

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**OLASI BİR NÜKLEER SALDIRI DURUMUNDA YSA
İLE KURTULMA ALGORİTMASI**
(Yüksek Lisans Tezi)

Tezi Hazırlayan:
Sermet KÖSE

İSTANBUL, 2018

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**OLASI BİR NÜKLEER SALDIRI DURUMUNDA YSA
İLE KURTULMA ALGORİTMASI**
(Yüksek Lisans Tezi)

Tezi Hazırlayan:
Sermet KÖSE

Öğrenci No:
160820052

Danışman:
Dr.Öğr.Üyesi Atınç YILMAZ

İSTANBUL, 2018

YEMİN METNİ

Yüksek Lisans Tezi Olarak Sunduğum “Olası Bir Nükleer Saldırı Durumunda YSA İle Kurtulma Algoritması” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmanın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım 25.05.2018



Sermet KÖSE

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi 160820052 no'lu
SERMET KÖSE için 11/6/18 tarihinde yapılan tez savunma sınavı¹
sonucunda 45 dakika süreyle sunduğu ve savunduğu tezi hakkında² oybirliğiyle,
KABUL kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

Tez Başlığı³ : OLASI BİR NÜKLEER SALDIRI DURUMUNDA YSA İLE
KURTULMA ALGORİTMASI

Tez Sınav Jürisi

Öğretim Üyesi

İmza

Danışman : Dr. Atay Yılmaz

Üye : Prof. Dr. Bekhan JICAHTAROV

Üye : Dr. Ediz ŞAYKOL

¹Jüri üyeleri, söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tezsınavına alır. Tez savunma sınav süresi en az 45, en çok 90 dakikadır. Jüri üyeleri, sınav öncesi yapılacak toplantıda, kendilerinin danışmanından bir üyeyi başkan seçer. Tezsınavı, tez çalışmasının sunulması ve buna ilişkin soru-cevap bölümünden oluşur. Tezsınavı, öğretim elemanları, lisansüstü öğrenciler ve alanın uzmanlarından oluşandırlayıcıların katılımına açık ortamlarda gerçekleştirilir. Belirlenengündeyapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda, jüri en geç on beş gün içinde toplanarak adayın tez savunma sınavına alır. (05 Ağustos 2017 tarihli 30145 sayılı Resmi Gazetede Yayınlanan Değişiklik-Madde 29-3)

²Tezsınavının tamamlanmasından sonra jüri, tez hakkında salt çoğunlukla "kabul", "düzeltme" veya "ret" karar verir. Jüri başkanı, jüri üyelerince imzalanmış karar tutanağını, tezsınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve birinci fıkradaki süre gereği jüri önünde yeniden savunur. Süresi içerisinde "düzeltme" savunmasını girmeyen öğrencinin enstitü ile ilişkisi kesilir. (Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği-Madde 29-4)

³İleriye doğabilecek aksaklıkların engellenmesi için tezin başlığını yazılması gerekmektedir.

Adı ve Soyadı : Sermet KÖSE

Danışmanı : Dr. Öğr. Üyesi Atınç YILMAZ

Türü ve Tarihi : Yüksek Lisans, 2018

Alanı : Bilgisayar Mühendisliği

Anahtar Kelimeler : Yapay Sinir Ağları, Nükleer, İstanbul, Sivil Savunma, Tensorflow, Xamarin, Weka, Anfis, İstanbul

ÖZ

OLASI BİR NÜKLEER SALDIRI DURUMUNDA YSA İLE KURTULMA ALGORİTMASI

İstanbul iline karşı gerçekleşecek olası bir havadan yere nükleer saldırı sonucu, erken uyarı sistemleri sayesinde gelecek bilgiyi kullanarak kullanıcının bulunduğu konuma göre kendisine yapay sinir ağları kullanılarak bir kaçma kurtulma algoritması oluşturmak. Saldırıdan sonra hayat idame için kullanıcıya destek ve arama kurtarma ekiplerinin işini kolaylaştırmak da çalışmanın diğer hedefidir.

Hesaplamalar erken uyarı sisteminden gelecek verilere göre(yaklaşık konum) gerçekleştirileceğinden mütevellit yerden yere tabir edilen terörist saldırılar kapsam dışıdır.

Olası saldırı senaryoları için uygun veri seti oluşturularak,yapay sinir ağları ile hata payı düşene kadar iterasyona uğratılacaktır. Çıkan sonuç kullanıcının anlayacağı düzeye indirgenerek görüntülenecektir.

Çalışma alanı olarak İstanbul seçilmiştir.

Name and Surname : Sermet KÖSE

Adviser : Asst. Prof. Dr. Atınç YILMAZ

Degree and Date : Masters, 2018

Major : Computer Engineering

Keywords : Neural Networks, Nuclear, Istanbul, Civil Defense
Tensorflow, Xamarin, Weka, Anfis, Istanbul

ABSTRACT

DECISION TO ESCAPE OF A POSSIBLE NUCLEAR ATTACK WITH USING ARTIFICIAL NEURAL NETWORKS

A possible air-to-surface attack against the city of Istanbul would result in an escape algorithm based on the user's current position, using neural networks through early warning systems. After the attack, life support and facilitating search and rescue teams' work are another goals of this study.

Calculations will be done according to incoming data given by the early warning system (approximate position) because of that the terrorist attacks are out of scope. The type of terrorist attacks is not air-to-surface.

By creating data set for possible attack scenarios, artificial neural networks will be iterated until the error margin decreases. The results will be listed as user-friendly.

Istanbul was chosen as the study area.

İÇİNDEKİLER

ÖZ	i
ABSTRACT	ii
TABLolar LİSTESİ	v
ŞEKİLLER LİSTESİ	vi
1. GİRİŞ	1
2. NÜKLEER SALDIRILAR	4
2.2. Havadan Yere Nükleer Saldırıları	4
2.2.1. Teknik Bilgiler	4
2.2.2. Hava Patlaması	4
2.2.3. Yeryüzü Patlaması	5
2.2.4. Mantar Bulut	5
2.3. Radar Verileri	6
2.3.1. Dış Kaynak	6
2.3.2. Yerel Kaynak	7
3. UYGULANAN YÖNTEM VE TEKNOLOJİLER	9
3.1. Yöntem	9
3.2. Yapay Sinir Ağları.....	9
3.2.1. Çok Katmanlı İleri Beslemeli Ağlar	9
3.3. Xamarin Cross-Platform Framework	10
3.3.1. Xamarin Geliştirme Ortamları ve Windows Gereksinimleri.....	11
3.3.2. Xamarin Forms	12
3.4. Mobil Öykünücü.....	13
3.4.1. Android Emulator	13
3.4.2. Remoted iOS Simulator	14
3.5. Dış Hizmetler	14
3.5.1. Meteoroloji Hizmeti.....	14
3.5.2. Custom Vision Hizmeti	14
3.6. Tensorflow.....	15
3.7. Weka Explorer.....	16
3.7.1. Trainable Weka Segmentation.....	16
3.8. Senaryo ve Veri Kaynakları	17
3.8.1. Olası Senaryolar.....	17
3.8.2. Kullanıcı Girişleri	17
3.8.3. Mobil Bildirimler.....	19

3.9. Saldırı Sonrası Hesaplamalar	19
3.9.1. YSA Veri Seti	19
3.9.2. YSA Ağının Eğitilmesi	21
3.9.3. Modelin Çalışma Yapısı	24
3.9.4. Eğitim Sonuçları Ve Test.....	26
3.9.5. Multilayer Perceptron Sonuçları	28
3.9.6. Anfis Sonuçları	31
3.9.7. Diğer Hesaplamalar	32
3.10. Uygulama Mimarisi.....	32
3.10.1. Çalışma Ortamı	32
3.10.2. Kullanılan Eklentiler	32
3.10.2. Deployment Diyagram.....	33
3.10.3. Sınıf Diyagramı.....	33
3.10.4. Merkezi Yapı	35
3.11. Saldırı Sonrası Bilgilendirmeler	35
4. SONUÇ	38
KAYNAKLAR	40
EKLER	43
Ek 1. Ekran Görüntüleri	43

TABLÖLAR LİSTESİ

	Sayfa No.
Tablo 1: Xamarin Geliştirme Ortamları ve Gereksinimleri	11
Tablo 2: Xamarin Windows Gereksinimleri	11
Tablo 3: Çalışmada Kullanılan Veri Setinin Son Hali	21
Tablo 4: Etiketleme Sonucu Son Durum.....	24
Tablo 5: ImageClassifier Sınıfı Kod Bloğu 1	24
Tablo 6: ImageClassifier Sınıfı Kod Bloğu 2	25
Tablo 7: ImageClassifier Sınıfı Kod Bloğu 3	26
Tablo 8: Etiket Bazında Eğitim Sonuçları	26
Tablo 9: Etiket Bazında Eğitim Sonuçları	29
Tablo 10: Çok Katmanlı Algılayıcı İçin Çıkarılan Veri Seti İlk 3 Örnek	29

ŞEKİLLER LİSTESİ

Sayfa No.

Şekil 1: Nevada Çölündeki Örnek Patlamaya Ait Mantar Bulut Görüntüsü(1957)	5
Şekil 2: Mantar Bulut Çizimleri	6
Şekil 3: Geiger Sayacı	8
Şekil 4: İleri Beslemeli Ağlar	10
Şekil 5: Xamarin Mimarisi	12
Şekil 6: Android Emulator.....	13
Şekil 7: Meteoroloji xHTML Görünümü	14
Şekil 8: Custom Vision Hizmeti.....	15
Şekil 9: Tensorflow	15
Şekil 10: Weka Explorer.....	16
Şekil 11: Trainable Weka Segmentation	17
Şekil 12: Kullanıcı Veri Giriş Ekranı	18
Şekil 13: Mobil Bildirim Gönderme Ekranı	19
Şekil 14: Custom Vision Eğitim Ağı Etiketleme Ekranı	22
Şekil 15: Örnek Test Sonuçları.....	27
Şekil 16: Mobil Uygulama Test Sonuçları	27
Şekil 17: Görüntü Sınıflandırma Ekranı	28
Şekil 18: Oluşan Plot Matrix	30
Şekil 19: Multi Layer Perceptron Kullanılarak Oluşan Ağ	30
Şekil 20: Anfis Sonuçları.....	31
Şekil 21: Anfis Ağ Yapısı.....	31
Şekil 22: Deployment Diyagram	33
Şekil 23: Android Katmanı Sınıf Diyagramı	34
Şekil 24: Xamarin Forms Katmanı Sınıf Diyagramı	34
Şekil 25: Çalışma Kapsamından Çıkarılan Merkezi Hesap Sistemi.....	35
Şekil 26: Örnek Bildirim Ekranı.....	36

KISALTMALAR

ANFIS	: Adaptive Network Based Fuzzy Inference System
API	: Application Programming Interface
BMEWS	: Ballistic Missile Early Warning System
EEG	: Elektroensefalografi
GPS	: Global Positioning System
iOS	: iPhone Operating System
MIRV	: Multiple Independently Targetable Reentry Vehicle
MLP	: Multilayer Perceptron
SDK	: Software Development Kit
T-SQL	: Transact Structured Query Language
UI	: User Interface
XAML	: Extensible Application Markup Language
xHTML	: Extensible HyperText Markup Language
YSA	: Yapar Sinir Ağları

1. GİRİŞ

Bu çalışma, yapay sinir ağıları kullanılarak olası bir nükleer saldırı sonrası hayatta kalma yöntemlerini ve bu yöntemlerin kullanılması ile geliştirilmiş uygulamayı anlatmaktadır. Çalışmanın amacı elde edilen bilgileri kullanarak, kullanıcıları saldırı sonrası oluşacak serpinli ve radyoaktif etkiden olabildiğince uzak tutmaktır.

Günümüz teknolojisi ile beraber erişimin daha da kolaylaşmasıyla nükleer saldırı gücü ülkeler arası tehdit haline gelmiş durumdadır. Bu çalışmada tehditlere karşı, gelişen son kullanıcı teknolojilerinden de faydalanarak kullanıcıyı yönlendirici ve bilgilendirici yöntemler üzerinde durulmuştur.

Giriş olarak ilişkili konular kısmında uygulamayla doğrudan alakalı olan konular incelenecektir. Yapay sinir ağıları ve kullanılacak teknolojiler açıklanacak, erken uyarı sistemleri hakkında bilgi verilecek, hesaplamalarda kullanılacak yardımcı sistemler tanıtılacak. Son olarak da bu yöntemlerle yapılmış diğer çalışmaların bu çalışmaya olan etkileri Literatür Taraması başlığı altında anlatılacaktır.

Yöntem bölümünde, giriş bölümünde bahsedilen yöntemler ve konuyla alakalı ilgili detayları anlatılacaktır. İlk olarak hesaplama algoritması için kullanılacak ana ve yardımcı veri kaynaklarının denklemdaki yerleri gösterilecek daha sonra bu durumlar için yapılan YSA modellemeleri detaylı bir şekilde açıklanacaktır.

Son bölümde ise, yapay zekanın öğrenmesi ile elde edilen veriden hareketle oluşturulacak simülasyon sonuçları sunulacak ve bu çıkarımlardan bir tanesi mobil ortamda görüntülenecektir. Yapay sinir ağlarının çalışmadaki yeri, eski saldırı verileri üzerinde bir eğitim seti oluşturulup olası bir saldırı için çıkarımda bulunmaya yardımcı olmasıdır.

Çalışmanın ilk aşaması saldırı öncesi evre olan erken uyarı aşamasıdır. Bu aşamada amaç kullanıcıyı veri toplama aşaması olan ikinci aşamaya hazırlamaktır. Kullanıcıya önceden gidecek bildirimle birlikte saldırı sonrası yapması gerekenler yönlendirilecektir.

Çalışmanın ikinci aşaması ise saldırı sonrası kullanıcıyı hayatta kalmasına yardımcı olan, nükleer serpinti ve radyoaktif etkilerden nasıl uzak duracağına dair bilgi ve yönlendirmeler içeren ve bunu yapay sinir ağları modellemesiyle yapan kısımdır.

İlk aşamadan bilgilenen kullanıcı uygulamanın yönlendireceği veri toplama aşamalarını gerçekleştirecektir.

Yapay sinir ağlarının uygulama alanı geniş olmakla beraber tahmin çalışmalarında etkin olarak kullanılmaktadır.

2015 yılında Genel Tıp Dergisinde yayınlanmış bir çalışmada, bir hastalık tanısı koyma ve bu tanıya bağlı olan hayatta kalma oranı araştırılmıştır. Bahsedilen çalışmadaki verileri 150 hastayla alakalı yaş, cinsiyet, ameliyat öncesi odyometrik sonuçlar, kulak patolojisi ve cerrahi işlem ayrıntıları oluşturmakla birlikte hastalar için ameliyat sonrası işitme eşiği hesaplanmıştır. Bu çalışmada geliştirilen en iyi sinir modelleri belirlenen test verileri için %84 doğru tahminler elde etmiştir[1]. Kadir Tok'a ait 2017 yılında yayınlanan başka bir çalışmada ise eğitim ve veri seti olarak romatoid artrit hastası olan ve olmayan insanlardan alınan veriler kullanılmıştır. Kullanılan ağ 0 hataya ortalama 500 ve 1500 iterasyon aralığında ulaşmıştır. Backpropagation algoritması ile %82, Perceptron algoritması ile ise de %71 doğruluk yüzdesine ulaşılmıştır[2]. Ali Aslantaş'ın sağlık alanında gerçekleştirmiş olduğu çalışmada ise yapay sinir ağları kullanılarak yapılan metastaz ayırımında %92.3 doğruluk oranı, %94 duyarlılık oranı ve %86,67 özgüllük oranı tespit edilebilmiştir[3]. Mustafa Coşkun'un 2013 yılında yapmış olduğu çalışmada ise çok katmanlı algılayıcı kullanılarak EEG tabanlı anestezi derinliği tahmini yapılmış ve %95 doğruluk oranına erişilmiştir[4]. Kemik erimesi teşhisi ile ilgili 2013 yılında yapay sinir ağları kullanılarak gerçekleştirilen Mustafa İstanbullu'ya ait çalışmada ise yapılan sınıflandırma sonucu %86 başarıyla erken dönüm görüntülerine osteoporoz teşhisi konulabilmiştir[5]. Epilepsi teşhisi için 2012 yılında yapılan başka bir yapay sinir ağları çalışmasında ise Nesibe Yalçın parçacık sürüsü optimizasyonu yöntemini kullanarak %98 sınıflandırma doğruluğuna ulaşmıştır[6]. Gülhan Toğa'nın 2012 yılında koroner arter hastalarına stent uygulandıktan sonra oluşan ölümlerin oranını indirgenmemiş veri seti ile oluşturulan yapay sinir ağları kullanarak incelediği çalışmada ise hastane içi ölümlerin %97.23'ü, taburcu sonrası ölümlerin %94.44'ü ve

eş zamanlı sınıflandırma modelindeki ölümlerin %90.87'si doğru sınıfa atanmıştır[7]. Biyomedikal işaretlerden hareketle yapay sinir ağlarını aracıyla fibromiyalji sendromunun teşhis çalışmasını gerçekleştiren Özhan Özkan ise 2012 yılında bu çalışmada eğitim verileri için doğruluk yüzdesini %100, test verileri ise bu yüzdeyi %86,4 olarak elde etmiştir[8]. Fatma Aşkın'ın 2011 yılında yapmış olduğu sol atriyum çapını kıyasladığı temel bileşenler regresyonu, kısmi en küçük kareler regresyonu ve yapay sinir ağları tahminlemelerinde en başarılı sonucu yapay sinir ağı modellemesiyle almıştır[9]. Telemedikal elektrokardiyografi sinyallerinin yapay sinir ağları ile Yashar M. Jwmah tarafından değerlendirildiği çalışmada ise gerçekleştirilen sınıflandırma sistemlerinde elde edilen sonuçlar; temel bileşen analizi %99.28, bağımsız bileşen analizi %95.71 ve dalgacık dönüşümü %98.92 olarak elde edilmiştir[10]. Hakan Aydemir'in apneli hastaları ayırt etme çalışmasında yapay sinir ağları ile tanıyı belirleyebilme oranı %96'ya ulaşmıştır[11].

Dr. Selvakumar ve arkadaşının 2010 yılında yayınlanan çalışmasında yapmış olduğu nükleer görüntü sınıflandırmasında ise nükleer görüntülerden patlamanın enerjisini hesaplamak üzere bir çalışma gerçekleştirilmiştir. Fotoğraflardan enerji ölçümlerinde %90 oranında bir doğruluk saptanmıştır[12].

Sumaya Abusaleh ve arkadaşlarının 2017 yılında yayınlanan patlamalar için video görüntüleri üzerinden gerçekleştirdikleri sınıflandırma çalışmasında %94.8 başarı oranına ulaşmışlardır[13].

Daniel T.Schmitt ve arkadaşının 2016 yılında yayınlanan çalışmalarında video üzerinden gerçekleştirdikleri yapıda %86.03 tahmin oranına ulaşmışlardır[14].

Robert ve arkadaşları, 2015 yılında yayınlanan çalışmalarında nükleer ateş topları üzerinde görüntüden sıcaklık analizi gerçekleştirmişlerdir[15].

2. NÜKLEER SALDIRILAR

2.2. Havadan Yere Nükleer Saldırıları

Nükleer teknoloji her ne kadar gelişmiş durumda olsa da, dünya devletlerine oranlanırsa hala sayıca az ülkenin sahip olduğu bir güçtür. Nükleer başlık sayısına göre bu güce sahip olan ülkeler büyükten küçüğe doğru Amerika, Rusya, İngiltere, Fransa, Çin, Hindistan, Pakistan, İsrail ve Kuzey Kore'dir[16].

Saldırı olasılığının olup olmaması başka bir araştırma konusu olmakla beraber, bu çalışmada baz alınacak senaryolar havadan yere gerçekleşecek saldırıları kapsamaktadır. Sebebi ise, radar ve tespit sistemlerinin yalnızca havadan yere olan saldırılarda ikaz halinde bulunabilmesidir.

Yerden yere şeklinde tabir edilen muhtemel terörist saldırılar çalışmanın kapsamı dışındadır.

Çalışma alanı olarak ise ülke içerisinde nüfus oranı göz önüne alınarak İstanbul ili tercih edilmiştir.

2.2.1. Teknik Bilgiler

Havadan yere nükleer saldırılar, bir uçak aracılığıyla havadan bırakılabileceği gibi, bir balistik füze aracılığıyla da yapılabilir.

Balistik füzenin menzili arttıkça, ulaştığı irtifa ve buna bağlı olarak hedefine doğru dalışa geçtiğinde sürati de artmaktadır.

Örneğin, 250-300 km civarında menzile sahip bir balistik füze için 3 dakika, 1300 km menzile sahip bir füze içinse 10 dakikadan az bir uçuş süresi vardır. Uçaklarla kıyas açısından, aynı rakamlar 1300 km menzile gitmeleri söz konusu olduğunda savaş uçakları için 1 ila 1,5 saat arasındadır[17].

2.2.2. Hava Patlaması

Havada patlatılan bombalar patlamalar teknik tabir olarak air burst diye tabir edilen patlama cinsine sahip bombalar için kullanılmaktadır. Buradaki temel amaç bombanın yıkıcı zararını arttırmaktır[18]. İleriki safhalarda gerçekleştirilen hesaplamalarda bu veri de göz önüne alınacaktır.

2.2.3. Yeryüzü Patlaması

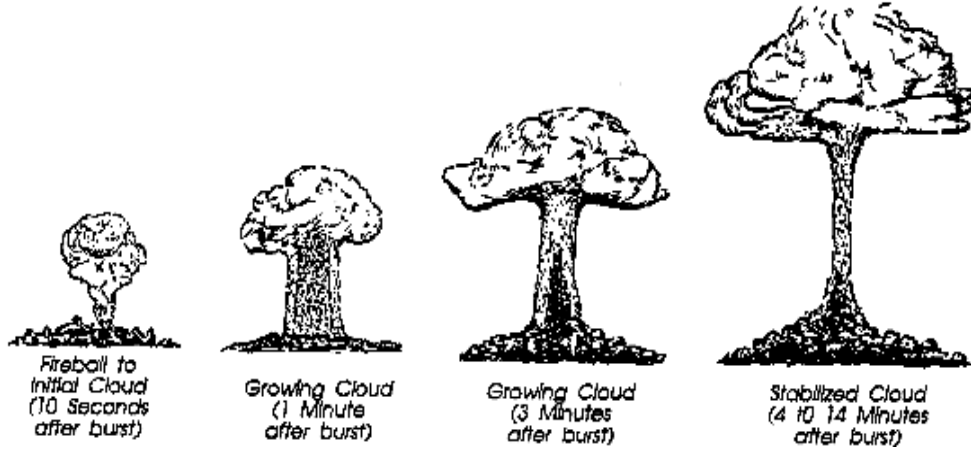
Yerde patlatılan bombalar için ise teknik olarak surface burst diye bahsedilmektedir. Adından da anlaşılacağı üzere bomba yer yüzüne ulaştığında patlamakta ve bu da patlama esnasında diğer yöntemle kıyasla daha çok toz, toprak ve beton parçası saçmaktadır[18]. Bu da nükleer kirlilik açısından diğer yöntemle kıyasla daha çok kirlilik yaratacaktır. Şekil 1’de Nevada çölündeki örnek bir patlamaya ait mantar bulut görüntüsü paylaşılmıştır[19].



Şekil 1: Nevada Çölündeki Örnek Patlamaya Ait Mantar Bulut Görüntüsü(1957)

2.2.4. Mantar Bulut

Mantar bulut, patlamadan sonra gökyüzünde oluşan, Rayleigh-Taylor Kararsızlığı[20] adı altında açıklanan bir fizik olaydır. Şekil 2’de mantar bulutunun oluşma aşamaları görüntülenmektedir[21].



Şekil 2: Mantar Bulut Çizimleri

Nükleer patlamadan sonra oluşan mantar bulutlar patlama gücüyle alakalı tahmini bilgiler verebilmektedir. Mantar bulutlar patlama tipine ve bölgesine göre renklerinde koyuluk/açıklık içermektedirler. Şehir içinde gerçekleşen patlamalarda etraftaki yıkıntı ve tozlardan ötürü mantar bulutun kafası koyu renkte olmaktadır. Mantar bulutlar patlamanın tipi hakkında da bilgi verebilmektedirler. Örneğin: bulut kökten daha açık renkteyse veya kök kısmı parçalanmış veya kırılmışsa(maddesel olarak bulutla herhangi bir bağlantı kurmuyorsa) hava saldırıdır. Diğer taraftan eğer bulutun kök kısmı kalın ve karanlıksa(mantar kısmıyla maddesel bir bağlantısı varsa) yeryüzü patlamasıdır[21].

2.3. Radar Verileri

Radar verileri saldırı esnasında uygulamanın başlangıcı vermek için kullanılabilir. Çalışmadaki hesaplamalarda öğretilmiş verilerden(eski saldırılardan hareketle) tahminle yapılacağından ötürü bu bölümde bu verilerden sadece bilgi amaçlı bahsedilmiştir. Çalışmada direkt olarak kullanılması da, ülkemizdeki mevcut radarlar ve çalışma yapıları çalışmanın mantığını kurabilmek adına önem taşımaktadır.

2.3.1. Dış Kaynak

Olası balistik füze saldırılarında veri kaynağı olarak kullanılan en temel kaynak, ülkemizde de konuşlu halde duran NATO BMEWS erken uyarı sistemidir. Bu

sistemler NATO'nun 29 üye ülkesinde bulunmakla beraber, rampadan ayrılan bir füzeyi direkt olarak saptayıp iyonosfer tabakasına ulaşana kadar füze tipi, yönü ve hedef noktasını belirli bir olasılıkla tespit edebilme kapasitesine sahiptirler[22].

Havadan yere atılan balistik bir füze hızlanarak troposfere çıktığında terminal irtifasına eriştiğinde kendi hedeflemesine uyarak bir noktaya doğru yatmaya başlar. Bu yatma anı esnasında konik bir projeksiyon alındığında dünyadaki olası Nato hedeflerinin yarısından çoğu hedef safhasından elenir. Örneğin 9.1 varyo(düşey sürat) ile Krasnoyarsk çıkışlı bir füze batıya doğru yatıyorsa Kuzey Amerikan hedef skalasında artık görülmez. Zira o başlık çoktan Avrupa'ya doğru yönelmiş demektir. Bundan sonraki saniyelerde yatışı iyice netleşerek ortaya bir yörünge çıkartır. Bu yörüngenin ortaya çıkmasıyla da kapsam alanına düştüğü 474L radarları uyarı vermeye başlar. Muhtemel hedefi tespit olasılığı yükselir.

Diğer yandan zaten bir MIRV[23] başlığıyla Avrupa'da bulunan iki şehir tek füzeyle vurulabileceğinden daha kesin bir netleştirmeye gidilemez.

2.3.2. Yerel Kaynak

Olası bir nükleer saldırıda Sivil Savunma Genel Müdürlüğü bilgi sağlamakta yükümlü ana kurumdur[24]. Bu çalışmada saldırı senaryoları ve kurtulma olasılıkları inceleneceğinden bu kaynaklar bilgi amaçlı anlatılmıştır. Hesaplamalar bölüm 3.8.1'de belirtilen senaryolara göre gerçekleştirileceklerdir.

Eldeki tahrip bilgisi ile meteorolojiden elde edilen rüzgar verisi birleştirilerek, kullanıcıya serpentinin ne kadar sürede ulaşacağı hesaplanacaktır. Eğer patlama yeryüzünde veya yeryüzüne yakın gerçekleştiyse havada patlamasından çok daha fazla serpinti üretir. Serpintideki parçacıklar gözle görülse bile özel bir cihaz kullanmadan radyasyon oranlarını hesaplamak mümkün değildir[25]. Şekil 3'te gösterilen Geiger sayacı gibi ölçümleme yapan özel aletler ile radyasyon ölçülebilir[26]. Serpinti hesaplamasındaki bu kısıtlamalardan dolayı uygulama safhasında serpinti ile ilgili sadece ulaşacağı süre hesaplanacaktır. Serpinti oranı hesaplama dışıdır.



Şekil 3: Geiger Sayacı

3. UYGULANAN YÖNTEM VE TEKNOLOJİLER

3.1. Yöntem

Çalışmada İstanbul kapsam içine alınarak, bu şehre olacak havadan yere tabir edilen nükleer saldırılara karşı erken uyarı sistemlerinden faydalanarak kullanıcıları uyarma ve saldırı sonrası kullanıcıların hayatta kalması için yönlendirici hesaplamalar gerçekleştirilecektir. Çalışma kapsamında bu hesaplamalar mobil uygulama çatısında derin öğrenme kütüphanesi olan Tensorflow kütüphanesinden yararlanılarak gerçekleştirilecektir.

Uygulama için gerekli altyapıyı hazırlamak adına ilk olarak senaryo ve buna bağlı kullanılacak hesaplamadan bahsedilecek. Sonra Tensorflow kütüphanesi ile yapılan işlemler ayrıntılandırılacaktır.

Bölümün son kısmında ise çalışma için tasarlanacak uygulamanın mimarisi ve teknik özelliklerinden bahsedilecektir.

3.2. Yapay Sinir Ağları

Yapay sinir ağları insandaki biyolojik sinir ağlarının basitleştirilmiş modelleri olmakla beraber hesaplamalarını bundan hareketle insan beyninin gerçekleştirdiği adımlamalara benzer şekilde gerçekleştirmektedir[27].

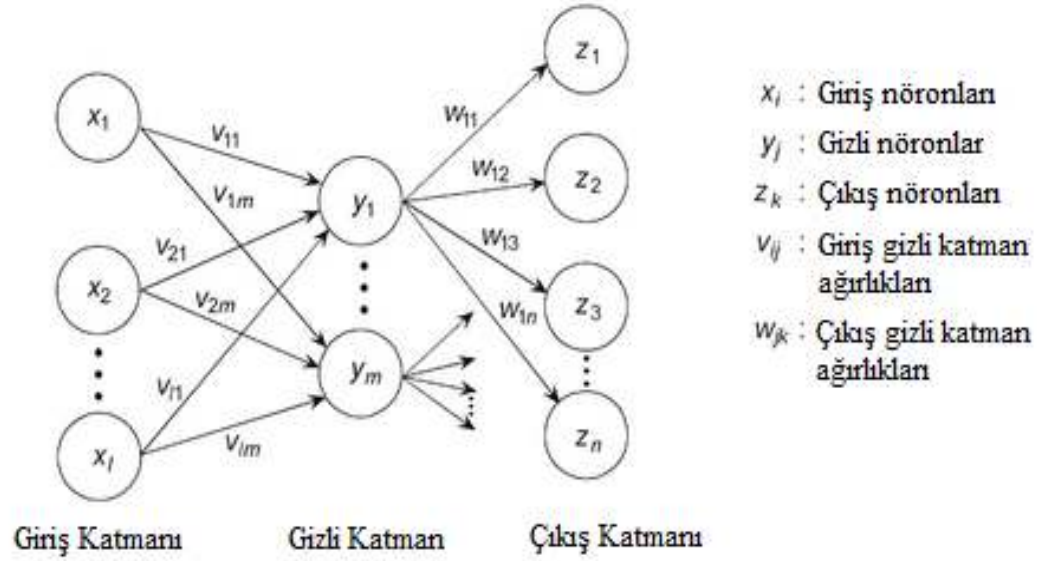
Yapay sinir ağları, beyin mimarisinde bir nöron olarak adlandırılan çok sayıda işleme elemanın birbirine bağlı şekilde çalıştığı bir ağdır. Bu sinir ağları paralel işlem yapabilme yetisine sahiptir[27].

Yapay sinir ağları örneklerle öğrenir. Bilinen bir problem ile ilgili örnekler aracılığıyla eğitilebilirler. Eldeki ağ uygun bir şekilde eğitildikten sonra eğitimsiz örnekleri çözmede etkin bir şekilde kullanılabilir[27].

3.2.1. Çok Katmanlı İleri Beslemeli Ağlar

İleri beslemeli ağlar, birden çok katmandan oluşur. Giriş ve çıkış katmanının yanı sıra gizli katman olarak anlandırılan bir ara katmana da sahiptir. Gizli katman, girişi çıkışa yönlendirmeden önce başka hesaplamaları gerçekleştirmeye yardımcı

olur. Giriş katmanı nöronları gizli katmanın nöronlarına bağlanır ve bu bağlantılardaki ağırlıklardan geçerek ilerler. Şekil 4’de ağıın yapısı gösterilmektedir[27].



Şekil 4: İleri Beslemeli Ağlar

3.3. Xamarin Cross-Platform Framework

Xamarin, Cross Platform ve Native kavramlarını bir araya getiren, Visual Studio kullanarak C# dili ile geliştirilebilen UI Frameworktür.

Projenin yapısına, ihtiyaçlarına ve Xamarin’i kullanma şeklinize bağlı olarak Android, iOS ve WindowsPhone platformlarına yönelik oluşturulacak uygulamaların kodlarını ortaklaştırarak genel proje üretim süresine olumlu yönde etki eder.

Günümüzde uygulama geliştirme işi için C# dilinin yaygın kullanıldığından hareketle, Xamarin’in sağladığı en önemli avantajlardan biri bu ortak dili kullanarak Android, iOS ve Windows Phone için Native uygulama geliştirilmesini sağlamasıdır[28].

3.3.1. Xamarin Geliştirme Ortamları ve Windows Gereksinimleri

Xamarin'ın geliştirme ortamları ve platform karşılıkları[28] Tablo 1'de gösterilmiştir:

	MacOS	Windows
Geliştirme Ortamı	Visual Studio for Mac	Visual Studio
Xamarin.iOS	Evet	Evet (Mac Bilgisayar Şartıyla)
Xamarin.Android	Evet	Evet
Xamarin.Forms	Sadece iOS ve Android	Android, iOS (Mac Bilgisayar Şartıyla)

Tablo 1: Xamarin Geliştirme Ortamları ve Gereksinimleri

Çalışmada kullanılacak olan Windows ortamında, geliştirme yapılması için için Microsoft tarafından tavsiye edilen Windows gereksinimleri Tablo 2'de belirtilmiştir:

	Tavsiye Edilen	Notlar
İşletim Sistemi	Windows 10	Minimum tavsiye edilen versiyon Windows 7. Xamarin.Forms için minimum 8.1.
Xamarin.iOS	iOS 10 SDK	Windows cihaza bağlamak üzere, ağ erişimi olan bir Mac bilgisayar.
Xamarin.Android	Android 6.0 / API level 23	Geriye dönük sürümleri için de uygulama geliştirilebilir.

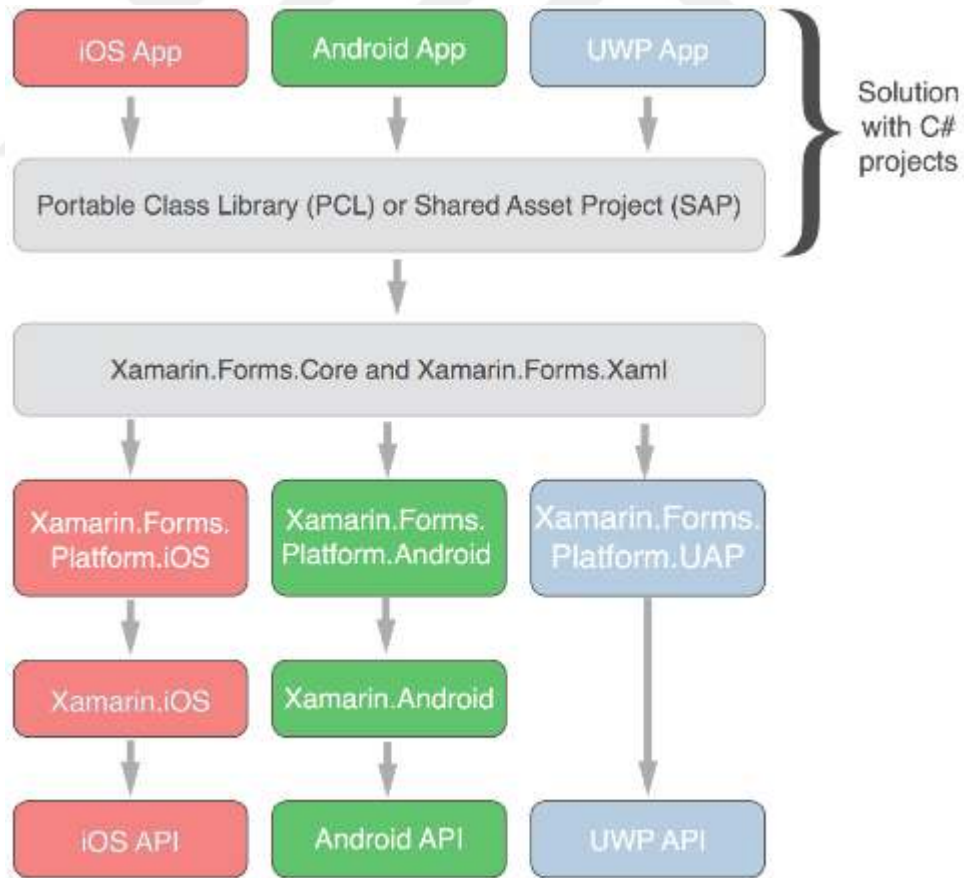
Tablo 2: Xamarin Windows Gereksinimleri

3.3.2. Xamarin Forms

Xamarin platformuyla geliştirme yapabilmek için iki yol ayrımı vardır: Birincisi Traditional Xamarin, ikincisi ise bu çalışmada kullanılacak olan Xamarin Forms'tur.

Traditional Xamarin kullanımı iOS ve Android platformlarının doğal geliştirme ortamlarına ait AppDelegate, UIView, Activity, Intent gibi özel kavramların C# üzerinde de kullanılmasını sağlar.

Xamarin Forms arabirimi ise CrossPlatform işleme yaparak arayüz nesnelerinin dağıtıldığı platforma(iOS, Android) özgü görünmesini sağlayan bir arabirimdir. C# geliştiricilerinin kullanmaya aşina olduğu ContentPage, View, Picker, Entry, Button gibi tanıtık kavramlarla yazılımcının geliştirme ortamına olan adaptasyonunu hızlandırır. Arayüz katmanını oluştururken XAML ya da C# kullanmaya olanak tanır[28]. Şekil 5'te platform bağımsızlığından bahsedilmektedir.



Şekil 5: Xamarin Platform Bağımsızlığı

3.4. Mobil Öykünücü

Çalışmada mobil cihazlar çalışma simülasyonu yapılabilmesi adına öykünücü kullanılması gerekmektedir. Xamarin'ın kullanılacağı platform olan Visual Studio 2017 sürümünde iki platform için de öykünücü gelmekte.

3.4.1. Android Emulator

Visual Studio 2017 ile gelen bu öykünücü, Android için derlenen çalışmaların simüle edilebilmesine olanak verir. Bu öykünücü ile birlikte gerçek cihazda sağlanan ivmeölçer, ekran yönü, hafıza kartı, pil, çok noktalı dokunma, GPS, kamera, ses ve klavyeyi içeren sensörleri kullanılabilir. Şekil 6'da örnek simüle ortam görüntüsü mevcuttur.

Cihaz profilleri menüsü özelliği ile de test ortamındaki Android sürümünü, ekran boyutlarını ve diğer donanım özelliklerini değiştirilebilir[28].



Şekil 6: Android Emulator

3.4.2. Remoted iOS Simulator

Bu öykünücüyü kullanmak için Mac bir cihaza bağlı olma şartı vardır. Xamarin ile iOS için derlenecek olan uygulamada bu öykünücü kullanılarak simüle edilecektir[28]. Android öykünücü için bahsedilen özelliklerin birçoğu iOS platformu için de gerçekleştirilebilir.

3.5. Dış Hizmetler

3.5.1. Meteoroloji Hizmeti

Saldırı sonrası hayatta kalma evresinde lazım olacak verilerden biri olarak ise rüzgar bilgisi de meteorolojik kaynaklarından edinilecektir. Şekil 7’de meteoroloji kaynağından sağlayacak hizmetin görüntüsü bulunmaktadır[29].

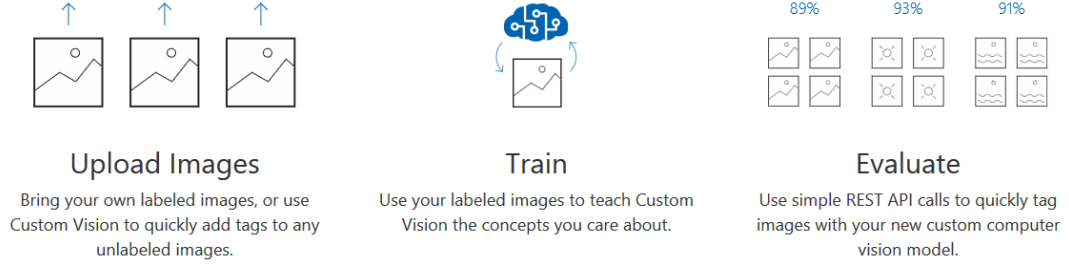


Şekil 7: Meteoroloji xHTML Görünümü

3.5.2. Custom Vision Hizmeti

Görsellerden oluşturulmuş olan veri setini bünyesine alarak, bunları belirtilen etiketlerle birlikte yükleyerek Tensorflow kütüphanesi aracılığıyla ağ oluşturup ve bunları eğiten, Microsoft tarafından sağlanan hizmetin adıdır. Şekil 8’de hizmetin çalışma yapısı gösterilmektedir.

Bu serviste oluşturulan modelin çıktısını alıp kullanabilir veya sağladığı API aracılığıyla servisi çağırıp dosya veya dosya adresi gönderilip tahminleme sonucu geri döndürülebilir.



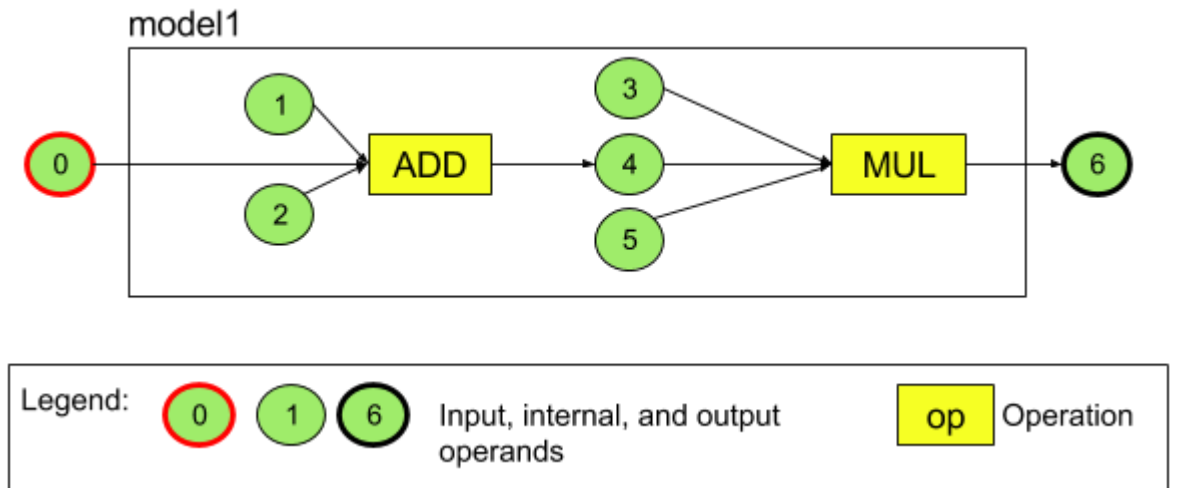
Şekil 8: Custom Vision Hizmeti

3.6. Tensorflow

Google tarafından geliştirilen, ve 2015 yılının Kasım ayından itibaren açık kaynak paylaşımlı olarak dağıtılan, grafikleri kullanarak sayısal hesaplamalar gerçekleştiren bir derin öğrenme kütüphanesidir[30].

Çalışmada kullanılan kapsamları yöntem bölümünde detaylandırılacaktır.

Şekil 9’da Tensorflow kütüphanesinin çalışma yapısı bulunmaktadır.

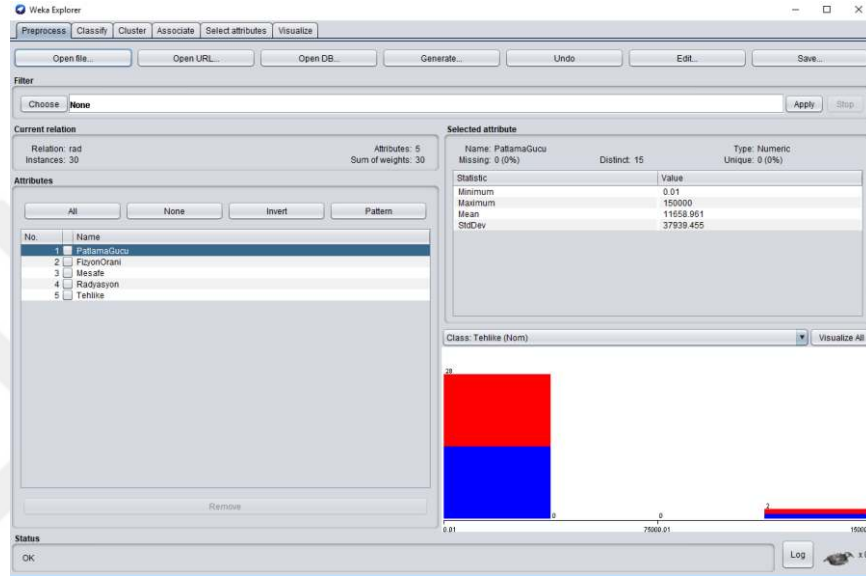


Şekil 9: Tensorflow

3.7. Weka Explorer

Waikato Üniversitesi tarafından Java dilinde hazırlanmış, açık kaynak kodlu bir pakettir. Hali hazırda gelen modellemeler kullanılarak tahminleme yaptırılabilir.

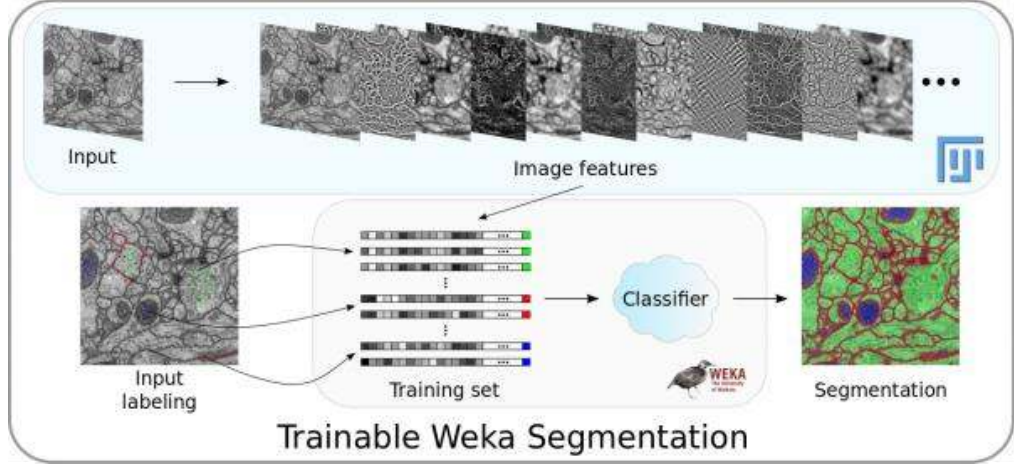
Veri setleri kullanarak bu modellemeler aracılığıyla makine öğrenimi işlemleri gerçekleştirilebilir[31]. Şekil 10'da uygulamanın arayüzü bulunmaktadır.



Şekil 10: Weka Explorer

3.7.1. Trainable Weka Segmentation

Weka Explorer uygulamasının bir eklentisidir. Görselleri sınıflandırmayı sağlar[32]. Piksel bazında 2 boyutlu, vokselle bazında 3 boyutlu sınıflandırma gerçekleştirilebilir. Şekil 11'de eklentinin görsel sınıflandırma yapısı görüntülenmektedir. İçerisinde 3 boyutlu sınıflandırma özellikleri olarak Gaussian blur, Hessian, Derivatives, Laplacian, Structure, Edges, Difference of Gaussian, Mean ve Variance sınıflandırmalarını bulundurmaktadır.



Şekil 11: Trainable Weka Segmentation

3.8. Senaryo ve Veri Kaynakları

Çalışmanın amacı ve hedefi daha anlaşılır olması adına bu bölümde olası saldırı senaryoları açıklanacak ve incelenecektir.

3.8.1. Olası Senaryolar

İlk adım olarak, erken uyarı sisteminin İstanbul hedefi için 12 dakika erkenden uyardığı ve mobil uygulamaya saldırının başladığı bilgisi geldiğinden hareketle; uygulama bu safhadan sonra kullanıcıya yönlendirme yaparak gerekli veri girişini sağlayacaktır.

Kullanıcıdan alınan girdiler ışığında, uygulama elindeki veriler yardımıyla kullanıcıyı nükleer serptinin etkilerinden uzak tutarak hayatı idame etmesine yardımcı olacak bilgiler sunacaktır.

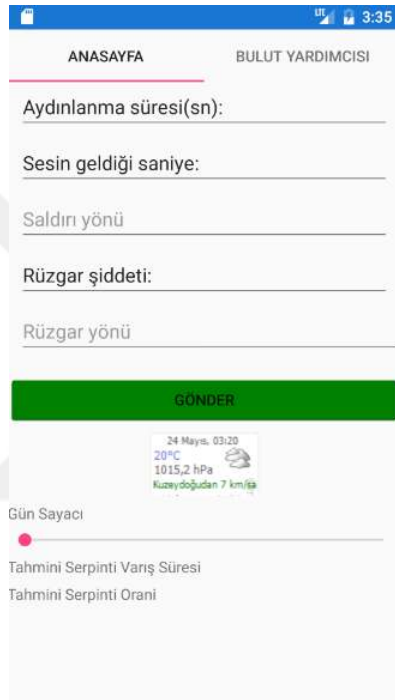
Bu özelliklere ek olarak kullanıcıyı adım adım yapacakları hakkında bilgi notları paylaşılacaktır. Bu adımlar uygulama şarj performansı için interaktif olarak tasarlanmayacaktır(Örneğin, soğuran maddeler hakkında bilgilendirme yapılması).

3.8.2. Kullanıcı Girişleri

Saldırı sonrası hayatı idame için kullanıcıya gönderilen uyarı bildiriminin ardından kullanıcıdan bazı girişler alınacaktır. Bunla Şekil 12’de de gösterildiği üzere:

Aydınlanma süresi(sn): Görüş açısının kısıtlılığı olduğu veya daha karanlık saatler boyunca patlamanın gücü aydınlanma zamanı ölçülerek tahmin edilebilir[33].

Beyazlık geçtikten sonra gelen ses süresi(saniye cinsinden): Patlamanın ilk habercisi olan beyazlığın süreceği zamanda kullanıcı saymaya başlar ve ses geldiğinde durursa, elde ettiği sayı değerinin ses hızıyla çarpımı kullanıcının patlama sıfır noktasına olan uzaklığını verecektir[33].



Şekil 12: Kullanıcı Veri Giriş Ekranı

Saldırı yönü: Saldırı yönü rüzgar hesaplamasında kullanılacaktır.

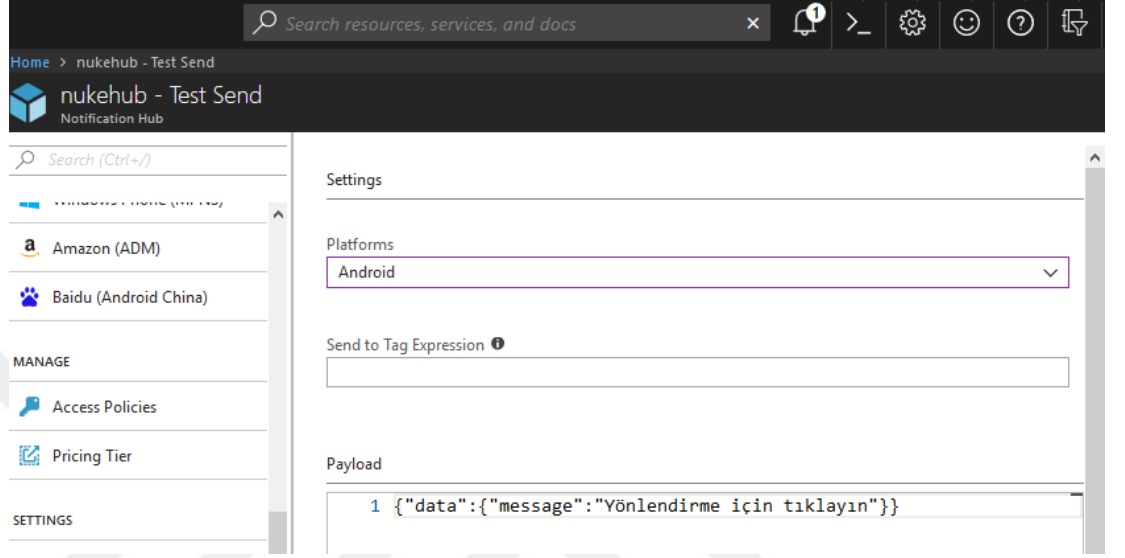
Rüzgar şiddeti: Rüzgar şiddeti anasayfada gelen meteoroloji verileri olacaktır.

Rüzgar yönü: Rüzgar yönü de yine meteorolojiden sağlanan veriyi içerecektir.

Stabil hâle gelen mantar bulutun fotoğrafı: Gövde kalınlığına göre yeryüzü veya hava patlamasına göre tahminleme gerçekleştirecektir.

3.8.3. Mobil Bildirimler

Uygulamada kullanıcılara erkenden uyarı yapabilmek adına mobil bildirimler kullanılmıştır. Bu bildirimler Microsoft'un sağladığı Notification Hub servisi altından gönderilebilmektedirler. Şekil 13'te bu servis üzerinden gönderilen test bildirim sayfası görüntülenmektedir.



Şekil 13: Mobil Bildirim Gönderme Ekranı

3.9. Saldırı Sonrası Hesaplamalar

3.9.1. YSA Veri Seti

Çalışmada veri seti olarak geçmiş nükleer patlamalar[34] ana veri kaynağı olarak kullanılmıştır. Fotoğraf kaynağı için ise Washington And Lee Üniversitesinin dijital kütüphanesi olan ALSOS'tan yararlanılmıştır. Toplanan 528 tane veri elenme işlemine tabi tutulmuştur. Bu işlemler sırasıyla:

- Kilotonu 0 olanlar çıkartıldı.
- Çalışma tipine uygun olmayanlar(denizaltı saldırısı) çıkartıldı.
- Fotoğrafları resmi resmi kaynaklardan doğrulanamayanlar çıkartıldı.
- Doğrulan fotoğraflardan, görsel olarak kullanılamayacak durumda olanlar çıkartıldı.
- Kalan fotoğraflardan bulutları stabilize olmayan görseller çıkartıldı.

- Fotoğraflardaki yabancı öğeler temizlendi(ağaç, insan).

Tüm işlemler sonucu veri seti uzunluğu 27 olarak nihai halini almış oldu. Son hali Tablo 3'te paylaşılmıştır:

Test Adı	PY(m)	Kiloton	Test Tipi
Buster-Jangle-Able	30	0,05	Atmosferik, kule
PLUMBOB-Franklin	91	0,14	Atmosferik, kule
Teapot-Wasp	232	1	Atmosferik, havadan atılan
HARDTACK II-Socorro	440	6	Atmosferik, balon
Buster-Jangle-Charlie	345	14	Atmosferik, havadan atılan
BUFFALO R1/One Tree	31	15	Atmosferik, kule
Mosaic G1	31	15	Atmosferik, kule
Dione	275	15	Atmosferik, balon
Crossroads-Able	158	19,1	Atmosferik, havadan atılan
Trinity	30	19,3	Atmosferik, kule
Buster-Jangle-Dog	432	21	Atmosferik, havadan atılan
Fat Man	503	21	Atmosferik, havadan atılan
Antler Round 3-Taranaki	300	26,6	Atmosferik, balon
Plumbob-Priscilla	210	37	Atmosferik, balon
Orion	400	72	Atmosferik, balon

Tablo 3: Çalışmada Kullanılan Veri Setinin Son Hali

Betelgeuse	600	120	Atmosferik, balon
Rigel	3	150	Atmosferik, su yüzeyinde mavnaya atılan
Sirius	10	300	Atmosferik, su yüzeyinde mavnaya atılan
Ivy-King	451	500	Atmosferik, havadan atılan
Toucan	500	594	Atmosferik, balon
Licorne	60	1000	Atmosferik, balon
Procyon	600	1200	Atmosferik, balon
Grapple Y	2350	2000	Atmosferik, havadan atılan
Canopus	600	2600	Atmosferik, balon
Hardtack I-Oak	0	8900	Atmosferik, su yüzeyinde mavnaya atılan
Ivy-Mike	11	10400	Atmosferik, yüzey
Tsar Bomba	4000	58000	Atmosferik, havadan atılan

Tablo 4: Çalışmada Kullanılan Veri Setinin Son Hali(Devamı)

3.9.2. YSA Ağının Eğitilmesi

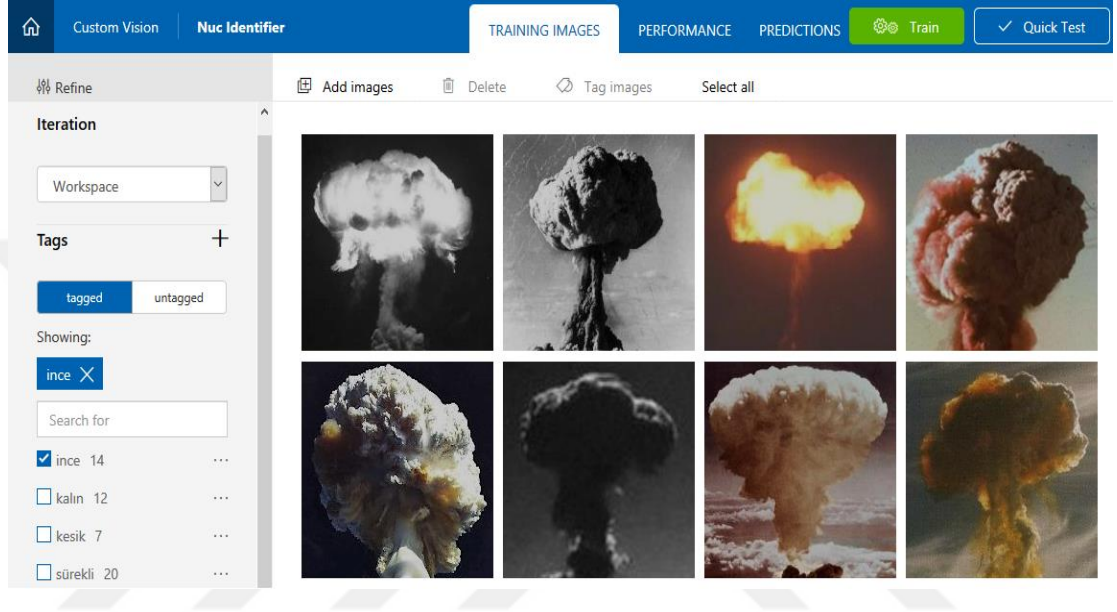
Veri setinin nihai halinde elde kalan 27 adet fotoğraf eğitilmek üzere Custom Vision Hizmeti'ne yüklenerek etiketleme yapıldı. Etiketleme işlemlerinin yapıldığı ekran Şekil 14'te gösterilmektedir.

Etiketleme yaparken sınıflandırmada kararsız kalınanlarda çift etiket uygulandı;hem kesik hem sürekli gibi(Örnek: Betelgeuse).

Fotoğraflar 2 değişik etiket ile 4 tipte etiketlendi. Bunlar mantar gövdesinin mantar kafasına oranına göre: kalın ve ince; mantar gövde sürekliliğine göre sürekli ve kesik.

Kararsız kalınan etiket tipi boş bırakıldı(Örnek: Mosaic G1).

Etiketleme sonucu oluşan veri seti Tablo 4’te paylaşılmıştır.



Şekil 14: Custom Vision Eğitim Ağı Etiketleme Ekranı

Test Adı	Etiket1	Etiket2
Buster-Jangle-Able	ince	kesik
PLUMBOB-Franklin	ince	sürekli
Teapot-Wasp	ince	kesik
HARDTACK II-Socorro	ince	kesik

Tablo 5: Etiketleme Sonucu Son Durum

Buster-Jangle-Charlie	ince	sürekli
BUFFALO R1/One Tree	ince	sürekli
Mosaic G1	kalın	
Dione	ince	kesik
Crossroads-Able	kalın	sürekli
Trinity	kalın	sürekli
Buster-Jangle-Dog	kalın	kesik
Fat Man	ince	kesik
Antler Round 3-Taranaki	ince	sürekli
Plumbob-Priscilla	ince	sürekli
Orion	ince	sürekli
Betelgeuse	kesik	sürekli
Rigel	kalın	sürekli
Sirius	kalın	sürekli
Ivy-King	kalın	sürekli
Toucan	kalın	sürekli
Licorne	ince	sürekli
Procyon	kalın	sürekli

Tablo 6: Etiketleme Sonucu Son Durum(Devamı)

Grapple Y	kalın	sürekli
Canopus	ince	sürekli
Hardtack I-Oak	kalın	sürekli
Ivy-Mike	kalın	sürekli
Tsar Bomba	ince	sürekli

Tablo 7: Etiketleme Sonucu Son Durum(Devamı)

3.9.3. Modelin Çalışma Yapısı

Eğitilen ağ TensorFlow modeli olarak mobil uygulamaya entegre edilmek üzere indirildi. Tablo 5’te ifade edilen kod bloğunda edinilen model ve etiketler TensorFlow arayüzüne gösterildi.

```
using using Org.Tensorflow.Contrib.Android;

var assets = Application.Context.Assets;

inferenceInterface = new TensorFlowInferenceInterface(assets,
"model.pb");
var sr = new StreamReader(assets.Open("labels.txt"));

var labels = sr.ReadToEnd()

        .Split('\n')

        .Select(s => s.Trim())

        .Where(s => !string.IsNullOrEmpty(s))

        .ToList();
```

Tablo 8: ImageClassifier Sınıfı Kod Bloğu 1

TensorFlow modelini test edebilmek için cihaz üzerine yüklenmiş örnek görsellerden faydalandı. Uygulamadaki bulut yardımcısı sayfasından “Bulut Seç(Test)” denilerek tahminde kullanılacak olan görsel seçildi. Seçilen görsel oluşturulan modelin anlayabilmesi için uygun formatta boyutlandırıldı.

Oluşan model için 227x227 piksel boyutuna indirgenen görselde her bir piksel için kırmızı, yeşil ve mavi değerleri belirlendi.

Bu işlemleri gerçekleştirebilmek için yüklenen görsel Bitmap tipi resime dönüştürüldü. İşlemlerim adımlandığı kod bloğu Tablo 6’da gösterilmiştir.

```
var resizedBitmap = Bitmap.CreateScaledBitmap(bitmap, 227, 227, false)

    .Copy(Bitmap.Config.Argb8888, false);

var floatValues = new float[227 * 227 * 3];

var intValue = new int[227 * 227];

resizedBitmap.GetPixels(intValue, 0, 227, 0, 0, 227, 227);

for (int i = 0; i < intValue.Length; ++i)

{

    var val = intValue[i];

    floatValues[i * 3 + 0] = ((val & 0xFF) - 104);

    floatValues[i * 3 + 1] = (((val >> 8) & 0xFF) - 117);

    floatValues[i * 3 + 2] = (((val >> 16) & 0xFF) - 123);

}
```

Tablo 9: ImageClassifier Sınıfı Kod Bloğu 2

Kullanıcıdan gelen girdiyi modelin anlayabileceği formata çektikten sonra bu girdi modele iletilen kısım Tablo 7’de gösterilmektedir.

```
var outputs = new float[labels.Count];

inferenceInterface.Feed("Placeholder", floatValues, 1, 227, 227, 3);

inferenceInterface.Run(new[] { "loss" });

inferenceInterface.Fetch("loss", outputs);
```

Tablo 10: ImageClassifier Sınıfı Kod Bloğu 3

Çıktı dizisi modeldeki tüm etiketler için değerleri sırayla ayrı ayrı doldurur.

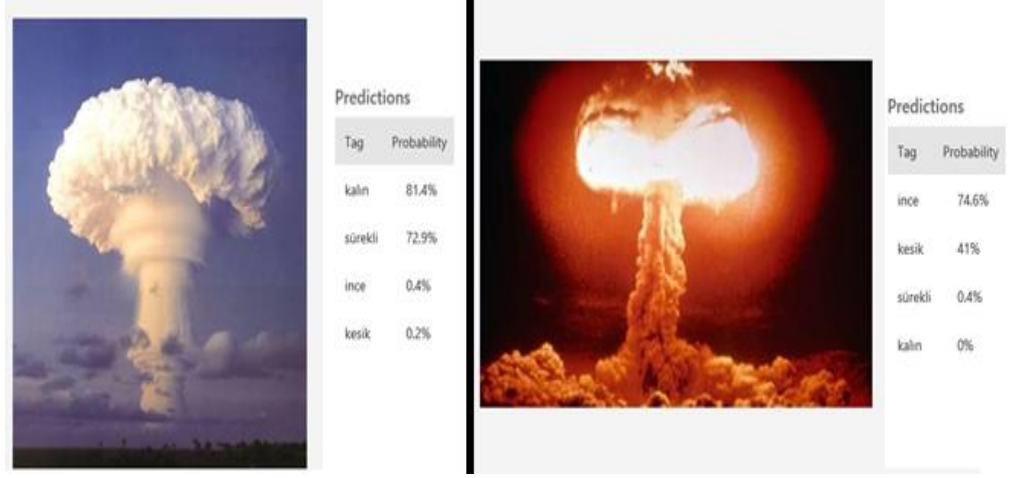
3.9.4. Eğitim Sonuçları Ve Test

Sınıflandırılan veri setinin eğitimi tamamlandığında ortaya çıkan sonuçlar(Olasılık eşiği: %44 olmak üzere): Precision: %76.3 ve Recall: %60.2 olarak ortaya çıkmıştır.

Etiket bazında eğitim sonuçları Tablo 8’de, Şekil 15 ve Şekil 16’da ise eğitim sonucu gerçekleştirilen test görüntüleri bulunmaktadır.

Etiket	Precision	Recall
ince	77.8%	43.3%
kalın	82.2%	83.3%
kesik	77.8%	44.4%
sürekli	83.8%	66.7%

Tablo 11: Etiket Bazında Eğitim Sonuçları



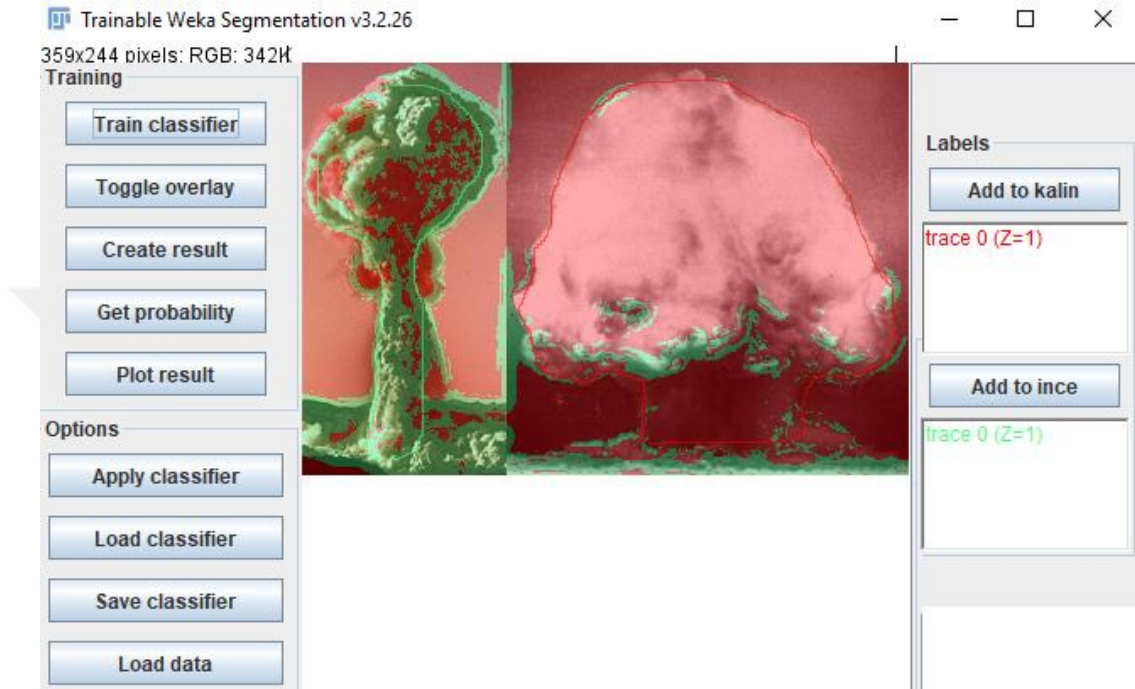
Şekil 15: Örnek Test Sonuçları



Şekil 16: Mobil Uygulama Test Sonuçları

3.9.5. Multilayer Perceptron Sonuçları

Sınıflandırma işlemi için kullanılan araç üzerinden görseller kenarlık bazında işaretlenerek seçilmiş ve veri seti oluşturulmuştur. Sınıflandırma işlemi her görsel için Şekil 17’de gösterilen ekran üzerinden tek tek gerçekleştirilmiş olup, ayrı ayrı kaydedilen veriler en sonda birleştirilmiştir.



Şekil 17: Görüntü Sınıflandırma Ekranı

Eldeki veri setinden birleştirilerek ince ve kalın olmak üzere sınıfa ayrılan ve örnek ilk 3 verisi Tablo 9’da gösterilen ve 4426 satırdan oluşan veri setinin çok katmanlı algılayıcı ile modellenmesi sonucu tahminleme başarısı %85.90 olarak ortaya çıkmıştır. Etiket bazında tahminleme başarısı Tablo 9’da gösterilmiştir. Bu veri setindeki değişkenlerden:

Mean: Görüntünün tamamı için bireysel vokselle yoğunluğunu,

Variance: Komşu vokselde olan farkı ifade etmektedir

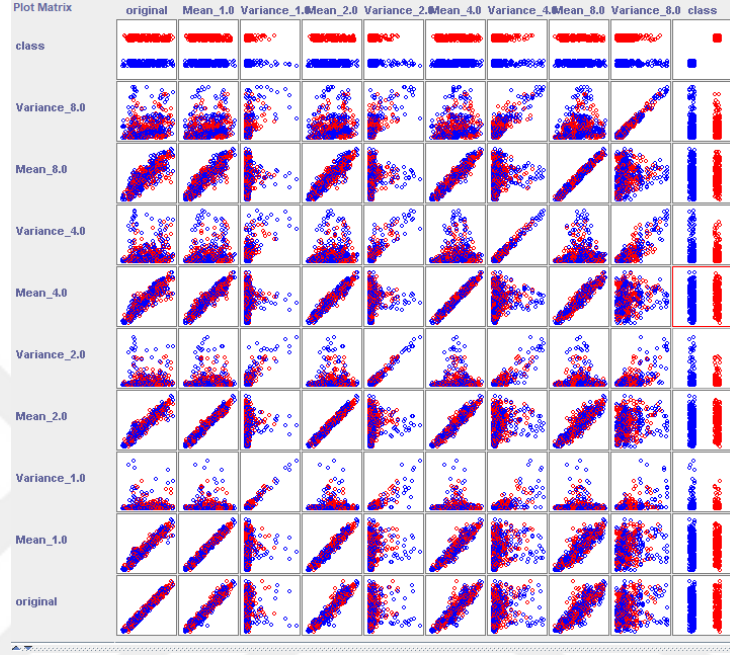
Etiket	Precision	Recall
ince	91,7%	81.3%
kalin	80%	91.4%

Tablo 12: Etiket Bazında Eğitim Sonuçları

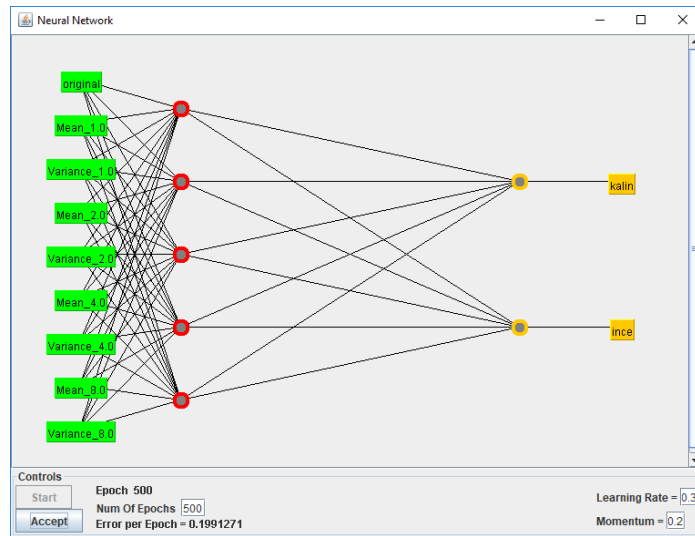
@relation segment
@attribute original numeric
@attribute Mean_1.0 numeric
@attribute Variance_1.0 numeric
@attribute Mean_2.0 numeric
@attribute Variance_2.0 numeric
@attribute Mean_4.0 numeric
@attribute Variance_4.0 numeric
@attribute Mean_8.0 numeric
@attribute Variance_8.0 numeric
@attribute class {kalin,ince}
@data
188.333328,192.333328,1,194,4,192.666672,6.666667,178.666672,50.333332,kalin
187.666672,188,0.666667,189,1,192.333328,5.333333,179.333328,36.666668,kalin
185.333328,184.666672,0.333333,185,1.333333,188,5,177,37.666668,kalin

Tablo 13: Çok Katmanlı Algılayıcı İçin Çıkarılan Veri Seti İlk 3 Örnek

Çalışma sonucu ortaya çıkan, değişkenlerin çift olarak birbirleriyle olan ilişkisini bir taban üzerinde görüntüleye Plot Matrix gösterimi Şekil 18’de, oluşan YSA modellemesi ise Şekil 19’da paylaşılmıştır. Çizimde her olası kombinasyon için bir görüntüleme vardır. Çizimdeki renklerden kırmızı olan “İnce”, mavi olan “Kalın” sınıfını içermektedir.



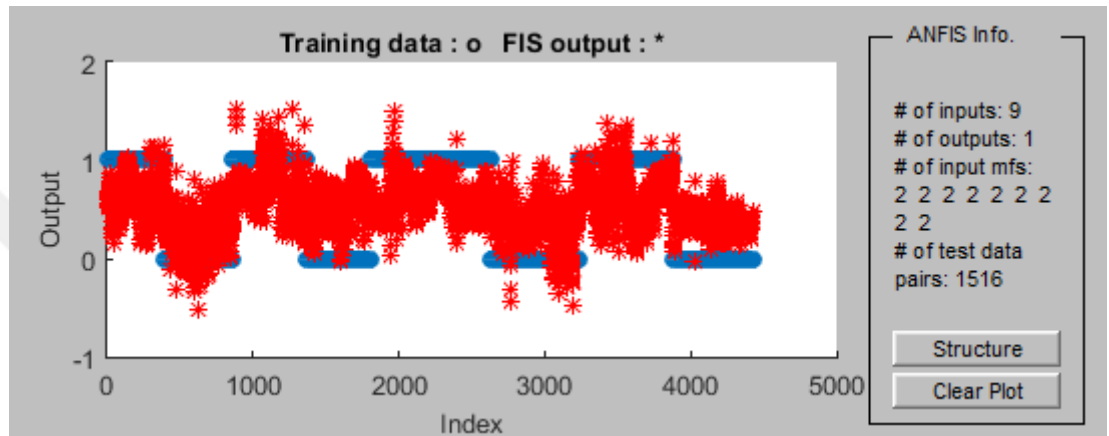
Şekil 18: Oluşan Plot Matrix



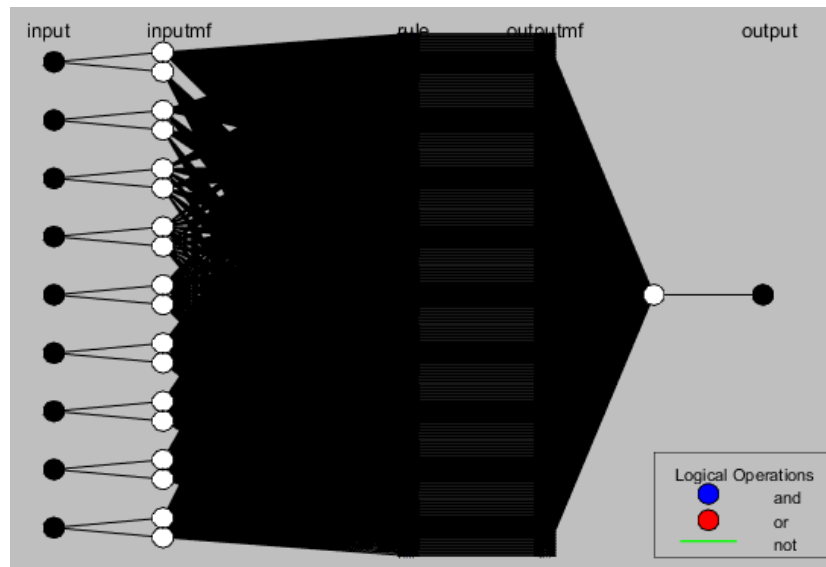
Şekil 19: Multi Layer Perceptron Kullanılarak Oluşan Ağ

3.9.6. Anfis Sonuçları

Aynı veri seti baz alınarak anfis üzerinde gerçekleştirilen çalışmada doğruluk oranı %57 olarak hesaplanmıştır. Şekil 20’de 0 ve 1 aralığında kalan kırmızı noktalar test verilerini, mavi çizgiler ise eğitim verilerini ifade etmektedir. 0 ince, 1 ise kalın’ı ifade etmektedir. Şekil 21’de ise oluşan ağ yapısı gözükmemektedir. Ağ yapısındaki girdiler Multilayer Perceptron yapısındaki girdilerle aynı olmakla beraber katman sayısı daha fazladır. Ağın eğitimlerinde üyelik fonksiyonu olarak gaussmf kullanılmıştır. Eğitim sırasında gerçekleşen iterasyon sayısı 30’dur.



Şekil 20: Anfis Sonuçları



Şekil 21: Anfis Ağ Yapısı

3.9.7. Diğer Hesaplamalar

YSA hesaplamasından gelen mantar bulut tipiyle beraber kullanıcıdan girilen veriler birleştirilerek kullanıcıya serpentinin tahmini ulaşma süresi ve patlama gücü hesaplanacaktır.

Tahmini Serpinti Varış Süresi: Bu değer bombanın gücünden bağımsız bir şekilde aynı süreyi göstermektedir. Bombanın gücü serpentinin varış süresini değiştirmemekle birlikte serpinti miktarıyla doğru orantılıdır.

Tahmini Serpinti Varış Süresi(saat): $\text{Kullanıcı Uzaklığı(km cinsinden)} / \text{Aktif Rüzgar Hızı(km/s cinsinden)}$ [33].

3.10. Uygulama Mimarisi

3.10.1. Çalışma Ortamı

Çalışma aşağıdaki özelliklere sahip ortamlarda gerçekleştirilmiştir:

- Intel i5 6200U, 12 GB Ram, 1 TB SSD ve 720P Ekran Windows 10 bilgisayar
- Android(minimum 7.1 OS) mobil cihaz
- Visual Studio 2017 Community
- Minimum 114 Kbps mobil internet bağlantısı

3.10.2. Kullanılan Eklentiler

Gorilla Player: Xaml önizleyicisi

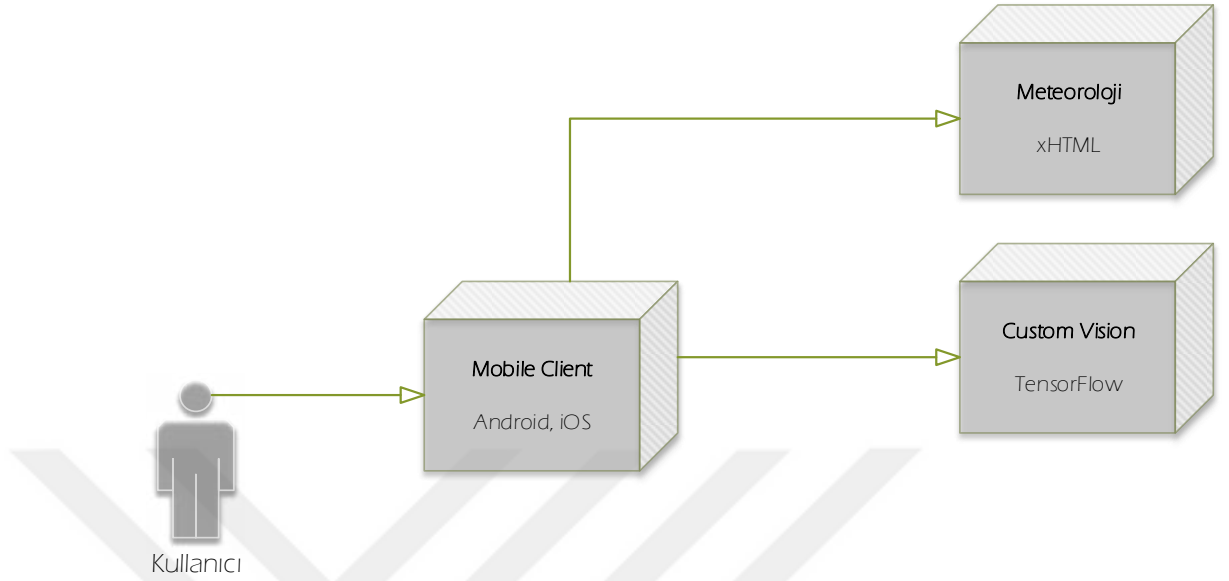
Xam.Android.TensorFlow: Android için TensorFlow kütüphanesi

Xam.Plugin.Media: Kamera ile fotoğraf çekme ve kaydetme işlemleri.

Xamarin.Azure.NotificationHubs.Android: Azure portal üzerinden Android uygulamaya bildirim gönderebilmek için.

3.10.2. Deployment Diyagramı

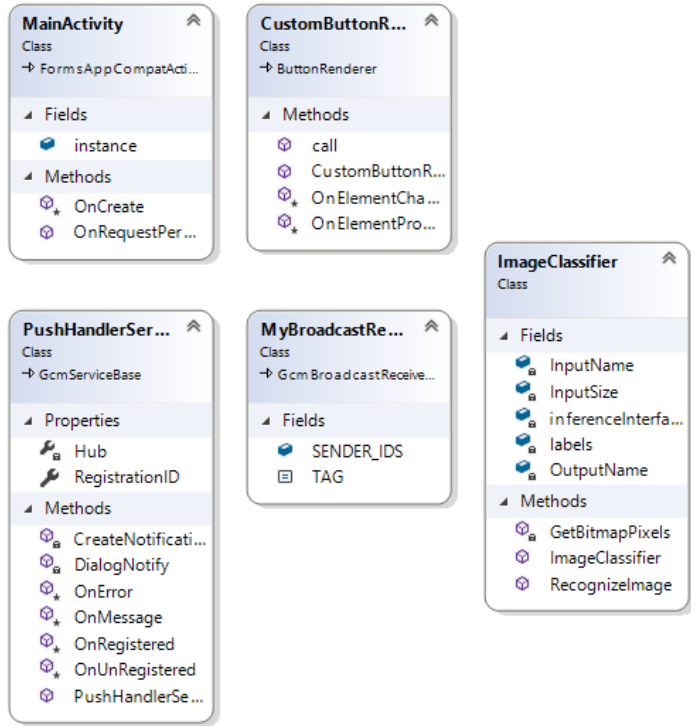
Uygulamaya ait deployment diyagramı Şekil 22’de paylaşılmıştır:



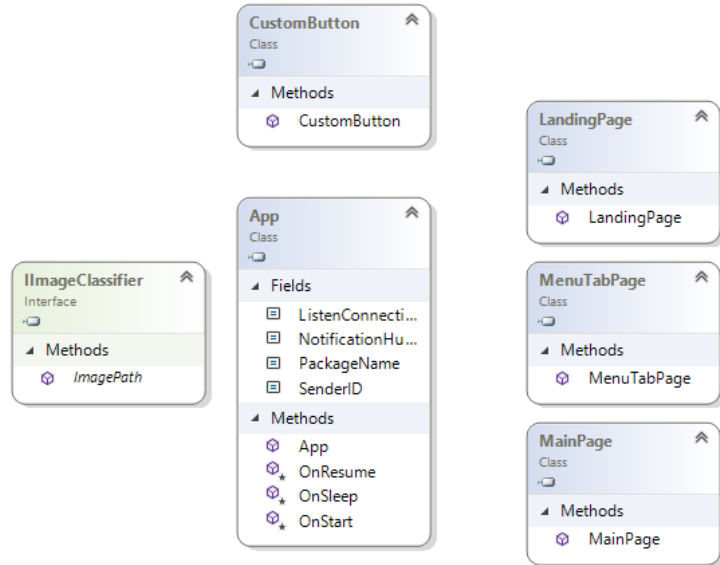
Şekil 22: Deployment Diyagramı

3.10.3. Sınıf Diyagramı

Uygulamada kullanılan sınıflara ait diyagramlar Android katmanı için Şekil 23’te, Xamarin Forms katmanı için ise Şekil 24’te paylaşılmıştır.



Şekil 23: Android Katmanı Sınıf Diyagramı

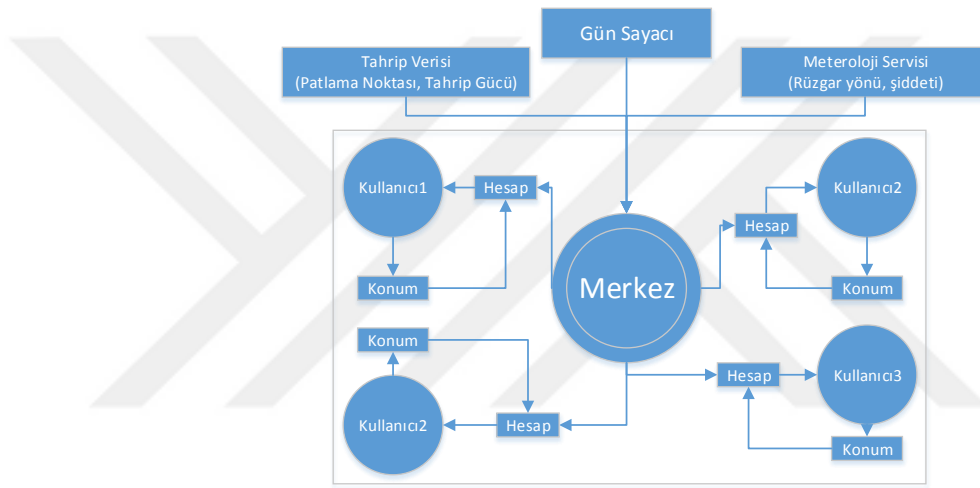


Şekil 24: Xamarin Forms Katmanı Sınıf Diyagramı

3.10.4. Merkezi Yapı

Saldırı sonrası hesaplamalarda kullanılacak olan hesaplamaların cihaz batarya ömrünü yoracağı öngörülmüştür. Bundan dolayı saldırı sonrası hesaplama kısmını cihazdan bağımsız merkezi bir sistemde yapıp cihazlara ayrı ayrı gönderilmesi de Şekil 25’te gösterildiği değerlendirmeye üzere alınmıştır.

Bu yapıda tahrip bilgisi ve meteoroloji servisini harmanlanarak, gün sayacını da merkezde tutarak kullanıcılara ortak bir değer gönderecekti. Değeri edinen kullanıcılar ise cihazın konumundan hareketle nükleer serpentinin bulundukları bölgeye ne kadar ulaştığını tahmini olarak görüntüleyebilecekti.



Şekil 25: Çalışma Kapsamından Çıkarılan Merkezi Hesap Sistemi

Ancak böyle bir felaket akabinde bu merkezi yapıya internet üzerinden erişim mümkün olmayacağından dolayı bu seçenek uygulama yapısına dahil edilmemiştir.

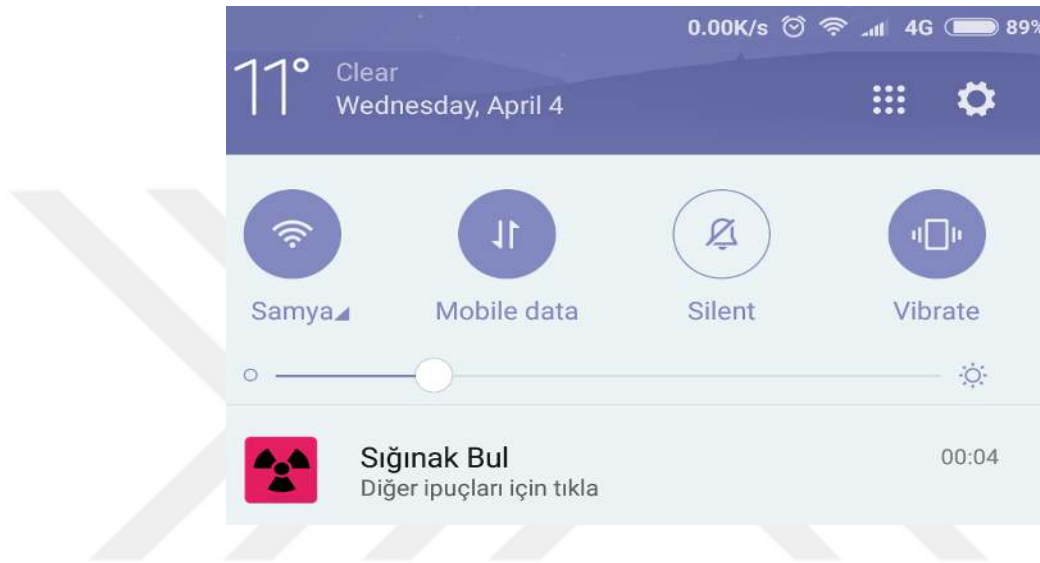
3.11. Saldırı Sonrası Bilgilendirmeler

Yukarıdaki hesaplama aşamalarından sonra kullanıcıya kılavuzluk yapılabilmesi adına temel bilgiler de sabit olarak sağlanacaktır. Bu bilgiler aşama aşama kullanıcıya uygulama içerisinde iletilecektir. Sadece kullanıcıya saldırı öncesi bildirim-eğer mevcutsa- Şekil 26’daki gibi iletilecektir.

- Erken uyarı süresince bulabildiğin su, yiyecek, varsa el radyosu ve el fenerini bir poşete doldur ve sığınağa(bodruma) in.

- Açık alandaysan gelen ışığa kesinlikle bakma, gözlerini ışıktan korumak için etrafta siper alabilecek herhangi bir şey bulun.

- Işık başlarken sayın, saymayı ses geldiğinde durdurun, elde ettiğiniz sayıyı “Sesin geldiği saniye” alanına girin.(Ses hızı havada ortalama 340m/sn’dir[35]).



Şekil 26: Örnek Bildirim Ekranı

- Işık bittikten sonra mantar bulutunu gözlemleyin ve mantar stabil hale geldiğinde hava veya yer saldırısı için tahmin edilmek üzere uygulamadan fotoğrafını çekin.

- Yardım gelene kadar sığınaktan ayrılmak zorunda kalırsanız döndüğünüzde bulunan mevcut su ve sabunla temizlenin. Su ve sabunun radyoaktif temizleyici özelliği bulunmaktadır[36].

- Sığınak hava girebilecek yerlerini ıslak bezlerle kapatın. Varsa klima gibi hava ünitelerini de kapatın.

- Sığınak içerisinde bulabildiğiniz soğurucu maddeleri kullanarak kendinize alan oluşturun.

- Soğurma sırasına göre örnek maddeler: Kurşun, plastik, beton, tuğla[37].
- Tuvalet ihtiyacını kesinlikle sığınakta gör.
- Dışarıya çıkarken saç köklerin dahil kapatabildiğinde her yerini kapat. Saç için varsa kapşonlu kıyafet kullanın.
- Kesinlikle toprak olmayan, asfalt gibi bir yoldan ilerleyin.



4. SONUÇ

Nükleer saldırı sonrası patlama ve patlamanın ısı alanından sağ çıkan kullanıcılar için, radyoaktif serpintiden kaçınmalarını sağlamak amacıyla ceplerinde taşıyabilecekleri bir mobil cihaz tarafından yönlendirilmeleri fikri araştırma ve uygulama konusu için ilgi çekici gelmiştir. Çalışmada yola çıkılan temel nokta, böyle bir saldırının serpinti aşamasından en az zararla kurtulabilmek belli birtakım matematiksel hesaplamalarla ve kestirmelerle mümkünken tüm kullanıcıların bu şekilde profesyonel eğitim almadığı ve yardıma ihtiyaç duyabilecekleri düşüncesi olmuştur. Bütün matematiksel hesaplamaları onların yerine yapan ve onları yönlendirecek bir yardımcı asistan çalışması, pratikte çok ihtiyaç olmayacaksa da teoride kullanıcılara faydalı olacağı düşünülmüştür. Kullanıcılar bu hesaplamaları ve özellikleri eğitimini almasalar dahi uygulamadaki yönergeleri takip ederek daha iyi durumda olacakları kesindir.

Kullanıcının patlamadan sonra oluşan mantar bulutu yorumlayabilmesi için geçmiş saldırılardan elde edilen veri setinin görselleştirilerek eğitim modeline uygulanması sonucu mantar bulutu tipinin tespitinde bulunan başarı oranı türlü iterasyonlar sonucu %76.3 çıkmıştır. Başarı oranı artması için görseller tekrar tekrar etiketlenmiş ve yeniden eğitilmişlerdir. Çalışmada yapay sinir ağlarının yeri sadece görsel tanımlamada kullanılmıştır. Görsel tahmininden önceki ve sonraki safhalarda kesin kurallar çerçevesinde hesaplanacak olan “serpintinin kullanıcıya ulaşması oranı” oran orantı hesaplamaları kullanıcıya gösterilecektir. Bulut tahmininde ikinci hesaplama yöntemi olarak multilayer perceptron da kullanılmıştır. Multilayer perceptron uygulanırken ilk YSA algoritmasındaki veri setinden farklı olarak sınıflandırma kalın ve ince olarak ikiye indirgenmiş ve görseller voksel bazında elle sınıflandırılmıştır. Mean ve variant özellikleriyle sınıflandırılan görsellerden çıkan veri seti multilayer perceptron kullanılarak hesaplanmış ve sonuç olarak tahminleme oranı %85.90 çıkmıştır. Aynı veri seti için Anfis ağında yapılan hesaplamada ise bu oran %57 olarak gerçekleşmiştir. Anfis’te çıkan sonuç diğer iki yöntemle göre düşük olarak ortaya çıkmıştır.

Görseller etiketlenirken patlamadan sonra stabil hale gelmiş olan mantar bulutunun gövdesinin kafasına oranına göre ve mantar gövde sürekliliğine göre ayrılmıştır. Eldeki veri setinde uyuşan verilerin azlığı nedeniyle mantar kirliliğine göre etiketleme yapılmamıştır. Şehirde yüzeye yakın gerçekleşen patlamalarda etrafta oluşan tozdan dolayı mantarda koyuluk oluşmaktadır. Ancak veri setindeki testlerin büyük bir kısmı şehir dışında gerçekleştiğinden ötürü veri bütünlüğünü bozmamak adına koyuluk oranı etiketlenmemiştir.

Çalışma ileride başka yöntemler kullanılarak genişletebilir, veya başka bir araştırmanın konusunda da kullanılabilir. Alternatif yapılacak olan çalışmalarda yöntem olarak nükleer saldırının yarattığı sismik etkileri[38] karar destek mekanizması kullanabilmek; diğeri ise radarlar aracılığıyla bulut mantar boyunu ölçümlemek olabilir[33]. Bunlara yardımcı olarak ise “rule of thumb” olarak adlandırılan baş parmak aracılığıyla mantar bulut tahminlemesi kullanılabilir. Ek olarak çalışmanın 3.10.4. numaralı bölümde bahsedilen merkezi yapı da diğer çalışmalara konu olabilir. Yapılan araştırmalar yöntemlerin de benzer çalışmalarda konu olarak kullanılabileceğini ortaya koymuştur[39].

Uygulamanın bilgilendirici kısımlarını dikkate alarak kullanıcılar bu konu hakkında sivil savunma eğitimi almış kadar saldırılar hakkında bilgi sahibi olacaklarından ötürü bu durum sivil savunma ekiplerinin işine de fayda sağlayacağı düşünülmüştür.

KAYNAKLAR

- [1] Y. Gönül *et al.*, “Yapay sinir ağları ve klinik araştırmalarda kullanımı,” *Genel Tıp Derg.*, vol. 25, pp. 104–111, 2015.
- [2] K. Tok, “Hastalık Teşhisi İçin Bir Yapay Sinir Ağları Yazılımının Tasarlanması Ve Gerçekleştirilmesi,” Selçuk Üniversitesi, 2016.
- [3] A. Aslantaş, “Kemik Metastazlarının Görüntü İşleme Ve Yapay Sinir Ağları Yöntemleri İle Tespiti,” Sakarya Üniversitesi, 2015.
- [4] M. Coşkun, “Yapay Sinir Ağları Yardımı İle Eeg Tabanlı Anestezi Derinliği Tahmini İçin Yöntemler,” Balıkesir Üniversitesi, 2013.
- [5] M. İstanbullu, “Yapay Sinir Ağları Ve Destek Vektör Makineleri İle Kemik Erimesinin Teşhisi,” İstanbul Teknik Üniversitesi, 2013.
- [6] N. Yalçın, “Sezgisel Algoritma Öğrenmeli Yapay Sinir Ağları İle Epilepsi Hastalığının Teşhisi,” Selçuk Üniversitesi, 2012.
- [7] G. Toğa, “Koroner Arter Hastalarında Stent İmplantasyonu Sonrası Ölümünün Yapay Sinir Ağları İle Modellenmesi,” Erciyes Üniversitesi, 2012.
- [8] Ö. Özkan, M. Yıldız, and E. Köklükaya, “Fibromiyalji Sendromunun Teşhisinde Kullanılan Laboratuvar Testlerinin Sempatik Deri Cevabı Parametreleriyle Desteklenerek Teşhis Doğruluğunun Arttırılması,” *SAÜ Fen Bilim. Derg.*, vol. 15, no. 1, p. 6, 2011.
- [9] F. Aşkın, “Sol Atriyum Çapının Temel Bileşener Regresyonu, Kısmi En Küçük Kareler Regresyon ve Yapay Sinir Ağları İle,” Fırat Üniversitesi, 2011.
- [10] Y. M. JWMAH, “Telemedikal Elektrokardiyografi Sinyallerinin Yapay Sinir Ağları İle Değerlendirilmesi,” İstanbul Üniversitesi, 2010.
- [11] H. Aydemir, “Horlama Kayıtlarından Doğrusal Öngörüm Kodlaması Ve Yapay Sinir Ağları Kullanılarak Apneli Hastaların Ayırt Edilmesi,” Ankara Üniversitesi, 2009.
- [12] Dr.R.K.SelvaKumar and M. D. Karthik, “Analysing Parameter for Nuclear Explosion Using Digital Images,” *Adv. Inf. Sci. Serv. Sci.*, vol. 2, no. 4, p. 59, 2010.
- [13] S. Abusaleh, A. Mahmood, K. Elleithy, and S. Patel, “A Novel Vision-Based Classification System for Explosion Phenomena,” *J. Imaging*, p. 1, 2017.
- [14] D. T. Schmitt and G. L. Peterson, “Feature detection and matching on atmospheric nuclear detonation video,” *IET Comput. Vis.*, p. 3, 2016.
- [15] R. C. Slaughter, T. R. Peery, and J. W. McClory, “Two-Dimensional Temperature Analysis of Nuclear Fireballs Using Digitized Film,” *J. Appl. Remote Sens.*, 2015.
- [16] H. M. Kristensen, “The Status of Nuclear Weapons: Arsenals, Modernizations, and Operations,” in *17th Castiglione International Conference*, 2017, p. 2.

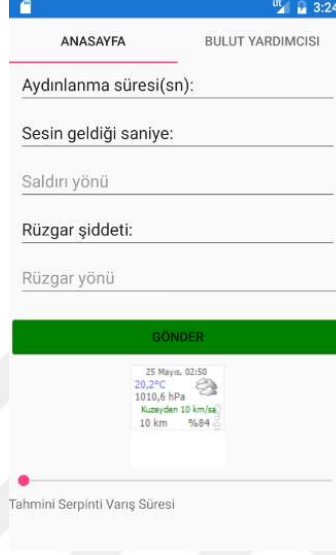
- [17] E. Sıtkı and S. Güvenç, “NATO’ nun Füze Savunma Sistemi ve Türkiye,” *Ortadoğu Anal.*, vol. 4, no. 40, p. 20, 2012.
- [18] S. Glasstone and P. Dolan, “The Effects of Nuclear Weapons,” *Eff. Nucl. weapons*, 1977.
- [19] *Nevada Test Site Guide B*. National Nuclear Security Administration Nevada Site Office, 2005.
- [20] W. C. Hinds, A. Ashley, N. J. Kennedy, and P. Bucknam, “Conditions For Cloud Settling And Rayleigh-Taylor Instability,” *Aerosol Sci. Technol.*, vol. 36, no. 12, p. 1133, 2010.
- [21] U. S. A. F. US Army, United States Marine Corps, United States Navy, “Multiservice Tactics, Techniques, And Procedures For Chemical, Biological, Radiological, And Nuclear Contamination Avoidance,” p. G-9,G-11, 2006.
- [22] M. L. Stone and G. P. Banner, “Radars for the Detection and Tracking of Ballistic Missiles , Satellites , and Planets,” *Lincoln Lab. J.*, vol. 12, no. 2, p. 223, 2000.
- [23] L. Caston *et al.*, “The Future of the U.S. Intercontinental Ballistic Missile Force,” *Proj. Air Force*, p. 25, 2014.
- [24] “5902 Afet ve Acil Durum Yönetimi Başkanlığının Teşkilat Ve Görevleri Hakkında Kanun,” *Resmi Gazete Sayısı*, vol. 48, no. 27261, pp. 10453–10454, 2009.
- [25] Office of Civil Defense Dept. of Defense and R. S. McNamara, *Fallout Protection What Do Know And Do About Nuclear Attack*. ABD, 1961.
- [26] S. Dönmez, “Radyasyon Tespiti ve Ölçümü,” *Nükleer Tıp Semin.*, no. 3, p. 174, 2017.
- [27] S. Rajasekaran and G. A. V. Pai, *Neural Networks, Fuzzy Logic, And Genetic Algorithms*. Hindistan: PHI Learning, 2011.
- [28] C. Petzold, *Creating Mobile Apps with Xamarin.Forms Preview Edition 2 (Developer Reference)*. ABD: Microsoft Press, 2015.
- [29] Meteoroloji, “Sitemize Link Vermek İçin.” [Online]. Available: <https://www.mgm.gov.tr/site/link-vermek-icin.aspx>. [Accessed: 05-Nov-2017].
- [30] A. Fandongo, *Mastering TensorFlow 1.x: Advanced Machine Learning And Deep Learning Concepts Using Tensorflow 1.x And Keras*. İngiltere: Packt Publishing, 2018.
- [31] R. Kirkby and E. Frank, “WEKA Explorer User Guide for Version 3-4-3.” University of Waikato, Yeni Zelanda, p. 2, 2004.
- [32] I. Arganda-Carreras *et al.*, “Trainable Weka Segmentation: A Machine Learning Tool For Microscopy Pixel Classification,” *Bioinformatics*, vol. 33, no. 15, pp. 2424–2426, 2017.

- [33] US Marine Corps Department Of The Army Commandant Headquarters, "Nuclear Contamination Avoidance," pp. 2-7,3-5,3-12,3-23, 1994.
- [34] N.-O. Bergkvist and R. Ferm, *Nuclear Explosions 1945-1998*. İsveç: Defence Research Establishment (Försvarets forskningsanstalt (FOA)), Division of Systems and Underwater Technology, 2000.
- [35] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous*. Wiley-Blackwell, 2015.
- [36] Army Department Of The USA, *Handling and Disposal of Unwanted Radioactive Material*. ABD: University Of Virginia Library, 1991.
- [37] C. Başığit and A. Kaçar, "Bazı Yapı Malzemelerinin Radyasyon Tutuculuk Özellikleri," *Süleyman Demirel Üniversitesi Fen Bilim. Enstitüsü Derg.*, vol. 10, no. 2, p. 310, 2006.
- [38] A. B. Bolt, *Nuclear Explosions And Earthquakes: The Parted Veil*. ABD: W H Freeman & Co, 1977.
- [39] FEMA, "Unit 4 Nuclear Threat FEMA Training," ABD: US Federal Emergency Management Institute.

EKLER

Ek 1. Ekran Görüntüleri

Mobil Uygulama Anasayfa



Mobil Uygulama Bulut Yardımcısı Sayfası



TensorFlow Test Ekran Görüntüsü



or

[Browse local files](#)

File formats accepted: jpg, png, bmp
File size should not exceed: 4mb

Predictions

Tag	Probability
ince	79.6%
sürekli	63.3%
kesik	0.5%
kalin	0%

Resim Sınıflandırma Ekran Görüntüsü

Trainable Weka Segmentation v3.2.26

548x252 pixels: RGB: 539H

Training

[Train classifier](#)

[Toggle overlay](#)

[Create result](#)

[Get probability](#)

[Plot result](#)

Options

[Apply classifier](#)

[Load classifier](#)

[Save classifier](#)

[Load data](#)

[Save data](#)

[Create new class](#)

[Settings](#)





Labels

[Add to ince](#)

trace 0 (Z=1)

[Add to kalın](#)

trace 0 (Z=1)

ÖZGEÇMİŞ

1991 yılında İstanbul’da doğdu. İstanbul’da Taşdelen İMKB Anadolu Teknik Lisesi’ni bitirdikten sonra Samsun 19 Mayıs Myo Bilgisayar Programcılığı ve ardından İstanbul’da Beykent Üniversitesi Yazılım Mühendisliğini bitirdi. Askerlik görevinden sonra özel bir firmada T-SQL üzerine başladığı kariyerine mobil yazılım geliştirici olarak devam etmektedir.



Sermet KÖSE