T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SİBER GÜVENLİK ANA BİLİM DALI
SİBER GÜVENLİK YÜKSEK LİSANSI

ANOMALİ TABANLI AĞ SALDIRI TESPİTİ

DÖNEM PROJESİ

Onur Fırat ÖZTÜRK

İSTANBUL 2023

T.C.
MARMARA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
SİBER GÜVENLİK ANA BİLİM DALI
SİBER GÜVENLİK YÜKSEK LİSANSI

ANOMALİ TABANLI AĞ SALDIRI TESPİTİ

DÖNEM PROJESİ

Onur Fırat ÖZTÜRK

Proje Yürütücüsü: Doç. Dr. Kazım YILDIZ

İSTANBUL 2023

İÇİNDEKİLER

ÖNSÖZ

Bu çalışma Marmara Üniversitesi Fen Bilimleri Enstitüsü Siber Güvenlik Yüksek Lisans Eğitimini dönem projesi olarak hazırlanmıştır. Bu projenin hazırlanmasında desteklerini esirgemeyen sayın Doç. Dr. Kazım Yıldız hocamıza teşekkürlerimi sunarım.

TABLO LİSTESİ

ŞEKİL LİSTESİ

KISALTMALAR

KNN K-Nearest Neighbor

SVM Support Vector Machines

ÖZET

İnsalık tarihinin başlangıcından beri insanlar çatışma içinde olup bunun için çeşitli saldırı ve savunma araçları geliştirmiştir. Teknolojinin gelişmesiyle fiziksel uzaydaki bu saldırılar siber uzaya taşınmış olup bu saldırılar için de savunma araçları üretmişlerdir. Sanal dünyadaki saldırıları fiziksel dünyadaki saldırılara kıyasla tespit etmesi zor olup kullanılan savunma araçlarının performansı ve saldırı doğru sınıflandırması önem teşkil etmektedir. Bu savunma araçlarından biri de Ağ Saldırısı Tespit Sistemleridir.

Bu proje çalışmasının amacı ağdaki anormal trafiği algılayan ve bunu sınıflandıran makine öğrenmesi tabanlı bir ağ saldırısı tespit sistemi modeli oluşturmaktır. K-Nearest Neighbor (KNN) ve Lineer Support Vector Machines (SVM) algoritmalarının çalışma sistemi diğer algoritmalara kıyasla daha az karmaşık olduğu için bu algoritmalar tercih edilmiştir. K-Nearest Neighbor (KNN) ve Lineer Support Vector Machines (SVM) algoritmalarını eğitebilmek için sanal lab ortamında normal ve çeşitli saldırıların ağ trafiği paketleri toplanmıştır. Veriler notlandıktan sonra normalize edilmiş ve K-Nearest Neighbor (KNN) ve Lineer Support Vector Machines (SVM) gibi algoritmalar denetimli öğrenme algoritmaları olduğundan veri seti etiketlenmiştir [21,22]. Doğrulukları maksimum yapan model parametreleri bulunmuş ve model puanları kıyaslanmıştır. K-Nearest Neighbor (KNN) modelleri, Lineer Support Vector Machines (SVM) modellerine kıyasla daha başarılı sonuçlar üretmiştir. “normal” etiketinin de dahil olduğu ve çoklu sınıflama yapan K-Nearest Neighbor (KNN) modeline karşı Hibrit model oluşturulmuş, bu iki modelin sonuçları da yarıca kıyaslanmıştır.

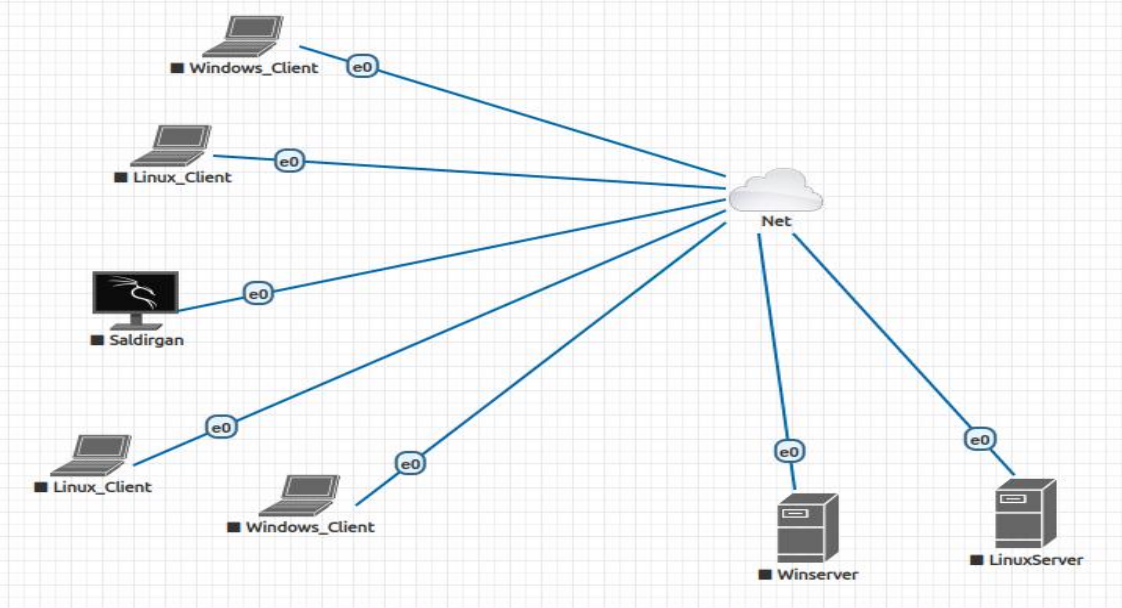
SUMMARY

Since the beginning of human history, humans have been engaged in conflicts and have developed various attack and defense tools for this purpose. With the advancement of technology, these physical attacks have been extended to the cyber domain, leading to the development of defense mechanisms against such attacks. Detecting attacks in the virtual world is more challenging compared to physical attacks, making the performance of defense tools and accurate classification of attacks crucial. One of these defense tools is the Network Intrusion Detection System.

The aim of this project is to create a machine learning-based network intrusion detection system that detects and classifies abnormal traffic in a network. The K-Nearest Neighbor (KNN) and Linear Support Vector Machines (SVM) algorithms were chosen because their working mechanisms are less complex compared to other algorithms. To train the K-Nearest Neighbor (KNN) and Linear Support Vector Machines (SVM) algorithms, network traffic packets of normal and various attack types were collected in a virtual lab environment. After the data was labeled, it was normalized, and since K-Nearest Neighbor (KNN) and Linear Support Vector Machines (SVM) are supervised learning algorithms, the dataset was labeled [21, 22]. The model parameters that maximize accuracy were found, and the model scores were compared. The K-Nearest Neighbor (KNN) models produced more successful results compared to the Linear Support Vector Machines (SVM) models. A Hybrid model was created by combining the K-Nearest Neighbor (KNN) model, which performs multi-classification including the "normal" label, and the results of these two models were also compared.

1. AĞ TOPOLOJİSİ VE TOPLANAN VERİLER

Labı oluşturmak için eve-ng yazılımı kullanılmıştır. Labdaki cihazlar dış ağdaki cihazlara erişebilir olup, dış ağdaki cihazarın lab cihazlarına erişimi kapatılmıştır. Ağda 1 tane Ubuntu Server yüklü web sunucu, 1 tane Windows Server 2012 yüklü Domain Controller, 2 tane Windows 10 yüklü istemci, 2 tane Arch Linux yüklü istemci ve 1 tane Kali Linux yüklü saldırı cihazı mevcuttur. 1. Şekilde verilerin toplandığı ağ topolojisi görülmektedir.



Şekil 1. Verilerin Toplandığı Ağ Topolojisi.

1.1 Uç Nokta Cihaz Trafiği

Normal bir uç nokta cihazda ağ trafiğini tarayıcı, ajan ve syslog yazlımları oluşturur. Lab ortamında ArchLinux ve Windows 7 cihazlar uç nokta cihazlar olarak kullanılmıştır. Bu uç nokta cihazlarda websitesi gezintileri yapılmış ve cihazlardaki loglar “fluent-bit” yazılımı aracılığı ile uzaktaki syslog sunucusuna 60514 TLS bağlantısı ile aktarılmıştır [1,27,28].

1.2 Web Sunucusu Trafiği

Web sunucusu için Ubuntu Server 20.04 sürümü kurulmuş ve 80 portundan dışarıya açılmıştır [2]. Sunucuya ağdaki saldırgan tarafından “hulk” isimli araçla dos saldırısı gerçekleştirilmiştir [3,26].

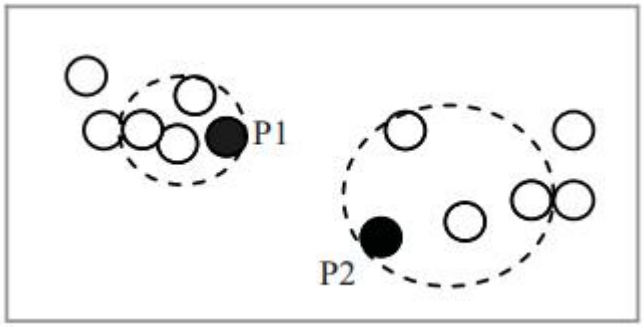
1.3 Domain Controller Trafiği

Windows Server 2012 R2 kurulmuş Samba ve Kerberos servisleri aktif edilmiştir [29,30]. Nmap aracı ile sık kullanılan portların taraması yapılmıştır [4,25]. SMB protokolündeki Ms17-010 zafiyeti, Metasploit frameworkü ile sömürülmüştür [5,24]. “enum4linux” aracı ile SMB bilgileri toplanmıştır [6]. “kerbrute” aracı ile kullanıcı bilgileri toplanmıştır [7]. Server üzerindeki kullanıcılarda kerberos pre-authentication aktif edilmemiş olup, “GetNPUsers.py” aracı ile ASREPRoasting saldırısı gerçekleştirilmiştir [8,23].

1. MODEL ALGORİTMALARI

1.1 KNN Algoritması

K-Nearest Neighbor (KNN), parametrik olmayan makine öğrenmesi algoritmasıdır [20]. Diğer algoritmalara göre daha basittir. Etiketli eğitim verisinde belirli sayıdaki komşunun istatistiki hesaplamasıyla tahminleme yapar [13,15]. 2. Şekilde temsili S_x kümesi gösterilemtedir.



Şekil 2. K-Nearest Neighbor, S_x kümesi örneği. [16]

d uzay boyutu, M sınıf etiketi sayısı, D eğitim verisi kümesi olsun.

$$D = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), \ldots, (x_n, y_n)\} \subseteq X^d \times y \quad (1)$$

$$y = \{1,2,3,4, \ldots, M\} \quad (2)$$

$x \in X^d$ için S_x , $S_x \subseteq D$ ve $|S_x| = k$ olarak tanımlayabiliriz [9]. Burada k S_x kümsesi içerisindeki eleman sayısıdır.

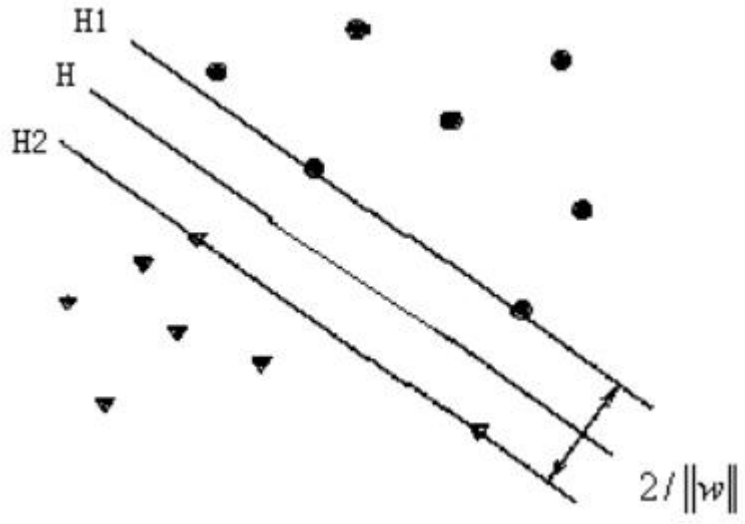
$$\text{dist}(x, x') \geq \max_{(x'',y'') \in S_x} \text{dist}(x, x''), \forall (x', y') \in D \setminus S_x \quad (3)$$

Yani, D'deki ancak S_x'te olmayan her nokta x'ten en az S_x'teki en uzak nokta kadar uzak olmalıdır. S_x kümesindeki en yaygın etiketi verecek sınıflayıcı fonksiyonu 4. deklemdeki gibi tanımlayabiliriz [9].

$$h(x) = \text{mode}(\{y'' : (x'', y'') \in S_x\}) \quad (4)$$

2.2 Lineer SVM Algoritması

Lineer Support Vector Machines (SVM) bir denetimli makine öğrenmesi algoritmasıdır. Uzayı, optimize edilmiş hiperdüzlem ile 2'ye bölerek tahminleme yapar [17]. Şekil 3'te örnek bir düzlem görülmektedir.



Şekil 3. Verilerin düzlem ile ayrılması. [19]

Veri setini 5. denklemdeki gibi tanımlayalım. [12]

$$x_i^T = (x_{i1}, \ldots, x_{id}) \in R^d \quad (5)$$

i = 1,2, ... , m için 5. denklemde d boyutlu uzayda m tane gözlem verisi tanımlıdır. Etiket ise y ∈ { − 1, + 1}'dir. Eğer x_i pozitif sınıfa atanmış ise ise y_i +1, x_i negatif atanımış ise y_i -1'dir. w ağırlık vektörü ve b ise bias olmak üzere, hiperdüzlem H_1 ve H_2 sırasıyla 6. ve 7. deklemde tanımlıdır [12].

$$H_1: (w^T x_i + b) = 1 \quad (6)$$

$$H_2: (w^T x_i + b) = -1 \quad (7)$$

Hiper düzlemler üzerinde 1'er nokta alalım. Bu noktalar sırası ile P_1 ve P_2 olsun.

$$W^T(P_2 - P_1) = 2 \quad (8)$$

$$(P_2 - P_1) = \frac{2}{||w||} \quad (9)$$

Düzlemler arası mesafe 9. denklemde ki gibi çıkar. Böylelikle problem $\frac{||w||}{2}$'yi minimize etmeye, $\frac{2}{||w||}$'yi ise maksimize etmeye dönüşür [12].

$$\alpha_i \geq 0 \quad (10)$$

$$\mathrm{L}(\mathrm{w}, \mathrm{b}, \alpha) = \frac{||w||}{2} - \sum_{i=1}^{m} \alpha_i[(w^T x_i + b)y_i - 1] \quad (11)$$

$$\underset{w,b}{argmin} \quad \underset{\alpha}{argmax} \quad L(w, b, \alpha)$$

Bunun için 10. denklemde ki gibi problemi Lagrange çarpanları metodu ile çözebiliriz [13].

$$\frac{\partial L}{\partial w} = 0 = w - \sum_{i=1}^{m} \alpha_i x_i y_i \quad (12)$$

$$\frac{\partial L}{\partial b} = 0 = \sum_{i=1}^{m} \alpha_i y_i \quad (13)$$

11. denklemdeki Lagrange fonksiyonu denklem 12'ye göre düzenlendiğinde Lagrange fonksiyonunun yeni hali denklem 14'teki gibi olur.

$$\mathrm{L}(\alpha) = \sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_{\mathrm{i}} - \frac{1}{2} \sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{m}} \alpha_{\mathrm{i}} \alpha_{\mathrm{j}} \mathrm{y}_{\mathrm{i}} \mathrm{y}_{\mathrm{j}} \mathrm{x}_{\mathrm{i}}^{\mathrm{T}} \mathrm{x}_{\mathrm{j}} \quad (14)$$

$$\underset{\alpha}{argmax} \; L(\alpha)$$

Lineer tahmin modelimiz $\mathrm{f}(\mathrm{x}) = \mathrm{sgn}(\mathrm{w}^{\mathrm{T}}\mathrm{x} + \mathrm{b})$ iken 12. denklem ile birlikte 15. denkleme dönüşür.

$$\mathrm{f}(\mathrm{x}) = \mathrm{sgn}\left(\sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_{\mathrm{i}} \mathrm{y}_{\mathrm{i}} \mathrm{x}_{\mathrm{i}}^{\mathrm{T}} \mathrm{x} + \mathrm{b}\right) \quad (15)$$

3. MODELLER VE ÖZELLİK SEÇİMİ

3.1 Backward Elimination

Backward elimination modelin doğruğuluğunu arttırmak için kullanılan bir tekniktir [10]. Modelleri ilk olarak eğitirken özellik seçimi yapılmadı. Her eğitimde 1 özellik çıkartıldı ve modellerin accuracy, f1, precision, recall puanları kaydedildi [18]. Eğitimler bittikten sonra bu puanları maksimum yapan KNN için k parametresinin ve Lineer SVM için C parametresinin değeri model sonuçları için baz alındı.

3.2 İkili Sınıflandırma

İkili sınıflandırma için toplanan Syslog trafiği ve Browser trafiği verileri normal ve Exploit, ASREPRoasting, Nmap, Dos, Samba bilgi toplama ve Kerberos kullanıcı adı toplama trafikleri anormal olarak etiketlendi.

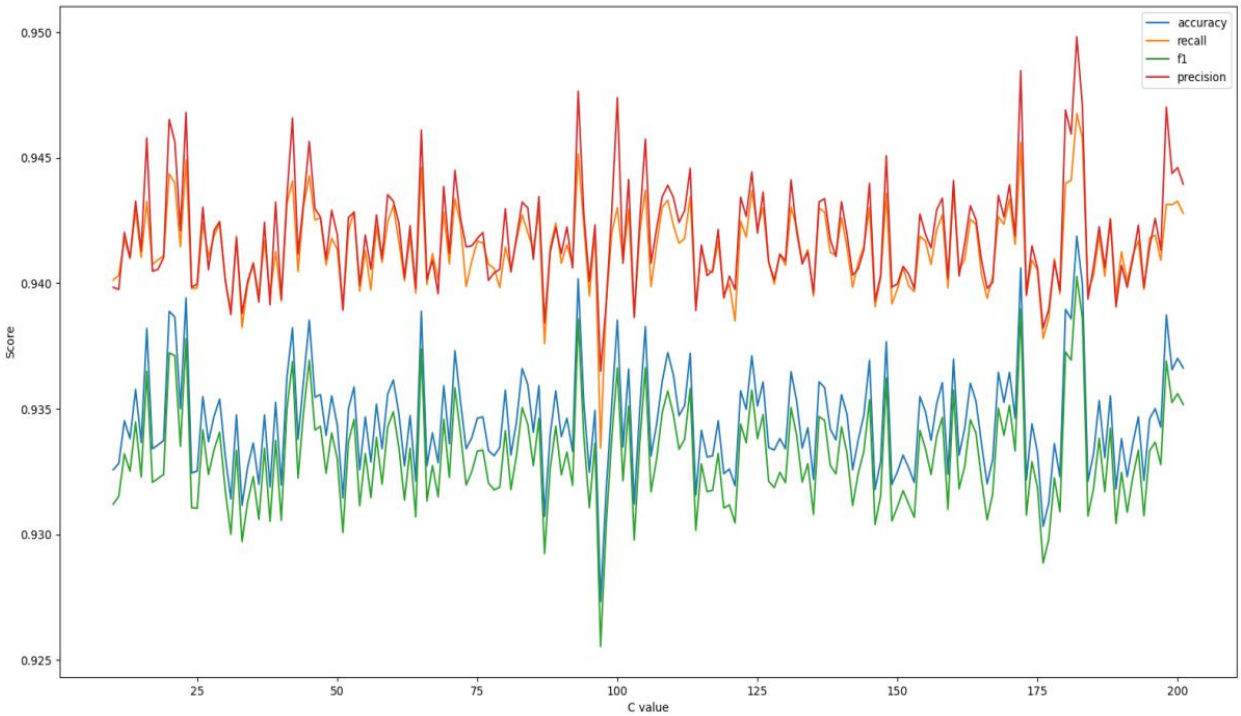
3.2.1 Lineer SVM

Lineer SVM algoritması matematiksel yapısı gereği KNN algoritmasına kıyasla modeli eğitmesi biraz daha yavaştır. Performansa etki eden tek şey algoritmanın kendisi değildir, bunu yanında özellik sayısını da göz önünde bulundurmak gerekir. Modeli eğitirken Backward elimination metodu ile doğruluğu maksimum yapan özellikler bulundu. Veri setini ham halinde özellik sayısı 25'dir. Bu metodu uyguladığımızda elde ettiğimiz özellikler 1. Tablodaki gibidir.

Tablo 1. Lineer SVM algoritmasında ikili sınıflama için kullanılan özellikler.

Sıra Numarası	Paket Özellikleri
1	ip.ttl
2	tcp.window_size
3	tcp.ack
4	tcp.seq
5	tcp.stream
6	frame.time_relative
7	tcp.time_relative
8	tcp.port

Doğruluğu etkileyen parametrelerden biri de C parametresinin değeridir. Maksimum yapan değeri bulmak için $10 \leq C \leq 200$ aralığında birden fazla kez model eğitildi. 4. Şekilde accuracy, recall, f1 ve precision değerleri gösterilmektedir ve görüldüğü üzere maksimum olduğu değer 182'dir.



Şekil 4. İkili sınflandırmada kullanılan Lineer SVM modelinde puanlarının C değerlerine göre değeri.

C=182 için karmaşıklık matrisi Tablo 2'deki gibi çıkmıştır.

Tablo 2. İkili sınflama için Lineer SVM modelinin karmaşıklık matrisi.

	Tahmin		
Gerçek		anormal	normal
	anormal	15185	464
	normal	1830	21988

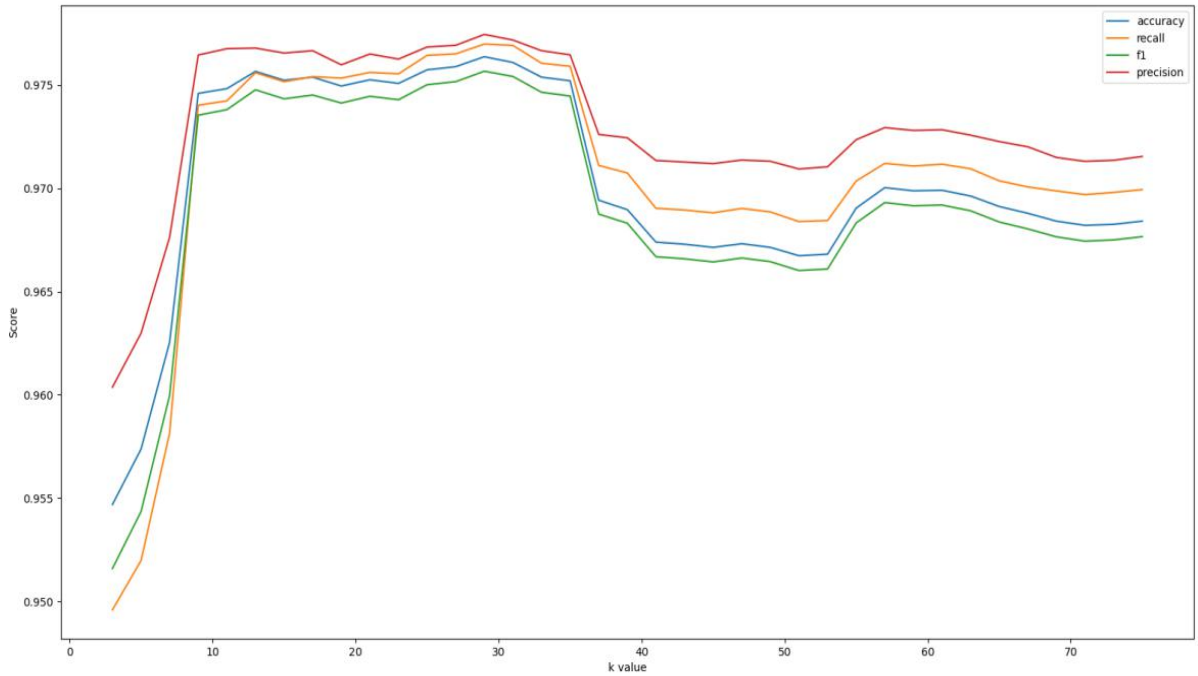
3.2.2 KNN

Bu modelin özellik seçimi için Backward elimination tekniği kullanıldığında en yüksek doğruluk için elde edilen özellikler Tablo 3'deki gibidir.

Tablo 3. İkili sınıflamada KNN için kullanılan özellikler.

Sıra Numarası	Paket Özellikleri
1	ip.ttl
2	tcp.window_size
3	tcp.ack
4	tcp.seq
5	tcp.stream
6	frame.time_relative
7	tcp.time_relative
8	tcp.port

Model için öklid metriği kullandı ve k parametresinin belirlenmesi için bu özellikler ile birlikte model $3 \leq k \leq 75$ aralığında model birden fazla kere eğitildi. Şekil 5'te accuracy, recall, f1 ve precision değerleri gösterilmektedir.



Şekil 5. İkili sınflandırmada kullanılan KNN modelinde puanlarının k değerlerine göre değeri.

Grafik k=29 değeri civarında yerel maksimuma sahiptir. Modelin k=29 için, 15649 tane anormal ve 23818 normal paketin sınıflandırma tablosu Tablo 4'teki gibidir.

Tablo 4. İkili sınflama için KNN modelinin karmaşıklık matrisi.

	Tahmin		
Gerçek		anormal	normal
	anormal	15335	314
	normal	619	23199

3.3 Çoklu Sınıflandırma

Çoklu sınıflandırma için veri setlerine göre 2'şer model oluşturuldu. 1. veri seti için 6 tane saldırı etiketi ve 1 tane sağlıklı veri etiketi, 2. veri seti için sadece 6 tane saldırı etiketi oluşturuldu. Bu veri setlerinin özellikleri için yine Backward elimination yöntemi kullandı.

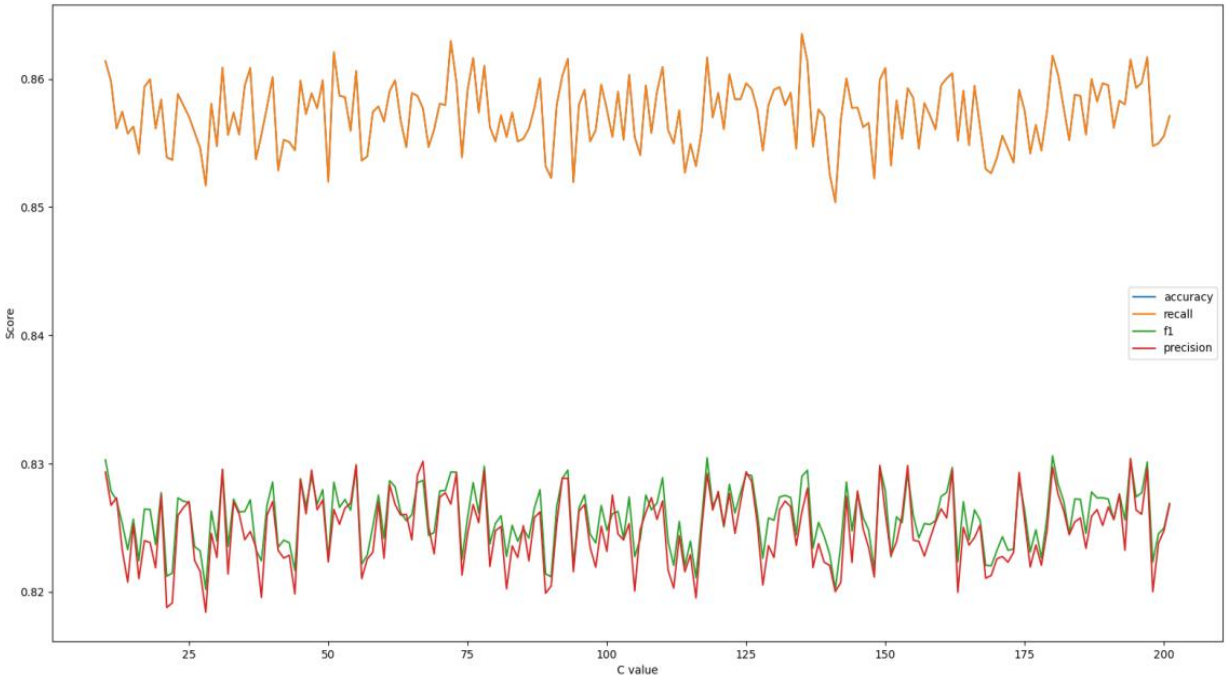
3.3.1 Lineer SVM

Tüm etiketler ile birlite Backward elimination metodu uygulandığında doğruluğu maksimum yapan etiketler Tablo 5'deki gibidir.

Tablo 5. "normal" etiketinin de olduğu çoklu sınıflandırma için Lineer SVM modelinde kullanılan özellikler.

Sıra Numarası	Paket özelliği
1	ip.ttl
2	tcp.window_size
3	tcp.ack
4	tcp.seq
5	tcp.len
6	tcp.stream
7	tcp.flags
8	frame.time_relative
9	frame.time_delta
10	tcp.time_relative
11	tcp.time_delta
12	tcp.port

Modeli $10 \leq C \leq 200$ aralığında birden çok defa eğittiğimizde C=135 için doğruluğun maksimum olduğu görülmüştür. Şekil 6'da accuracy, recall, f1 ve precision değerleri gösterilmektedir.



Şekil 6. "normal" etiketinin de olduğu çoklu sınflandırmada kullanılan Lineer SVM modelinde puanlarının C değerlerine göre değeri.

Modelin karmaşıklık matrisi ise Tablo 6'daki gibi bulunmuştur. Tabloda 7 etiket için tahmini değerlerin dağılımı gösterilmektedir..

Tablo 6. "normal" etiketinin de olduğu çoklu sınflandırmada kullanılan Lineer SVM modelinin karmaşıklık matrisi.

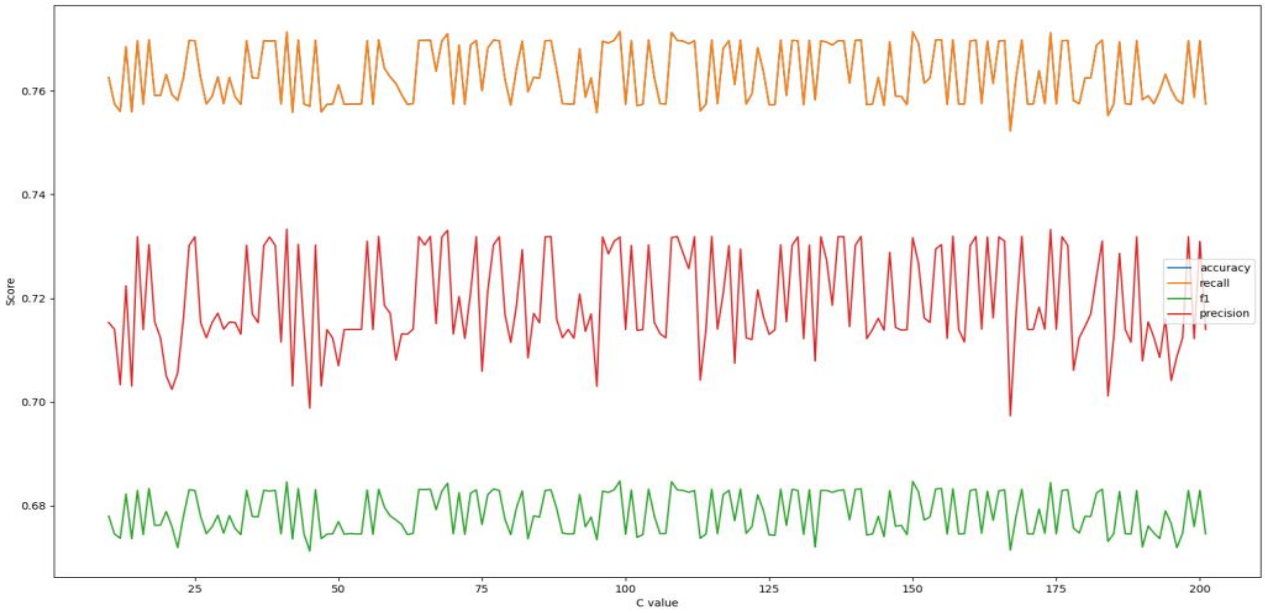
	Tahmin							
Gerçek		asreproasting	dos	exploit-samba	kerberos-userenum	nmap	normal	samba-enum
	asreproasting	0	93	0	3	0	56	3
	dos	0	9384	0	33	263	132	0
	exploit-samba	0	128	1	4	9	176	0
	kerberos-userenum	0	18	0	1923	0	9	0
	nmap	0	2908	0	108	38	0	0
	normal	0	477	0	602	0	22734	5
	samba-enum	0	165	0	23	0	172	0

Veri setinde “normal” etiketi yokken Backward elimination metodu kullanıldığında doğruluğu maksimum yapan etiketler Tablo 7’deki gibidir.

Tablo 7. Sadece saldırı etiketlerinin olduğu Lineer SVM modelinde kullanılan özellikler.

Sıra Numarası	Paket Özelliği
1	ip.ttl
2	ip.proto
3	tcp.window_size
4	tcp.seq
5	tcp.stream
6	tcp.flags
7	udp.port
8	frame.time_relative
9	tcp.time_relative
10	tcp.time_delta
11	tcp.port

Model $10 \leq C \leq 200$ aralığında birden fazla kez eğitildiğinde C=99 için doğruluğun maksimum olduğu görülmüştür. Şekil 7’de accuracy, recall, f1 ve precision değerleri gösterilmektedir.



Şekil 7. Sadece saldırı etiketlerinin olduğu Lineer SVM modelinde puanların C değerlerine göre değeri.

Bu değer için karmaşıklık matrisi Tablo 8'deki gibi bulunmuştur. Tabloda saldırı verilerinin tahmin dağılımları gösterilmektedir.

Tablo 8. Sadece saldırı etiketlerinin olduğu Lineer SVM modelinin karmaşıklık matrisi.

		Tahmin					
Gerçek		asreproasting	dos	exploit-samba	kerberos-enum	nmap	samba-userenum
	asreproasting	0	124	0	0	0	31
	dos	0	9811	0	0	0	1
	exploit-samba	0	218	94	2	2	2
	kerberos-enum	0	27	0	1923	0	0
	nmap	0	2931	0	24	99	0
	samba-userenum	0	194	0	21	0	145

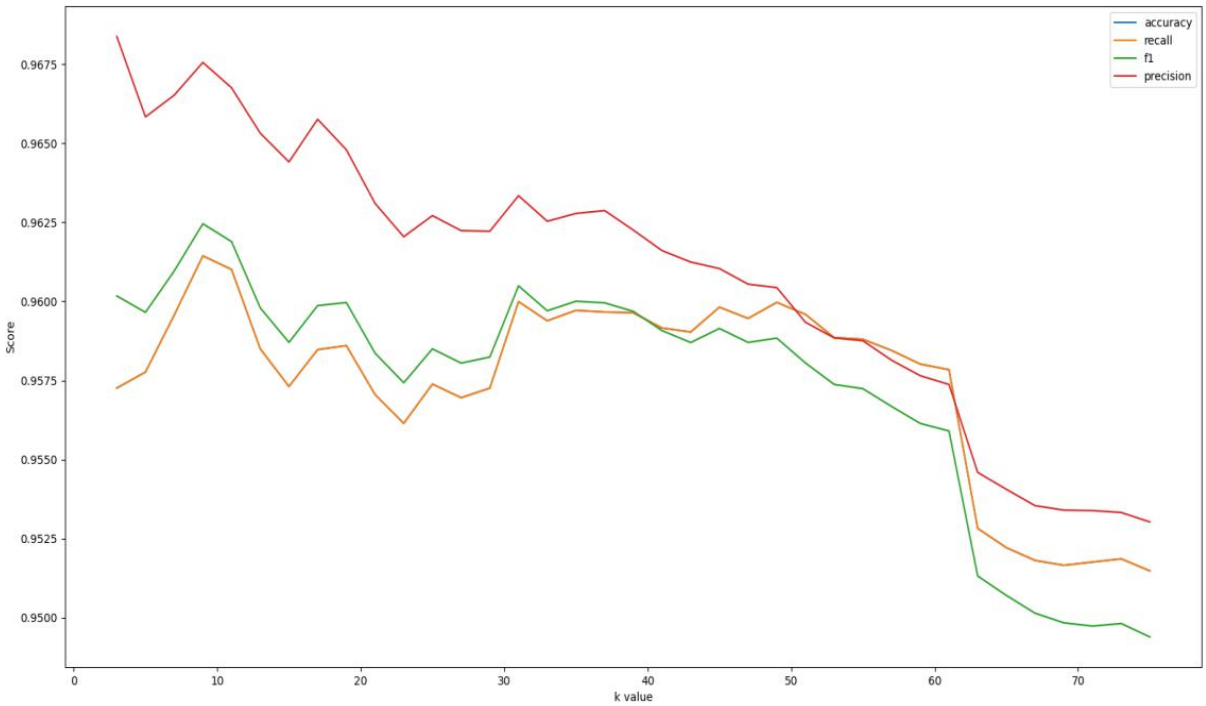
3.3.2 KNN

"normal" etiketinin dahil olduğu model için Backward elimination kullanıldığında elde edilen özellikler Tablo 9'daki gibi bulunmuştur.

Tablo 9. "normal" etiketinin de olduğu çoklu sınıflandırma için KNN modelinin özellikleri.

Sıra Numarası	Paket Özelliği
1	ip.ttl
2	tcp.window_size
3	tcp.ack
4	tcp.seq
5	tcp.stream
6	frame.time_relative
7	tcp.time_relative
8	tcp.port

"normal" etiketinin dahil olduğu modelde $3 \leq k \leq 75$ aralığında birden fazla eğitim gerçekleştirilmiştir. Doğruluğu maksimum yapan k parametresinin değeri 9 çıkmıştır. Şekil 8'de accuracy, recall, f1 ve precision değerleri gösterilmektedir.



Şekil 8. "normal" etiketinin de olduğu çoklu sınıflandırma için kullanılan KNN modelinde puanların k değerlerine göre değeri.

Modelin karmaşıklık matrisi Tablo 10'daki gibi elde edilmiştir. Tabloda tüm verilerin tahmini dağılımları gösterilmektedir

Tablo 10. "normal" etiketinin de olduğu çoklu sınıflandırma için kullanılan KNN modelinin karmaşıklık matrisi.

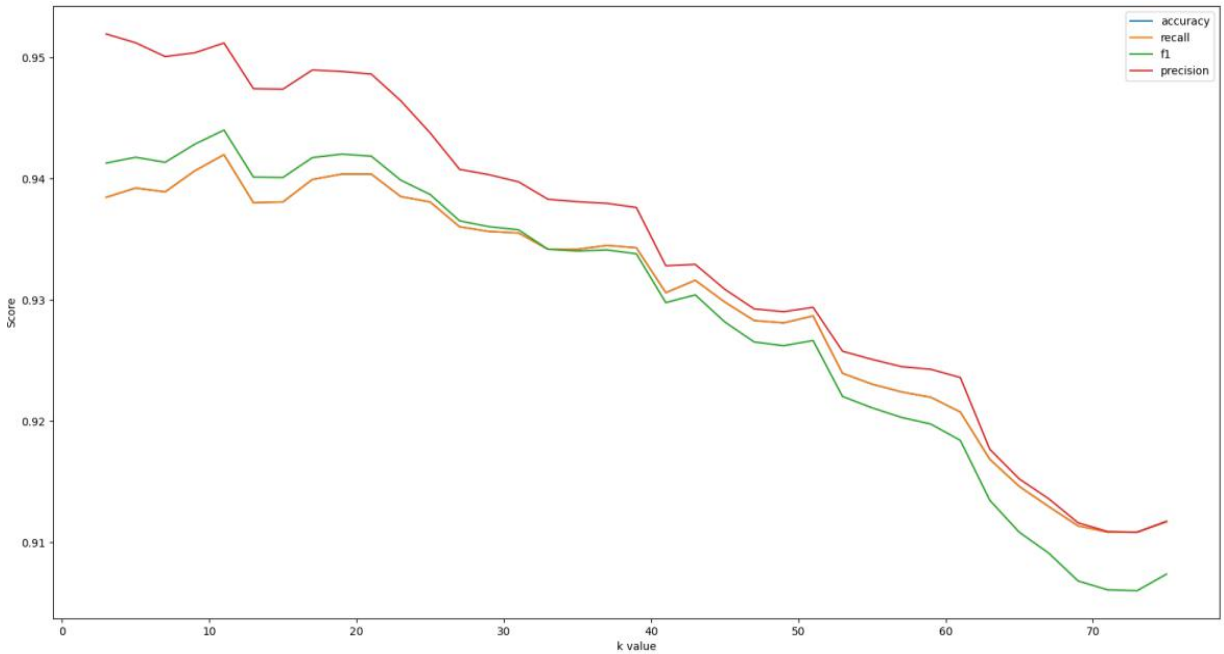
	Tahmin							
Gerçek		asreproasting	dos	exploit-samba	kerberos-userenum	nmap	normal	samba-enum
	asreproasting	150	0	0	0	2	1	2
	dos	0	9282	0	0	529	1	0
	exploit-samba	0	0	188	0	4	124	2
	kerberos-userenum	12	0	0	1923	15	0	0
	nmap	13	235	0	20	2693	66	27
	normal	12	20	78	9	277	23381	41
	samba-enum	0	0	0	6	13	13	328

“normal” etiketnin olmadığı sadece saldırı etiketlerini olduğu veri setinde Backward elimination kullanıldığında doğruluğu maksimum yapan özellikler Tablo 11’deki gibi belirlenmiştir.

Tablo 11. Sadece saldırı etiketlerinin olduğu KNN modelinde kullanılan özellikler.

Sıra Numarası	Paket Özelliği
1	tcp.flags
2	tcp.window_size
3	tcp.ack
4	tcp.stream
5	frame.time_relative
6	tcp.time_relative
7	tcp.port

“normal” etiketinin olmayıp sadece saldırı etiketlerinin bulduğu veri seti ile $3 \leq k \leq 75$ aralığında birden fazla kez eğitildiğinde doğruluğu maksimum yapan k parametresinin değeri 11 çıkmıştır. Şekil 9’da accuracy, recall, f1 ve precision değerleri gösterilmektedir.



Şekil 9. Sadece saldırı etiketlerinin olduğu KNN modelinde puanların k değerlerine göre değeri.

Modelin karmaşıklık matrisi Tablo 12'deki gibi elde edilmiştir. Tabloda saldırı verilerinin tahmin dağılımları gösterilmektedir.

Tablo 12. Sadece saldırı etiketlerinin olduğu KNN modelinin karmaşıklık matrisi.

	Tahmin						
Gerçek		asreproasting	dos	exploit-samba	kerberos-userenum	nmap	samba-enum
	asreproasting	153	0	0	0	2	0
	dos	0	9338	0	0	473	1
	exploit-samba	0	0	296	0	4	18
	kerberos-userenum	12	0	0	1923	15	0
	nmap	14	260	12	22	2709	37
	samba-enum	0	0	0	15	23	322

4. BULGULAR VE TARTIŞMA

İkili sınıflama için KNN ve Lineer SVM'in puan sonuçları birbirine çok yakındır. Lineer SVM yapısı gereği KNN algoritmasına göre daha yavaştır ve saldırı anında zaman çok önemlidir [15]. Lineer SVM'in yapısı gereği uzayı 2 parçaya böldüğünden çoklu sınıflama için bazı sağlıksız veri türlerini tespit edememiştir [15]. "normal" etiketinin de dahil olduğu çoklu sınıflamada KNN için sadece "exploit-samba" yetersiz sonuçlar göstermiştir. Bu sorunu aşmak için "exploit-samba" verilerini tekrar tahmin yapması adına ikili sınıflandırma yapan KNN modeline verdik. "normal" ve "anormal" ayrım dağılımı Tablo 13'deki gibi çıkmıştır. Tabloda saldırı verilerinin "normal" ve "anormal" olarak dağılımları gösterilmiştir.

Tablo 13. İkili sınıflandırma yapan KNN modelinin karmaşıklık matrisi.

		Tahmin	
Gerçek		anormal	normal
	exploit-samba	304	14
	nmap	3040	15
	asreproasting	155	0
	dos	9811	1
	kerberos-userenum	1940	10
	samba-enum	295	65
	normal	76	23742

Ardından "anormal" olarak tahmin edilen paketleri sadece saldırı tahminleri yapan KNN modeline verdik. Bu tahminlemelerin sonuçları da Tablo 14'deki gibi çıkmıştır. Tabloda tüm verilerin hibrit model sonucundaki tahminleri gösterilmiştir.

Tablo 14. Hibrit KNN Modelinin Karmaşıklık matrisi.

		Tahmin						
Gerçek		exploit-samba	nmap	asreproasting	dos	kerberos-userenum	samba-enum	normal
	exploit-samba	294	7	3	0	0	0	14
	nmap	1	2924	3	112	0	0	15
	asreproasting	1	8	146	0	0	0	0
	dos	0	0	0	9811	0	0	1
	kerberos-userenum	0	0	14	0	1926	0	10
	samba-enum	0	0	0	0	6	289	65
	normal	17	4	4	17	8	26	23742

Tablo 15. Hibrit KNN modelinin puanları.

Accuracy	Recall	Precision	F1
0.9461	0.9461	0.9573	0.9516

Tablo 14'deki tahmin dağılımında da görüldüğü üzere iki aşamalı model "exploit-samba" verilerini sınıflandırmada daha başarılı olmuştur. Bu model iki aşamalı olduğundan dolayı anormal trafiğin türünün tespiti diğerine göre daha yavaştır. Fakat saldırılarda ilk olarak anormal trafiğin tespiti daha önemlidir. Şekil 8'de, k parametresi 9 olduğunda çıkan puan değerlerine ve Tablo 15'deki puan değerlerine baktığımızda aradaki farkların %1 ile %1.5 arasında olduğunu görüyoruz. "normal" etiketinin de dahil olduğu 3.3.2. bölümündeki K-Nearest Neighbor modelinde "exploit-samba" etiketli ağ trafiğinin neredeyse yarısı "normal" olarak tahminlenmiştir. Buna karşın model puanları arasındaki fark çok azdır.

KAYNAKÇA

[1] https://docs.fluentbit.io/manual/pipeline/outputs/syslog

[2] https://ubuntu.com/tutorials/install-and-configure-apache#1-overview

[3] https://github.com/R3DHULK/HULK

[4] https://nmap.org/book/man.html

[5] https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/windows/smb/ms17_010_eternalblue.rb

[6] https://www.kali.org/tools/enum4linux/

[7] https://github.com/ropnop/kerbrute

[8] https://github.com/fortra/impacket/blob/master/examples/GetNPUsers.py

[9] https://www.cs.cornell.edu/courses/cs4780/2017sp/lectures/lecturenote02_kNN.html

[10] https://www.javatpoint.com/backward-elimination-in-machine-learning

[11] https://arshad-kazi.com/mathematics-behind-svm/

[12] Esperanzo Garcia-Gonzalo, Zulima Fernandez-Muniz, Paulino Jose Garcia Nieto, Antonio Bernardo Sanchez, Marta Menendez Fernandez, Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers, 2016, 10.3390/ma9070531

[13] R. Tyrrell Rocafellar, Lagrange Multipliers And Optimality, SIAM Review 35 (1993) Yihuo Liao, Rao V. Vemuri, Use of K-Nearest Neighbor Classifier For Intrusion Detection, 2002, 10.1016/S0167-4048(02)005514-X

[14] Dong Seong Kim, Jong Sou Park, Network Based Intrusion Detection with Support Vector Machines, 2003, 10.1007/978-3-540-45235-5_73

[15] Kashvi Taunk, Sanjukta De, Srishti Verma, Aleena Swetapadma, A Brief Review of Nearest Neighbor Algorithm for Learning and Classification, 2019, 10.1109/ICCS45141.2019.9065747

[16] Yun-lei Cai, Duo Ji, Dong-feng Cai, A KNN Research Paper Classification Method Based on Shared Nearest Neighbor, 2010

[17] Theodoros Evgeniou, Massimiliano Pontil, Support Vector Machines: Theory and Applicaitons, 2001, 10.1007/3-540-44673-7_12

[18] Journal Article, Powers, D. M. W., 2011, Journal of Machine Learning Technologies, metrics, nlp evaluation, 1, 37--63, Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation

[19] Yongli Zhang, Support Vector Machines Classification Algorithm and Its Application, 2012, 10.1007/978-3-642-34041-3_27
[20] Danilo Bzdok, Martin Krzywinski, Naomi Altman. Machine learning: Supervised methods, SVM and kNN. Nature Methods, 2018, pp.1-6. ffhal-01657491f
[21] T. Cover and P. Hart, "Nearest neighbor pattern classification," in IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, January 1967, doi: 10.1109/TIT.1967.1053964.
[22] Cortes, C., Vapnik, V. Support-vector networks. Mach Learn 20, 273–297 (1995), doi: 10.1007/BF00994018
[23] C. D. Motero, J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo and N. G. Gómez, "On Attacking Kerberos Authentication Protocol in Windows Active Directory Services: A Practical Survey," in IEEE Access, vol. 9, pp. 109289-109319, 2021, doi: 10.1109/ACCESS.2021.3101446.
[24] Liu, Zian. “Working mechanism of Eternalblue and its application in ransomworm.” International Conference on Cryptography and Security Systems (2021).
[25] M. Shah, S. Ahmed, K. Saeed, M. Junaid, H. Khan and Ata-ur-rehman, "Penetration Testing Active Reconnaissance Phase – Optimized Port Scanning With Nmap Tool," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-6, doi: 10.1109/ICOMET.2019.8673520.
[26] Thilagavathi, Mrs. S. and Dr. A. Saradha. “Impact Analysis of Dos & DDos Attacks.” IOSR Journal of Computer Engineering 16 (2014): 24-33.
[27] Sahoo, Prasanta & Chottray, R & Jena, Gunamani & Pattnaiak, S. (2019). Syslog a Promising Solution to Log Management.
[28] Xavier de Carné de Carnavalet and Paul C. van Oorschot. 2023. A survey and analysis of TLS interception mechanisms and motivations. ACM Comput. Surv. Just Accepted (January 2023). https://doi.org/10.1145/3580522
[29] Smith, Roderick. (2004). Samba and SMB/CIFS. 10.1007/978-1-4302-0683-5_2.
[30] Kohl, John T. and Clifford Neuman. “The Evolution of the Kerberos Authentication Service.” (1992).