate	0,29	0,35	0,36	0,36	0,94
mean-max_	0,29	0,35	0,36	0,37	0,98
N_IN_Conn_P_DstIP-drate	0,3	0,35	0,35	0,36	1,5
N_IN_Conn_P_DstIP-srate	0,29	0,35	0,35	0,34	0,91
N_IN_Conn_P_DstIP-max_	0,29	0,35	0,35	0,36	0,91
drate-srate	0,29	0,35	0,36	0,37	1
drate-max_	0,3	0,35	0,37	0,35	0,99
srate-max_	0,3	0,35	0,31	0,35	0,99

Tablo 6.10. Öznitelik çiftlerinin birim maliyeti(son beş algoritma)

Öznitelik Çiftleri / Metodlar	Random Forest	Neural Net	Ada Boost	Naive Bayes	QDA
sport-dport	1,12	0,38	1,94	0,38	0,38
sport-seq	0,91	0,39	1,9	0,31	0,38
sport-stddev	0,93	0,39	2,02	0,38	0,38
sport-N_IN_Conn_P_SrcIP	0,97	0,39	1,99	0,38	0,38
sport-min_	0,94	0,39	1,92	0,39	0,38
sport-state_number	0,93	0,39	1,96	0,38	0,38
sport-mean	0,94	0,38	1,94	0,38	0,38

Tablo 6.10. (Devamı)

sport-N_IN_Conn_P_DstIP	0,93	0,39	1,95	0,38	0,38
sport-drate	0,98	0,39	1,9	0,38	0,38
sport-srate	0,91	0,38	1,91	0,38	0,38
sport-max_	0,92	0,39	1,91	0,38	0,38
dport-seq	0,91	0,39	1,91	0,38	0,38
dport-stddev	0,91	0,39	1,91	0,07	0,07
dport-N_IN_Conn_P_SrcIP	0,92	0,39	1,91	0,38	0,38
dport-min_	0,93	0,39	1,94	0,38	0,38
dport-state_number	0,97	0,39	1,94	0,38	0,38
dport-mean	0,93	0,39	1,93	0,38	0,38
dport-N_IN_Conn_P_DstIP	1	0,39	1,95	0,38	0,38
dport-drate	0,95	0,39	1,94	0,38	0,38
dport-srate	0,98	0,39	1,99	0,38	0,38
dport-max_	0,97	0,39	1,94	0,38	0,38
seq-stddev	0,95	0,39	1,95	0,38	0,38
seq-N_IN_Conn_P_SrcIP	0,96	0,39	1,96	0,38	0,38
seq-min_	0,91	0,39	1,91	0,38	0,38
seq-state_number	0,95	0,39	1,98	0,38	0,38
seq-mean	0,95	0,39	1,98	0,38	0,38
seq-N_IN_Conn_P_DstIP	0,95	0,39	1,96	0,38	0,38
seq-drate	0,94	0,39	1,94	0,38	0,38
seq-srate	0,92	0,38	1,89	0,38	0,38
seq-max_	0,95	0,39	1,95	0,38	0,38
stddev-N_IN_Conn_P_SrcIP	0,93	0,29	1,92	0,38	0,38
stddev-min_	0,91	0,29	1,88	0,38	0,38
stddev-state_number	0,91	0,29	1,95	0,38	0,38
stddev-mean	0,92	0,38	1,91	0,38	0,38
stddev-N_IN_Conn_P_DstIP	0,9	0,39	1,9	0,38	0,38
stddev-drate	0,94	0,29	1,91	0,38	0,38
stddev-srate	0,91	0,38	1,92	0,38	0,38
stddev-max_	0,94	0,39	2,05	0,38	0,38
N_IN_Conn_P_SrcIP-min_	0,92	0,38	1,93	0,38	0,38
N_IN_Conn_P_SrcIP-state_number	0,92	0,39	1,89	0,38	0,38
N_IN_Conn_P_SrcIP-mean	0,91	0,39	1,9	0,38	0,38
N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP	0,97	0,39	1,91	0,38	0,38
N_IN_Conn_P_SrcIP-drate	0,9	0,38	1,91	0,38	0,38
N_IN_Conn_P_SrcIP-srate	0,92	0,39	1,91	0,38	0,38
N_IN_Conn_P_SrcIP-max_	0,93	0,29	1,97	0,38	0,38
min_-state_number	0,96	0,29	1,95	0,38	0,38
min_-mean	0,95	0,39	1,92	0,38	0,38
min_-N_IN_Conn_P_DstIP	0,91	0,39	1,98	0,38	0,07
min_-drate	0,9	0,38	1,89	0,38	0,38
min_-srate	0,97	0,39	1,91	0,38	0,38
min_-max_	0,93	0,38	1,91	0,38	0,38
state_number-mean	0,94	0,39	1,9	0,38	0,38
state_number-N_IN_Conn_P_DstIP	0,96	0,39	1,93	0,38	0,38
state_number-drate	0,96	0,39	1,91	0,38	0,38
state_number-srate	0,94	0,39	1,96	0,38	0,38
state_number-max_	0,94	0,39	1,96	0,38	0,38
mean-N_IN_Conn_P_DstIP	0,76	0,39	1,91	0,38	0,38
mean-drate	0,92	0,39	1,9	0,38	0,38
mean-srate	0,94	0,39	1,93	0,38	0,38

Tablo 6.10. (Devamı)

mean-max_	0,98	0,29	1,98	0,38	0,38
N_IN_Conn_P_DstIP-drate	1,5	0,39	1,97	0,38	0,38
N_IN_Conn_P_DstIP-srate	0,91	0,39	1,92	0,38	0,38
N_IN_Conn_P_DstIP-max_	0,91	0,39	2,02	0,38	0,38
drate-srate	1	0,39	1,98	0,38	0,38
drate-max_	0,99	0,39	1,96	0,31	0,31
srate-max_	0,99	0,39	2	0,38	0,38

Birim maliyet dikkate alındığında Random Forest ve AdaBoost Algoritmaları oldukça maliyetli olduğu görülmektedir. Diğer algoritmalar ise bu algoritmalara göre daha hızlı çalışmaktadır. Hem doğruluk hem birim maliyet dikkate alındığında Nearest Neighbour, RBF SVM, Gaussion Process ve Decision Tree Algoritmaları oldukça iyi sonuçlar vermektedir.

Tüm özniteliklerle elde edilen sonuçlar ve birim maliyetleri aşağıdaki Tablo 6.11. ve Tablo 6.12.'de belirtilmiştir:

Tablo 6.11. Tüm özniteliklerle elde edilen sonuçlar (ilk beş algoritma)

Metodlar	Nearest Neighbors	Linear SVM	RBF SVM	Gaussian Process	Decision Tree
Doğruluk Oranları	99,8	99,29	99,19	99,19	99,6
Precision	99,29	99,32	99,25	99,05	99,54
Recall	99,91	99,28	99,13	99,33	99,66
Fscore	99,6	99,3	99,19	99,19	99,6
Birim Maliyet	0,46	0,49	0,47	0,52	0,47

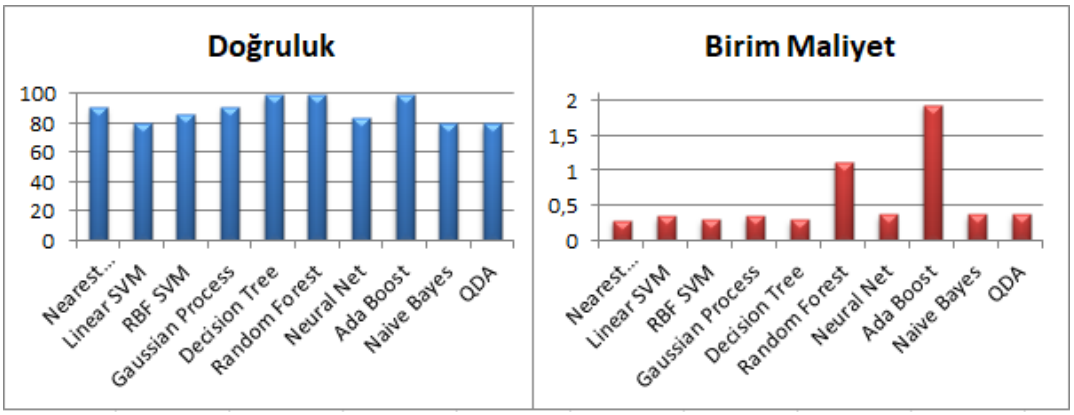
Tablo 6.12. Tüm özniteliklerle elde edilen sonuçlar(son beş algoritma)

Metodlar	Random Forest	Neural Net	Ada Boost	Naive Bayes	QDA
Doğruluk Oranları	99,8	99,5	99,6	92,43	86,88
Precision	99,36	99,85	99,56	92,23	85,24
Recall	99,64	99,35	99,44	91,31	83,44
Fscore	99,5	99,6	99,5	91,77	84,33
Birim Maliyet	1,35	0,53	2,32	0,52	0,53

Tüm özniteliklerle elde edilen sonuçlara bakıldığında birim maliyetin oldukça fazla olduğu görülmektedir. Bu çalışmada öznitelik çiftleri ile elde edilen sonuçlara bakıldığında hem doğruluk oranı olarak oldukça yüksek sonuçlar elde edildi hem de birim maliyet olarak kazanç sağlanıldı. Ortalama olarak %20 ile %30 arasında sistemin maliyeti azaltıldı.

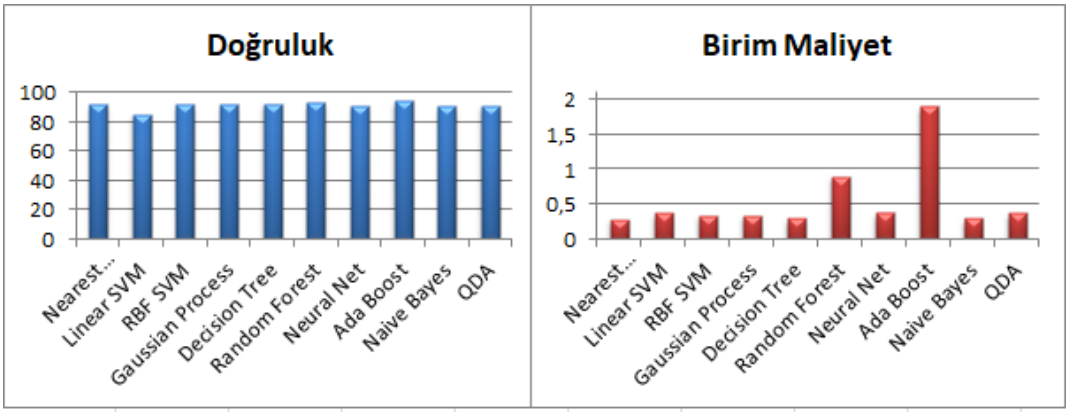
6.3. Topluluk Öğrenmesi

Topluluk öğrenimi, birden çok modelden gelen tahminleri birleştirerek daha iyi tahmine dayalı performans arayan makine öğrenimine yönelik genel bir meta yaklaşımdır. Bizim çalışmamızda hem doğruluk hem de birim maliyete bakılarak en optimal ve en etkili sonuç elde edilmiştir. Aşağıdaki şekillerde öznitelik çiftleriyle eğitilen sistemlerde elde edilen en başarılı algoritmalar tek tek açıklanmıştır. Şekillerde makine öğrenimi algoritmaları kullanılarak hem doğruluk oranı hem de birim maliyet matematiksel olarak gösterilmiştir.



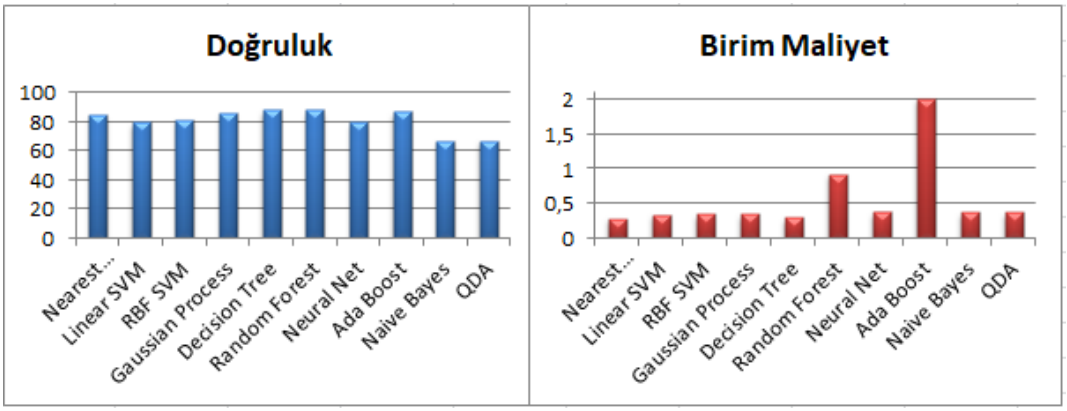
Şekil 6.6. Sport-dport öznitelik çifti doğruluk ve birim maliyet oranları

Sport-dport öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



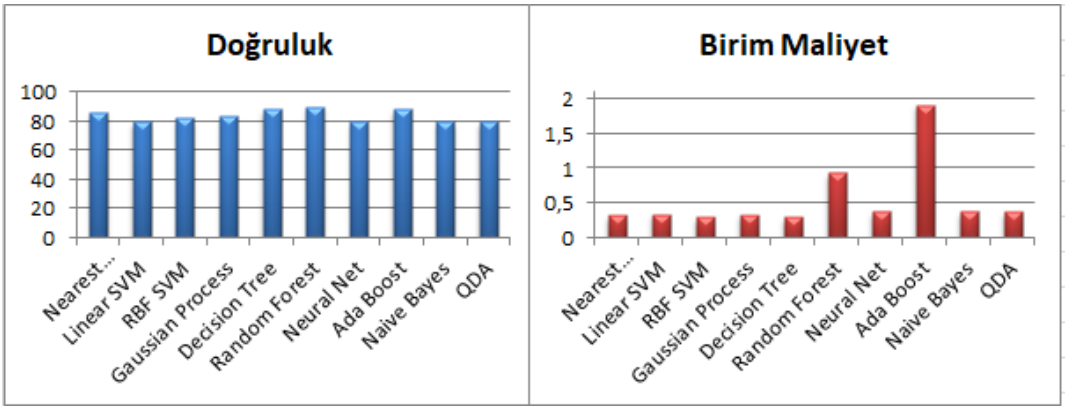
Şekil 6.7. Sport-seq öznitelik çifti doğruluk ve birim maliyet oranları

Sport-dport öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



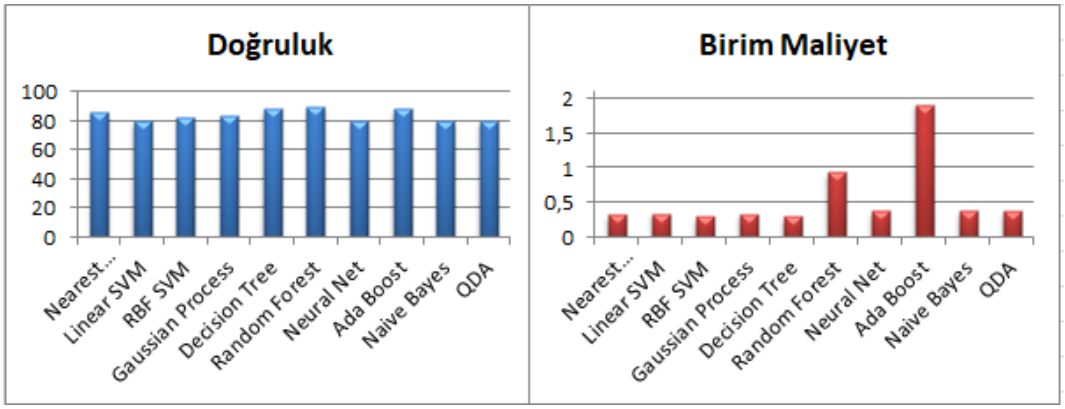
Şekil 6.8. Sport-stddev öznitelik çifti doğruluk ve birim maliyet oranları

Sport-stddev öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



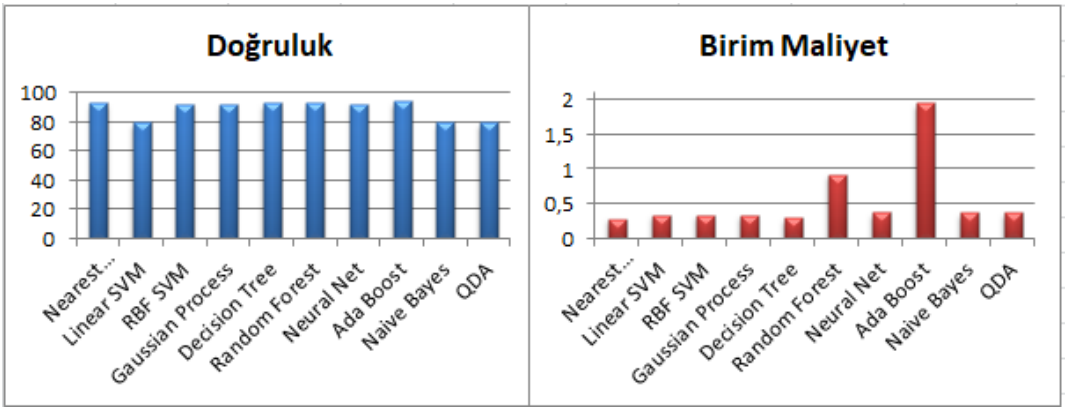
Şekil 6.9. Sport-N…SrcIP öznitelik çifti doğruluk ve birim maliyet oranları

Sport-N…SrcIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



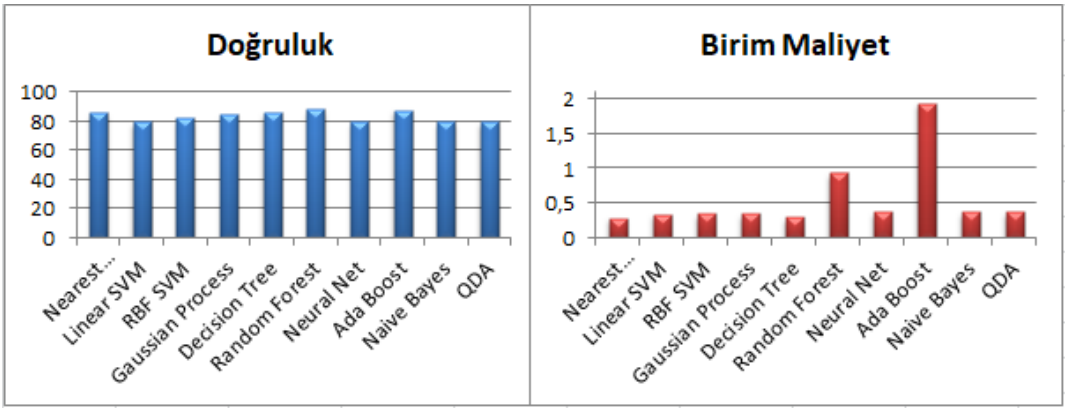
Şekil 6.10. Sport-min öznitelik çifti doğruluk ve birim maliyet oranları

Sport-min öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



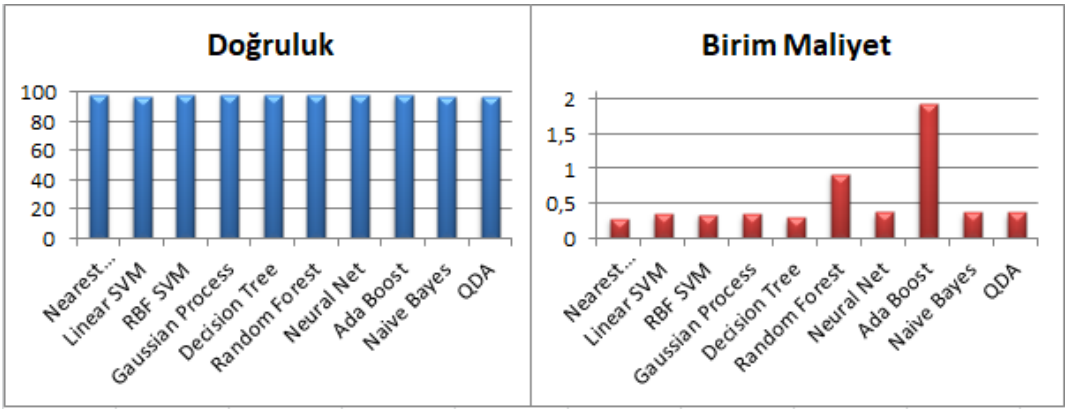
Şekil 6.11. Sport-state_number öznitelik çifti doğruluk ve birim maliyet oranları

Sport-state_number öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



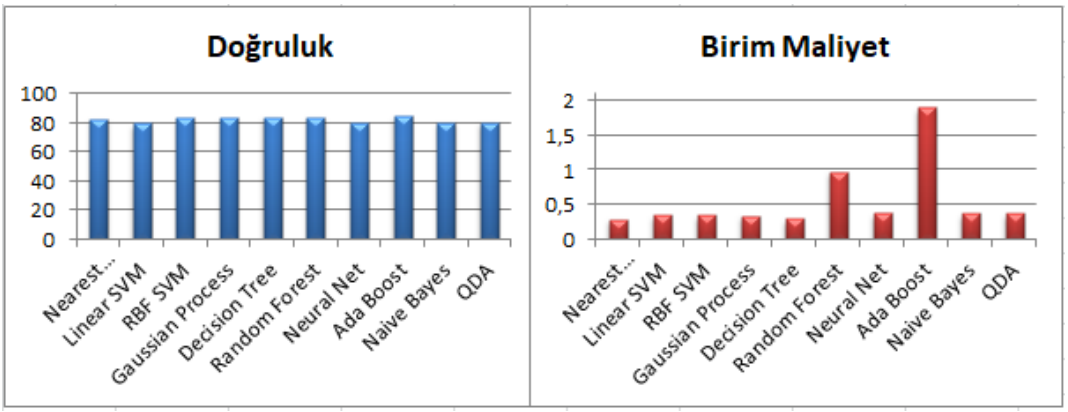
Şekil 6.12. Sport-mean öznitelik çifti doğruluk ve birim maliyet oranları

Sport-mean öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



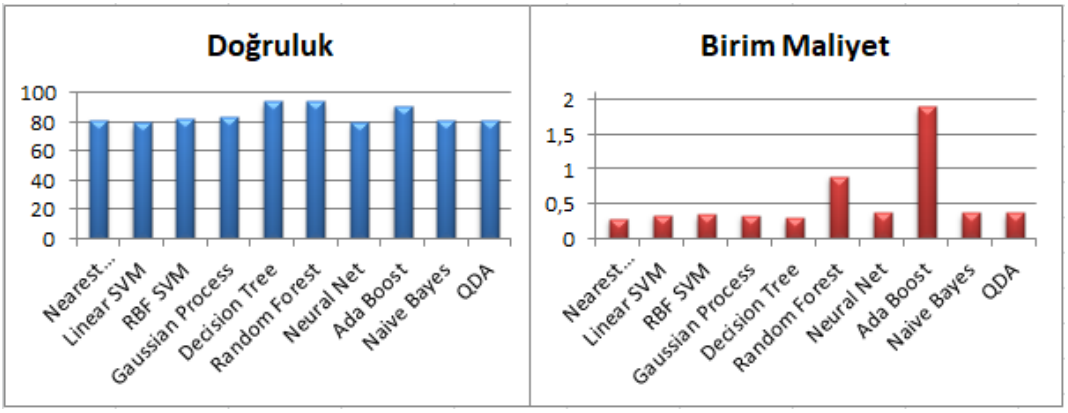
Şekil 6.13. Sport-N…DstIP öznitelik çifti doğruluk ve birim maliyet oranları

Sport-N…DstIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



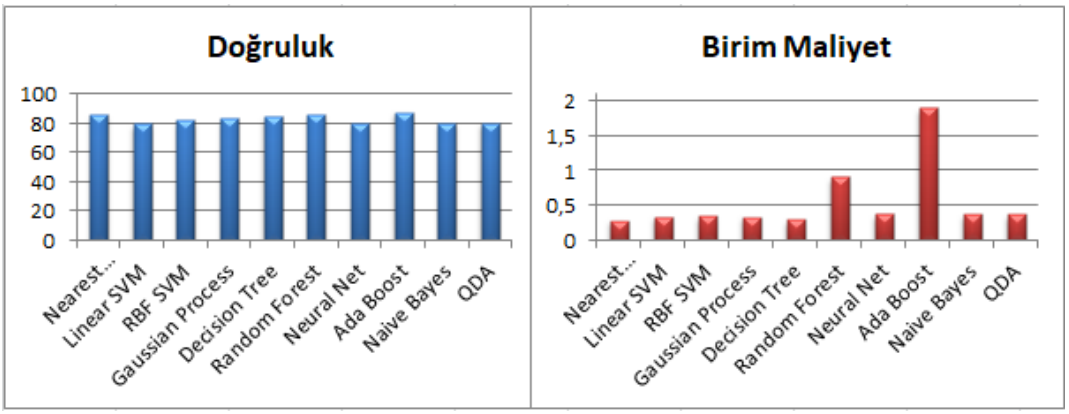
Şekil 6.14. Sport-drate öznitelik çifti doğruluk ve birim maliyet oranları

Sport-drate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



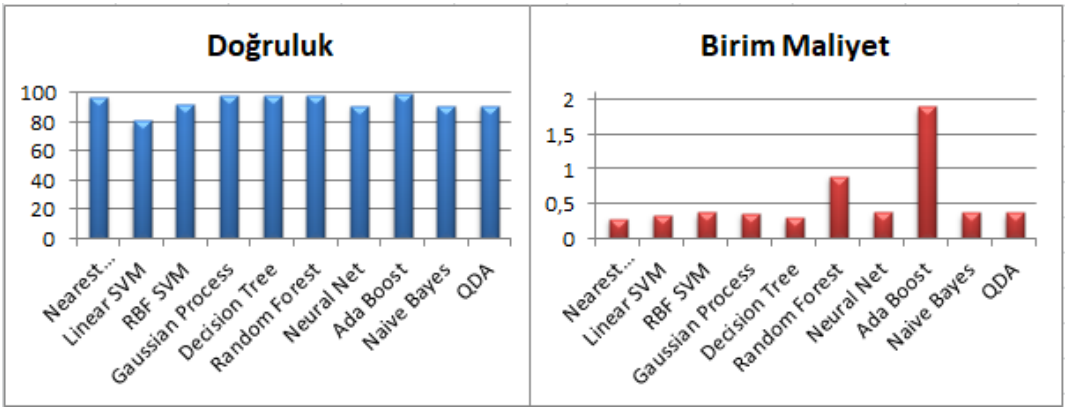
Şekil 6.15. Sport-srate öznitelik çifti doğruluk ve birim maliyet oranları

Sport-srate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



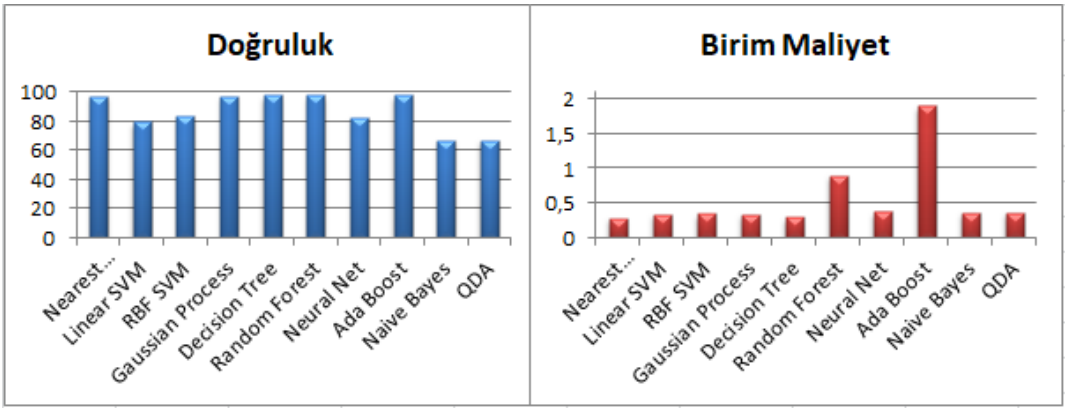
Şekil 6.16. Sport-max öznitelik çifti doğruluk ve birim maliyet oranları

Sport-max öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



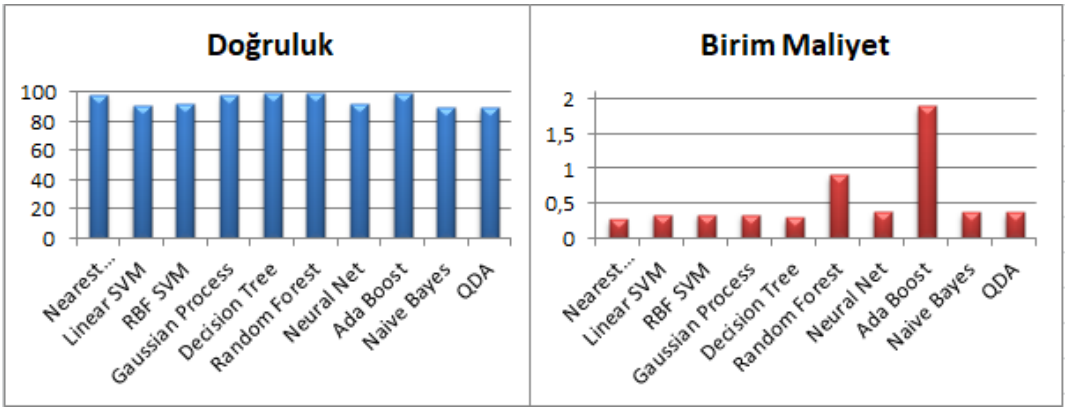
Şekil 6.17. Dport-seq öznitelik çifti doğruluk ve birim maliyet oranları

Dport-seq öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



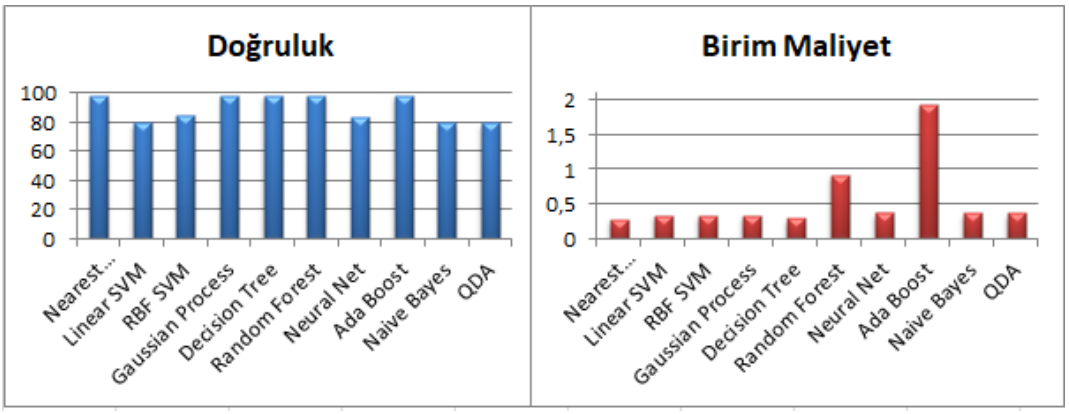
Şekil 6.18. Dport-stddev öznitelik çifti doğruluk ve birim maliyet oranları

Dport-stddev öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Decision Treee ve Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



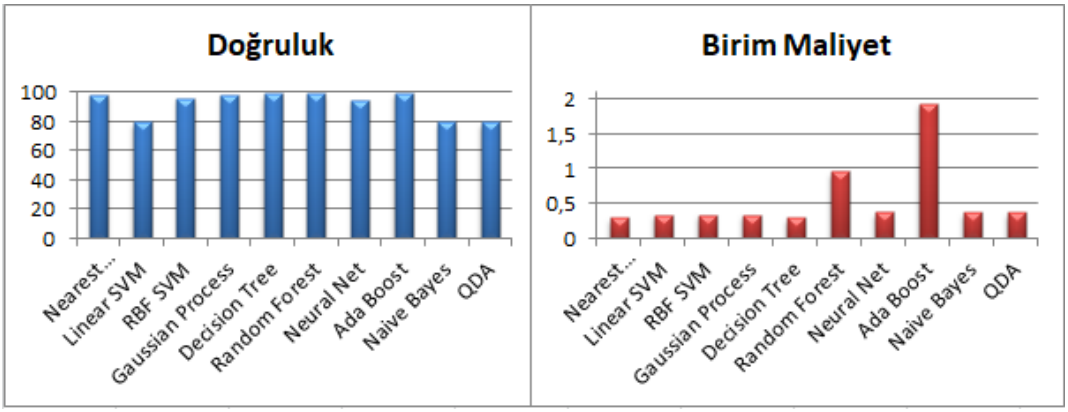
Şekil 6.19. Dport-N…SrcIP öznitelik çifti doğruluk ve birim maliyet oranları

Dport-N…SrcIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Decision Tree ve Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



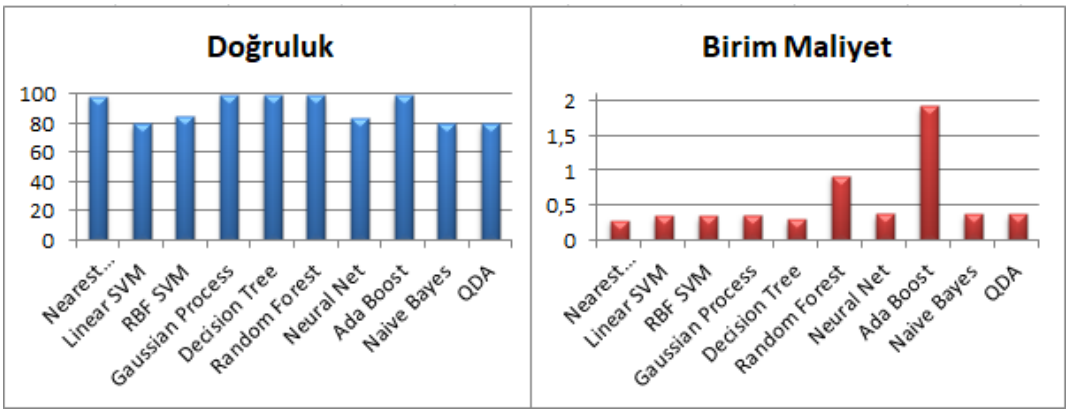
Şekil 6.20. Dport-min öznitelik çifti doğruluk ve birim maliyet oranları

Dport-min öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Decision Tree ve Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



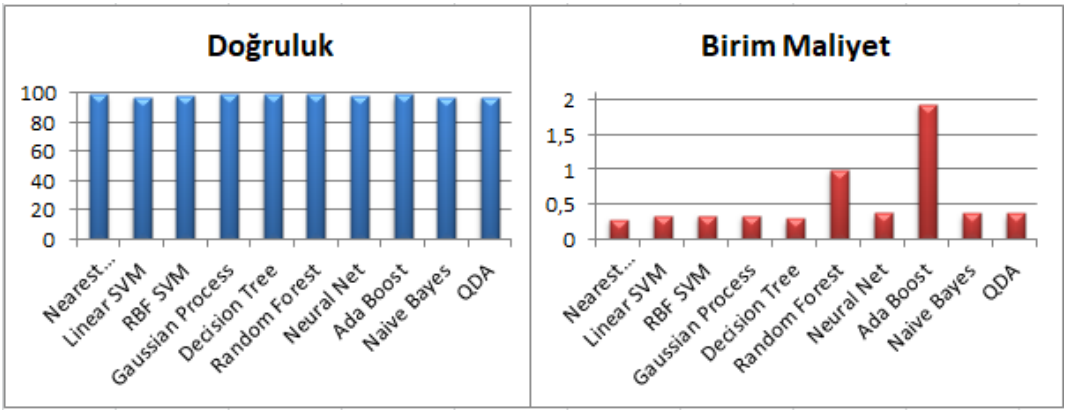
Şekil 6.21. Dport-state_number öznitelik çifti doğruluk ve birim maliyet oranları

Dport-state_number öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



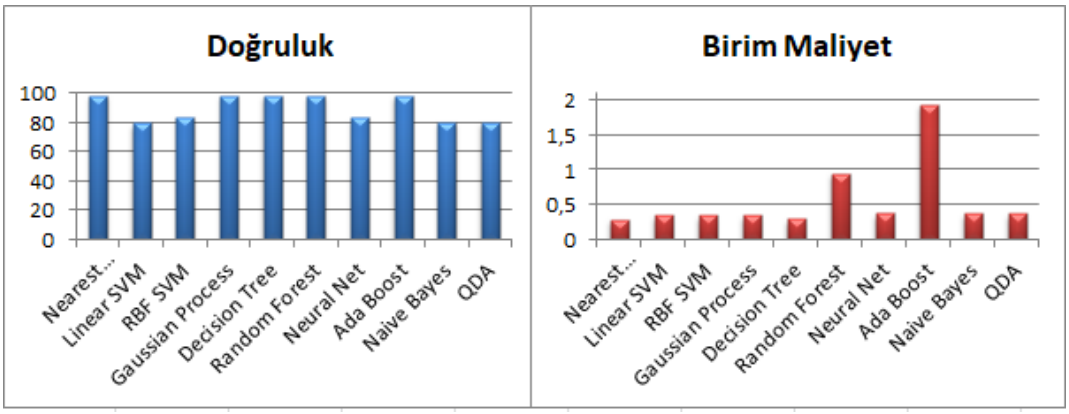
Şekil 6.22. Dport-mean öznitelik çifti doğruluk ve birim maliyet oranları

Dport-mean öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Decision Tree olmaktadır.



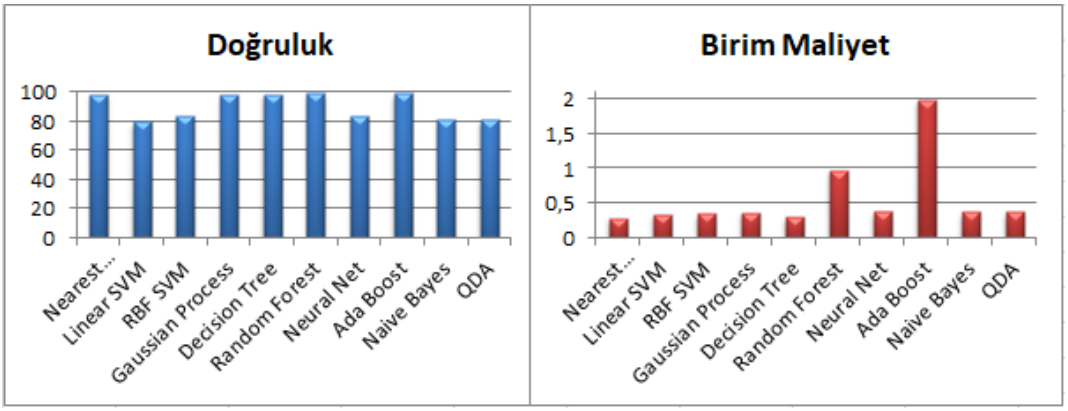
Şekil 6.23. Dport-N…DstIP öznitelik çifti doğruluk ve birim maliyet oranları

Dport-N…DstIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Gaussian Process ve Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Gaussian Process olmaktadır.



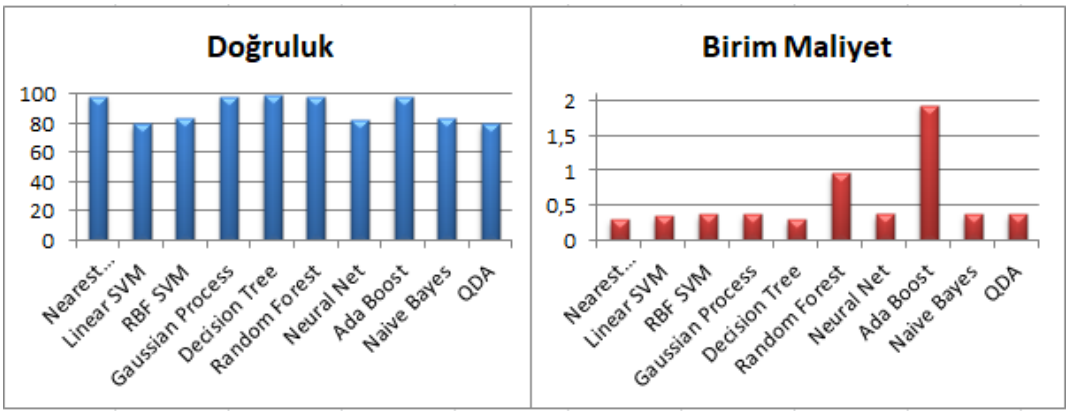
Şekil 6.24. Dport-drate öznitelik çifti doğruluk ve birim maliyet oranları

Dport-drate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



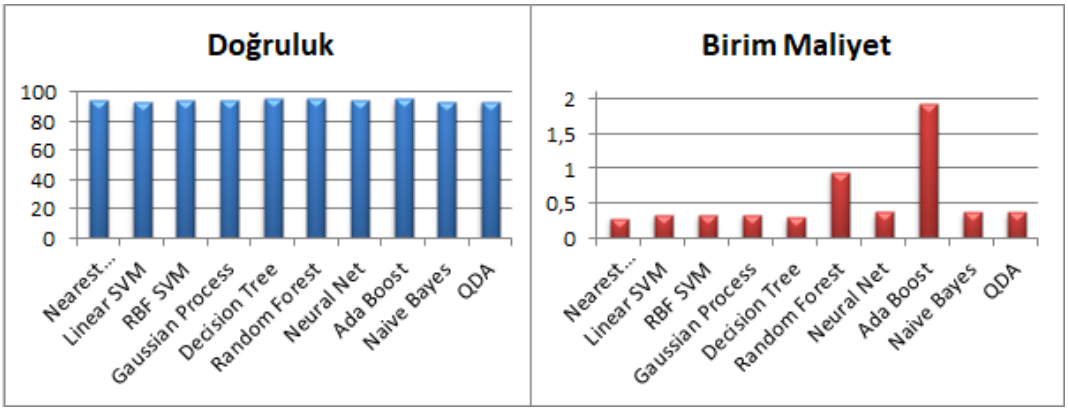
Şekil 6.25. Dport-srate öznitelik çifti doğruluk ve birim maliyet oranları

Dport-srate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



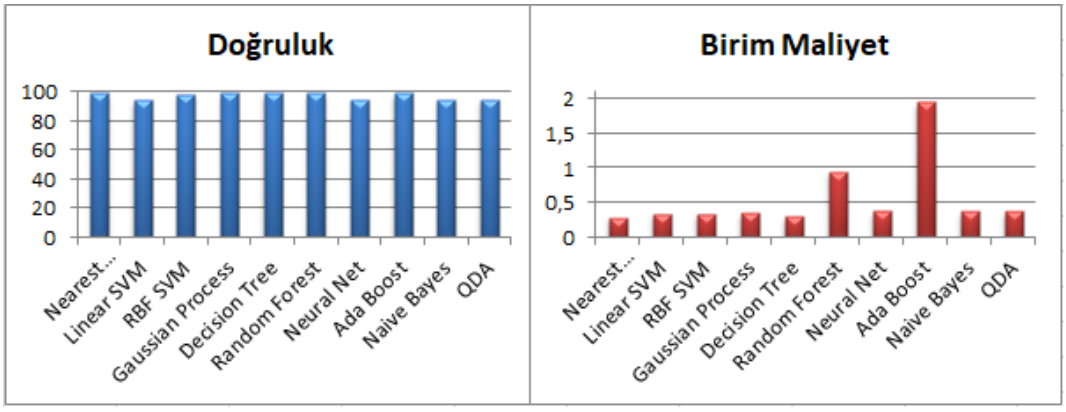
Şekil 6.26. Dport- max öznitelik çifti doğruluk ve birim maliyet oranları

Dport- max öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Decision Tree olmaktadır.



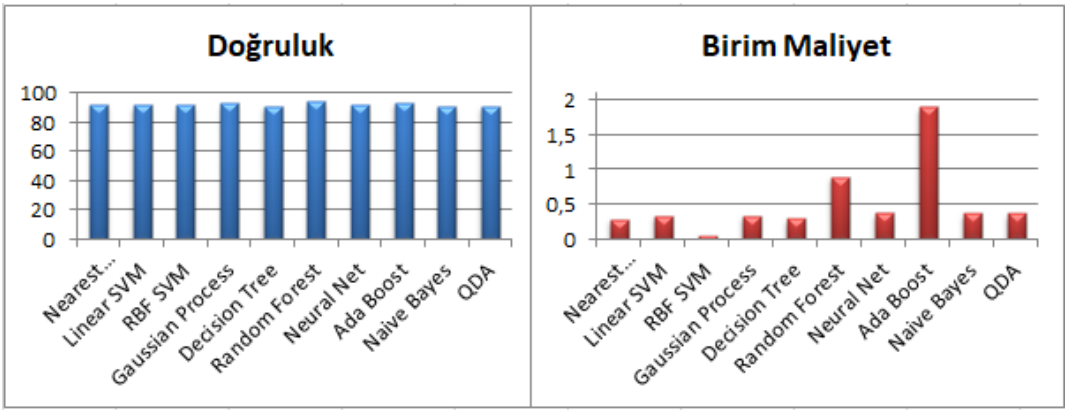
Şekil 6.27. Seq-stddev öznitelik çifti doğruluk ve birim maliyet oranları

Seq-stddev öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



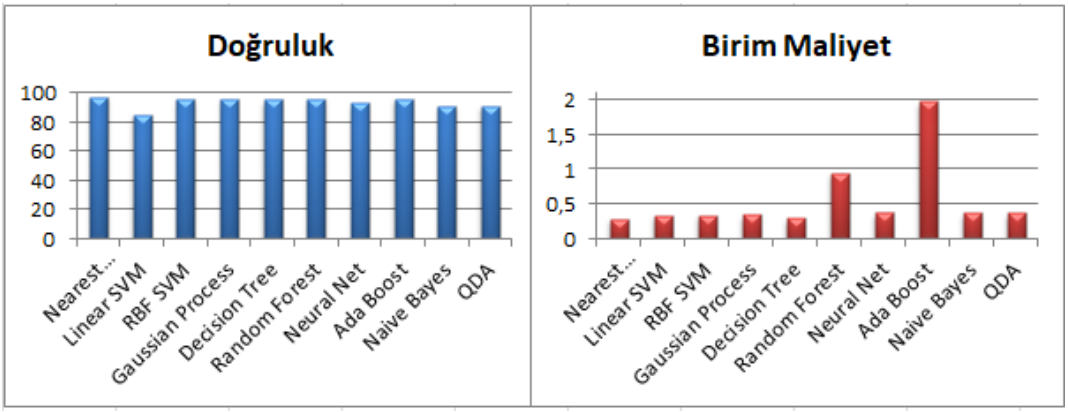
Şekil 6.28. Seq-N…SrcIP öznitelik çifti doğruluk ve birim maliyet oranları

Seq-N…SrcIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ve Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



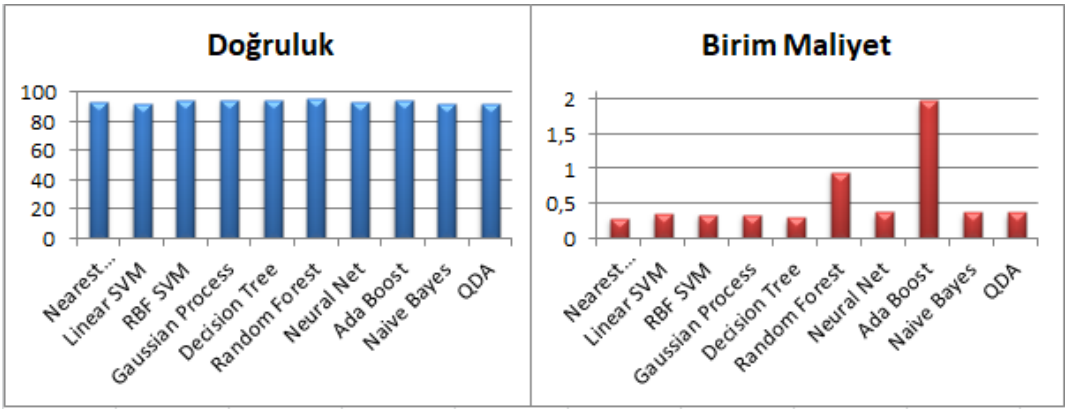
Şekil 6.29. Seq-min öznitelik çifti doğruluk ve birim maliyet oranları

Seq-min öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Gaussian Process olmaktadır.



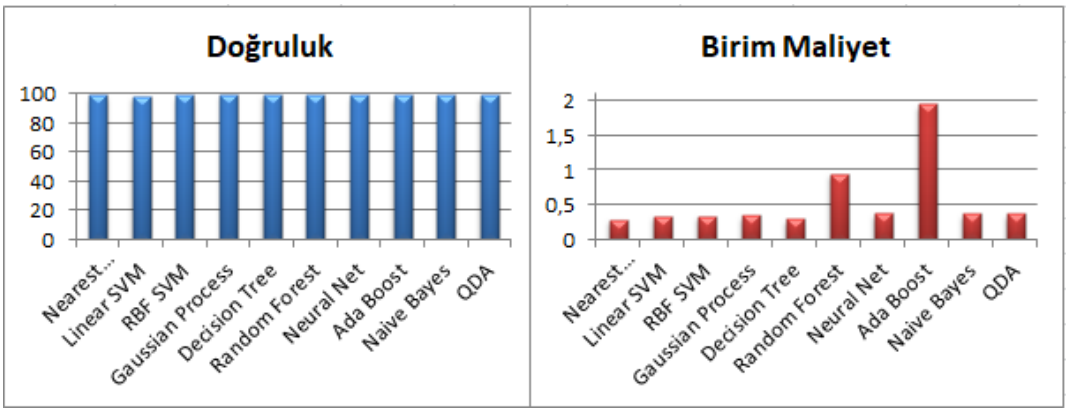
Şekil 6.30. Seq-state_number öznitelik çifti doğruluk ve birim maliyet oranları

Seq-state_number öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



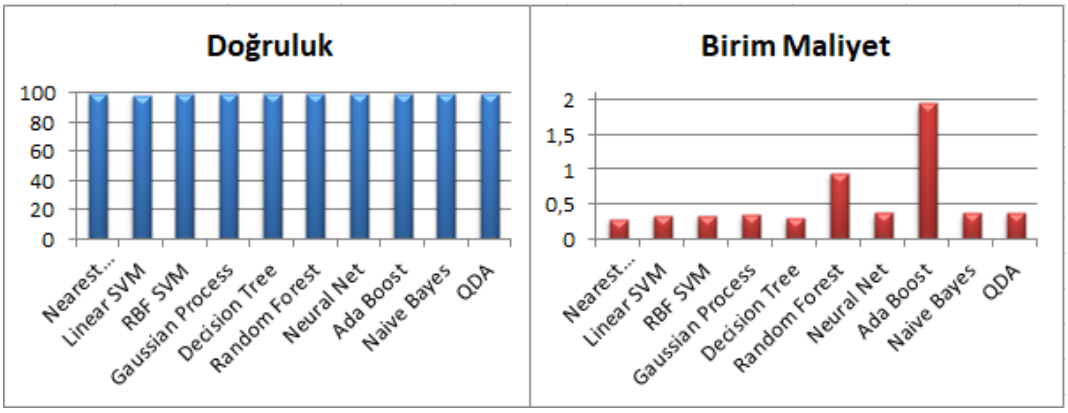
Şekil 6.31. Seq-mean öznitelik çifti doğruluk ve birim maliyet oranları

Seq-mean öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



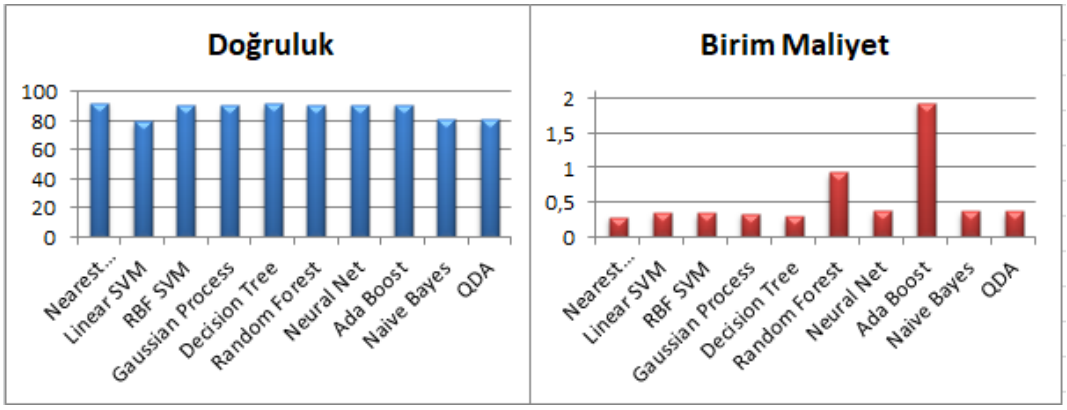
Şekil 6.32. Seq-N…DstIP öznitelik çifti doğruluk ve birim maliyet oranları

Seq-N…DstIP öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Decision Tree olmaktadır.



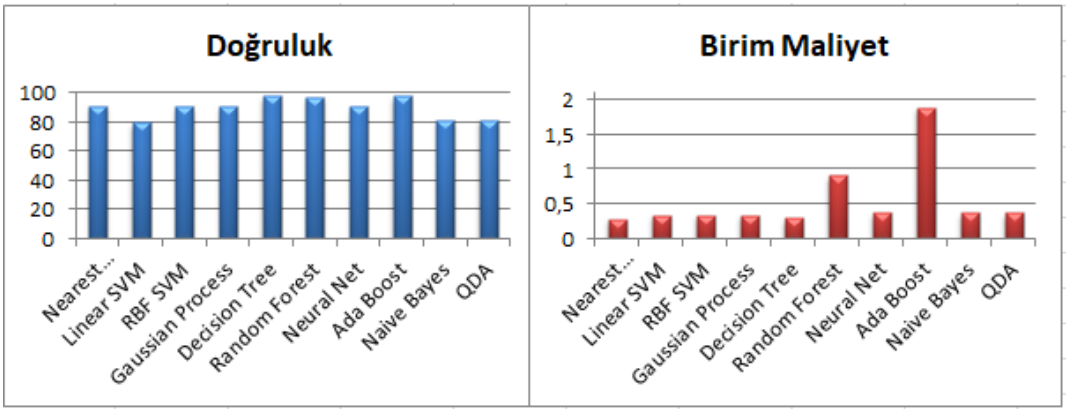
Şekil 6.33. Seq-drate öznitelik çifti doğruluk ve birim maliyet oranları

Seq-drate öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



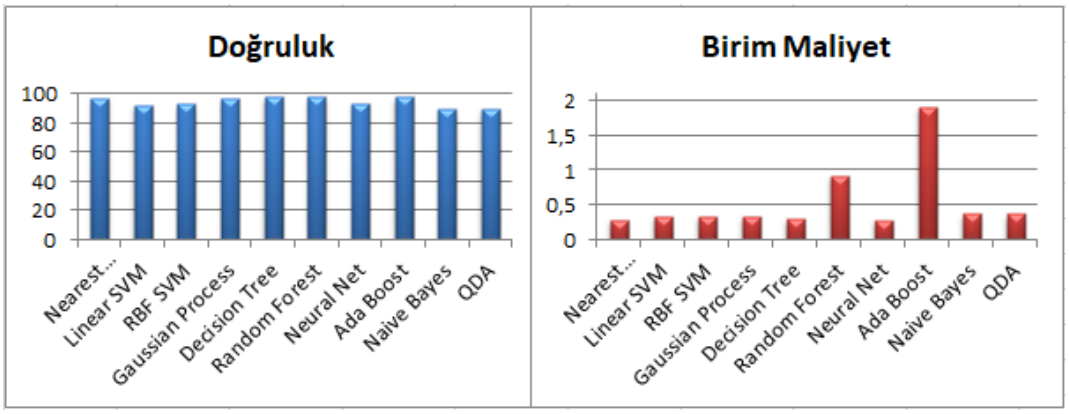
Şekil 6.34. Seq-srate öznitelik çifti doğruluk ve birim maliyet oranları

Seq-srate öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Decision Tree olmaktadır.



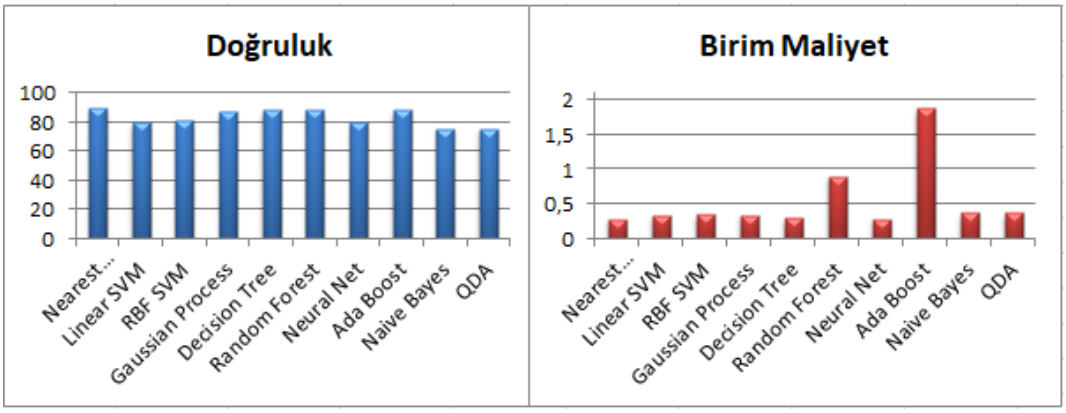
Şekil 6.35. Seq-max öznitelik çifti doğruluk ve birim maliyet oranları

Seq-max öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



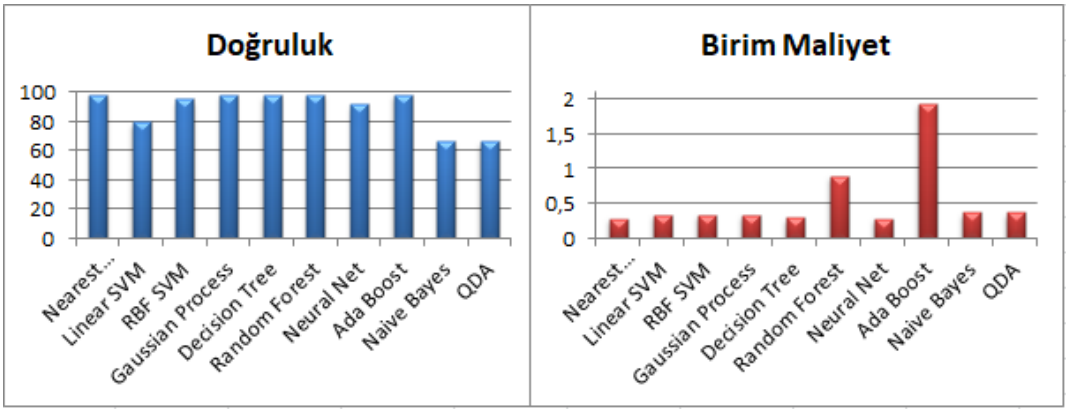
Şekil 6.36. Stddev-N…SrcIP öznitelik çifti doğruluk ve birim maliyet oranları

Stddev-N…SrcIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



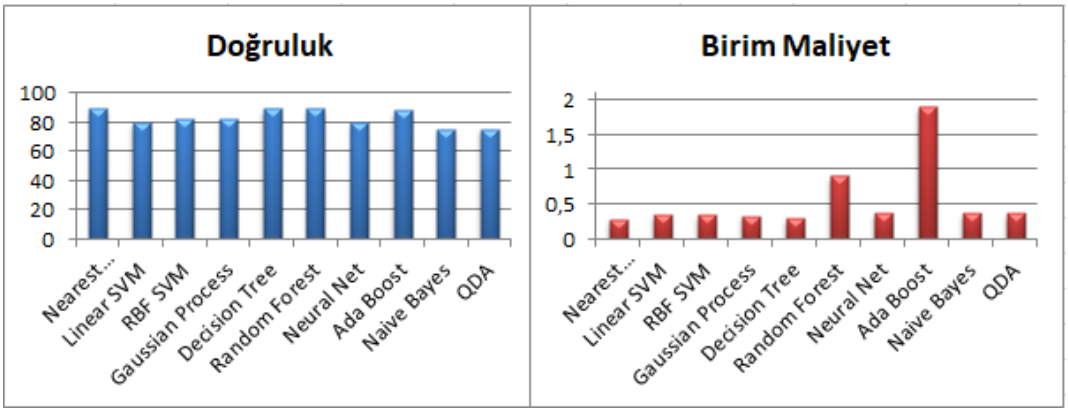
Şekil 6.37. Stddev-min öznitelik çifti doğruluk ve birim maliyet oranları

Stddev-min öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



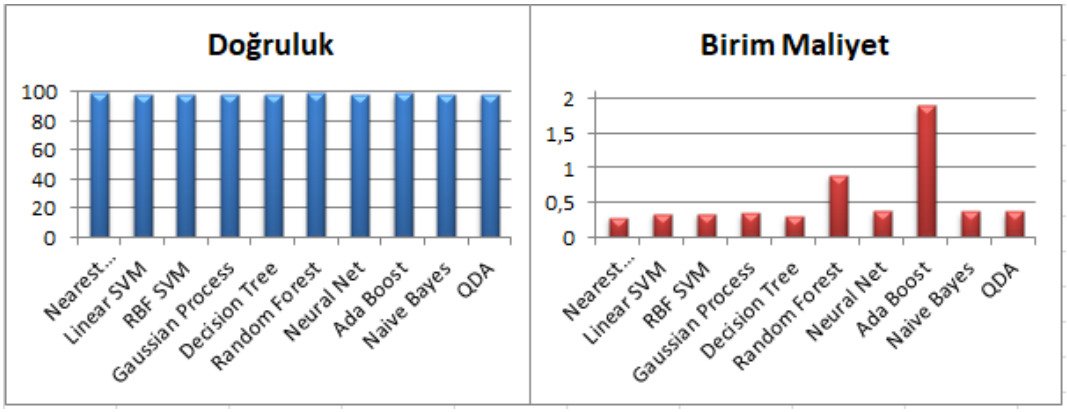
Şekil 6.38. Stddev-state-number öznitelik çifti doğruluk ve birim maliyet oranları

Stddev-state-number öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



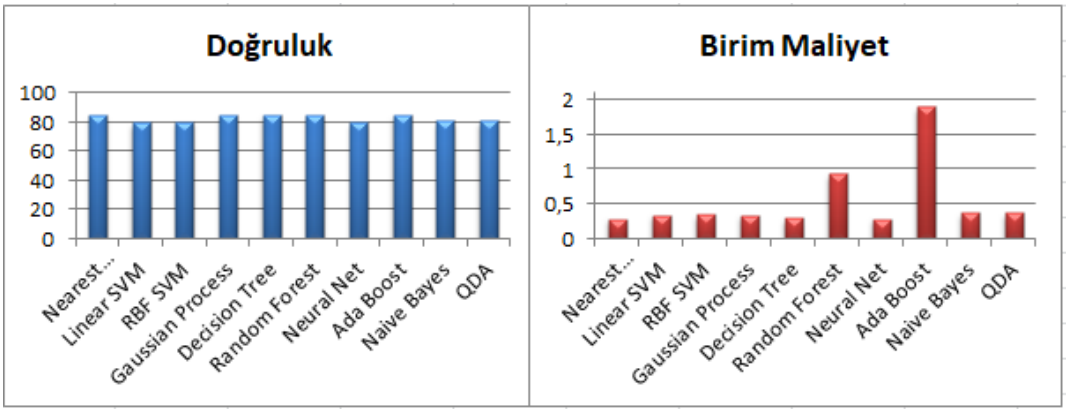
Şekil 6.39. Stddev-mean öznitelik çifti doğruluk ve birim maliyet oranları

Stddev-mean öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



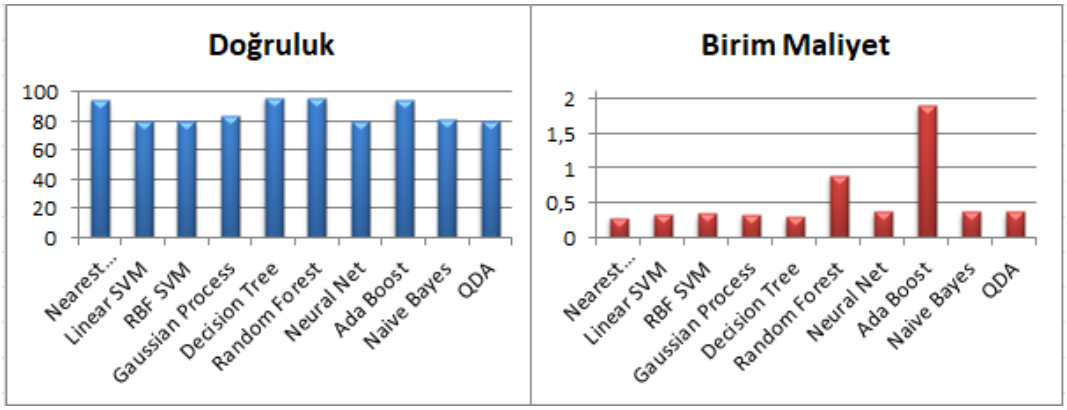
Şekil 6.40. Stddev-N…DstIP öznitelik çifti doğruluk ve birim maliyet oranları

Stddev-N…DstIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



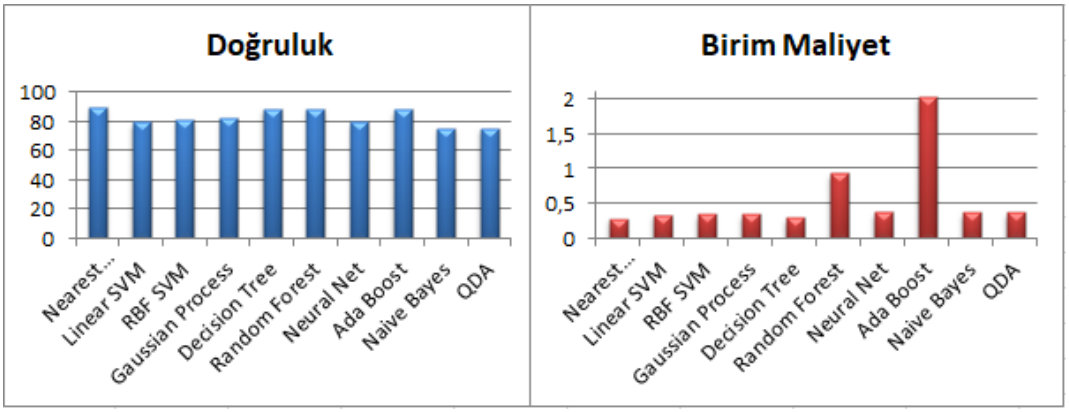
Şekil 6.41. Stddev-drate öznitelik çifti doğruluk ve birim maliyet oranları

Stddev-drate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Gaussian Process olmaktadır.



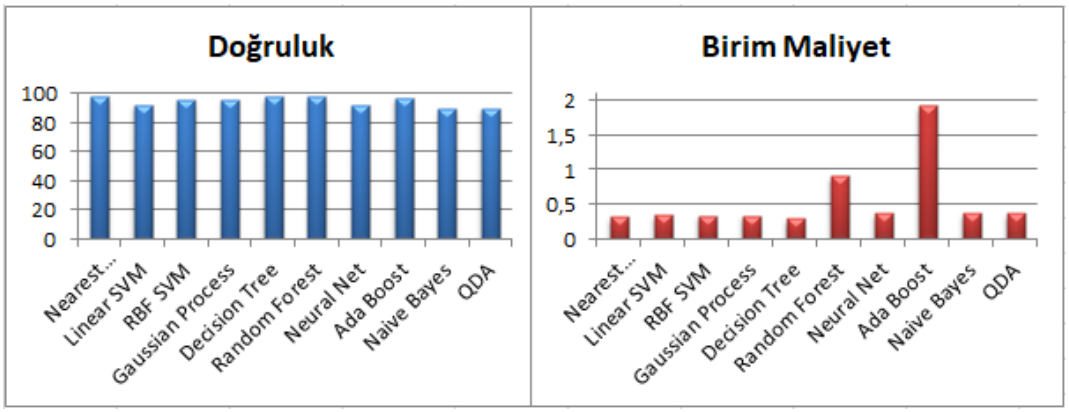
Şekil 6.42. Stddev-srate öznitelik çifti doğruluk ve birim maliyet oranları

Stddev-srate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Decision Tree ve Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



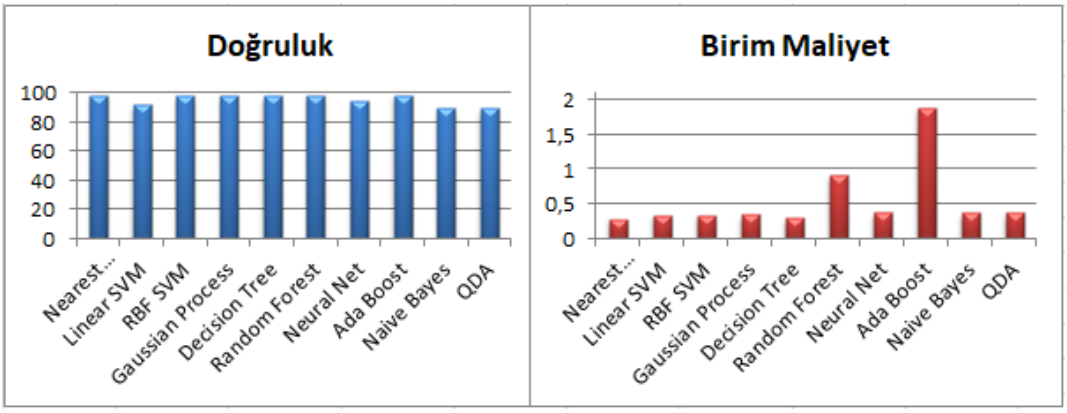
Şekil 6.43. Stddev-max öznitelik çifti doğruluk ve birim maliyet oranları

Stddev-max öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



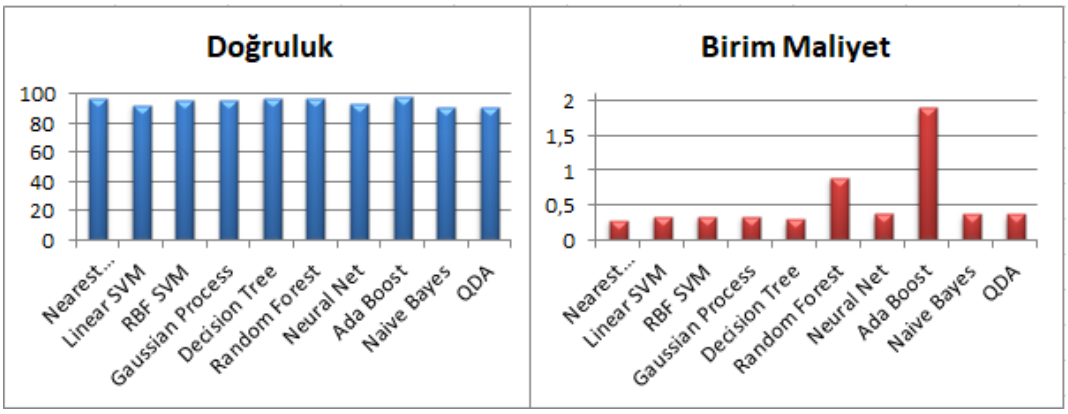
Şekil 6.44. N…SrcIP-min öznitelik çifti doğruluk ve birim maliyet oranları

N…SrcIP-min öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Decision Tree ve Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



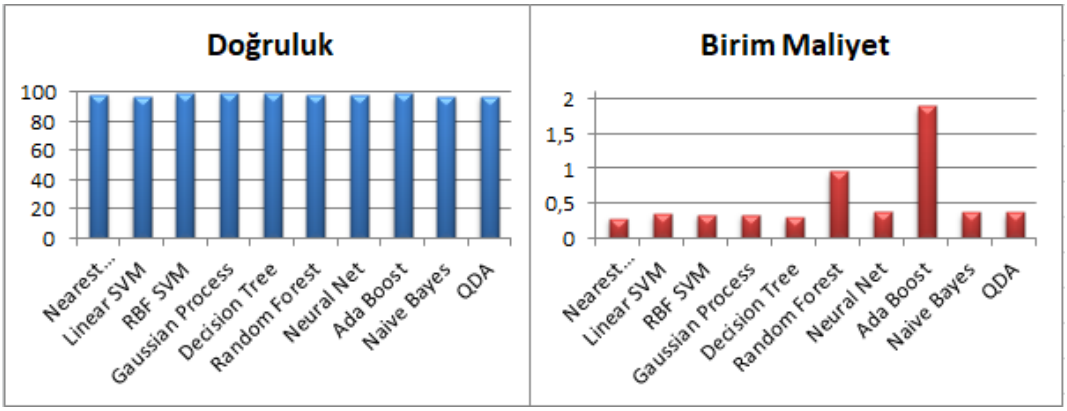
Şekil 6.45. N…SrcIP-state_number öznitelik çifti doğruluk ve birim maliyet oranları

N…SrcIP-state_number öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ve Nearest Neighbors ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



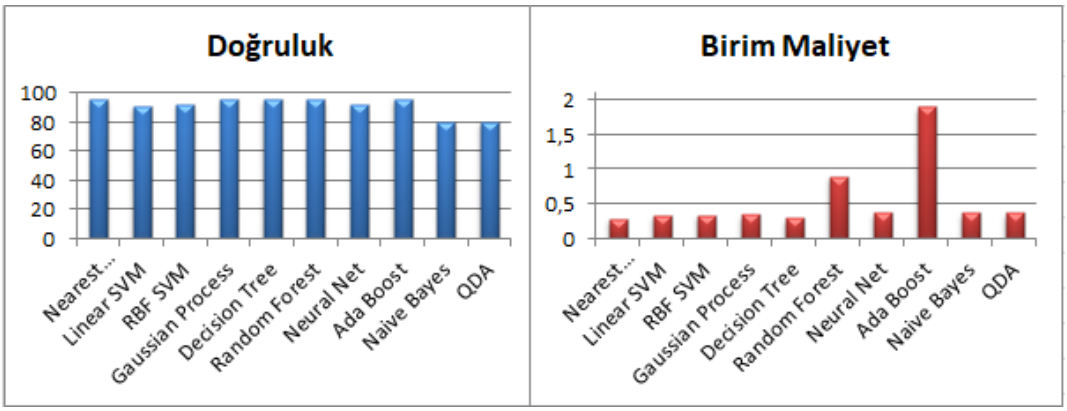
Şekil 6.46. N…SrcIP-mean öznitelik çifti doğruluk ve birim maliyet oranları

N…SrcIP-mean öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



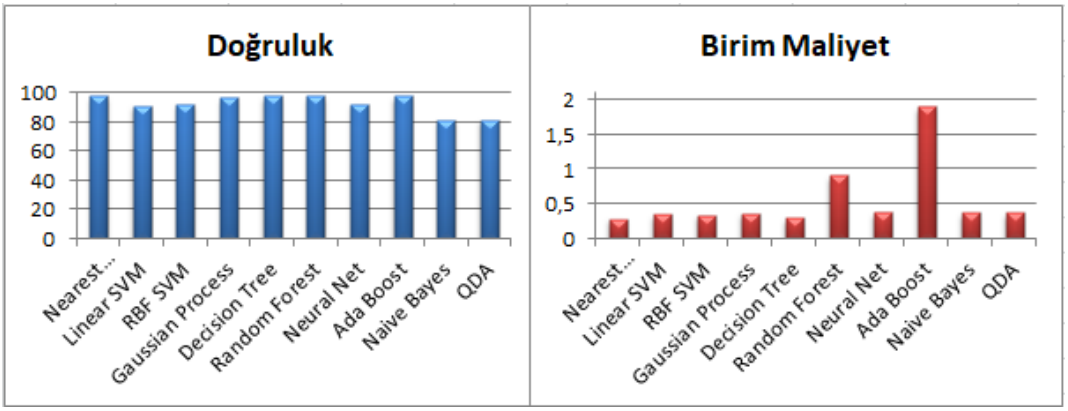
Şekil 6.47. N…SrcIP-N…DstIP öznitelik çifti doğruluk ve birim maliyet oranları

N…SrcIP-N…DstIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



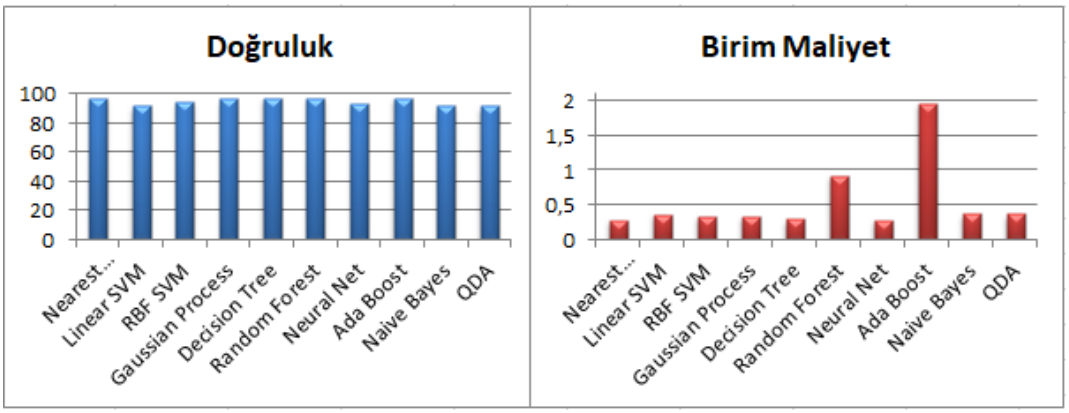
Şekil 6.48. N…SrcIP-drate öznitelik çifti doğruluk ve birim maliyet oranları

N…SrcIP-drate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



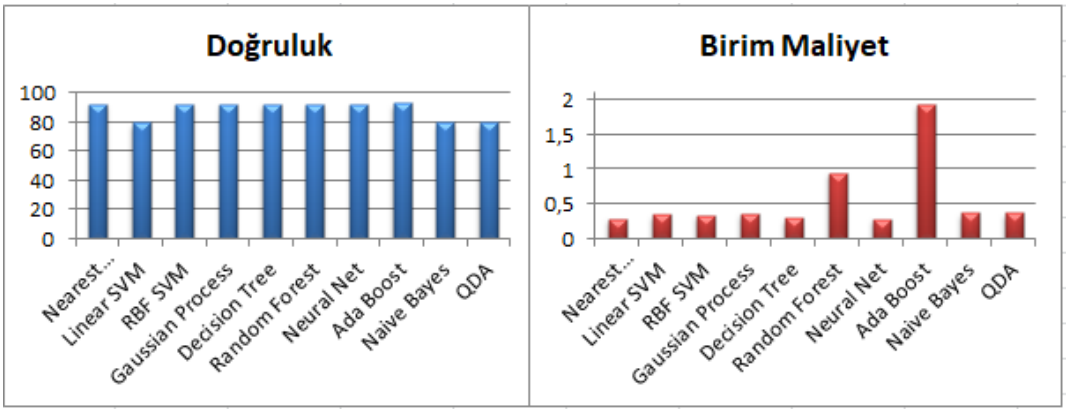
Şekil 6.49. N…SrcIP-srate öznitelik çifti doğruluk ve birim maliyet oranları

N…SrcIP-srate öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



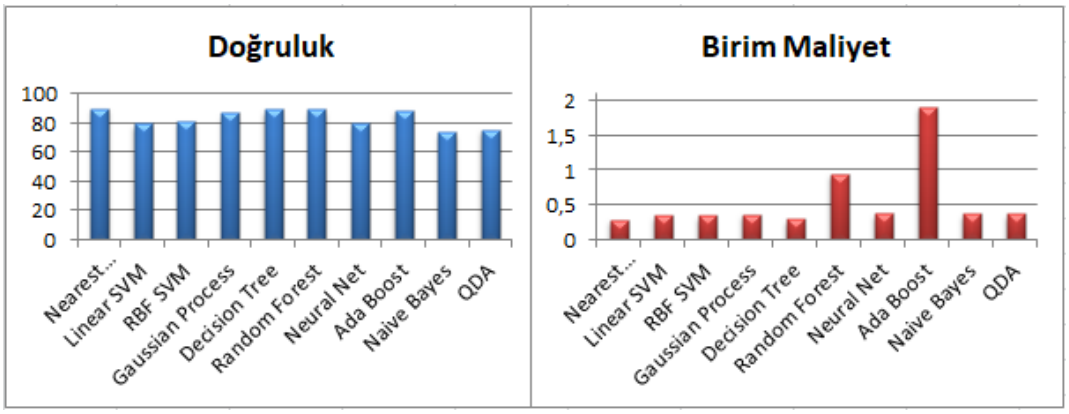
Şekil 6.50. N…SrcIP-max öznitelik çifti doğruluk ve birim maliyet oranları

N…SrcIP-max öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



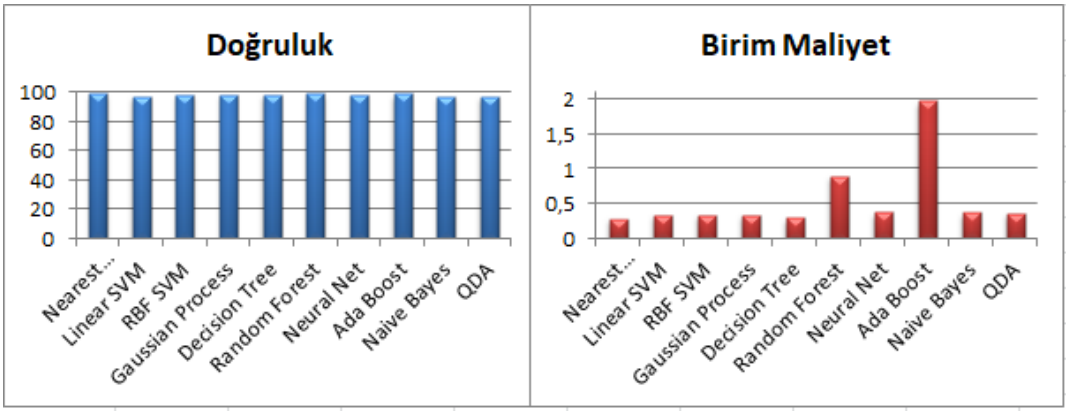
Şekil 6.51. Min-state_number öznitelik çifti doğruluk ve birim maliyet oranları

Min-state_number öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Gaussian Process olmaktadır.



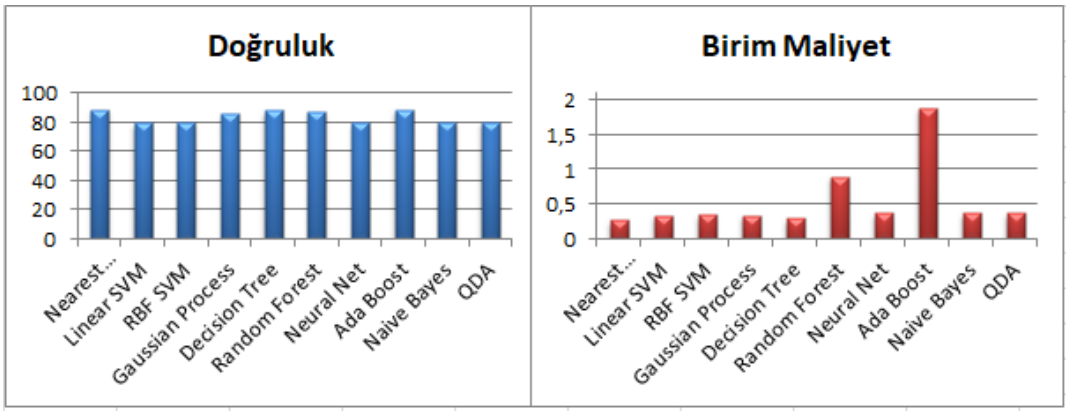
Şekil 6.52. Min-mean öznitelik çifti doğruluk ve birim maliyet oranları

Min-mean öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



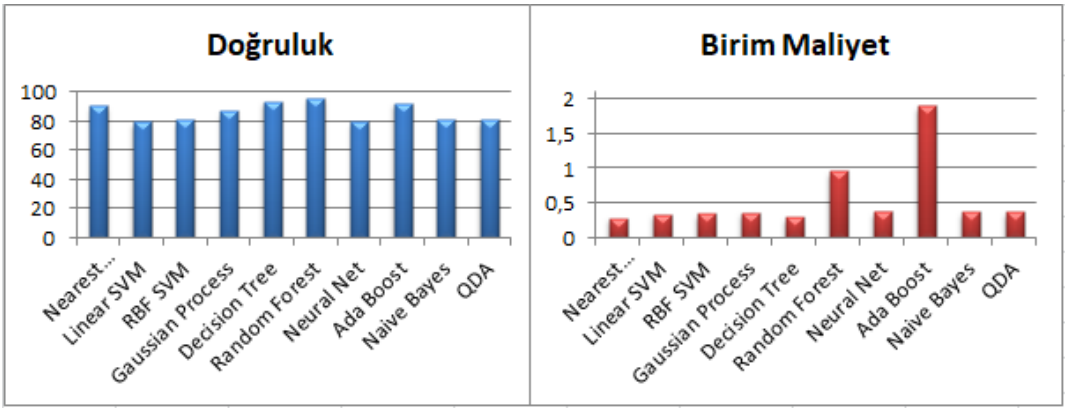
Şekil 6.53. Min-N...DstIP öznitelik çifti doğruluk ve birim maliyet oranları

Min-N…DstIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



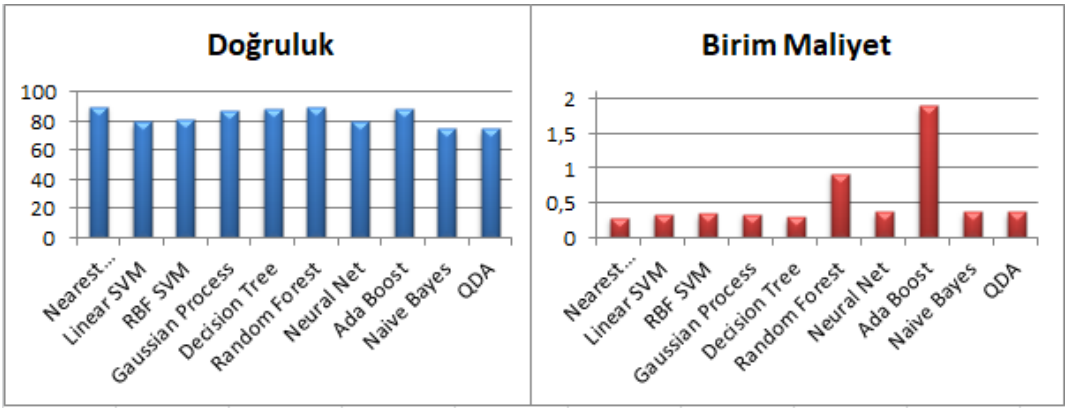
Şekil 6.54. Min-drate öznitelik çifti doğruluk ve birim maliyet oranları

Min-drate öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



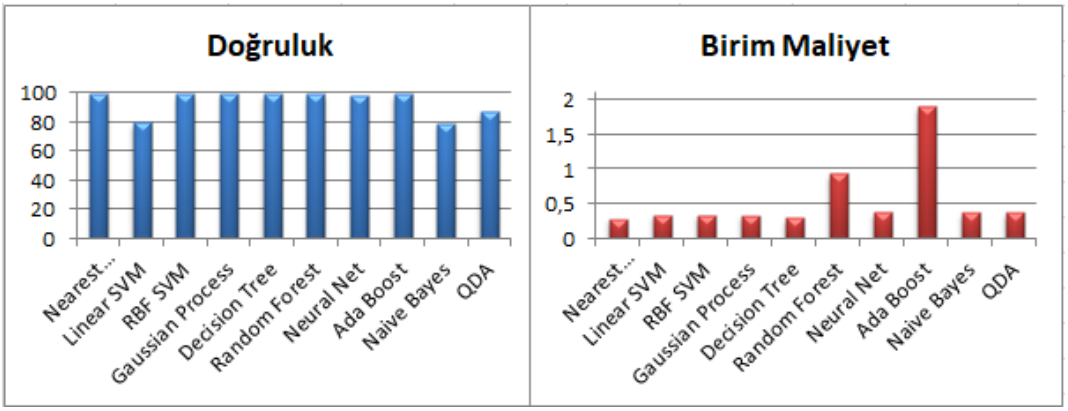
Şekil 6.55. Min-srate öznitelik çifti doğruluk ve birim maliyet oranları

Min-srate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



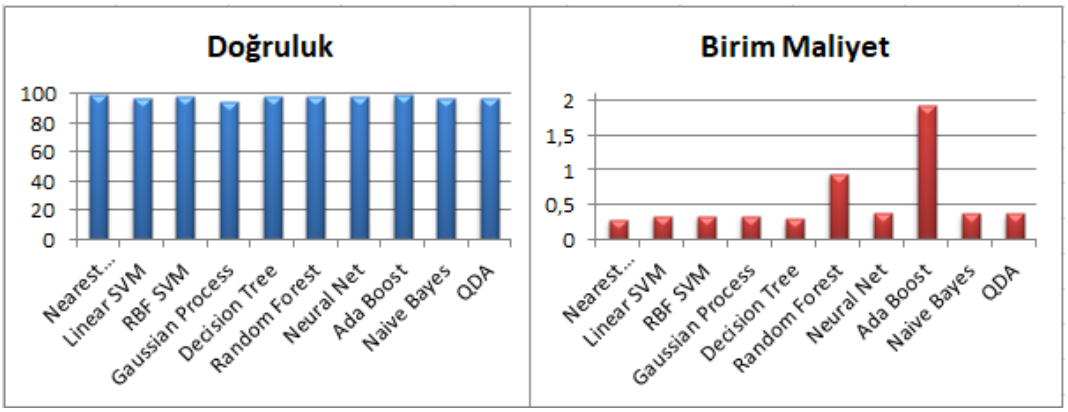
Şekil 6.56. Min-max öznitelik çifti doğruluk ve birim maliyet oranları

Min-max öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



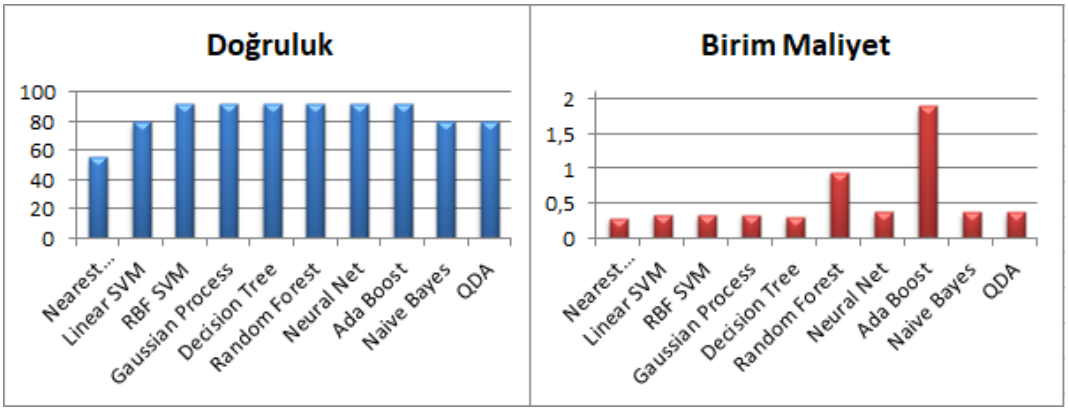
Şekil 6.57. State_number-mean öznitelik çifti doğruluk ve birim maliyet oranları

State_number-mean öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Gaussian Process olmaktadır



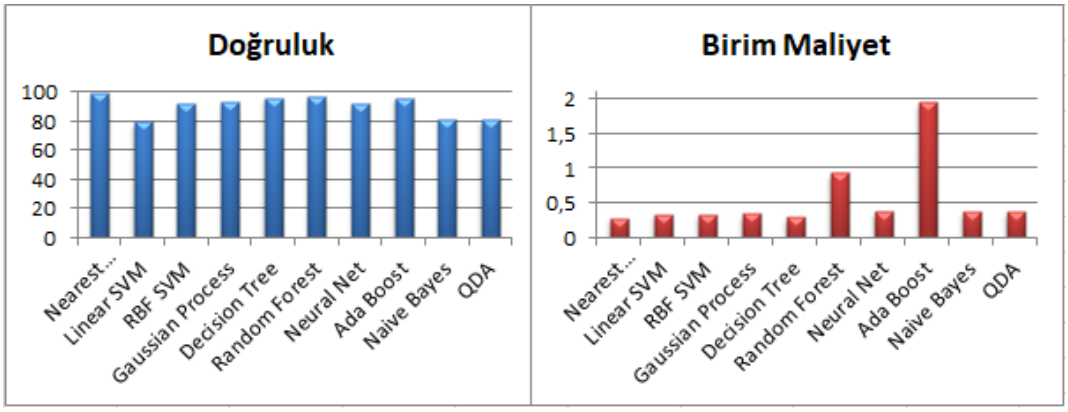
Şekil 6.58. State_number-N… DstIP öznitelik çifti doğruluk ve birim maliyet oranları

State_number-N…DstIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



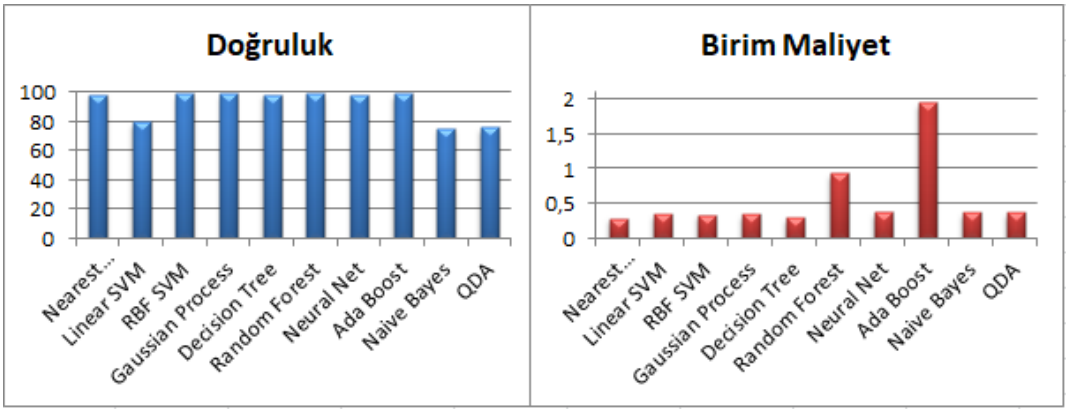
Şekil 6.59. State_number-drate öznitelik çifti doğruluk ve birim maliyet oranları

State_number-drate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu RBF SVM ve Gaussian Process ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Gaussian Process olmaktadır.



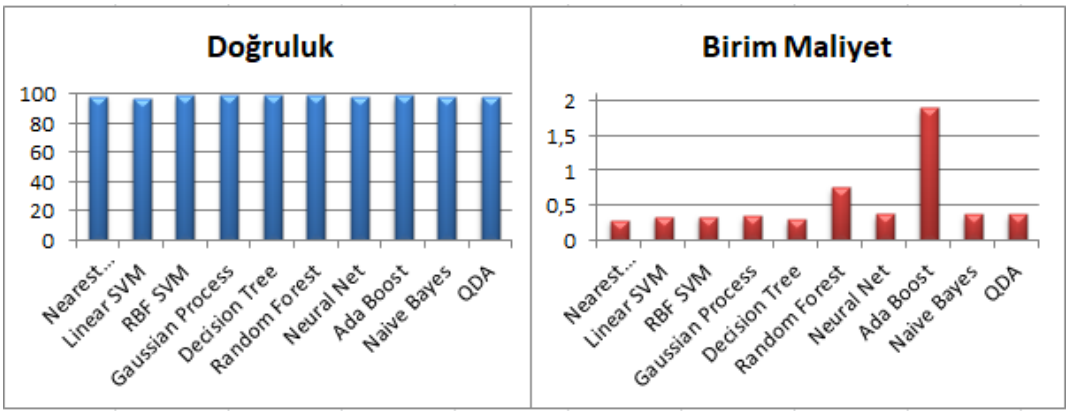
Şekil 6.60. State_number-srate öznitelik çifti doğruluk ve birim maliyet oranları

State_number-srate öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır



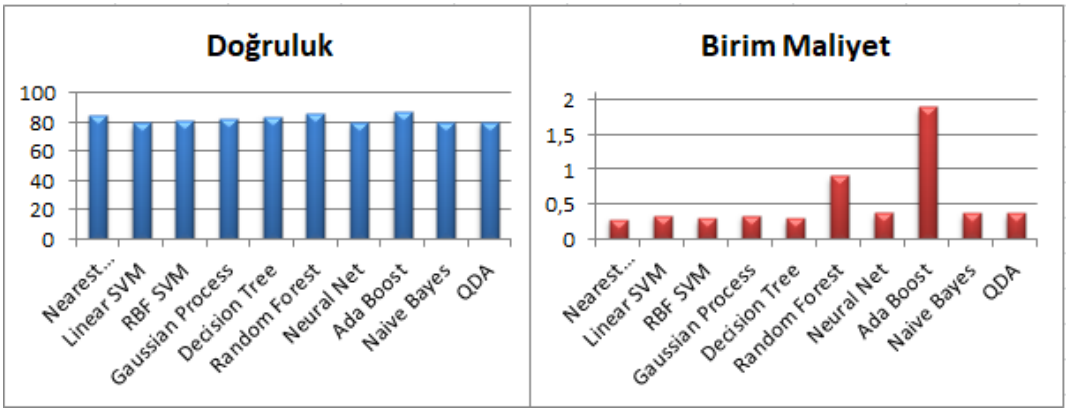
Şekil 6.61. State_number-max öznitelik çifti doğruluk ve birim maliyet oranları

State_number-max öznitelik çiftinde elde edilen en iyi doğruluk sonucunu RBF SVM ve Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod RBF SVM olmaktadır.



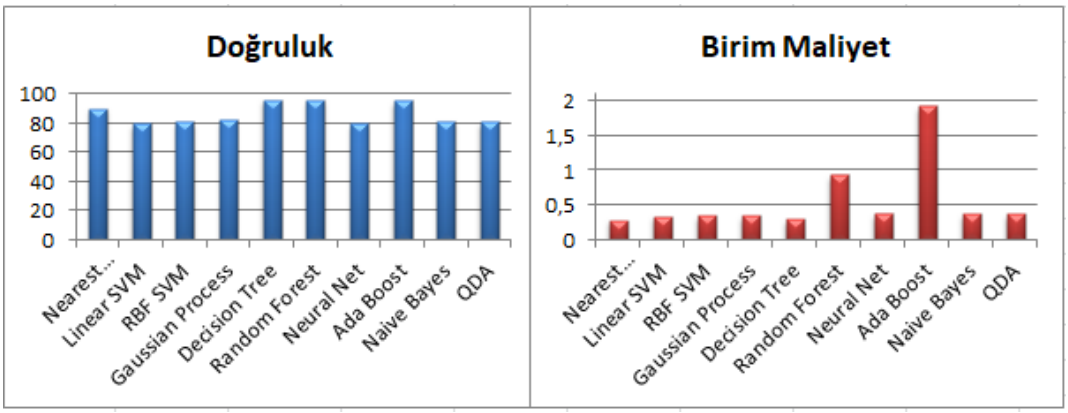
Şekil 6.62. Mean-NDstIP öznitelik çifti doğruluk ve birim maliyet oranları

Mean-NDstIP öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Gaussian Process ve Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Gaussian Process olmaktadır.



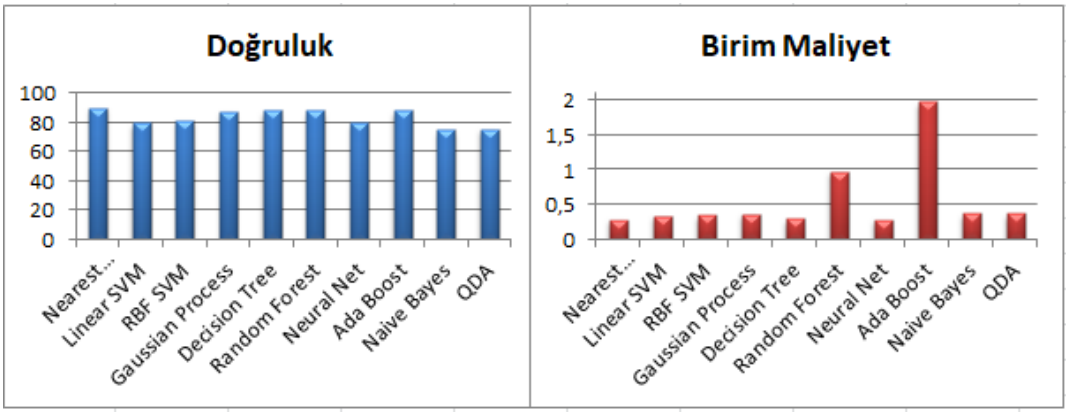
Şekil 6.63. Mean-drate öznitelik çifti doğruluk ve birim maliyet oranları

Mean-drate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



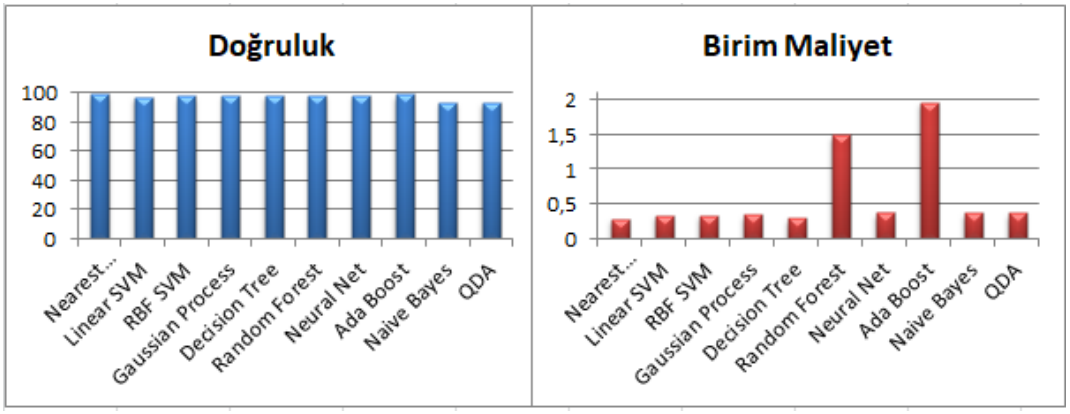
Şekil 6.64. Mean-srate öznitelik çifti doğruluk ve birim maliyet oranları

Mean-srate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



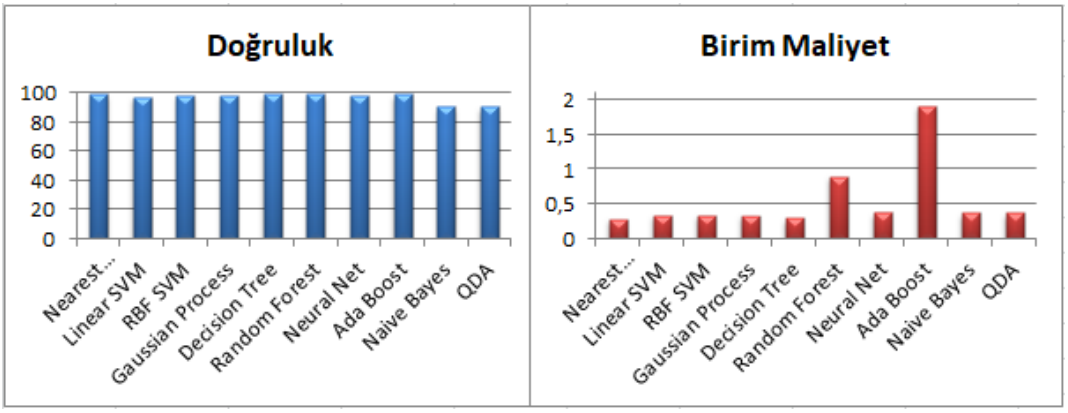
Şekil 6.65. Mean-max öznitelik çifti doğruluk ve birim maliyet oranları

Mean-max öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



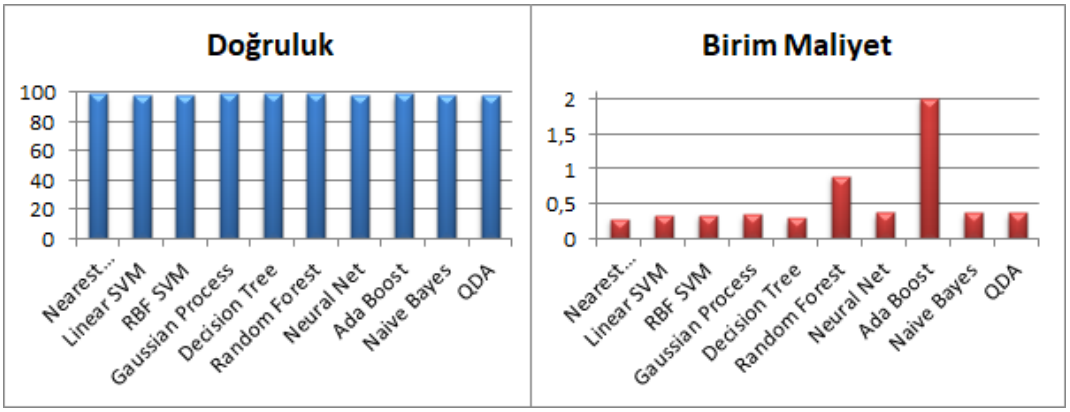
Şekil 6.66. NDstIP-drate öznitelik çifti doğruluk ve birim maliyet oranları

NDstIP-drate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Nearest Neighbors ve Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Nearest Neighbors olmaktadır.



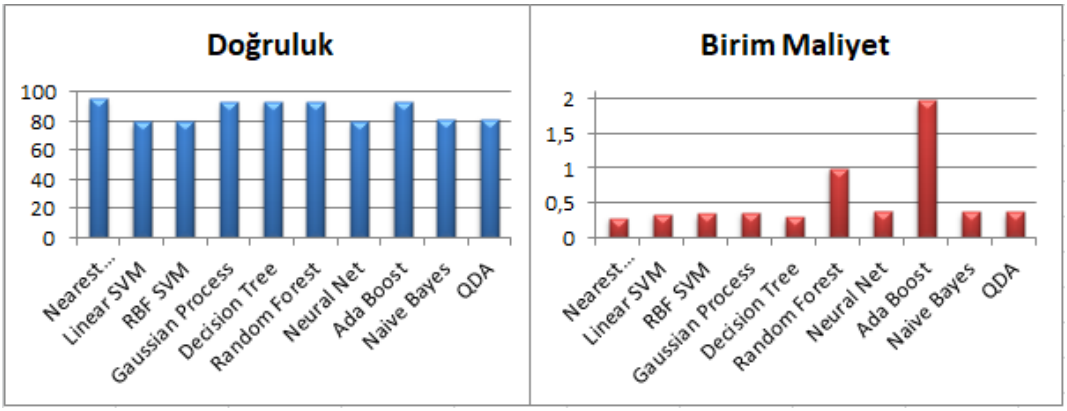
Şekil 6.67. NDstIP-srate öznitelik çifti doğruluk ve birim maliyet oranları

NDstIP-srate öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



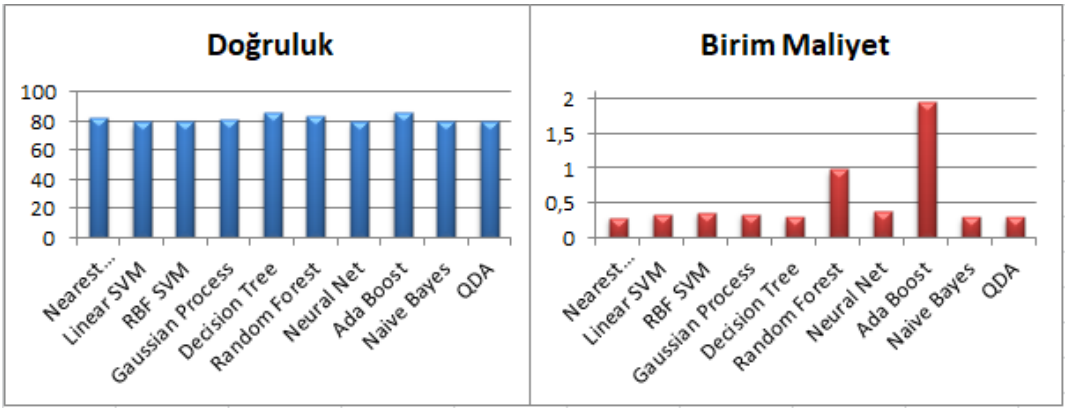
Şekil 6.68. NDstIP-max öznitelik çifti doğruluk ve birim maliyet oranları

NDstIP-max öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



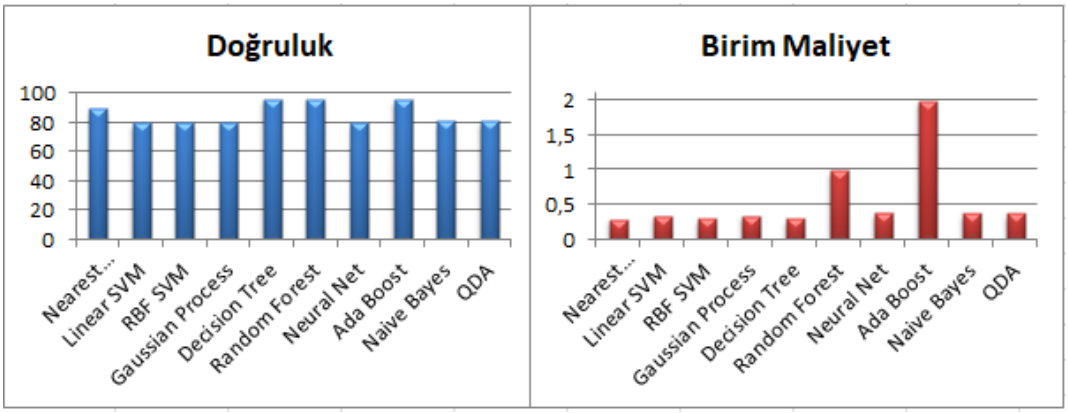
Şekil 6.69. Drate-srate öznitelik çifti doğruluk ve birim maliyet oranları

Drate-srate öznitelik çiftinde hem doğruluk oranında elde edilen sonuç hem de birim maliyette elde edilen sonuç dikkate alındığında; en iyi sonuçları veren metod Nearest Neighbors olmaktadır.



Şekil 6.70. Drate-max öznitelik çifti doğruluk ve birim maliyet oranları

Drate-max öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Ada Boost ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.



Şekil 6.71. Srate-max öznitelik çifti doğruluk ve birim maliyet oranları

Srate-max öznitelik çiftinde elde edilen en iyi doğruluk sonucunu Random Forest ile elde edilmesine rağmen; doğruluk ve birim maliyet birlikte dikkate alındığında son seçilen metod Decision Tree olmaktadır.

Aşağıdaki tabloda öznitelik çiftlerinin topluluk öğrenmesi sonuçları gösterilmiştir:

Tablo 6.13. Topluluk öğrenmesi sonuçları

Öznitelik Çifti	Son Seçilen Metod	Doğruluk Oranı	Birim Maliyet	İlk Seçilen Metod	Doğruluk Oranı	Birim Maliyet
sport-dport	Decision Tree	98,99	0,31	Ada Boost	99,19	1,94
sport-seq	Decision Tree	92,53	0,31	Ada Boost	94,45	1,9
sport-stddev	Decision Tree	88,29	0,31	Random Forest	88,4	0,93
sport-N_IN_Conn_P_SrcIP	Decision Tree	95,66	0,31	Ada Boost	95,86	1,99
sport-min_	Decision Tree	88,4	0,31	Random Forest	90,01	0,94
sport-state_number	Nearest Neighbors	93,44	0,29	Ada Boost	94,25	1,96
sport-mean	Nearest Neighbors	85,87	0,29	Random Forest	88,4	0,94
sport-N_IN_Conn_P_DstIP	Decision Tree	98,79	0,31	Random Forest	98,89	0,93
sport-drate	Decision Tree	84,16	0,31	Ada Boost	84,56	1,9
sport-srate	Decision Tree	94,25	0,31	Random Forest	94,35	0,91
sport-max_	Nearest Neighbors	86,07	0,29	Ada Boost	87,79	1,91
dport-seq	Decision Tree	98,79	0,31	Ada Boost	99,6	1,91
dport-stddev	Decision Tree	98,18	0,31	Decision Tree	98,18	0,31
dport-N_IN_Conn_P_SrcIP	Decision Tree	99,7	0,31	Decision Tree	99,7	0,31
dport-min_	Decision Tree	98,69	0,31	Decision Tree	98,69	0,31
dport-state_number	Decision Tree	99,19	0,31	Ada Boost	99,6	1,94
dport-mean	Decision Tree	99,6	0,31	Decision Tree	99,6	0,31
dport-N_IN_Conn_P_DstIP	Gaussian Process	99,6	0,35	Gaussian Process	99,6	0,35
dport-drate	Decision Tree	98,69	0,31	Ada Boost	98,79	1,94

Tablo 6.13. (Devamı)

sport-dport	Decision Tree	98,99	0,31	Ada Boost	99,19	1,94
sport-seq	Decision Tree	92,53	0,31	Ada Boost	94,45	1,9
sport-stddev	Decision Tree	88,29	0,31	Random Forest	88,4	0,93
sport-N_IN_Conn_P_SrcIP	Decision Tree	95,66	0,31	Ada Boost	95,86	1,99
sport-min_	Decision Tree	88,4	0,31	Random Forest	90,01	0,94
sport-state_number	Nearest Neighbors	93,44	0,29	Ada Boost	94,25	1,96
sport-mean	Nearest Neighbors	85,87	0,29	Random Forest	88,4	0,94
sport-N_IN_Conn_P_DstIP	Decision Tree	98,79	0,31	Random Forest	98,89	0,93
sport-drate	Decision Tree	84,16	0,31	Ada Boost	84,56	1,9
sport-srate	Decision Tree	94,25	0,31	Random Forest	94,35	0,91
sport-max_	Nearest Neighbors	86,07	0,29	Ada Boost	87,79	1,91
dport-seq	Decision Tree	98,79	0,31	Ada Boost	99,6	1,91
dport-stddev	Decision Tree	98,18	0,31	Decision Tree	98,18	0,31
dport-N_IN_Conn_P_SrcIP	Decision Tree	99,7	0,31	Decision Tree	99,7	0,31
dport-min_	Decision Tree	98,69	0,31	Decision Tree	98,69	0,31
dport-state_number	Decision Tree	99,19	0,31	Ada Boost	99,6	1,94
dport-mean	Decision Tree	99,6	0,31	Decision Tree	99,6	0,31
dport-N_IN_Conn_P_DstIP	Gaussian Process	99,6	0,35	Gaussian Process	99,6	0,35
dport-drate	Decision Tree	98,69	0,31	Ada Boost	98,79	1,94
seq-N_IN_Conn_P_DstIP	Decision Tree	99,7	0,31	Decision Tree	99,7	0,31
seq-drate	Nearest Neighbors	92,43	0,3	Nearest Neighbors	92,43	0,3
seq-srate	Decision Tree	97,98	0,31	Decision Tree	97,98	0,31
seq-max_	Decision Tree	95,46	0,31	Random Forest	96,37	0,95
stddev-N_IN_Conn_P_SrcIP	Decision Tree	97,88	0,31	Ada Boost	98,08	1,92
stddev-min_	Nearest Neighbors	89,71	0,29	Nearest Neighbors	89,71	0,29
stddev-state_number	Nearest Neighbors	98,49	0,29	Random Forest	98,89	0,91
stddev-mean	Nearest Neighbors	89,81	0,3	Nearest Neighbors	89,81	0,3
stddev-N_IN_Conn_P_DstIP	Nearest Neighbors	99,19	0,29	Ada Boost	99,29	1,9
stddev-drate	Gaussian Process	84,86	0,35	Ada Boost	84,96	1,91
stddev-srate	Decision Tree	96,27	0,31	Decision Tree	96,27	0,31
stddev-max_	Nearest Neighbors	90,01	0,29	Nearest Neighbors	90,01	0,29
N_IN_Conn_P_SrcIP-min_	Decision Tree	98,18	0,31	Decision Tree	98,18	0,31
N_IN_Conn_P_SrcIP-state_number	Nearest Neighbors	98,49	0,29	Nearest Neighbors	98,49	0,29
N_IN_Conn_P_SrcIP-mean	Nearest Neighbors	97,68	0,29	Ada Boost	97,78	1,9
N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP	Decision Tree	98,99	0,31	Ada Boost	99,09	1,91

Tablo 6.13. (Devamı)

N_IN_Conn_P_SrcIP-drate	Nearest Neighbors	96,17	0,3	Ada Boost	96,37	1,91
N_IN_Conn_P_SrcIP-srate	Nearest Neighbors	98,18	0,29	Nearest Neighbors	98,18	0,29
N_IN_Conn_P_SrcIP-max_	Nearest Neighbors	97,58	0,29	Nearest Neighbors	97,58	0,29
min_-state_number	Gaussian Process	92,84	0,37	Ada Boost	92,94	1,95
min_-mean	Nearest Neighbors	90,21	0,3	Nearest Neighbors	90,21	0,3
min_-N_IN_Conn_P_DstIP	Nearest Neighbors	99,09	0,29	Ada Boost	99,19	1,98
min_-drate	Nearest Neighbors	88,8	0,28	Nearest Neighbors	88,8	0,28
min_-srate	Decision Tree	93,64	0,31	Random Forest	96,47	0,97
min_-max_	Nearest Neighbors	89,91	0,3	Nearest Neighbors	89,91	0,3
state_number-mean	Gaussian Process	99,6	0,34	Gaussian Process	99,6	0,34
state_number-N_IN_Conn_P_DstIP	Nearest Neighbors	99,09	0,3	Ada Boost	99,29	1,93
state_number-drate	Gaussian Process	92,53	0,34	Gaussian Process	92,53	0,34
state_number-srate	Nearest Neighbors	99,29	0,29	Nearest Neighbors	99,29	0,29
state_number-max_	RBF SVM	99,29	0,35	RBF SVM	99,29	0,35
mean-N_IN_Conn_P_DstIP	Gaussian Process	99,5	0,36	Gaussian Process	99,5	0,36
mean-drate	Nearest Neighbors	84,86	0,29	Ada Boost	87,29	1,9
mean-srate	Decision Tree	96,17	0,31	Random Forest	96,27	0,94
mean-max_	Nearest Neighbors	89,91	0,29	Nearest Neighbors	89,91	0,29
N_IN_Conn_P_DstIP-drate	Nearest Neighbors	98,99	0,3	Nearest Neighbors	98,99	0,3
N_IN_Conn_P_DstIP-srate	Decision Tree	99,29	0,31	Random Forest	99,5	0,91
N_IN_Conn_P_DstIP-max_	Decision Tree	99,29	0,31	Ada Boost	99,5	2,02
drate-srate	Nearest Neighbors	95,86	0,29	Nearest Neighbors	95,86	0,29
drate-max_	Decision Tree	85,97	0,31	Ada Boost	86,68	1,96
srate-max_	Decision Tree	96,17	0,31	Random Forest	96,47	0,99

BÖLÜM 7. SONUÇLAR VE GELECEKTEKİ ÇALIŞMALAR

İnternet teknolojilerinde yaşanan gelişmeler ve hızla değişen trafik talepleri, yeni ağ güvenlik araçları ve desteği gerektirmektedir. Bu yüzden bu çalışmada geleneksel ağ yöntemi yerine YTA yöntemi baz alındı. Günümüzde kullanılan geleneksel ağ altyapısı bilişim dünyasında sürekli artan (büyük veri iletimi, hız, bulut bilişim, sanallaştırma) gereksinimlere cevap veremez hale geldiğinden OpenFlow protokolü temel alınarak YTA yöntemi kullanıldı.

Ayrıca çalışmamızda Bot-Iot veriseti kullanarak diğer verisetlerinin eksiklerinin giderilmesi amaçlanmıştır. Bu veri kümesi gerçekçi verilerden elde edilmiş ve en yeni versiyonu kullanılmıştır (2018 yılında laboratuvar ortamında elde edilen gerçek veriler kullanılmıştır). BoT-IoT veri seti UNSW Canberra Cyber Merkezinin Cyber Range Lab'ında gerçekçi bir ağ ortamı tasarlanarak üretilmiştir.

Son on yılda makine öğrenimi geleneksel yöntemlere göre, anomali tespitinde daha iyi doğruluk sonucu verdiği tespit edilmiştir. Geliştiriciler tarafından ağlarda çok sayıda anomali tabanlı saldırı tespit ve önleme tekniği geliştirilmiştir. Farklı saldırı senaryolarını tespit ederek ağlar üzerinden koruma sağlamak, herhangi bir geleneksel yöntemle zorlu bir iştir. Saldırı tespit sistemleri, veri alışverişi için daha güvenli bir ortam sağlamak amacıyla ağlara yapılan farklı saldırı türlerini analiz etmek ve belirlemek için ortaya çıkmıştır. Bu çalışmada en popüler on makine öğrenimi algoritmalarının(Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes ve QDA) çalışma sistemi açıklandı. Bu algoritmaların öznitelikleri, avantajları ve dezavantajları madde madde Bölüm 4'te açıklanmıştır.

Platformda kullanılacak programların avantajları tek tek açıklanmıştır. Bu çalışmada VirtualBox, POX, Python ve Mininet kullanılacaktır. Veriseti olarak UNSW Canberra Cyber Merkezinin Cyber Range Lab'ında gerçekçi bir ağ ortamı tasarlanarak üretilen Bot-Iot veriseti kullanılacaktır. Bu verisetinin diğer verisetlerinden farkları ve kullanılacak öznitelikler detaylı bir şekilde anlatılmıştır.

Bu çalışmada, en etkili ve optimal öznitelik çiftlerinin ve makine öğrenimi algoritmalarının belirlenmesine yardımcı olan yeni bir yaklaşım geliştirildi. 66 öznitelik çiftinin her bir öznitelik çifti ile 10 üzerinden her bir ML algoritmasını değerlendirildi. Bu amaçla 10 ML algoritması ve öznitelik çifti oluşturma modülünü içeren yeni bir sistem geliştirildi. Sistem belirli sayıda öznitelikten otomatik olarak öznitelik çiftleri oluşturdu ve oluşturulan öznitelik çiftleriyle kendini eğitti; daha sonra oluşturulan öznitelik çiftleri ile yöntemler eğitildiğinde 10 algoritmayı doğruluk ve birim maliyet açısından değerlendirmiştir. Bizim çalışmamızda en etkili 12 öznitelik kullanıldı. Sistem 66 benzersiz öznitelik çifti üretti ve her bir öznitelik çiftini belirli bir makine algoritması ile beslendi. Ardından, her bir anomali tespit algoritmasının doğruluğunu ve performansını değerlendirdi. Sonuç olarak, sport–N_IN_Conn_P_SrcIP, seq–stddev, seq–min ve sport–N_IN_Conn_P_DstIP dahil 37 öznitelik çiftinin birbirleriyle çok iyi dağıldıkları için tüm algoritmalarda çok iyi performans gösterdiğini belirlendi. Dört makine öğrenimi algoritması; Nearest Neighbor, Decision Trees, Random Forests and Ada-Boost % 95'in üzerinde yüksek bir doğruluk elde etti.

Birim maliyet dikkate alındığında Random Forest ve AdaBoost Algoritmaları oldukça maliyetli olduğu görülmektedir. Diğer algoritmalar ise bu algoritmalara göre daha hızlı çalışmaktadır. Hem doğruluk hem birim maliyet dikkate alındığında Nearest Neighbour, RBF SVM, Gaussion Process ve Decision Tree algoritmaları oldukça iyi sonuçlar vermektedir.

Tüm özniteliklerle elde edilen sonuçlara bakıldığında birim maliyetin oldukça fazla olduğu görülmektedir. Bu çalışmada öznitelik çiftleri ile elde edilen sonuçlara bakıldığında hem doğruluk oranı olarak oldukça yüksek sonuçlar elde edildi hem de

birim maliyet olarak kazanç sağlanıldı. Ortalama olarak %20 ile %30 arasında sistemin maliyeti azaltıldı.

Gelecekte ise yeni türetilen özniteliklerin başarım oranına etkileri ve saldırı var ise hangi türde (DoS,DDoS, Reconnaissance, Exploits) oldukları tespit edilecektir ve bunlara karşı önleme sistemi geliştirilecektir.

KAYNAKLAR

[1] Habeeb R.A.A., Nasaruddin F., Gani A., Hashem I.A.T., Ahmed E., Imran M. Real-time big data processing for anomaly detection: A survey Int. J. Inf. Manage., 45 (2019), pp. 289-307

[2] Muhammad Burhan, Rana Asif Rehman, Bilal Khan, and Byung Seo Kim. IoT elements, layered architectures and security issues: A comprehensive survey. Sensors (Switzerland), 18(9):1–37, 2018.

[3] Ibrahim Mashal, Osama Alsaryrah, Tein yaw Chung, Cheng-Zen Yang, Wen-Hsing Kuo, and Dharma Agrawal. Choices for interaction with things on internet and underlying issues. Ad Hoc Networks, 28:68–90, 01 2015.

[4] Miao Yun and Bu Yuxin. Research on the architecture and key technology of internet of things (IoT) applied on smart grid. In 2010 International Conference on Advances in Energy Engineering, pages 69–72, 2010.

[5] Omar Said and Mehedi Masud. Towards internet of things: Survey and future vision. International Journal of Computer Networks, 5:1–17, 02 2013.

[6] Ramaswamy Rajendran Somayya Madakam and Siddharth Tripathi. Internet of things (IoT): A literature review. Journal of Computer and Communications, 3:164–173, 05 2015.

[7] R. Khan, S. U. Khan, R. Zaheer, and S. Khan. Future internet: The internet of things architecture, possible applications and key challenges. In 2012 10th International Conference on Frontiers of Information Technology, pages 257–260, 2012.

[8] Mohamed Faisal Elrawy, Ali Ismail Awad, and Hesham F. A. Hamed. Intrusion detection systems for iot-based smart environments: a survey. Journal of Cloud Computing, 7(1):21, Dec 2018.

[9] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. IEEE Access, 8:104650–104675, 2020.

[10] R. Want. An introduction to rfid technology. IEEE Pervasive Computing, 5(1):25–33, 2006.

[11] R.Want. Near field communication. IEEE Pervasive Computing, 10(3):4– 7, 2011.

[12] P. McDermott-Wells. What is bluetooth? IEEE Potentials, 23(5):33–35, 2005.

[13] E. Ferro and F. Potorti. Bluetooth and wi-fi wireless protocols: a survey and a comparison. IEEE Wireless Communications, 12(1):12–26, 2005.

[14] G. V. Crosby and F. Vafa. Wireless sensor networks and lte-a network convergence. In 38th Annual IEEE Conference on Local Computer Networks, pages 731–734, 2013.

[15] W. Drira, ÃL'. Renault, and D. Zeghlache. Towards a secure social sensor network. In 2013 IEEE International Conference on Bioinformatics and Biomedicine, pages 24–29, 2013.

[16] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Generation Computer Systems, 100:779 – 796, 2019.

[17] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani. Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. IEEEWireless Communications, 24(3):10–16, 2017.

[18] S.M. Erfani, M. Baktashmotlagh, S. Rajasegarar, S. Karunasekera, C. Leckie, R1SVM: a randomised nonlinear approach to large-scale anomaly detection, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 432–438.

[19] Sarah M Erfani, Mahsa Baktashmotlagh, Sutharshan Rajasegarar, Vinh Nguyen, Christopher Leckie, James Bailey, and Kotagiri Ramamohanarao. R1STM: Oneclass Support Tensor Machine with Randomised Kernel. In Proceedings of SIAM International Conference on Data Mining (SDM), 2016.

[20] Sarah M. Erfani, Mahsa Baktashmotlagh, Masud Moshtahgi, Vinh Nguyen, Christopher Leckie, James Bailey, and Kotagiri Ramamohanarao. Robust Domain Generalisation by Enforcing Distribution Invariance. In Proceedings of International Joint Conference on Artificial Intelligence(IJCAI), 2016.

[21] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection : A Survey," ACM Computing Surveys, vol. 41, no. 3, pp. 15:1–15:58, September 2009.

[22] N. K. Ampah, C. M. Akujuobi, M. N. O. Sadiku, and S. Alam, "An intrusion detection technique based on continuous binary communication channels," International J. Security and Networks, vol. 6, no. 2/3, pp. 174–180, November 2011.

[23] Z. Cai, Z. Wang, K. Zheng, and J. Cao, “A Distributed TCAM coprocessor architecture for integrated longest prefix matching, policy filtering, and content filtering,” IEEE Transactions on Computers, vol. 62, no. 3, pp. 417–427, 2013.

[24] K. Zheng, Z. Cai, X. Zhang, Z. Wang, and B. Yang, “Algorithms to speedup pattern matching for network intrusion detection systems,” Computer Communications, vol. 62, pp. 47–58, 2015.

[25] Y Yu Y., Long J., Liu F., Cai Z. (2016) Machine Learning Combining with Visualization for Intrusion Detection: A Survey. In: Torra V., Narukawa Y., Navarro-Arribas G., Yañez C. (eds) Modeling Decisions for Artificial Intelligence. MDAI 2016.

[26] Close Farahnakian F., Heikkonen J. A deep auto-encoder based approach for intrusion detection system 2018 20th International Conference on Advanced Communication Technology, ICACT (2018).

[27] E. Özer and M. İskefiyeli, "Detection of DDoS attack via deep packet analysis in real time systems," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 1137-1140.

[28] Yeo M., Koo Y., Yoon Y., Hwang T., Ryu J., Song J., Park C. Flow-based malware detection using convolutional neural network 2018 International Conference on Information Networking, ICOIN (2018), pp. 910-913.

[29] Sun X., Dai J., Liu P., Singhal A., Yen J. Using Bayesian networks for probabilistic identification of zero-day attack paths IEEE Trans. Inf. Forensics Secur., 13 (10) (2018), pp. 2506-2521.

[30] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. Ddos in the iot: Mirai and other botnets. Computer, 50(7):80–84, 2017.

[31] S. Rathore, P. K. Sharma, V. Loia, Y.-S. Jeong, and J. H. Park, “Social network security: Issues, challenges, threats, and solutions,” Inf. Sci., vol. 421, pp. 43–69, 2017.

[32] K. Kaur et al., “Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay,” IEEE Commun. Mag., vol. 56, no. 2, pp. 44–51, Feb. 2018.

[33] D. He, S. Chan, X. Ni, and M. Guizani, “Software-defined-networkingenabled traffic anomaly detection and mitigation,” IEEE Internet Things J., vol. 4, no. 6, pp. 1890–1898, Dec. 2017.

[34] H. Peng, Z. Sun, X. Zhao, S. Tan, and Z. Sun, “A detection method for anomaly flow in software defined network,” IEEE Access, vol. 6, pp. 27 809–27 817, 2018.

[35] T. Ha et al., “Suspicious traffic sampling for intrusion detection in software-defined networks,” Comput. Netw., vol. 109, pp. 172–182, 2016.

[36] Mnar Saeed Alnaghes, Fayez Gebali, 2015, “A Survey on Some Currently Existing Intrusion Detection Systems for Mobile Ad Hoc Networks”, 2nd International Conference on Electrical and Electronics Engineering, Clean Energy and Green Computing (EEECEGC2015), pp. 12-18.

[37] Zamani, Mahdi. “Machine Learning Techniques for Intrusion Detection.” ArXiv abs/1312.2177 (2013).

[38] Huang G., Song S., Gupta J.N., Wu C. Semi-supervised and unsupervised extreme learning machines IEEE Trans. Cybern., 44 (12) (2014), pp. 2405-2417

[39] Gomez, Jonatan & Dasgupta, Dipankar. (2002). Evolving Fuzzy Classifiers for Intrusion Detection. Proceedings of The IEEE - PIEEE.

[40] M. Gharbaoui, F. Paolucci, A. Giorgetti, B. Martini, and P. Castoldi, “Effective statistical detection of smart confidentiality attacks in multi-domain networks,” IEEE Trans. Netw. Serv. Manag., vol. 10, no. 4, pp. 383–397, 2013.

[41] H. X. Nguyen and M. Roughan, “Multi-observer privacy-preserving hidden markov models,” IEEE Trans. Signal Process., vol. 61, no. 23, pp. 6010–6019, 2013.

[42] G. Fernandes, L. F. Carvalho, J. J. P. C. Rodrigues, and M. L. Proença, “Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization,” J. Netw. Comput. Appl., vol. 64, pp. 1–11, 2016.

[43] Lei Y. Network anomaly traffic detection algorithm based on SVM 2017 International Conference on Robots Intelligent System, ICRIS (2017), pp. 217-220, 10.1109/ICRIS.2017.61

[44] Tolles, Juliana & Meurer, William. (2016). Logistic Regression: Relating Patient Characteristics to Outcomes. JAMA. 316. 533. 10.1001/jama.2016.7653.

[45] Bruno Bogaz ZarpelÃco, Rodrigo Sanches Miani, ClÃaudio Toshio Kawakani, and Sean Carlisto de Alvarenga. A survey of intrusion detection in internet of things. Journal of Network and Computer Applications, 84:25 – 37, 2017.

[46] Robert Mitchell and Ing-Ray Chen. A survey of intrusion detection in wireless network applications. Computer Communications, 42:1 – 23, 2014.

[47] Mark D. Wilkinson et al. The fair guiding principles for scientific data management and stewardship. Scientific Data, 3(1):160018, Mar 2016.

[48] Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, and Andreas Hotho. A survey of network-based intrusion detection data sets. Computers and Security, 86:147 – 167, 2019.

[49] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Generation Computer Systems, 100:779 – 796, 2019.

[50] Singh, Ashutosh & Srivastava, Shashank. (2018). A survey and classification of controller placement problem in YTA. International Journal of Network Management. 28. e2018. 10.1002/nem.2018.

[51] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," in IEEE Communications Magazine, vol. 51, no. 11, pp. 24-31, November 2013, doi: 10.1109/MCOM.2013.6658648.

[52] SDN Architecture Overview, Open Networking Foundation December 12, 2013.

[53] Latah, Majd & Toker, Levent. (2016). Application of Artificial Intelligence to Software Defined Networking: A Survey. Indian Journal of Science and Technology. 9. 10.17485/ijst/2016/v9i44/89812.

[54] Nowsin Amin Sheikh, Mohammad. " YTA-Based Approach to Evaluate the Best Controller: Internal Controller NOX and External Controllers POX, ONOS, RYU." Global Journal of Computer Science and Technology [Online], (2019): n. pag. Web. 20 Oct. 2021

[55] J. H. Cox et al., "Advancing Software-Defined Networks: A Survey," in IEEE Access, vol. 5, pp. 25487-25526, 2017, doi: 10.1109/ACCESS.2017.2762291.

[56] M. Alsaeedi, M. M. Mohamad and A. A. Al-Roubaiey, "Toward Adaptive and Scalable OpenFlow-YTA Flow Control: A Survey," in IEEE Access, vol. 7, pp. 107346-107379, 2019, doi: 10.1109/ACCESS.2019.2932422

[57] Li, C, Wu, Y, Yuan, X, et al. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based YTA. Int J Commun Syst. 2018; 31:e3497. https://doi.org/10.1002/dac.3497

[58] A. Mondal, S. Misra and I. Maity, "Buffer Size Evaluation of OpenFlow Systems in Software-Defined Networks," in IEEE Systems Journal, vol. 13, no. 2, pp. 1359-1366, June 2019, doi: 10.1109/JSYST.2018.2820745.

[59] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo and T. Guo, "Trustworthy Network Anomaly Detection Based on an Adaptive Learning Rate and Momentum in IoT," in IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6182-6192, Sept. 2020, doi: 10.1109/TII.2020.2975227.

[60] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty and V. Sravan Kiran, "Similarity Based Feature Transformation for Network Anomaly Detection," in IEEE Access, vol. 8, pp. 39184-39196, 2020, doi: 10.1109/ACCESS.2020.2975716.

[61] Gomez, Jonatan & Dasgupta, Dipankar. (2002). Evolving Fuzzy Classifiers for Intrusion Detection. Proceedings of The IEEE - PIEEE.

[62] M. Gharbaoui, F. Paolucci, A. Giorgetti, B. Martini, and P. Castoldi, “Effective statistical detection of smart confidentiality attacks in multi-domain networks,” IEEE Trans. Netw. Serv. Manag., vol. 10, no. 4, pp. 383–397, 2013.

[63] H. X. Nguyen and M. Roughan, “Multi-observer privacy-preserving hidden markov models,” IEEE Trans. Signal Process., vol. 61, no. 23, pp. 6010–6019, 2013.

[64] Feiping Nie, Wei Zhu, Xuelong Li. Decision Tree SVM: An extension of linear SVM for non-linear classification. Neurocomputing, Volume 401, 2020, Pages 153-159, ISSN 0925-2312, https://doi.org/10.1016/ j.neucom.2019.10.051.

[65] Waqas, Ahmad & Gilal, Abdul & Bhatti, Zeeshan & Mahesar, Abdul. (2013). Investigating ANNs and Applications. Indian Journal of Automation and Artificial Intelligence. 1. 65-69.

[66] Jian Zhou, Panagiotis G. Asteris, Danial Jahed Armaghani, Binh Thai Pham, Prediction of ground vibration induced by blasting operations through the use of the Bayesian Network and random forest models, Soil Dynamics and Earthquake Engineering, Volume 139, 2020, 106390, ISSN 0267-7261, https://doi.org/10.1016/j.soildyn.2020.106390.

[67] VMware vs. VirtualBox: Which is Better for Desktop Virtualization? https://tech nologyadvice .com/blog/information-technology/vmware-vs-virtualbox/

[68] Python vs Matlab http://www.pyzo.org/python_vs_matlab.html#matlab-and-python-and-their-ecosystems

[69] L Majd, T Levent, “Application of Artificial Intelligence to Software Defined Networking: A Survey”, Indian Journal of Science and Technology

[70] POX https://searchsdn.techtarget.com/definition/POX

[71] S. T. Brugger, J. Chow, An assessment of the darpa ids evaluation dataset using snort, UCDAVIS department of Computer Science 1 (2007) (2007) 22.

[72] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in: Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on, IEEE, 2009, pp. 1{6.

[73] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion tra_c characterization, in: Proceedings of fourth international conference on information systems security and privacy, ICISSP, 2018.

[74] Unibs, university of brescia dataset (2009). http://www.ing.unibs.it/ntw/tools/traces/

[75] M. H. Bhuyan, D. K. Bhattacharyya, J. K. Kalita, Towards generating reallife datasets for network intrusion detection. IJ Network Security 17 (6) (2015) 683{701.

[76] Lawrence berkley national laboratory (lbnl), icsi, lbnl/icsi enterprise tracing project (2005). http://www.icir.org/enterprise-tracing/

[77] U. o. n. B. Canadian Institute of Cybersecurity, Iscx dataset. http://www.unb.ca/cic/datasets/index.html

[78] P. Gogoi, M. H. Bhuyan, D. Bhattacharyya, J. K. Kalita, Packet and flow based network intrusion dataset, in: International Conference on Contemporary Computing, Springer, 2012, pp. 322{334.

[79] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 12(85):2825–2830, 2011.

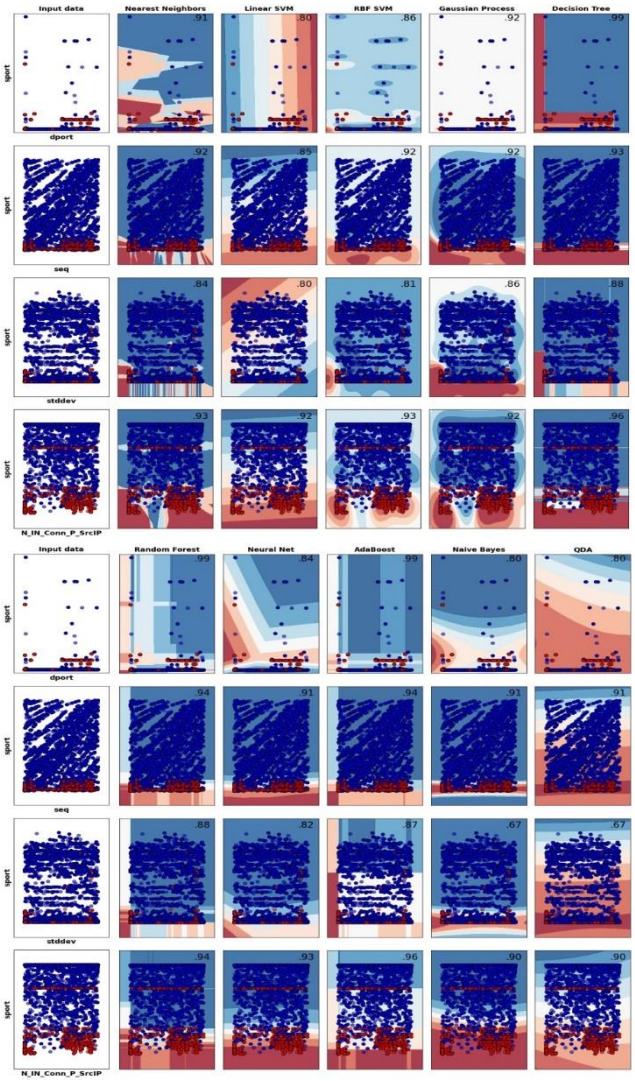
[80] S. van der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. Computing in Science Engineering, 13(2):22–30, 2011.

[81] Wes McKinney. Pandas: a foundational python library for data analysis and statistics. 2011.

[82] Paul Barrett, J. Hunter, J.T. Miller, J.-C Hsu, and P. Greenfield. matplotlib – a portable python plotting package. 12 2005.

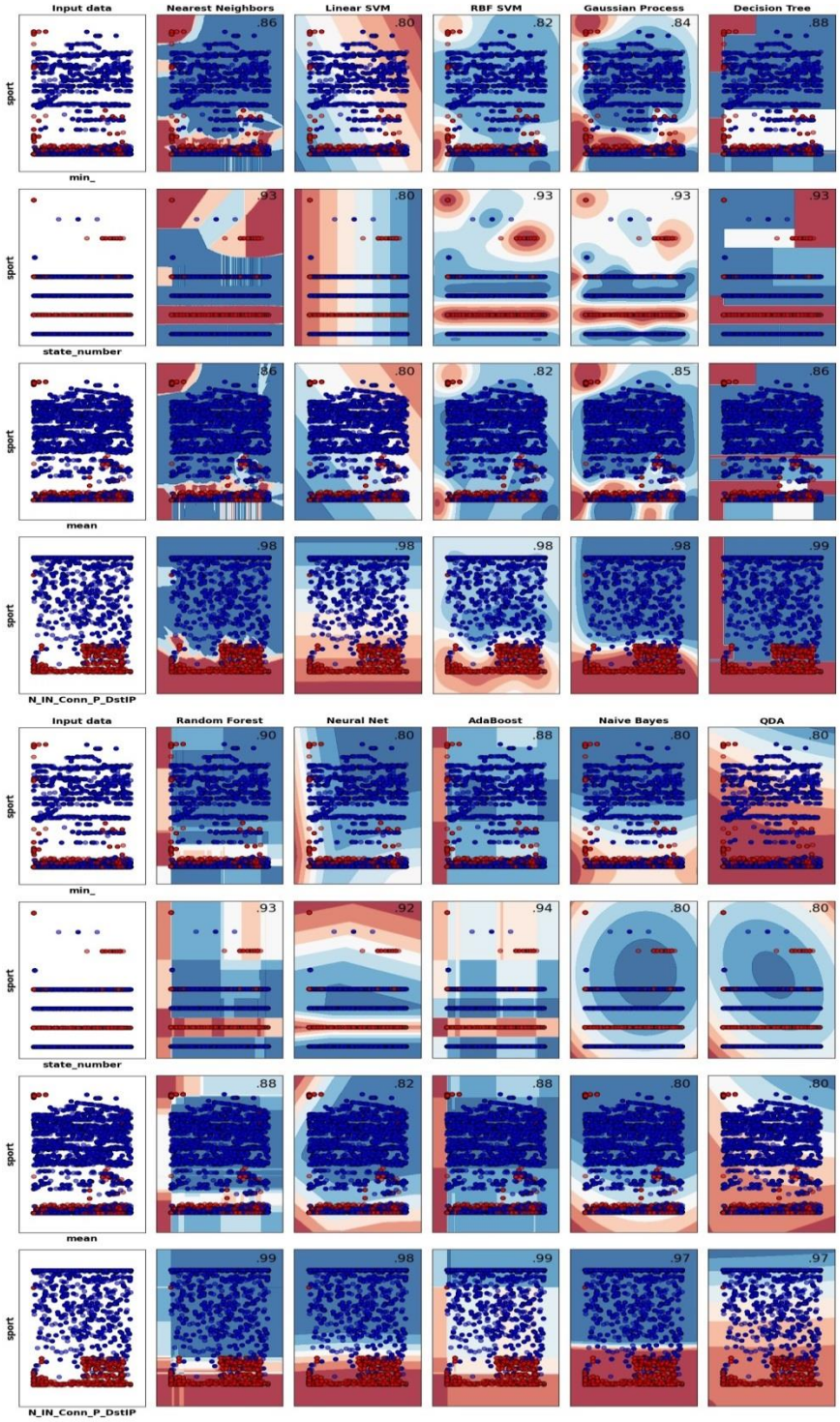
EKLER

EK A: Sonuçlar



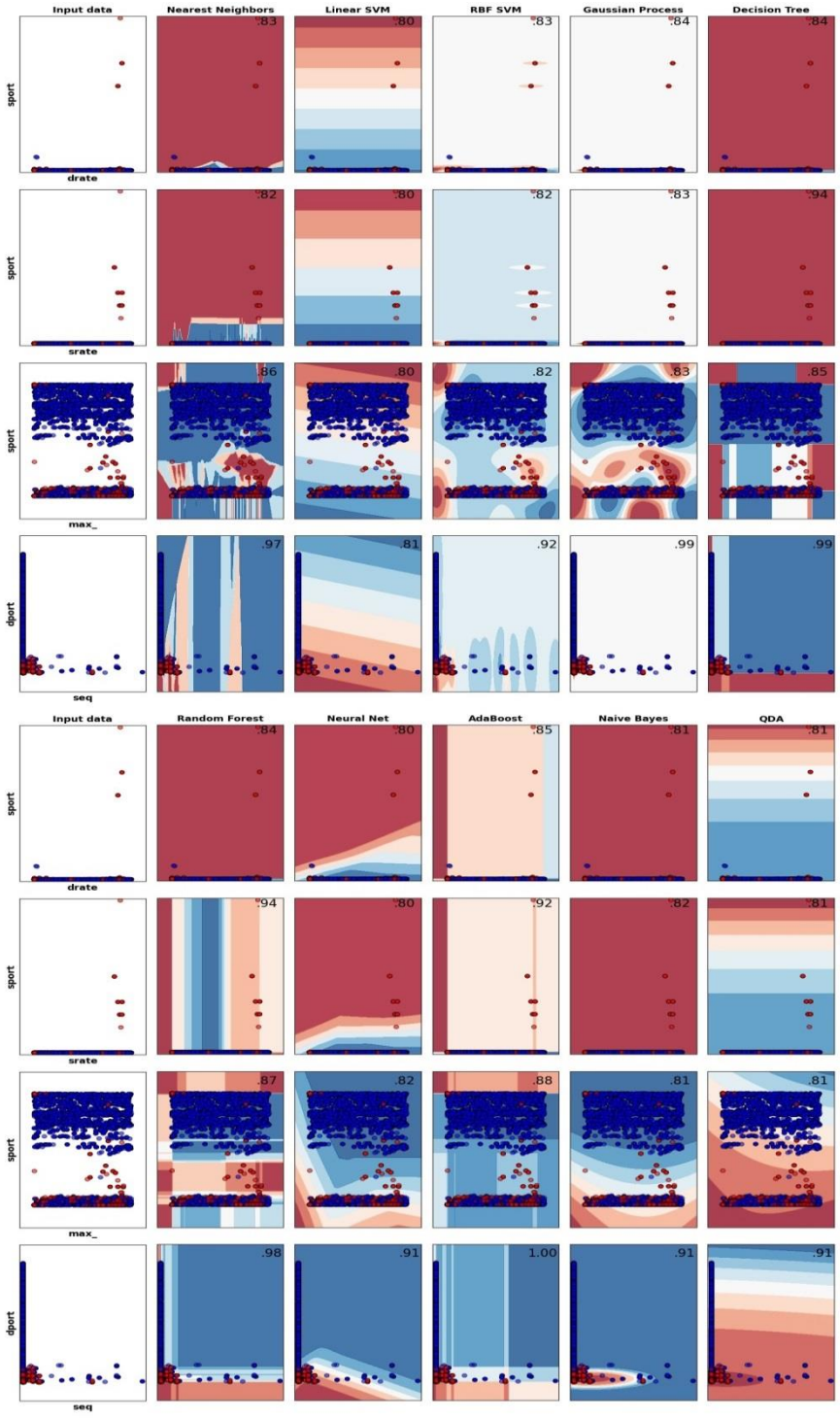
Şekil Ek A.1. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde sport-dport, sport-seq, sport-stddev, sport-N_IN_Conn_P_SrcIP sonuçları elde edilmiştir.



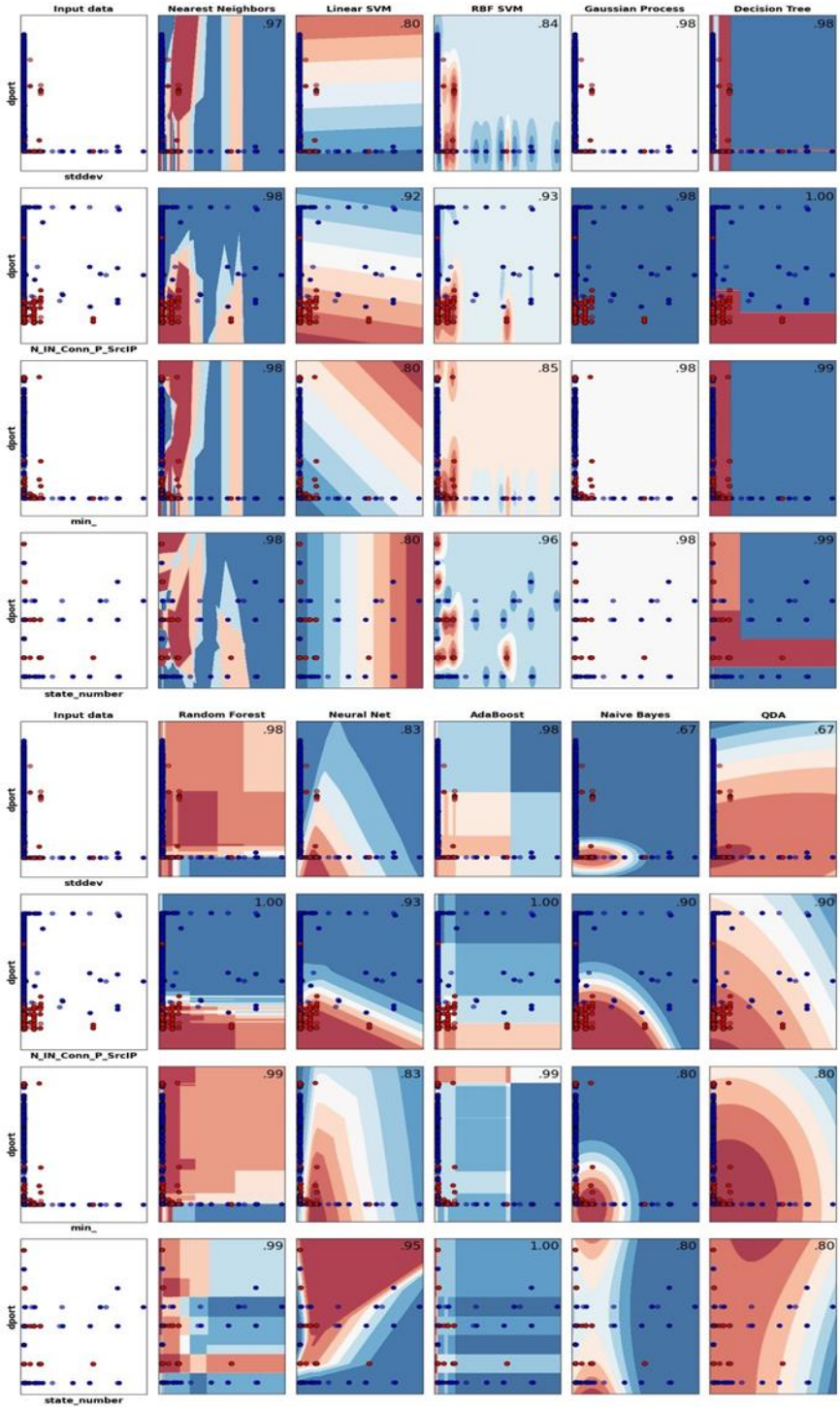
Şekil Ek A.2. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde sport-min, sport-state_number, sport-mean, sport-N_IN_Conn_P_Dst_IP sonuçları elde edilmiştir.



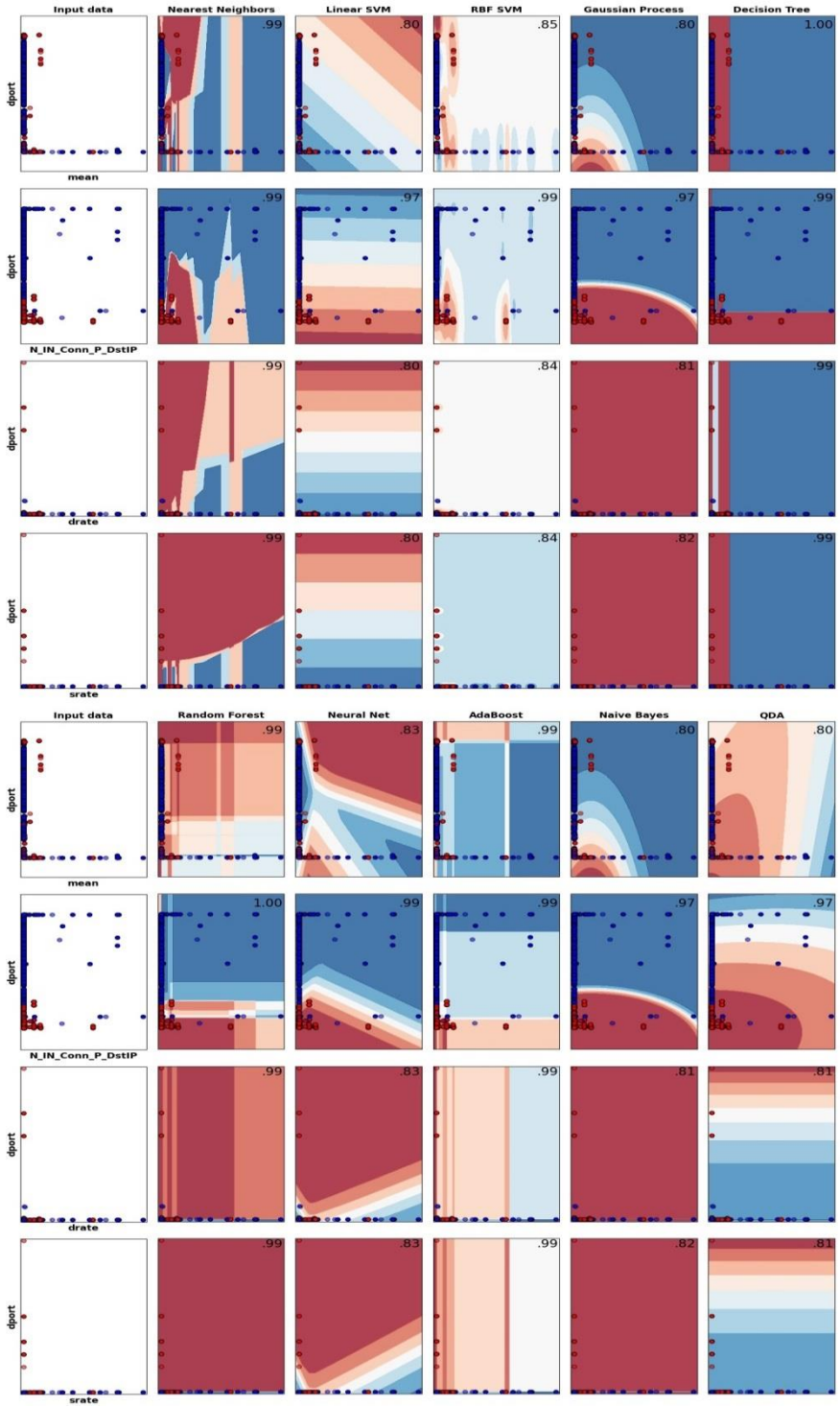
Şekil Ek A.3. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde sport-drate, sport-srate, sport-max, dport-seq sonuçları elde edilmiştir.



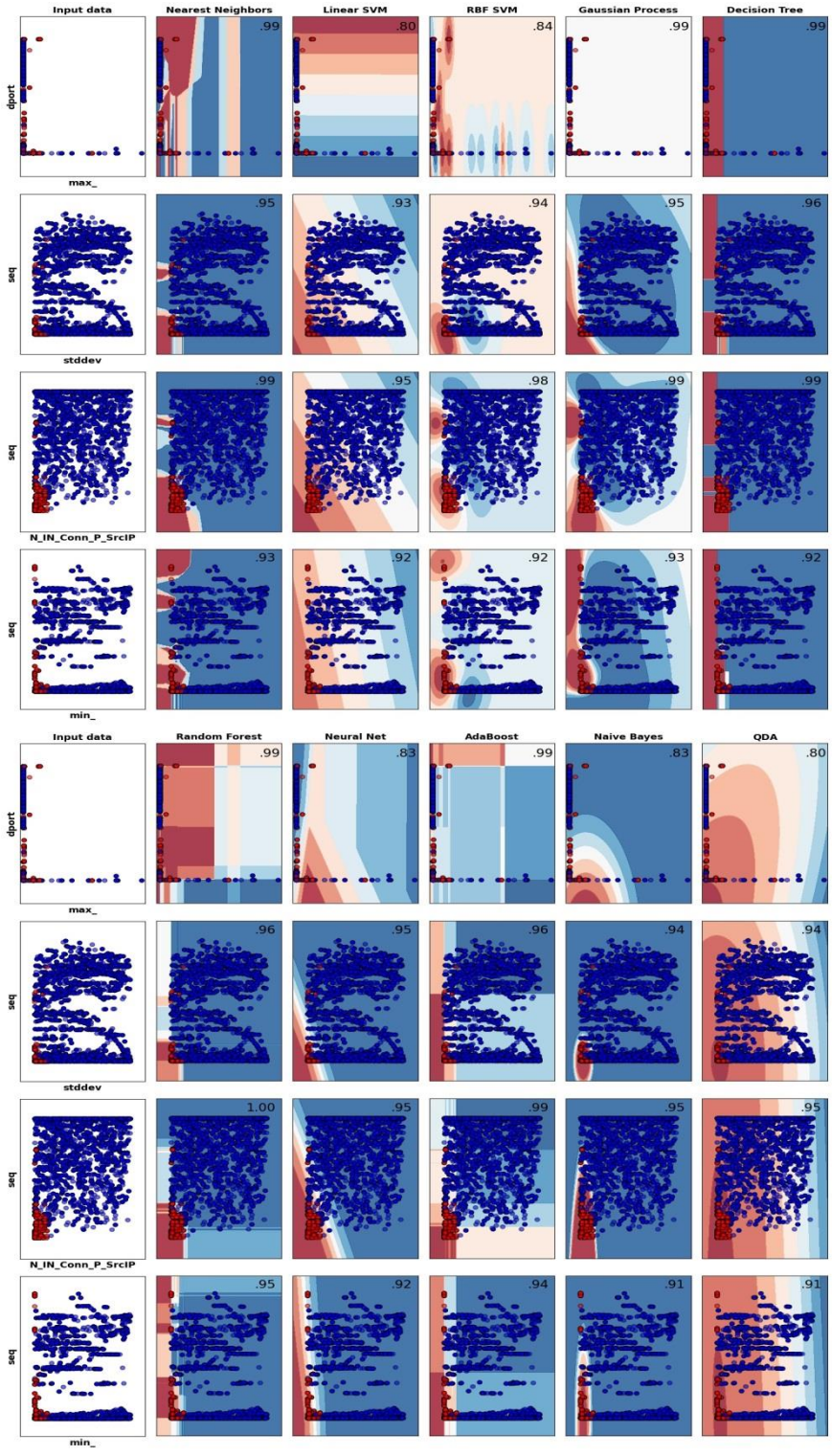
Şekil Ek A.4. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde dport-stddev, dport- N_IN_Conn_P_SrcIP, dport-min, dport-state_number sonuçları elde edilmiştir.



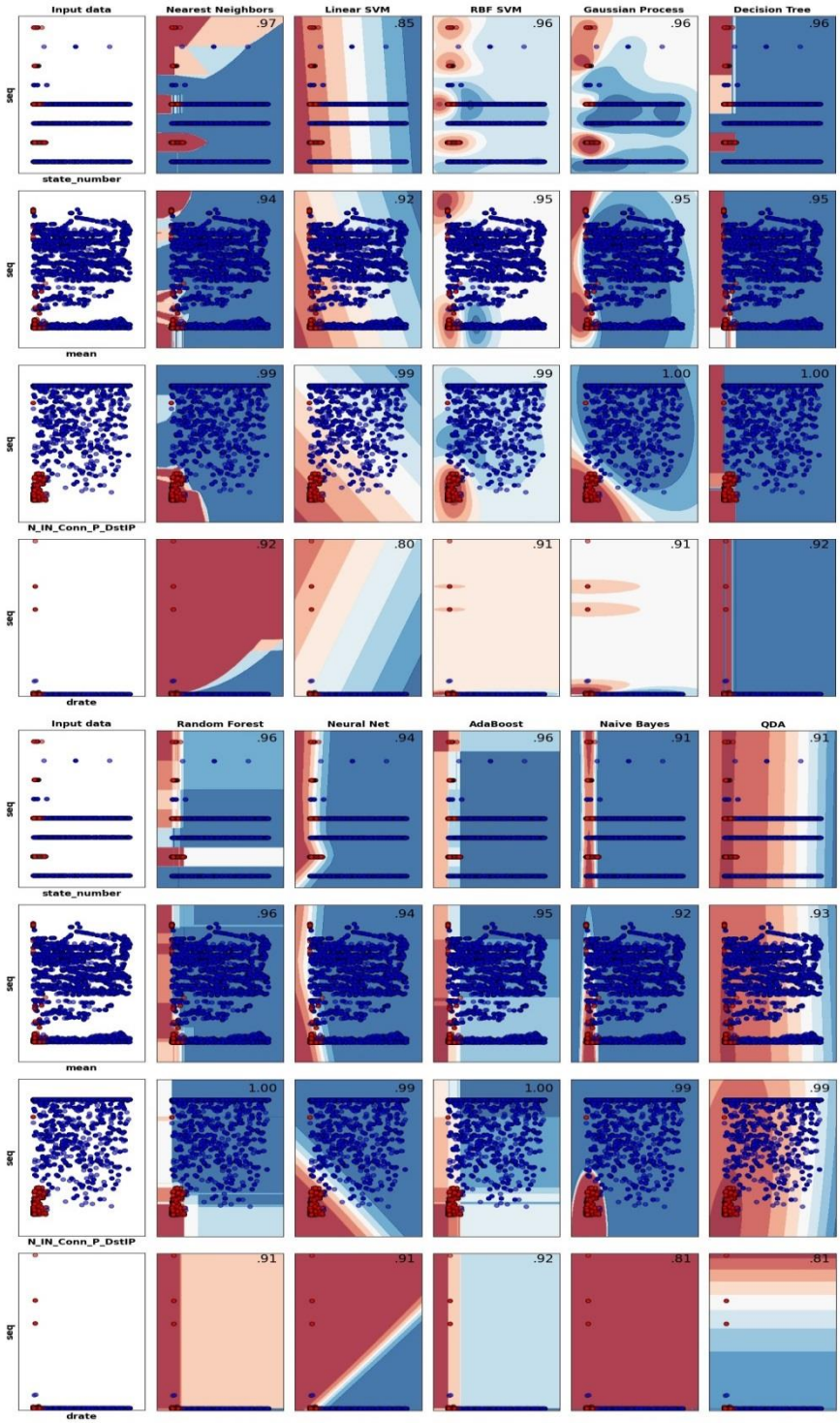
Şekil Ek A.5. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde dport-mean, dport- N_IN_Conn_P_SrcIP, dport-drate, dport-srate sonuçları elde edilmiştir.



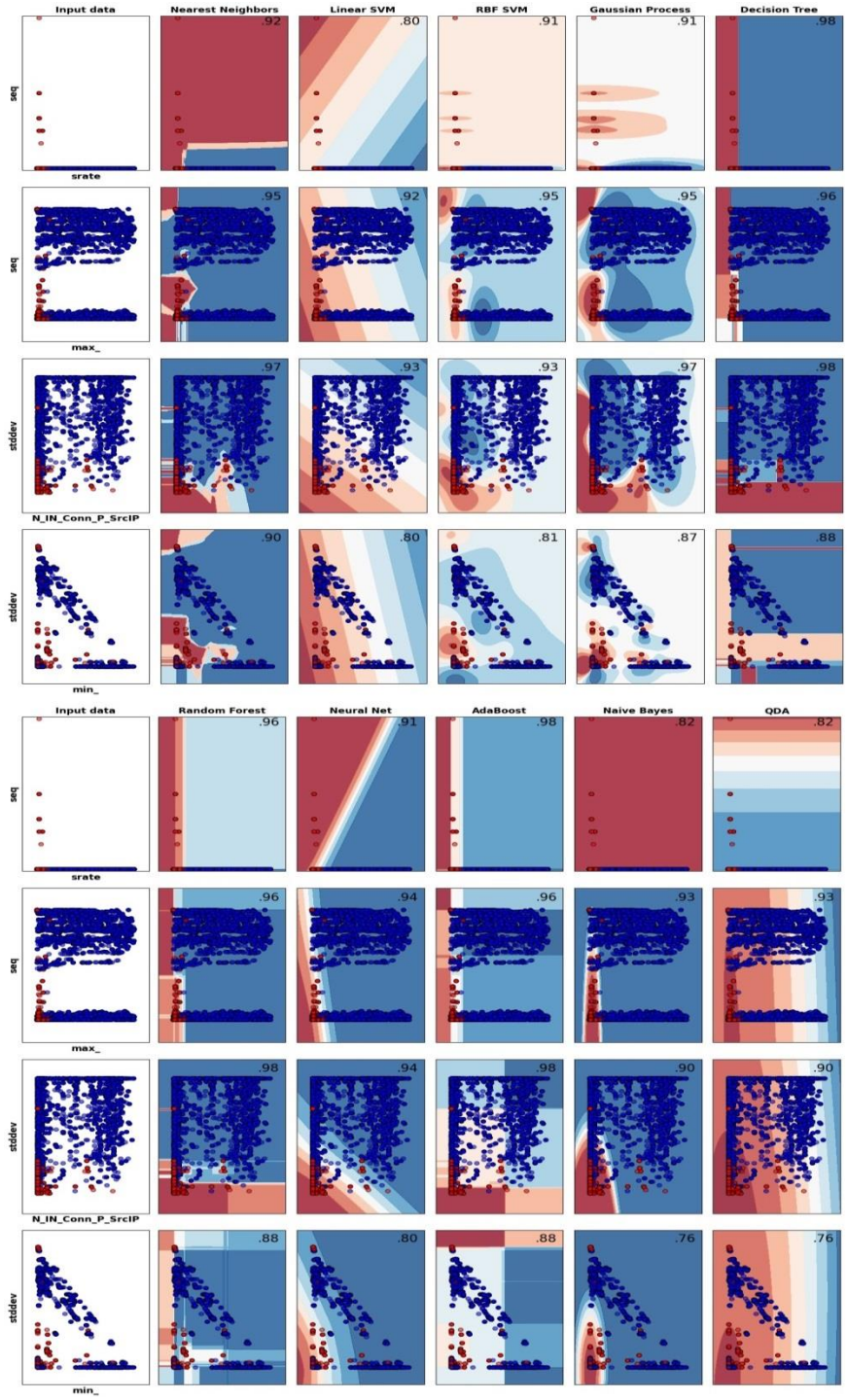
Şekil Ek A.6. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde dport-max, seq-stddev, seq- N_IN_Conn_P_SrcIP, seq-min sonuçları elde edilmiştir.



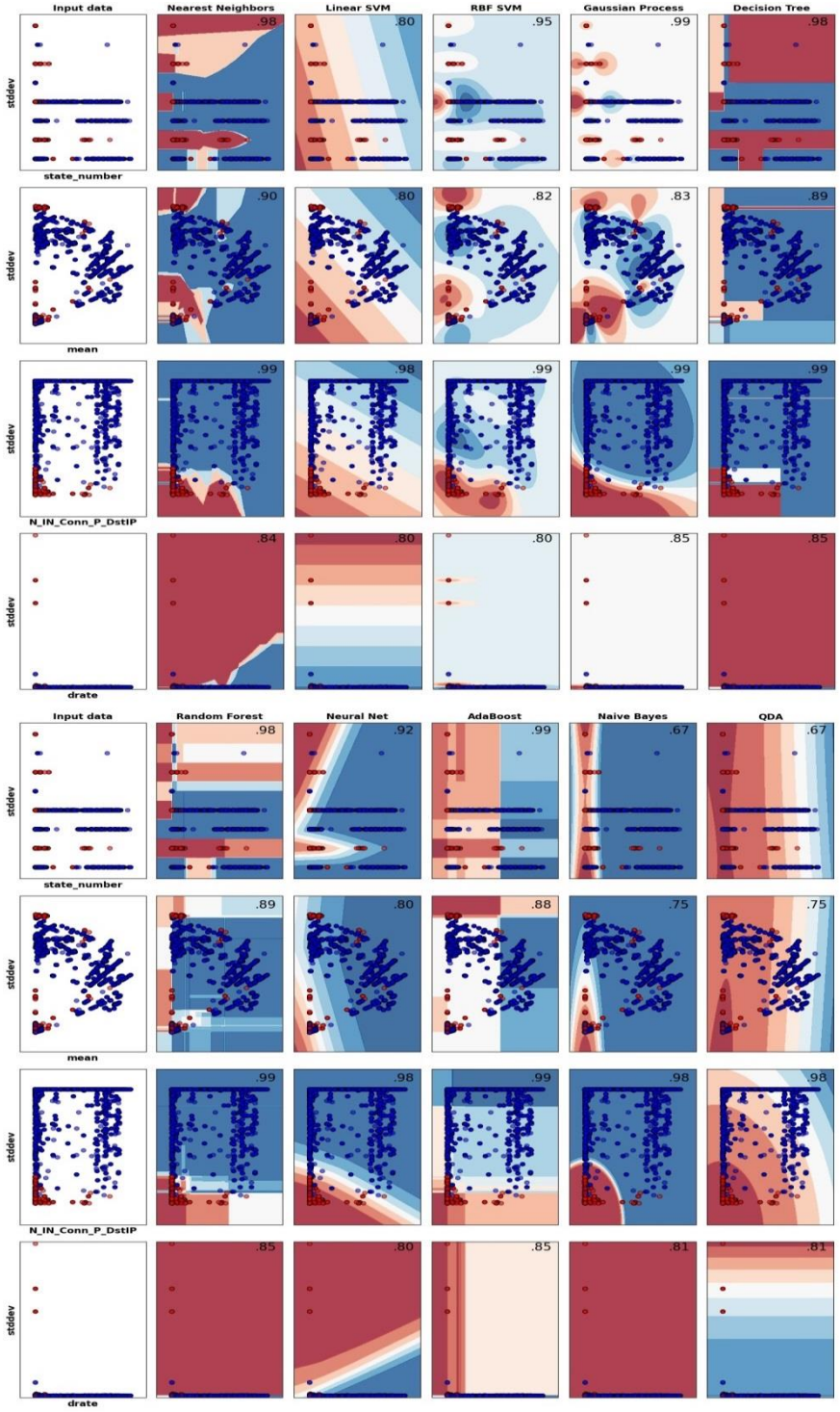
Şekil Ek A.7. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde seq-state_number, seq-mean, seq- N_IN_Conn_P_DstIP, seq-drate sonuçları elde edilmiştir.



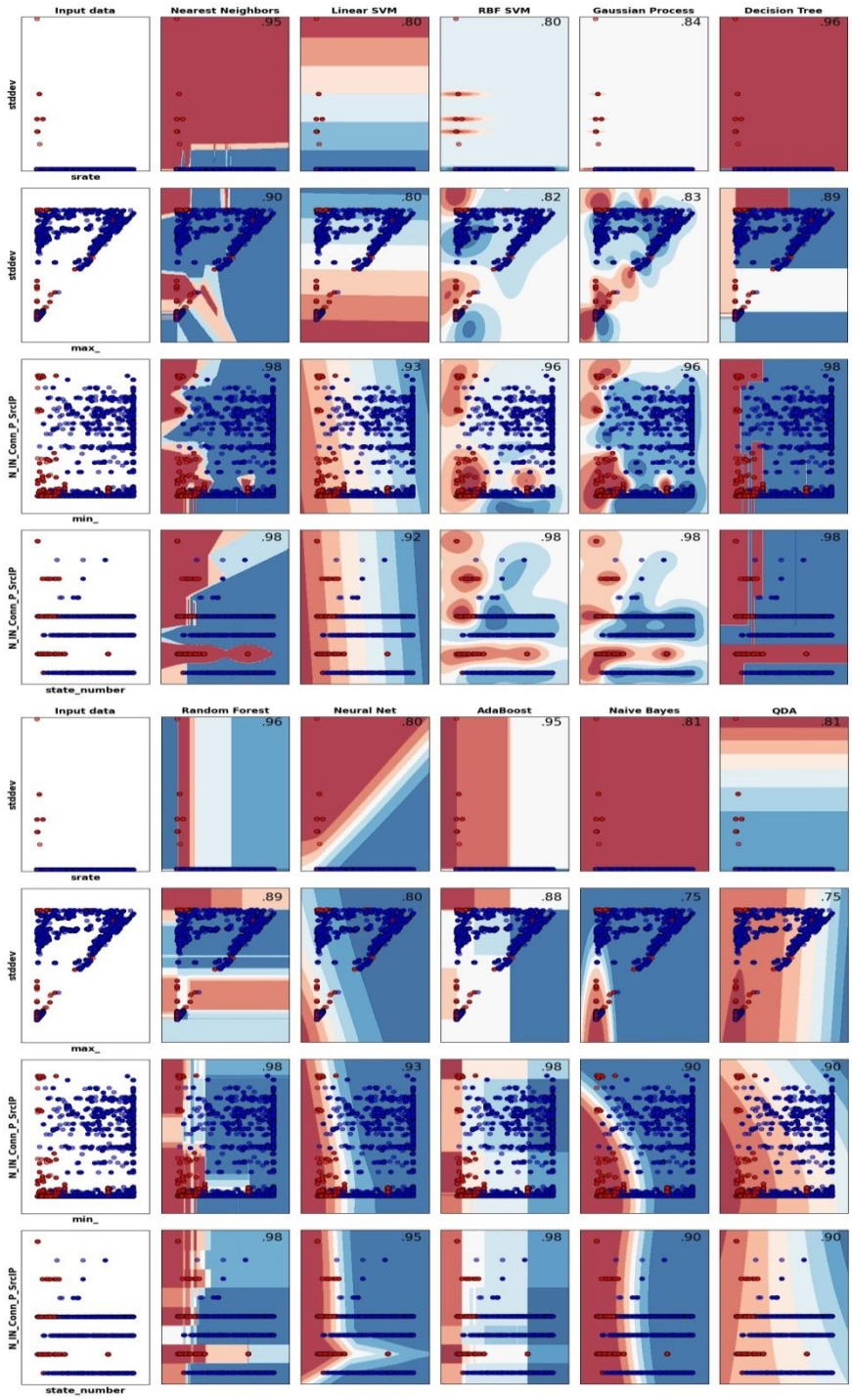
Şekil Ek A.8. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde seq-srate, seq-max, stddev- N_IN_Conn_P_SrcIP, stddev-min sonuçları elde edilmiştir.



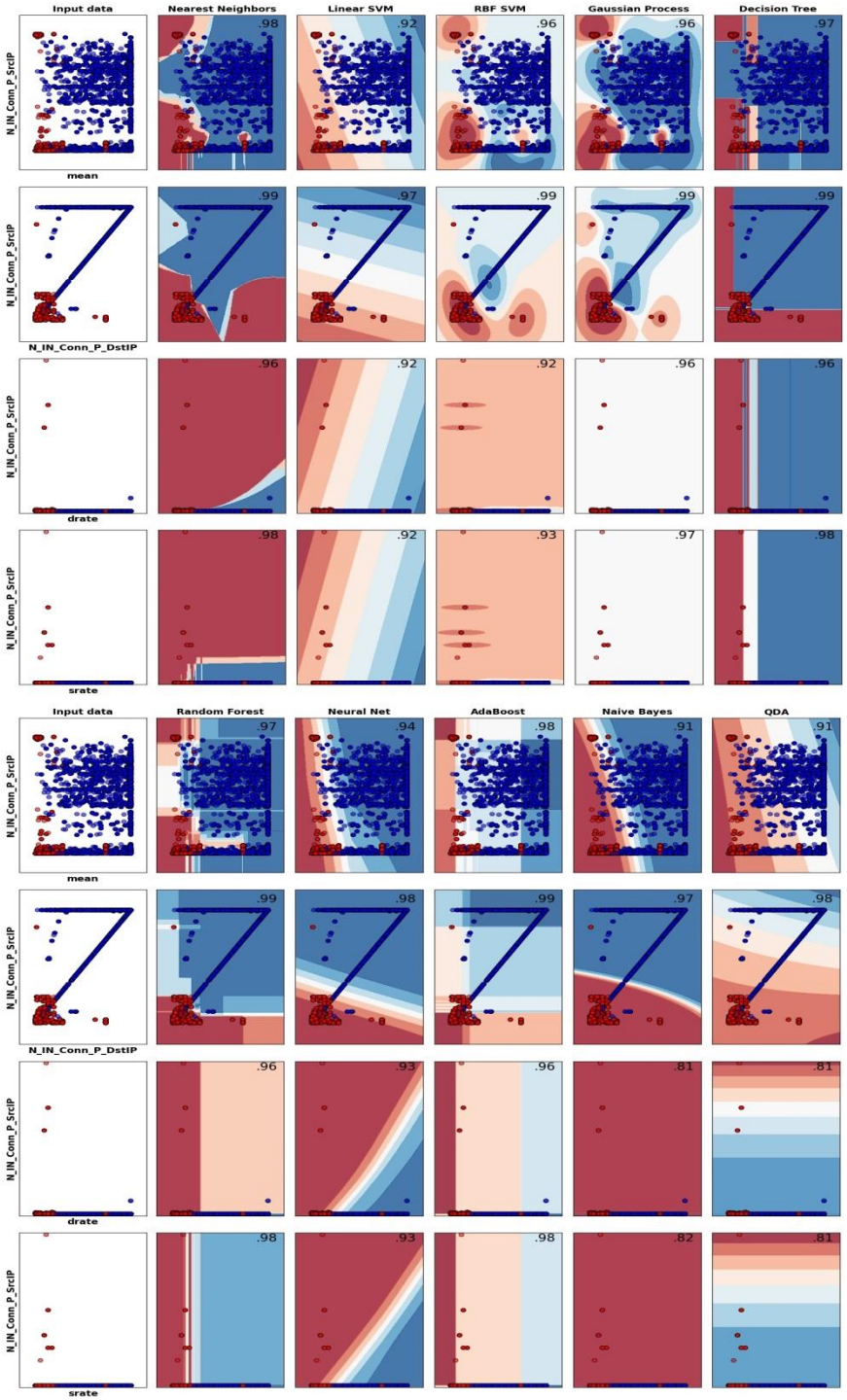
Şekil Ek A.9. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde stddev-state_number, stddev-mean, stddev- N_IN_Conn_P_DstIP, stddev-drate sonuçları elde edilmiştir.



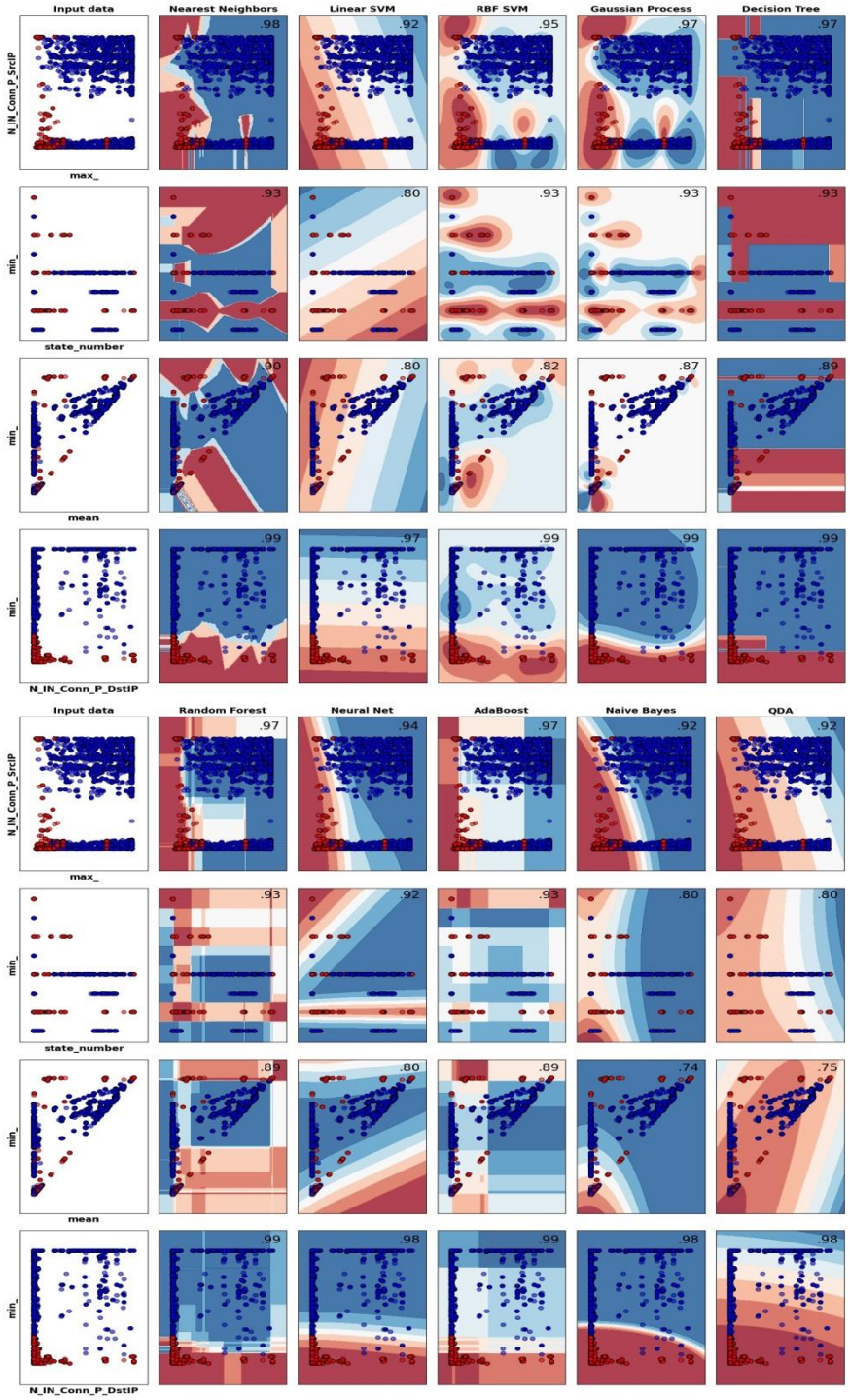
Şekil Ek A.10. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde stddev-srate, stddev-max, N_IN_Conn_P_SrcIP-min, N_IN_Conn_P_SrcIP-state_number sonuçları elde edilmiştir.



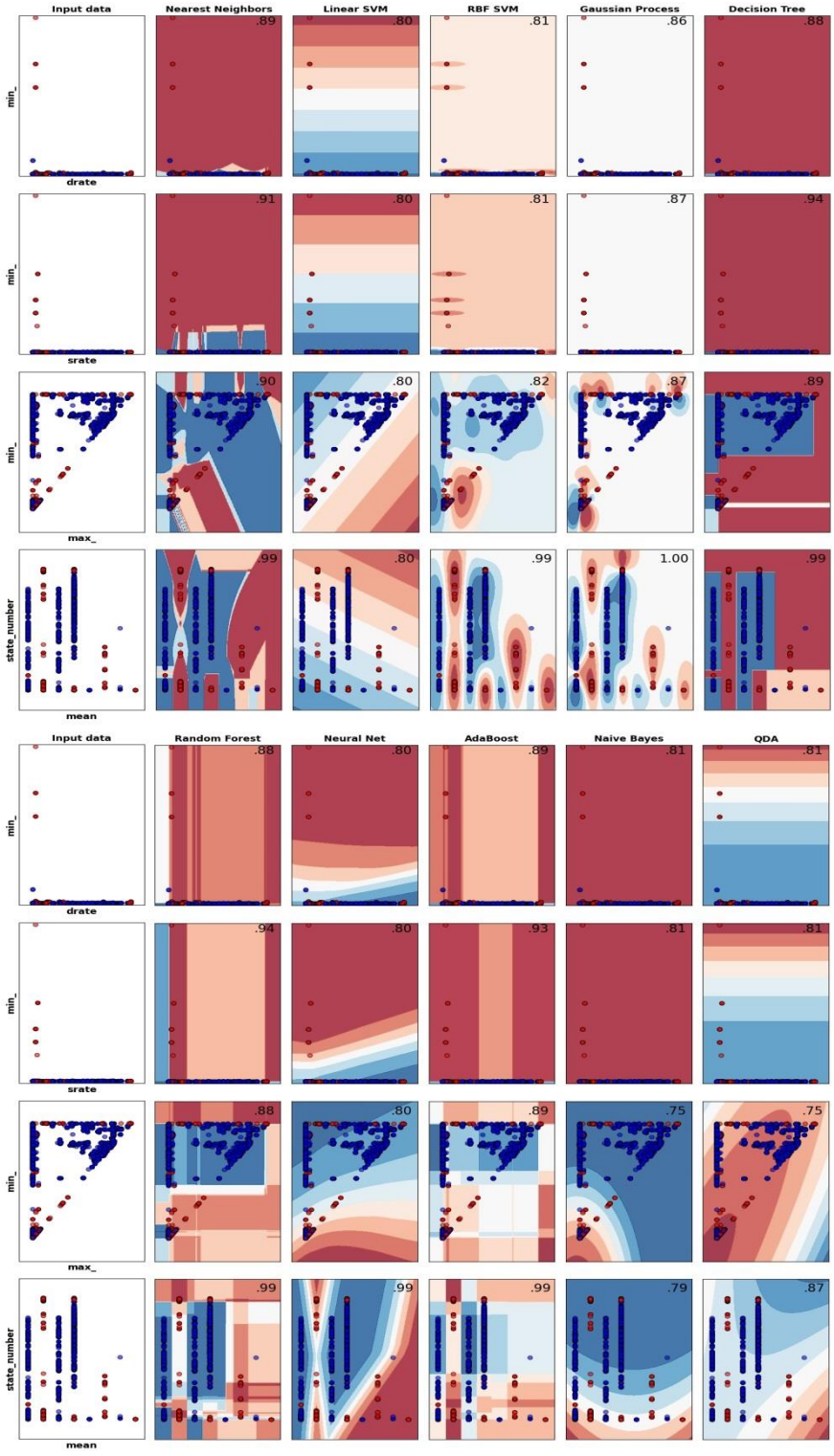
Şekil Ek A.11. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde N_IN_Conn_P_SrcIP-mean, N_IN_Conn_P_SrcIP-N_IN_Conn_P_DstIP, N_IN_Conn_P_SrcIP-drate, N_IN_Conn_P_SrcIP-srate sonuçları elde edilmiştir.



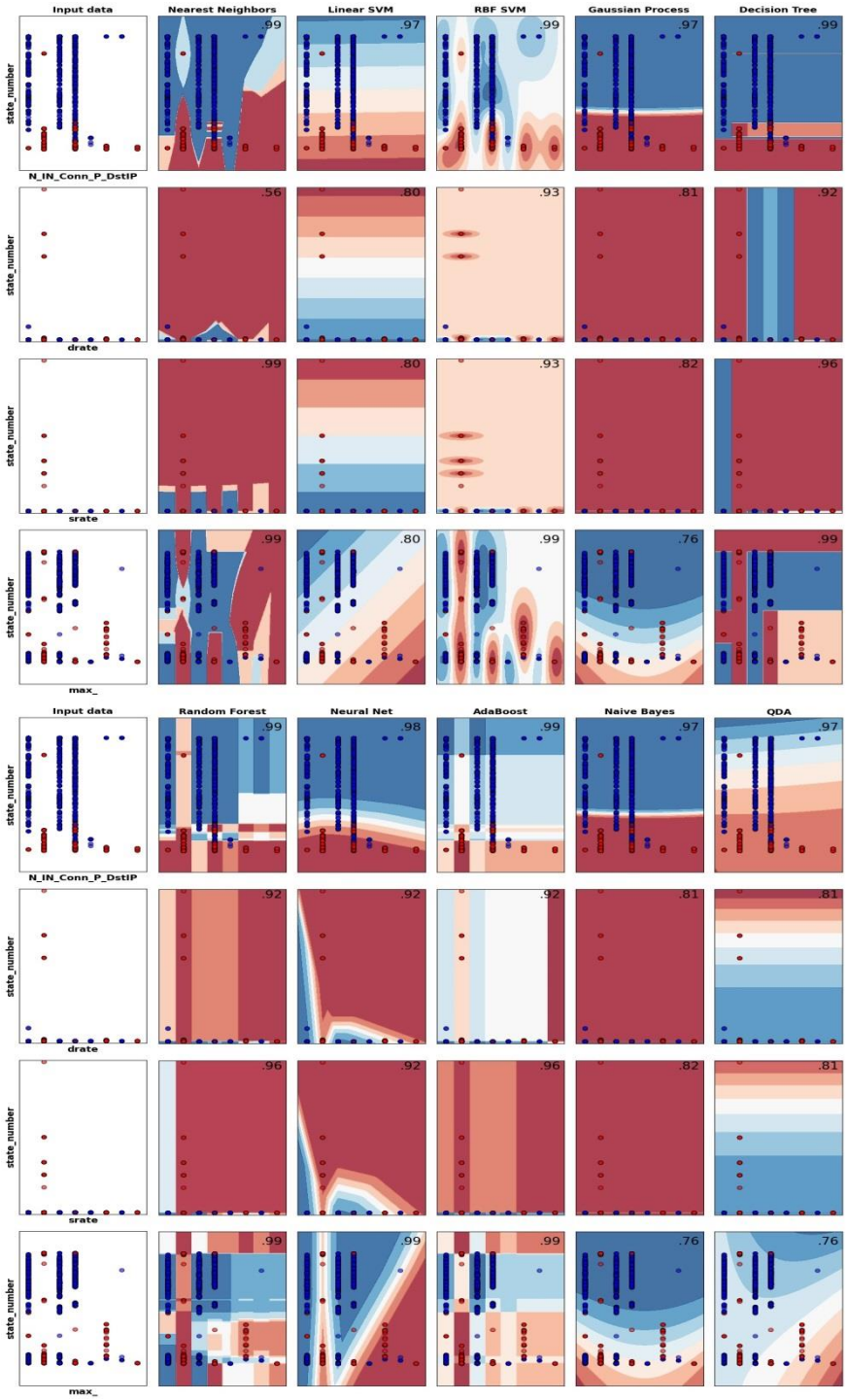
Şekil Ek A.12. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde N_IN_Conn_P_SrcIP-max, min-state-number, min-mean, min-N_IN_Conn_P_DstIP sonuçları elde edilmiştir.



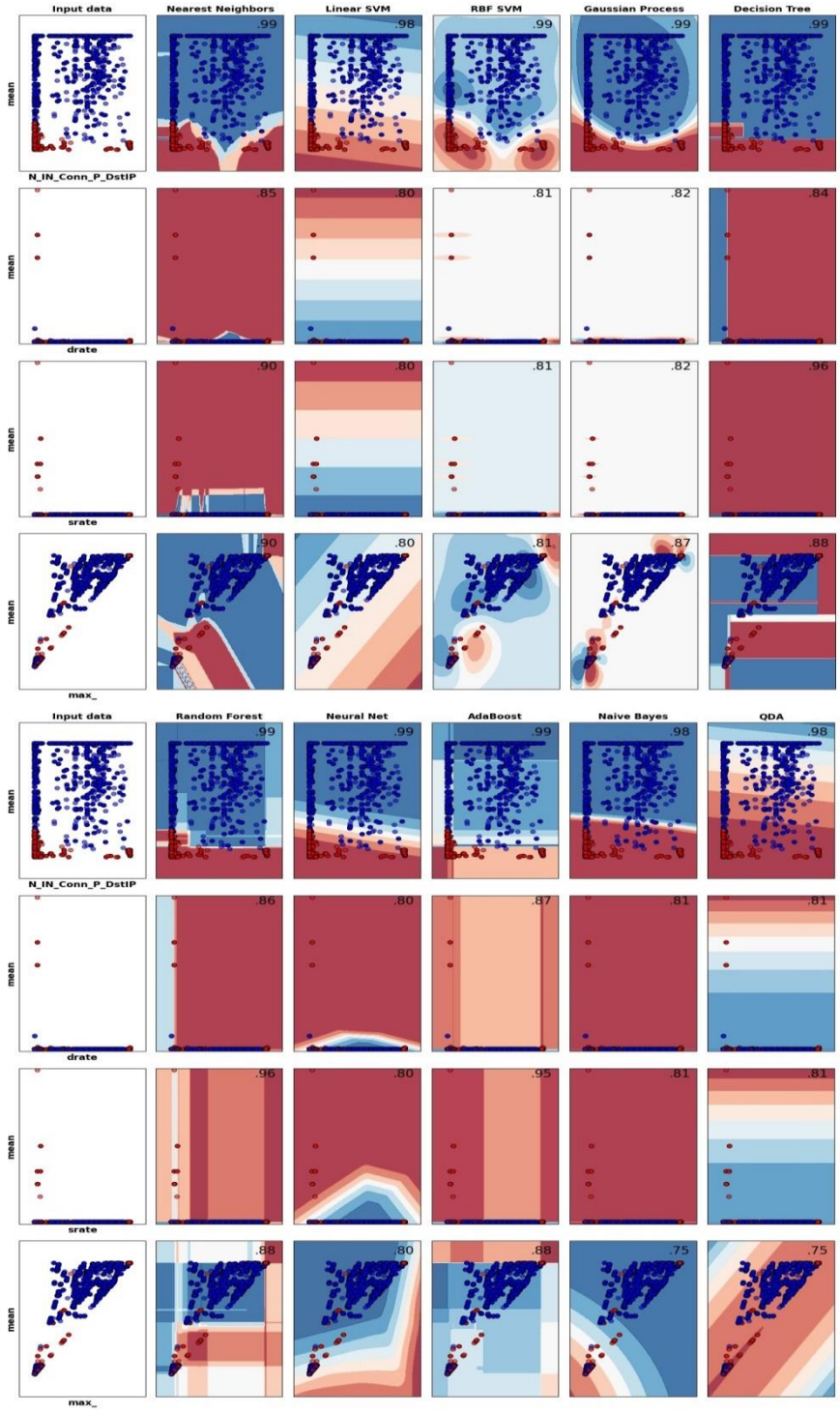
Şekil Ek A.13. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde min-drate, min-srate, min-max, state_number-mean sonuçları elde edilmiştir.



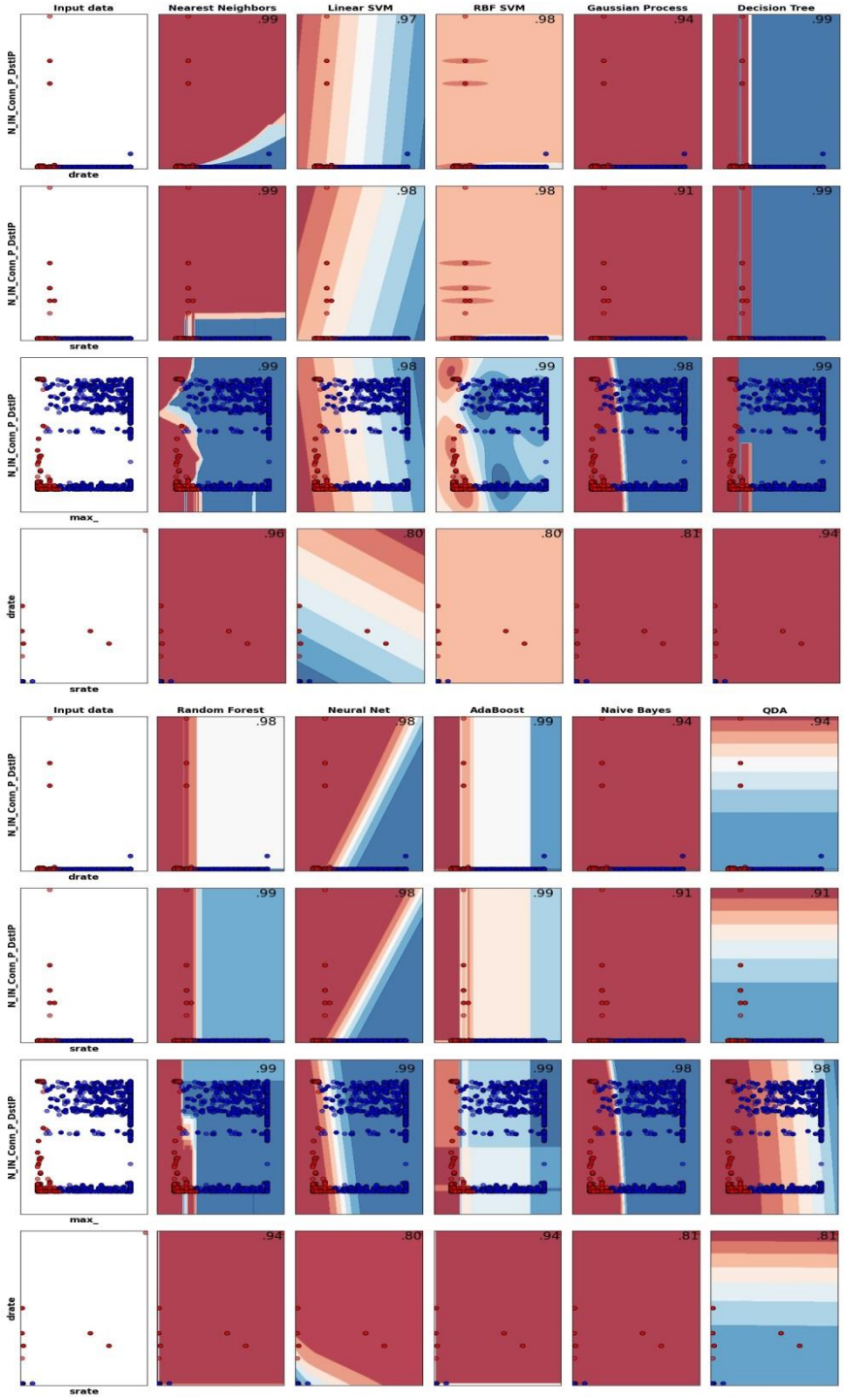
Şekil Ek A.14. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde state_number-N_IN_Conn_P_DstIP, state_number-drate, state_number-srate, state_number-max sonuçları elde edilmiştir.



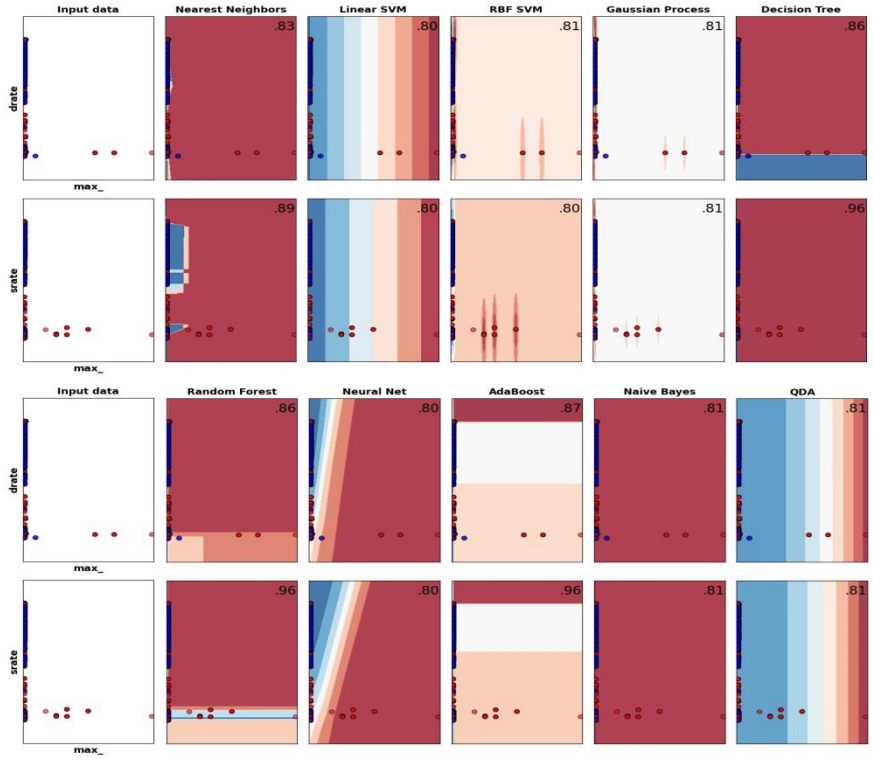
Şekil Ek A.15. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde mean-N_IN_Conn_P_DstIP, mean-drate, mean-srate, mean-max sonuçları elde edilmiştir.



Şekil Ek A.16. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde N_IN_Conn_P_DstIP-drate, N_IN_Conn_P_DstIP-srate, N_IN_Conn_P_DstIP-max, drate-srate sonuçları elde edilmiştir.



Şekil Ek A.17. Sonuçlar

Resimdeki şekilde öznitelik çiftleriyle makine öğrenmesi algoritmalarının eğitilmesiyle elde edilen doğruluk sonuçları gösterilmiştir. Bu şekilde drate-max, srate-max sonuçları elde edilmiştir.

ÖZGEÇMİŞ

Adı Soyadı **:** Erman ÖZER

ÖĞRENİM DURUMU

Derece	**Eğitim Birimi**	**Mezuniyet Yılı**
Doktora	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü/Bilgisayar ve Bilişim Mühendisliği	Devam ediyor
Yüksek Lisans	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü/Bilgisayar ve Bilişim Mühendisliği	2016
Lisans	Süleyman Demirel Üniversitesi/Mühendislik Fakültesi/Bilgisayar Mühendisliği	2012

İŞ DENEYİMİ

Yıl	**Yer**	**Görev**
2013 Şubat(1 ay)	Recep Tayyip Erdoğan Üniversitesi	Araştırma Görevlisi
2013-Halen	Sakarya Üniversitesi	Araştırma Görevlisi

ESERLER (makale, bildiri, proje vb.)

1. Özer, E., İskefiyeli, M., & Azimjonov, J. 2021,'Toward lightweight intrusion detection systems using the optimal and efficient feature pairs of the Bot-IoT 2018 dataset', International Journal of Distributed Sensor Networks, Vol.17, No.10, pp.15501477211052202.

2. E. Özer and M. İskefiyeli, "Detection of DDoS attack via deep packet analysis in real time systems," 2017 International Conference on Computer Science and Engineering (UBMK), 2017, pp. 1137-1140, doi: 10.1109/UBMK.2017.8093526.