

**T.C.
AKDENİZ ÜNİVERSİTESİ**



TINYML VE UÇ BİLGİ İŞLEME İLE KARGO DÜŞÜŞ TESPİTİ

Hüseyin Oğuzalp AKGÜL

**FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI
YÜKSEK LİSANS TEZİ**

OCAK 2024

ANTALYA

**T.C.
AKDENİZ ÜNİVERSİTESİ**



TINYML VE UÇ BİLGİ İŞLEME İLE KARGO DÜŞÜŞ TESPİTİ

Hüseyin Oğuzalp AKGÜL

FEN BİLİMLERİ ENSTİTÜSÜ

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI**

YÜKSEK LİSANS TEZİ

OCAK 2024

ANTALYA

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TINYML VE UÇ BİLGİ İŞLEME İLE KARGO DÜŞÜŞ TESPİTİ

Hüseyin Oğuzalp AKGÜL

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

Bu tez 02/02/2024 tarihinde jüri tarafından Oybirliği / Oyçokluğu ile kabul edilmiştir.

Dr. Öğr. Üyesi Yalçın ALBAYRAK (Danışman)

Prof. Dr. Selçuk HELHEL

Dr. Öğr. Üyesi Mehmet Erhan ŞAHİN

ÖZET

TINYML VE UÇ BİLGİ İŞLEME İLE KARGO DÜŞÜŞ TESPİTİ

Hüseyin Oğuzalp AKGÜL

Yüksek Lisans Tezi, Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Yalçın ALBAYRAK

Ocak 2024; 34 sayfa

Bu çalışmada, nakliye sırasında kargoda meydana gelebilecek olası hasarları tespit etmek ve izlemek için bir sistem geliştirmeye odaklanılmıştır. Sistem, veri toplamak, analiz etmek ve iletmek için sensörlerin, mikrodenetleyicilerin, makine öğrenimi modellerinin ve iletişim modüllerinin kullanımını içermektedir.

Tezde DS1307 RTC modülü, I2C iletişim protokolü, mikro SD kart modülü, SPI iletişim protokolü, ESP8266 WiFi modülü ve UART iletişim protokolü gibi çeşitli elektronik modüllerin ve iletişim protokollerinin kullanımı özetlenmektedir. Ayrıca, STM32F4 Discovery geliştirme kartı, STM32CubeIDE, STM32CubeMX, X-CUBE-AI paketi gibi geliştirme araçları ile ilgili bilgi verilmiştir.

Çalışmada, kargonun hareketi sırasında veri toplamak ve düşüş durumlarını tespit etmek için ivmeölçer kullanılan bir cihaz geliştirilmiştir. Toplanan verilerde sapmaları engellemek için Kalman filtresi kullanılmış ve yapay sinir ağı ile model eğitilmiştir. TinyML ile eğitilen model gömülü sistem üzerine yüklenmiştir. Ölçülen değerler yerel olarak cihazın üzerinde eğitilen model sayesinde sınıflandırılarak düşüş durumu tespit edilmektedir. Eğitilmiş model ile %97 başarı ile düşme doğru tahmin edilmiştir. Cihaz üzerinde bu tespitin yapılması ile uç bilgi işleme sağlanmış ve bulut yükü azaltılarak gerçek zamanlı algılama sağlanmıştır.

ANAHTAR KELİMELER: Düşüş tespiti, TinyML, Uç bilgi işleme, Yapay sinir ağı

JÜRİ: Dr. Öğr. Üyesi Yalçın ALBAYRAK

Prof. Dr. Selçuk HELHEL

Dr. Öğr. Üyesi Mehmet Erhan ŞAHİN

ABSTRACT

CARGO DROP DETECTION WITH TINYML AND EDGE COMPUTING

Hüseyin Oğuzalp AKGÜL

MSc. Thesis in Department of Electric-Electronic Engineering

Supervisor: Asst. Prof. Dr. Yalçın ALBAYRAK

January 2024; 34 pages

This study focuses on developing a system to detect and monitor potential damage to cargo during transportation. The system involves the use of sensors, microcontrollers, machine learning models and communication modules to collect, analyse and transmit data.

The thesis summarizes the use of various electronic modules and communication protocols such as DS1307 RTC module, I2C communication protocol, micro SD card module, SPI communication protocol, ESP8266 WiFi module and UART communication protocol. In addition, information about development tools such as STM32F4 Discovery development board, STM32CubeIDE, STM32CubeMX, X-CUBE-AI package is given.

In the study, a device using accelerometers was developed to collect data during the movement of the cargo and to detect fall situations. A Kalman filter was used to prevent deviations in the collected data and the model was trained with artificial neural networks. The model trained with TinyML was loaded onto the embedded system. The measured values are classified locally thanks to the model trained on the device and the fall is detected. The trained model correctly predicted the fall with 97% success. By making this determination on the device, edge information processing is provided and cloud load is reduced and real-time detection is provided.

KEYWORDS: Drop detection, TinyML, Edge computing, Artificial neural network

COMMITTEE: Asst. Prof. Dr. Yalçın ALBAYRAK

Prof. Dr. Selçuk HELHEL

Asst. Prof. Dr. Mehmet Erhan ŞAHİN

ÖNSÖZ

Çalışma sürecinde bilgi ve deneyimlerini benimle paylaşan, tez konumun belirlenmesinde rehberlik eden, öneri ve yardımlarıyla tezime önemli katkı sağlayan değerli danışmanım, Dr. Öğr. Üyesi Yalçın ALBAYRAK'a içten teşekkürlerimi sunar, saygılarımı iletiyorum.

Tez çalışmam boyunca beni destekleyen ve her aşamada güvenen eşim Gonca'ya, eğitim-öğretim ve çalışma hayatımda bana desteklerini esirgemeyen aileme minnettarlığımı ve teşekkürlerimi sunarım.



İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ	iii
AKADEMİK BEYAN	vi
SİMGELER VE KISALTMALAR.....	vii
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ	x
1. GİRİŞ	1
2. KAYNAK TARAMASI	3
3. MATERYAL VE METOT	6
3.1 DS1307 RTC Modülü	6
3.2 I2C Haberleşme Protokolü	6
3.3 Mikro SD Kart Modülü	7
3.4 SPI Haberleşme Protokolü	8
3.5 ESP8266 Wifi Modülü	9
3.6 AT Komutları	10
3.7 UART Haberleşme Protokolü	10
3.8 STM32F4 Discovery Geliştirme Kartı	11
3.9 STM32CubeIDE ve STM32CubeMX.....	12
3.10 X-CUBE-AI Paketi.....	13
3.11 Python Programlama Dili	14
3.12 Jupyter Notebook	14
3.13 TensorFlow Kütüphanesi	15
3.14 Pandas, Numpy, Matplotlib Kütüphaneleri	15
3.15 Java Programlama Dili	16
3.16 Java Spring Boot ve PostgreSQL	16
3.17 Java Script Programlama Dili.....	16
3.18 Kalman Filtresi	17
4. BULGULAR VE TARTIŞMA	18
4.1 Veri Seti Oluşturma.....	19
4.1.2 Cihazın tasarımı ve verilerin kayıt edilmesi	19

4.1.3 Verilerin filtreden geçirilmesi.....	22
4.1.4 Veri setini oluşturulması.....	24
4.2 Model Eğitimi ve Kullanımı.....	25
4.2.2 Verilerin bölünmesi ve yapay sinir ağı modeli.....	26
4.2.3 Modelin dönüştürülmesi ve gömülü sisteme aktarılması	28
4.3 Sonuçların Sunucuya Gönderilmesi ve Kullanıcı Arayüzü.....	29
4.3.2 Sunucunun oluşturulması	29
4.3.3 Sunucuya düşüş verilerinin gönderilmesi.....	30
4.3.4 Kullanıcı arayüzünün oluşturulması	31
6. SONUÇLAR	32
7. KAYNAKLAR	33
ÖZGEÇMİŞ	

AKADEMİK BEYAN

Yüksek Lisans Tezi olarak sunduğum “Tinyml Ve Uç Bilgi İşleme İle Kargo Düşüş Tespiti” adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak yazıldığını belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

02/02/2024

Hüseyin Oğuzalp AKGÜL

İmzası

SİMGELER VE KISALTMALAR

Simgeler

Tezde ondalık yazım olarak “.” işareti kullanılmıştır. (Örneğin: 21.01)

Kisaltmalar

IoT	: Nesnelerin interneti (Internet of Things)
JVM	: Java sanal makinesi (Java virtual machine)
MISO	: Ana giriş köle çıkış (master input slave output)
MOSI	: Ana çıkış köle giriş (master output slave input)
ReLU	: Doğrultulmuş Lineer Birim (rectified linear unit)
RTC	: Gerçek zamanlı saat (Real-time clock)
SCL	: Seri saat hattı (Serial clock line)
SCLK	: Seri saat (Serial clock)
SDA	: Seri veri hattı (Serial data line)
SPI	: Seri çevre arabirimi (Serial peripheral interface)
SS	: Köle seçimi (Slave select)
TinyML	: Minik makine öğrenimi (Tiny Machine Learning)
UART	: Evrensel asenkron alıcı verici (Universal asynchronous receiver transmitter)
UTİKAD	: Uluslararası Taşımacılık ve Lojistik Hizmet Üretenleri Derneği

ŞEKİLLER DİZİNİ

Şekil 1.1. Kargo hasar türleri	1
Şekil 2.1. Düşme algılama sisteminin şeması.....	3
Şekil 3.1. Gerçek zamanlı saat modülü.....	6
Şekil 3.2. I2C kullanan cihazların bağlantı şeması	7
Şekil 3.3. SD kart modülü.....	7
Şekil 3.4. SPI kullanan cihazların haberleşme protokolü	8
Şekil 3.5. ESP8266 wifi modülü.....	9
Şekil 3.6. İki UART cihazın bağlantı şeması.....	11
Şekil 3.7. STM32F4 Discovery geliştirme kartı	12
Şekil 3.8. Mikrodenetleyici pinleri yapılandırma ekranı	13
Şekil 3.9. X-CUBE-AI paketi ekleme ekranı.....	14
Şekil 3.10. Jupyter Notebook örnek bir kod bloğu	15
Şekil 4.1. Çalışmanın genel sistem mimarisi	18
Şekil 4.2. Veri setinin elde edilme aşamaları.....	19
Şekil 4.3. Bağlantı için pinlerin tanımlanması.....	20
Şekil 4.4. Tasarlanan cihazdaki pin bağlantı şeması.....	21
Şekil 4.5. Cihazın tamamlanmış halinin görseli	21
Şekil 4.6. Durağan cismin filtresiz ivme verileri	22
Şekil 4.7. Durağan ivme verisinin kalman filtreli ve filtresiz hali	23
Şekil 4.8. Hareketli ivme verisinin filtreli ve filtresiz hali	24
Şekil 4.9. Normal taşımada x, y ve z eksenleri ivme ölçer verileri grafikleri; a); b) sabit durumdaki veri grafiği c); d) hareket halindeki veri grafiği	24
Şekil 4.10. Düşüş anında x, y ve z eksenleri ivme ölçer verileri; a); b); c); d) kargo kutusunun farklı yönlerdeki düşüş ivme verileri.....	25
Şekil 4.11. Model eğitim ve kullanım aşamaları	25
Şekil 4.12. Kullanılan yapay sinir ağı temsili şeması	27
Şekil 4.13. Çıktı sonuçları örneği	27
Şekil 4.14. Eğitim ve doğrulama kayıpları grafiği.....	28
Şekil 4.15. Yapay sinir ağı ağırlık katsayıları.....	28
Şekil 4.16. Verilerin web arayüzünde gösterilme aşamaları.....	29
Şekil 4.17. Server istek adresleri.....	30

Şekil 4.18. Arayüz giriş ekranı	31
Şekil 4.19. Düşüş verilerinin gösterildiği ekran.....	31



ÇİZELGELER DİZİNİ

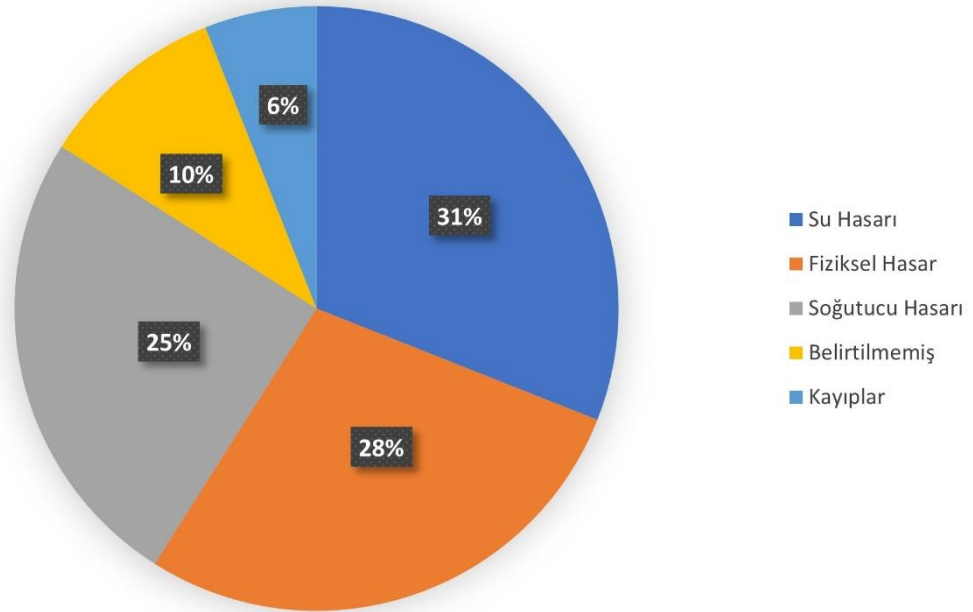
Çizelge3.1. ESP8266 için kullanılan bazı AT komutları.....	10
--	----



1. GİRİŞ

Dünya genelinde artan tüketici talepleri ve hızlı teslimat beklentileri ile kargo ve lojistik sektörü önemli bir ölçüde büyüme eğilimi göstermektedir. İnternet ve e-ticaretin gelişmesi bu büyümeye katkı sağlamaktadır. UTİKAD lojistik sektörü raporuna göre 2020 yılında pandemiden kaynaklı olarak küresel lojistik sektörü pazar payı 5.72 trilyon Euro'ya gerilemiş fakat bu rakam 2021 yılında 599 trilyon Euro'ya yükselmiştir ve bu büyümenin 2024 yılında 6.88 trilyon Euro'yu geçmesi beklenmektedir (UTİKAD Sektörel İlişkiler Departmanı, 2021). Lojistik sektöründeki bu büyüme taşımacılıktaki kargo güvenliği ve takibi problemlerini de yanında getirmektedir.

Günümüzde artan kargo taşımacılığı esnasında bazı olumsuz durumlar oluşabilmekte ve kargolar farklı şekillerde hasar alabilmektedir. Bu hasarlar kargo sevkiyat süreçlerinin artmasına, müşteri veya lojistik firmalarının ekonomik olarak zarar görmesine neden olabilmektedir. Yapılan bir araştırmaya bakıldığında kargo hasarlarının dağılım oranları %23 oranında fiziksel hasarlar oluşturmaktadır (Lorenc & Kuźnar, 2016). Bunu %22 oranında ıslanma hasarları ve %16 oranında kirlenme takip etmektedir. Bir başka araştırmada da kargolarda tazminat talep edilen durumlar %31 oranında su hasarları, %28 oranında fiziksel hasarlardır (Recoupex, 2021). Diğer hasar türleri Şekil 1.1'de gösterildiği gibidir. Fiziksel hasarların çeşitli nedenleri olabilir. Bunlardan bazıları kargoların taşımacılık sırasında özensiz taşınmaları, düzgün istiflemenin yapılmaması, kargonun taşınma koşullarının uyulmaması ve taşımanın yapıldığı araçta oluşan olumsuz durumlar neden olabilmektedir.



Şekil 1.1. Kargo hasar türleri (Recoupex, 2021)

Kargonun yolculuk sırasında düşme, sarsılma gibi kargoyu kötü etkileyebilecek eylemler kargo sağlam gözükmese durumunda tespit edilememektedir. Eğer kargo hasarlı gelirse bunun ne zaman gerçekleştiği bilinmemektedir. Bu gibi durumlar için kargoların üzerine cihazlar yerleştirilerek veriler toplanabilir ve bunların analizleri yapılarak hasarın nedenleri, ne zaman olduğu, şiddeti gibi sorular cevaplandırılabilir. Bu analizler formüller çıkarılarak veya önceden eğitilmiş olan makine öğrenimi modelleri ile gerçekleştirilebilir. Bu hesaplamalar cihaz üzerinde veya bulut ortamında yapılarak kargo firmalarına veya kargo sahiplerine bilgi verebilir.

Makine öğrenimi, verilmiş olan veri girdilerinin analizini yaparak kazanılmış olan deneyim ile çıktılar üretebilen yapay zeka alanının bir alt dalıdır. Makine öğrenimi sayesinde sistemleri istenilen birçok alanda eğitim gerçekleştirilebilir ve eğitilmiş modellerle karar mekanizmaları oluşturulabilir. Karmaşık formüllerden kurtararak deneyimler ile daha doğru sonuçlar vermesi sağlanabilir.

TinyML, makine öğreniminin küçük cihazlarda çalışmasına olanak sağlayan bir makine öğrenimi modelidir. IoT cihazlarında, mobil cihazlarda, giyilebilir teknolojiler gibi cihazlarda bu modeller ile makine öğrenimin kullanılmasına olanak sağlamaktadır. Bu sayede daha düşük enerjili cihazlar akıllı hale getirilebilmektedir.

Uç bilgi işleme (edge computing), veri işleme ve depolamasının bulut dışında yakın cihazlarda ve uç noktalarda bulut kullanılmadan yapılmasına olanak sağlayan bir bilgi işleme modelidir. Uç bilgi işleme ile işlem maliyetleri ve bulut maliyetleri azaltılmış olur. Ağdan bağımsız daha hızlı cihazlar üretilmektedir.

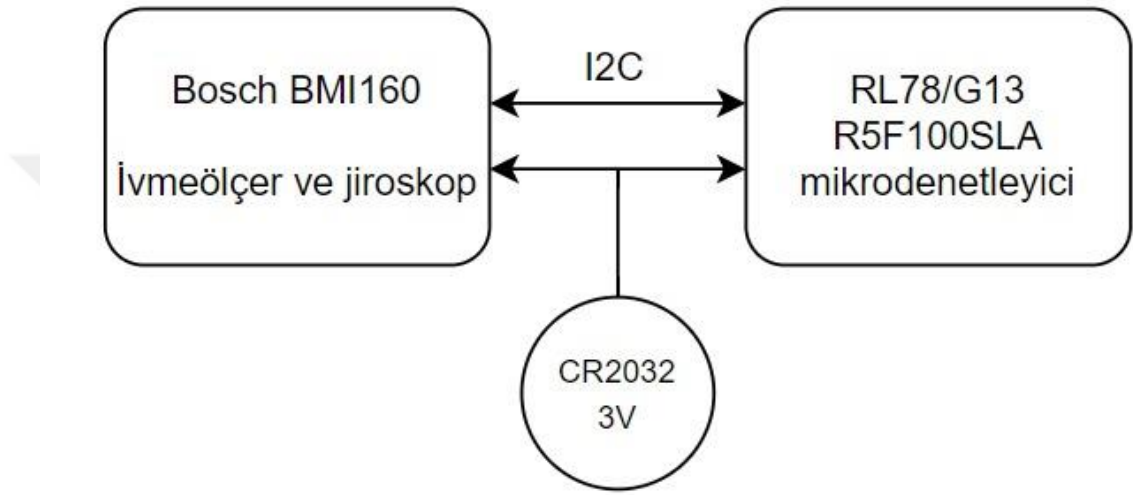
TinyML ve uç bilgi işleme teknolojileri bir arada kullanılarak uç noktalarda ağdan bağımsız akıllı cihazlar oluşturulabilmektedir. Bu cihazlar bulut üzerinde işlem yapmadan uç noktalarda eğitilmiş olan makine öğrenimi modellerinin çalıştırılarak enerji verimliliği, veri güvenliği ve gizliliği, karmaşık hesaplamaların yapılmadan istenen çıktılarının alınması gibi olanak sağlayabilmektedir.

Yapılacak olan çalışmada kargo ve lojistik sektöründeki fiziksel hasarların oluşumunun TinyML ve uç bilgi işleme teknolojilerini kullanarak tespit edilmesi ele alınacaktır. Bu teknolojilerle geliştirilen cihaz kargo üzerinde verileri toplayarak bir ağ kullanmadan kargo düşme durumunu makine öğrenimi ile tespit edip zamanı kaydedecek, kargo firmaları ve kargo sahipleri ne zaman gerçekleştiğini öğrenebilecektir. Bu sayede ilerleyen sevkiyatlarda neden olan durumlar tespit edilerek daha güvenli bir sevkiyat süreci sağlanabilecek ve kayıplar engellenmiş olacaktır.

Çalışma kapsamında öncelikle hazırlanmış olan cihaz bir kargo üzerine yerleştirilecektir. Cihaz üzerinde bulunan ivme ölçer ile sensör verileri toplanacaktır. Toplanan verilerin analizleri yapılacak ve filtreden geçirilerek eğitime hazır hale getirilecektir. Hazırlanmış olan veri setleri ile makine öğrenimi modeli oluşturulacaktır. Oluşturulan bu model gömülü yazılım içerisine aktarılarak kargo düşüşünün tespitini sağlayacaktır. Düşüşün tespit edildiği durumlarda üzerinde bulunan hafıza kartına zamanı ve düşüşün olduğunu kaydedecektir. Kaydedilen bu veriler hafıza kartı üzerinden veya web sitesi üzerinden kullanıcılara gösterilecektir.

2. KAYNAK TARAMASI

Erdoğan (2022), insanların alışveriş alışkanlıklarının değişmekte olduğunu ve internetten alışveriş yapmaya yönlendiğini belirtmektedir. Bu artan kargo miktarının bazı sorunları da beraberinde getirdiğinin ve sadece kırılabilen ürünlerin değil elektronik cihazlar gibi düşme sonrasında dış kısımda hasar gözükmese de düşüşün zararlara neden olabileceğini söylemektedir. Yapılmış olan çalışmada bu düşüşlerin tespit edilebilmesi için düşük güç ile çalışan bir cihaz geliştirilmiştir. Pille çalışan bu cihaz kargo üzerine sabitlenerek düşüş tespiti için kullanılmıştır.



Şekil 2.1. Düşme algılama sisteminin şeması (Erdoğan, 2022)

Şekil2.1’de düşüş tespiti için kullanılan cihazın şeması gösterilmiştir. Cihaz üzerinde ivmeölçer ve jiroskop sensörü kullanılmıştır. Öncelikle verileri toplamış ve Matlab üzerinde lojistik regresyon tekniği kullanılmıştır. Bu yöntem ile ivmeölçer ve jiroskop değerleri için karar sınırları belirlenmiş ve bu değerler gömülü yazılımda karar aşaması için kullanılmıştır. Cihaz üzerinde düşüş durumunu öğrenmek için üzerinde bulunan butona basılır ve led kırmızı yanarsa düşüş olmuş, yeşil yanarsa bir düşüş eylemi olmamış anlamına gelmektedir.

Gupta, Young, Rao (2022), malın bir yerden başka bir yere aktarılması sırasında alınan hasar miktarı ile ilgili bilgi sahibi olunursa, yaşanabilecek hasarlar alınacak önlemler ile azaltılabileceği tahmin edilmektedir. Çoğu zaman paketlerin kötü bir şekilde taşındığını ve yurtiçi, yurtdışı sigorta taleplerinin %50’si reddedildiği belirtilmiştir. Bu çalışmada kargonun transitinin gerçekleşecek olduğu coğrafi konuma bağlı olarak kargo üzerinde oluşabilecek etkilerin ön tespitini yapılmaktadır. Oluşabilecek hasar bilgisini sensörler kullanarak kargo durumu aktif olarak takip edilebilmektedir. Darbe ve titreşimleri önlemek için kullanılabilecek birçok sensör bulunmaktadır. Fakat bu sensörlerin fiyatları oldukça yüksektir. Bu çalışmada amaçlanan kullanılacak araçların olabildiğince uygun maliyetlerde olacak şekilde bir çözüm bulmaktır. Çalışmada titreşim ve darbe

verilerini yakalamak için gelişmiş ve pahalı sensörler yerine uygun maliyetli bir ivme ölçer sensörü kullanılmıştır. Projede kullanılan cihaz ivme ölçeri içermesinin yanında sıcaklık sensörü, nem sensörü, GPS ,depolama hafızası ve iletişim olanağı da sağlamaktadır. Kargonun farklı ülkelerden nakliyatının sağlanabileceği için her yerde bağlantı özelliğine sahip olunması gerekmektedir. Bunun için bağlantı olarak 2G/3G destekleyen cihaz seçilmiştir. 2G/3G kuleleri fazla bulunduğu için tercih edilmiştir. Toplanan veriler bulut ortamına aktarılmaktadır. İvme ölçer verileri her saniye 1'den fazla ivme verisini toplamaktadır. Bahsi geçen sensör kullanılarak birçok deney gerçekleştirilmiştir. Sarsıntı ve çarpışma anındaki hızlı ivme değişimi ile bir eşik değeri belirlenmiştir. Bu eşik değeri kargo içerisinde bulunan malzemenin hassasiyetine göre değişmektedir. Müşteri için oluşturulan canlı kontrol paneli, çevrimiçi sunularak müşteriye tüm mevcut gönderilerini aynı yerden takip edebilmesi için kolay erişim sağlanmıştır(Gupta vd., 2022).

Pambudi, Handayani, Hidayah (2018), konteynerlerde hasar tespiti üzerine çalışmıştır. Limana gelen hasarlı konteynerlerden dolayı limana dava açmaktadırlar. Bu nedenle hasarlı konteynerler için otomatik bir tespit sistemi geliştirmişlerdir. Çalışmada hasarlı konteyner görüntüleri ve normal konteyner görüntüleri kullanılmıştır. Görüntü işleme ile giriş yapan konteynerlerin hasar durumları tespit edilebilmiştir (Setio Pambudi vd., 2018).

Chen, Gao, Liang (2023), düşük güç tüketimli kestirimci bakım sistemi üzerine bir çalışma yapmışlardır. Makinaların gerçek zamanlı durumunun izlenmesi ve anormalliklerin tespitinin sağlanması amaçlanmıştır. Çalışmada düşük güç tüketimli kestirimci bakım sistemlerin geliştirilmesi ve değerlendirilmesi detaylarıyla anlatılmıştır. Hafif bir piezoelektrik kantileverin sensör olarak kullanılması, bu sensörlerin titreşimlere yanıt olarak gerilim sinyalleri üretmesi ve bu sinyallerin farklı yapay zeka modelleri kullanılarak analizlerinin yapılması ele alınmıştır. Çalışmanın sonucu bu sistemin endüstriyel uygulamalarda düşük maliyetli ve düşük güç tüketimli bir çözüm olarak potansiyelini göstermektedir (Chen vd., 2023).

Kim, Kyung, Kim, Ko (2022), yaptıkları çalışmada mevcut EKG izleme sistemlerinin gerçek zamanlı sınıflandırma ihtiyacını ele alarak gömülü sistemlerinin geliştirilmesini ele almışlardır. Zamanında tedavi için anormal EKG sinyallerinin sınıflandırılmasının önemi ve bilgilerin işlenmesi gereksinimleri nedeniyle sunucu tabanlı sınıflandırmanın sınırlamalarına vurgu yapılmaktadır. Geleneksel EKG izleme sistemlerinin sadece toplama rolünün olduğu ve gerçek zamanlı sınıflandırma sağlamak için gömülü sistem üzerinde çalışmalarını gerçekleştirmişlerdir. Yapılan çalışmada model ile yüksek doğruluklu sınıflandırma sağlanmıştır. Sonuç olarak mevcut sistemlerdeki sınıflandırma ihtiyacını ele alarak, gömülü sistemlerde TinyML'in kullanımını önermişlerdir. Bu yaklaşımın belirli noktalara olan ihtiyacı ortadan kaldırarak, ağ bant genişliği tüketimini azaltmayı ve gerçek zamanlı sınıflamayı mümkün kılmışlardır (Kim vd., 2022).

Zhou, Du, Chen, Li, Luan (2022), çalışmada demir yolu araçlarında çalışma durumunun gerçek zamanlı tespitinin akıllı demiryolu araçları için önemli bir teknoloji olduğu belirtilmektedir. Geleneksel sistemlerde tespit için araçtan buluta veriler gönderilmektedir. Bunun zahmetli ve verimsiz bir yöntem olduğuna değinmişlerdir. Çalışmada TinyML teknolojisi ile raylı araç çalışma durumu için akıllı bir durum tespit yöntemi önerilmekte ve küçük boyutlu ve düşük enerji tüketimli bir IoT sistemi

geliştirmişlerdir. Sistem ile algılanan sonuçlar görselleştirme için IoT sunucusuna yüklemişlerdir. Yapılan deneylerde altı karmaşık çalışma durumunu gerçek zamanlı olarak %99'un üzerinde doğrulukla tespit edilebildiğini göstermişlerdir (Zhou vd., 2022).

Durgun (2023), bu çalışmada yaşlı bireyler için düşme durumunda TinyML ve uç bilgi işleme teknolojileri kullanılarak düşme algılama sistemi üzerine çalışılmıştır. Yaşlı bireylerin evde yalnız yaşamalarının sağlık ve güvenlik açısından riskli olduğu vurgulanmaktadır. Düşmelerin fiziksel, psikolojik ve ekonomik yan etkilerinin olduğu belirtilmiştir. Çalışmada Arduino Nano 33 BLE ve yapay sinir ağı modeli kullanılmış ve başarılı sonuçlar elde edilmiştir. Yapılan testlerde, geliştirilen modelin %98,5 oranında başarı gösterdiği belirtilmiştir (Durgun, 2023).

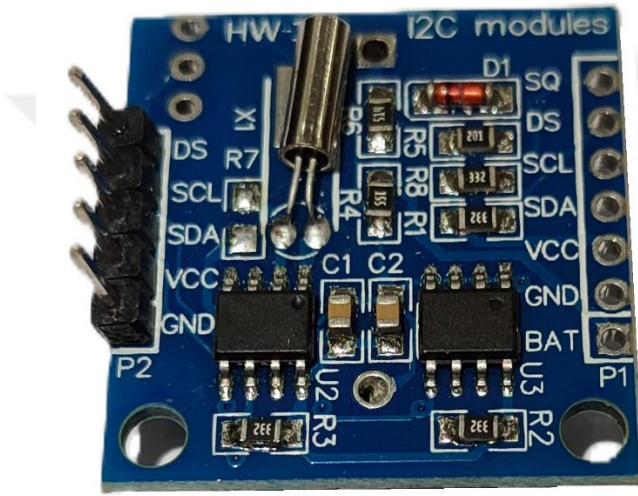
Yapılan çalışmalar incelendiğinde kargo sektöründeki çoğu çalışma yukarıdaki örneklerde olduğu gibi verilerin bir eşik değerinin bulunması ile gerçekleşmektedir. Bu eşik değeri ile düşme durum tespiti yapılmıştır. Yapay zekanın kullanıldığı örneklerde de eşik değeri tespiti için yapay zeka kullanılmıştır. Yapay zeka ile tahmin uygulamalarında da ham veriler bulut sisteme gönderilerek düşüş durumları tespit edilmiştir. Bu çalışmalar gerçek zamanlı olarak internet bağlantısı gerekmektedir ve cihaz sayısı arttıkça bulut üzerinde bir yük oluşturmaktadır. Eşik değerinin bulunması ve bunun karar mekanizması olarak kullanıldığı durumlarda da bazı karışık durumlarda doğru sonuçların çıkarılmasında yanıltıcı olabilmektedir. Görüntü ile yapılan hasar tespitlerinde de kargo düşme gibi durumlarında dış kısımda bir bozukluk meydana gelmezse tespit edilememektedir. Bu durum kargo durumu hakkında yanıltıcı sonuçlar çıkarmamıza neden olabilmektedir. Kargo sektörü dışında TinyML'in ve uç bilgi işleme teknolojilerinin kullanıldığı farklı sektörler bulunmaktadır. Yapılan çalışmalardaki cihazlar ve eğitim yöntemleri farklılık gösterebilmektedir.

Tüm bu çalışmalar göz önüne alındığında kargo sektöründeki bulut yoğunluğunun azaltılması ve karmaşık durumların tespitinde daha doğru sonuçların elde edilmesi için bu tezde farklı bir yaklaşımda gerçekleştirilmiştir. Cihaz içerisinde yerel olarak çalıştırılan yapay zeka modeli ile bulut yükü azaltmaktadır. Tasarlanan cihaz ve yaklaşımla gerçek zamanlı durum tespitinin doğruluk oranının artırılması sağlanmıştır.

3. MATERYAL VE METOT

3.1 DS1307 RTC Modülü

Ds1307 modülü bir RTC (gerçek zamanlı saat) modülüdür. Bazen projelerde gerçek zamanın kullanılması gerekmektedir. Bu gibi durumlarda bu modül kullanılabilir. I2C üzerinden mikrokontrolörler ile haberleştirilebilir. Enerji gitse dahi üzerinde bulunan pil sayesinde saat bilgileri güncel kalır. İlk başta zaman ve tarihin girilmesi yeterli olacaktır. Şekil3.1’de kullanılan modülün görseli bulunmaktadır.

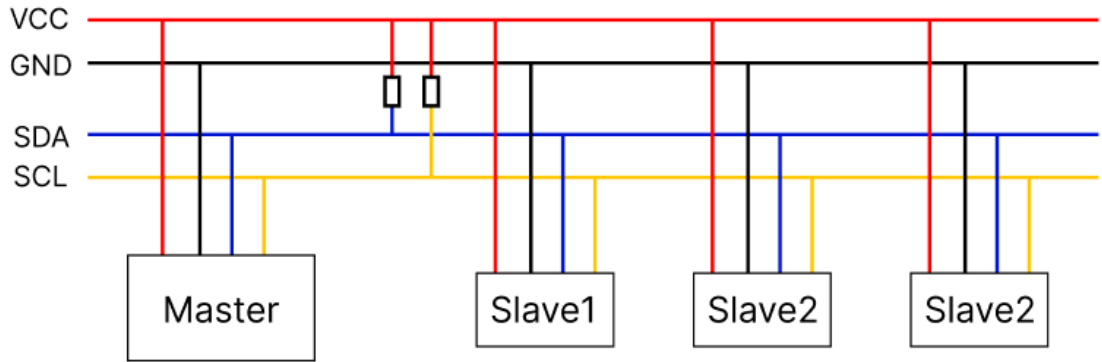


Şekil 3.1. Gerçek zamanlı saat modülü

5V dc besleme ile çalışmaktadır. I2C üzerinden haberleşme yapılabilmesi için VCC ve GND pinleri üzerinden 5V besleme yapıp SDA ve SCL pinleri üzerinden haberleşme sağlanabilir.

3.2 I2C Haberleşme Protokolü

I2C bir seri haberleştirme protokolüdür. Düşük veri hızının kullanılabileceği ve birden fazla cihazın bağlanacağı durumlarda kullanılabilir. Haberleşmenin sağlanabilmesi için hatta en az 1 adet master ve 1 adet slave cihaz bulunması gerekmektedir(Hemmanur, 2009).

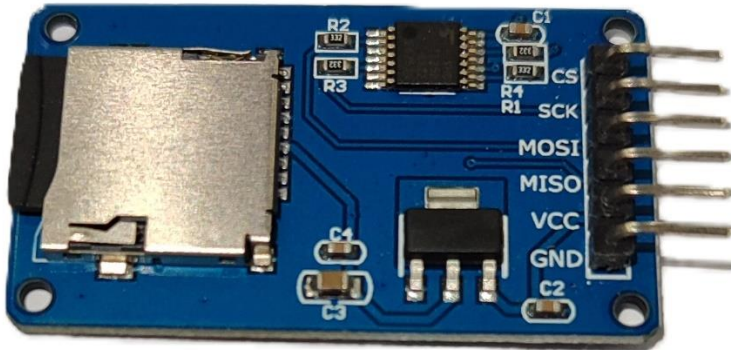


Şekil 3.2. I2C kullanan cihazların bağlantı şeması

Seri haberleşme 2 hat üzerinden gerçekleşmektedir. Bunlar çift taraflı SDA ve SCL hatlarıdır. Master cihaz SCL hattı üzerinden saat sinyali gönderir ve bu sinyal ile SDA hattı üzerindeki veri akışını kontrol etmektedir. İletişimin sağlanabilmesi için cihazların ortak GND hattına bağlı olması gerekmektedir. Hattın daha sağlıklı haberleşme sağlaması için de SDA ve SCL hatları pull-up dirençleriyle VCC hattına bağlanmalıdır. Şekil3.2’de cihazların birbirlerine bağlanma şeması gösterilmiştir.

3.3 Mikro SD Kart Modülü

Bazı projelerde okunan sensör verilerinin kaydının alınması, bazı bilgilerin saklanması veya bunlara benzer hafıza gerektiren durumlar olabilmektedir. Bu gibi durumlarda harici bir depolama kullanılabilir. Mikro SD kartlar bunlardan bir tanesidir. Mikro SD kart modülü sayesinde veriler okunabilir veya yazılabilir. Şekil 3.3’te kullanılan SD kart modülü gösterilmiştir.

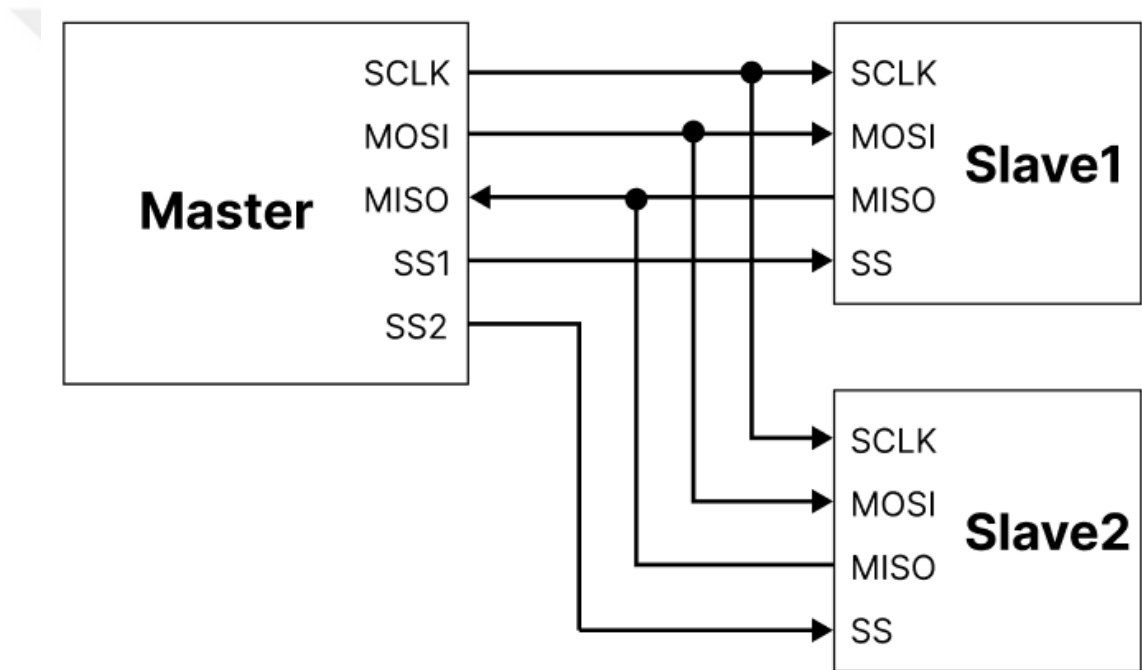


Şekil 3.3. SD kart modülü

Mikro SD kart modülü SPI protokolü ile haberleşmektedir. Üzerinde 6 adet pin bulunmaktadır. Vcc ve GND pinleri besleme için kullanılmaktadır. MISO, MOSI ve SCK pinler SPI ile haberleşme için ve CS pini de seçim pinidir.

3.4 SPI Haberleşme Protokolü

SPI bir seri haberleşme protokolüdür. Elektronik devrelerde cihazlar ile kısa mesafeli haberleşme için kullanılmaktadır. Haberleşme hattında bulunan master cihaz, slave cihazlar ile haberleşmeyi yönetmektedir. Haberleşme için 4 ayrı hat bulunmaktadır. Bunlar SCLK, MOSI, MISO ve SS/CS hatlarıdır.



Şekil 3.4. SPI kullanan cihazların haberleşme protokolü

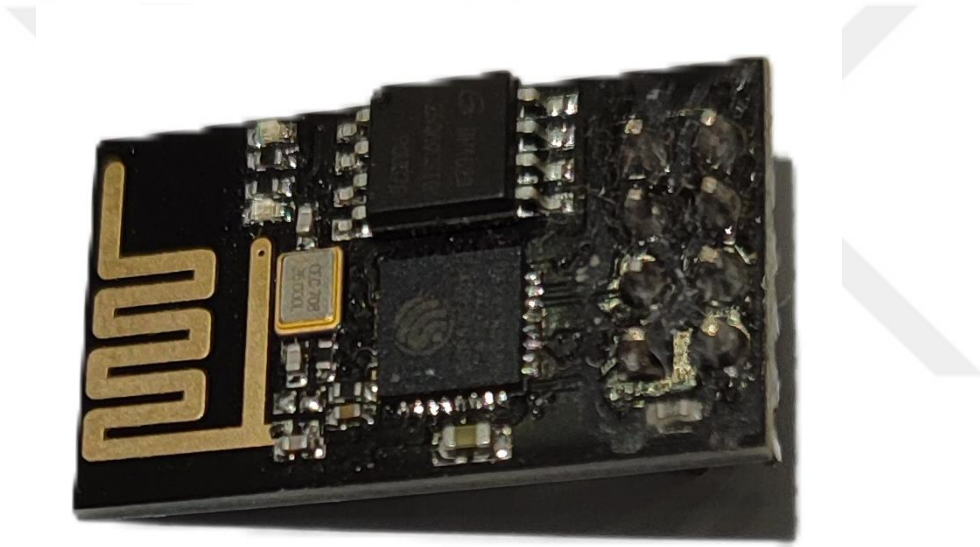
Master cihaz üzerinde bulunan SS pini üzerinden haberleşmek istenen slave cihazı seçilir. SS pini normal koşullarda yüksek voltaj seviyesindedir. Haberleşmesi istenen slave cihazın bulunduğu SS pini düşük voltaj konumuna getirilir. Böylece Master cihaz üzerinde MOSI ve MISO pinleri seri bağlanarak haberleşme sağlanabilir (Leens, 2009)

SCLK pini hat üzerindeki senkron haberleşmenin hızını belirleyen kısımdır. Cihazlara göre saat sinyalinin yükselen veya alçalan kenarında veriler iletilmektedir.

MOSI ve MISO hatları üzerinden cihazlar aralarında haberleşme sağlamaktadır. Master üzerinden slave cihaza veriler MOSI üzerinden gönderilir ve slave cihaz üzerinden alınır. Tam tersi durum için de MISO hattı kullanılmaktadır. Şekil3.4’ te örnek SPI haberleşme yapacak cihazların bağlantı şeması bulunmaktadır.

3.5 ESP8266 Wifi Modülü

ESP8266 modülü kablosuz ağ uygulamalarında, internete bağlanmanın gerektiği projelerde wifi noktasına bağlanmak için kullanılmaktadır. Aynı zamanda wifi noktası olarak veya her ikisi için de kullanılabilir. Küçük boyutu ve uygun fiyatı ile birçok uygulamada kullanılan popüler bir modüldür. Üzerinde dahili anten bulunmaktadır. Şekil3.5’te modülün gösterilmiştir.



Şekil 3.5. ESP8266 wifi modülü

Modül üzerinde SPI ve UART desteği bulunmaktadır. Genellikle UART üzerinden AT komutları ile kontrol edilmektedir. Üzerinde 8 adet pin bulunmaktadır. Cihazın UART ile kullanılabilmesi için VCC ve GND pinleri besleme hattına, TX ve RX pinleri UART haberleşmesi hattına bağlanmalı ve CH_EN pini HIGH yapılmalıdır. (Kocer vd., 2021)

3.6 AT Komutları

AT komutları, GSM, bluetooth, modem gibi haberleşme cihazlarında kullanılan bir seri haberleşme protokolüdür. “AT” kelimesi, “Attention” kelimesinin kısaltmasından gelmektedir.

Haberleşmenin sağlanacağı seri port üzerinden AT komutları gönderilerek cihaz kontrol edilir. Örneğin ESP8266 wifi modülünü kontrol etmek için RX ve TX bacaklarına kontrol edecek cihaz bağlanır. Seri port üzerinden AT komutlarını göndererek wifi modülünün durumu öğrenilebilir, çevredeki wifi cihazları sorgulanabilir, wifi modülüne reset atılabilir ve wifi modülünün desteklediği diğer tüm komutlar gönderilerek işlemler gerçekleştirilebilir(Espressif Systems IOT Team, 2016). Çizelge3.1’de bunların bir kısmı verilmiştir.

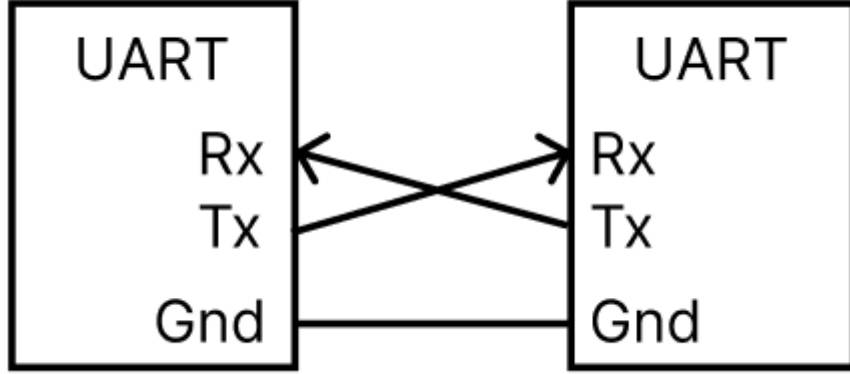
Çizelge3.1. ESP8266 için kullanılan bazı AT komutları

AT	ESP8266'nın çalışıp çalışmadığını kontrol etmek için kullanılır.
AT+CWJAP="SSID","password"	ESP8266'nın bir Wi-Fi ağına bağlanmasını sağlar.
AT+CIPSTART="TCP","ip_address",port_number	TCP protokolü kullanılarak belirtilen IP adresi ve port numarasına bir bağlantı başlatır.
AT+CIPSEND	Veri göndermek için bu komut kullanılır ve gönderilecek veri komutun hemen ardından gönderilir.
AT+RST	Bu komut, ESP8266'nın yeniden başlatılmasını sağlar.
AT+GMR	Bu komut, ESP8266'nın sürüm numarasını gösterir.

3.7 UART Haberleşme Protokolü

UART bir asenkron seri haberleşme protokolüdür. Haberleşme için 2 hat bulunmaktadır. Haberleşen cihazların vericisi diğer cihazın alıcısına bağlanması gerekmektedir. Ortak kullanılan bir saat sinyali yoktur onun yerine önceden belirlenmiş bir hız ile iki cihaz haberleşebilmektedir. 1200, 2400, 4800, 9600, 115200 yaygın olarak

kullanılan iletişim hızlarıdır. 9600 değeri saniyede gönderilen bit sayısının 9600 olduğu anlamına gelmektedir. Birimi bps'dir. (Peña & Legaspi, 2020)



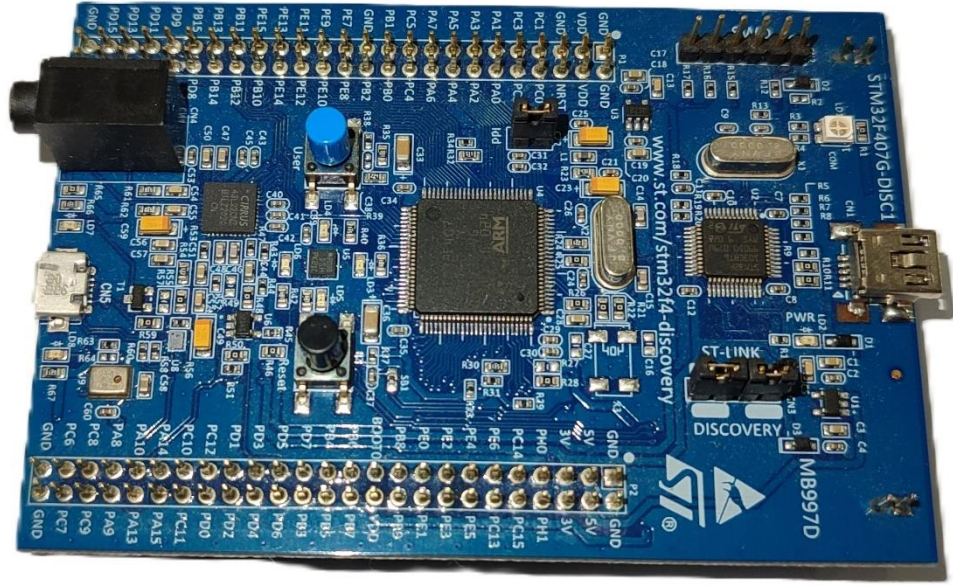
Şekil 3.6. İki UART cihazın bağlantı şeması

UART protokolü ile veriler iletilirken veri paketinin ilk biti başlangıç bitidir. Sonrasında veriler gelmektedir. Verilerin hatasız geldiğinin kontrolünü yapmak için veri bitlerinden sonra eşlik biti gönderilebilir. Son olarak da 1 veya 2 bitlik bitiş biti ile veri paketi sonlandırılır. Cihazlar bağlanırken gönderim hattı diğerinin alıcı hattına bağlanması gerekmektedir. Şekil3.6'da örnek olarak bu çapraz durum gösterilmiştir.

3.8 STM32F4 Discovery Geliştirme Kartı

STMicroelectronics tarafından üretilen bir geliştirme kartıdır. Bu kart STM32F4 mikrodeneleyicisi ile yapılacak uygulamaları geliştirmek ve test etmek için kullanılmaktadır. Her seviyeden geliştiriciye hitap etmektedir.

Üzerinde 32 bit ARM Cortex-M4 işlemci bulunmaktadır. Aynı zamanda dahili olarak bulunan st-link/v2 ile hata ayıklama yapılabilir ve kartın programlanması sağlanabilir. UART, SPI, I2C ve CAN gibi farklı haberleşme protokollerini desteklemektedir. Dahili olarak MP45DT02 mikrofon, LIS302DL veya LIS3DSH ivmeölçer ve CS43L22 ses çıkışı, buton ve led bulunmaktadır. USB üzerinden 5V ile veya harici uygulamalar için 3V ve 5V ile beslenebilir (STMicroelectronics, 2020). Şekil3.7'de cihazın görseli gösterilmiştir.



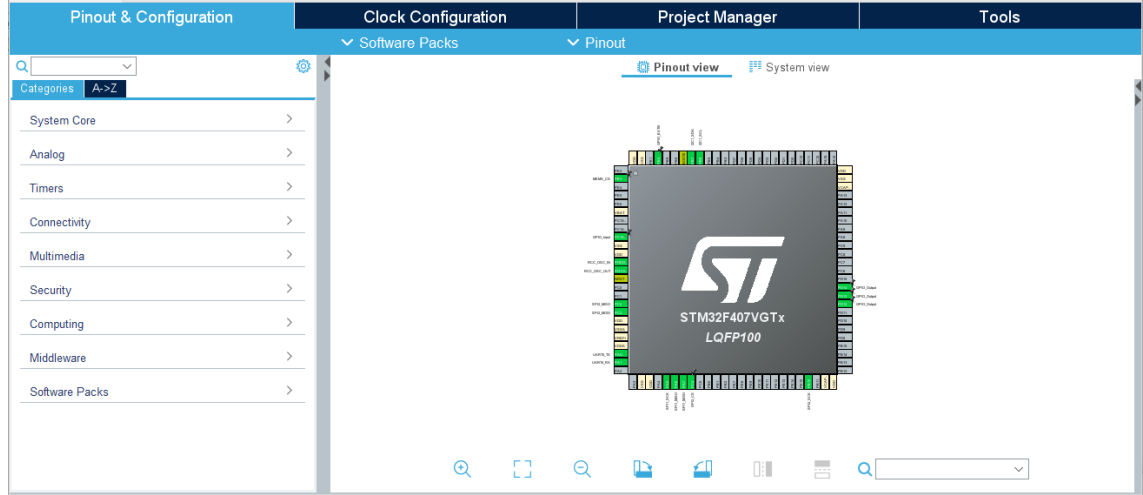
Şekil 3.7. STM32F4 Discovery geliştirme kartı

3.9 STM32CubeIDE ve STM32CubeMX

STM32CubeIDE ve STM32CubeMX programları STM32 mikrodeneleyicileri için kullanılabilen yazılım geliştirme araçlarıdır.

STM32CubeIDE , mikrodeneleyiciler için C/C++ ile uygulama yazma olanağı sağlamaktadır. Bu geliştirme aracı ile yazılan uygulamalar derlenerek mikrodeneleyicilerin içerisine atılabilir. Aynı zamanda gerçek zamanlı olarak hata ayıklama imkanı da sunmaktadır. Bu sayede program içerisindeki hatalar kolayca bulunabilmekte ve programın akışı takip edilebilmektedir. Geliştirme ortamı kullanmaya başlanırken kullanılacak olan mikrodeneleyicinin modeli seçilir ve geliştirmeye mikrodeneleyiciye uygun konfigürasyonlar ile başlanır.

STM32CubeMX, STM32 mikrodeneleyicilerinin yapılandırılmasını kolaylaştırabilen bir araçtır. Seçilmiş olan denetleyicinin bacaklarının özelliklerine rahatlıkla ekran üzerinden ulaşılabilir ve bu pinlerin yapılandırma işlemleri yapılabilir. Şekil3.8’te bu ekran verilmiştir. Görsel yapılandırma ile giriş çıkış pinleri isimlendirilebilir, haberleşme pinleri belirlenebilir ve bunlar gibi başka mikrodeneleyici yapılandırma işlemi kolaylıkla gerçekleştirilebilir.

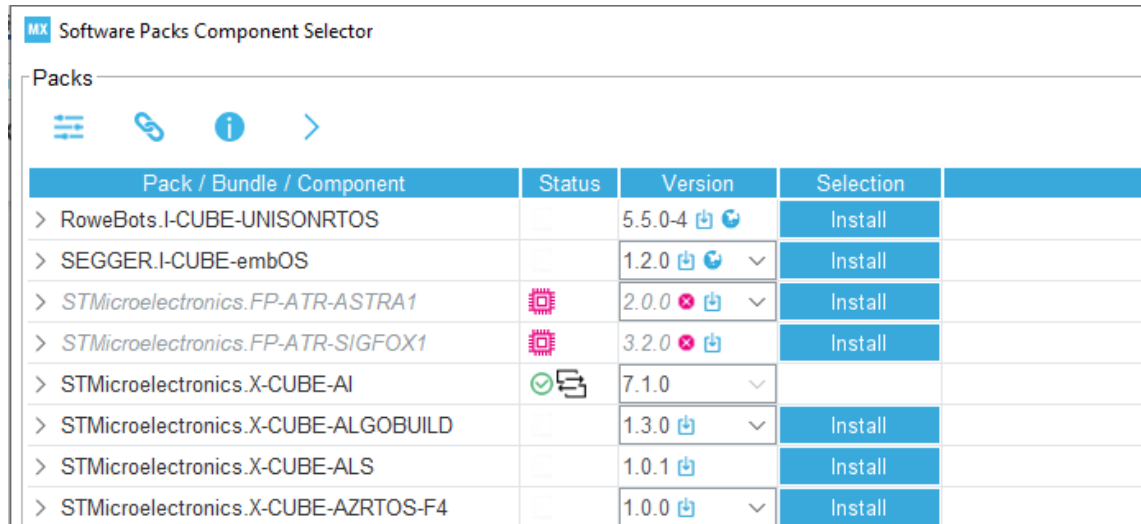


Şekil 3.8. Mikrodenetleyici pinleri yapılandırma ekranı

3.10 X-CUBE-AI Paketi

X-CUBE-AI paketi STMicroelectronics firmasının sunmuş olduğu bir genişletme paketidir. Yazılım paketi önceden eğitilmiş sinir ağları ve makine öğrenimlerinin dönüştürülerek STM32 mikrodenetleyicilerde kullanılabilir hale getirilmesini sağlar. Aynı zamanda STM32CubeIDE üzerinde geliştirilen projelerde önceden eğitilip dönüştürülmüş olan makine öğrenimi ve sinir ağları algoritmasını proje içerisine entegre etmektedir(STMicroelectronics, 2023).

STM32CubeIDE projesinin .ioc dosyasına girdikten sonra içerisinde bulunan “Software Packs” kısmından yüklenebilecek yazılım paketlerine ulaşılabilir. Açılan yazılım paketleri seçim ekranında yükleme butonuna basarak paket kolaylıkla projeye dahil edilebilmektedir. Şekil3.9’da paket yükleme kısmı gösterilmiştir.



Şekil 3.9. X-CUBE-AI paketi ekleme ekranı

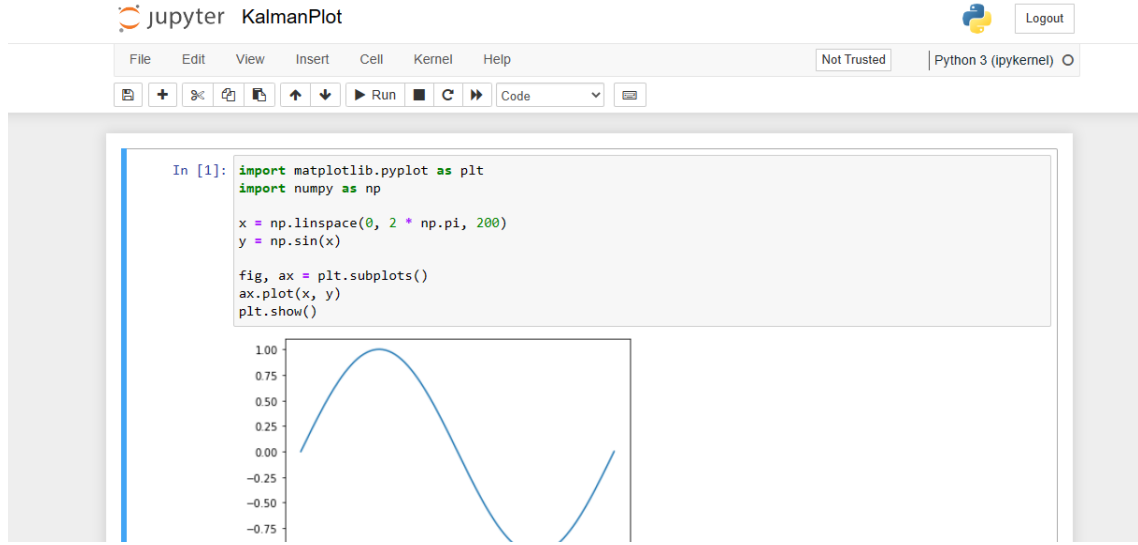
3.11 Python Programlama Dili

Python, yüksek seviyeli nesne tabanlı bir programlama dilidir. Veri analizi, yapay zeka, web uygulamaları ve bunun gibi birçok alanda kütüphane desteği sağlamaktadır. Linux, MacOS ve Windows işletim sistemleri üzerinde çalışabilmektedir. Nesneye yönelik programlama ve fonksiyonel programlama yapılabilir. Birçok alanda kullanılabilirliğinden ve kolay sözdizimi sayesinde hızla gelişmekte olan ve büyüyen bir programlama dilidir.

3.12 Jupyter Notebook

Jupyter Notebook, çeşitli programlama dilleri için kullanılan adımda da geçtiği gibi not defteri gibi yazıların yazılabildiği ve not defteri içerisinde kodların çalıştırılabildiği bir platformdur. İlk başta IPython Notebook adıyla bilinen ve sadece Python dilini destekleyen bir uygulamadır. Sonrasında farklı programlama dillerine de hizmet sunmaktadır (Pimentel vd., 2019).

Not defteri içerisinde belirli kısımları çalıştırabilme özelliği sayesinde kodlar blok halinde çalıştırılabilir. Kısmi yürütme özelliği sayesinde derin öğrenme, veri bilimi ve makine öğrenimi gibi projelerde sıklıkla kullanılmaktadır. Şekil3.10'da bir kod bloğu örneği verilmiştir.



Şekil 3.10. Jupyter Notebook örnek bir kod bloğu

3.13 TensorFlow Kütüphanesi

TensorFlow, açık kaynak kodlu ve ücretsiz bir makine öğrenimi kütüphanesidir. Python programlama dili ile uygulamalar gerçekleştirilir. Ses işleme, doğal dil işleme, finansal tahminler gibi çeşitli alanlarda kullanılabilmektedir (Abadi vd., 2016).

TensorFlow kütüphanesinin sağlamış olduğu yapay sinir ağı algoritmaları, sınıflandırma algoritmaları gibi yöntemlerle veriler eğitime sokularak eğitilmiş modeller oluşturulabilir. Eğitilmiş bu modelleri kaydedilip ilerleyen süreçlerde kullanılabilmektedir.

3.14 Pandas, Numpy, Matplotlib Kütüphaneleri

Veri analizleri yapılırken bazı araçlara ihtiyaç duyulmaktadır. Python dilinin sunmuş olduğu kullanışlı kütüphaneler bulunmaktadır. Bu araçlar ile veri yapıları kontrol edilebilir, düzenlenebilir ve görselleştirilebilirler. Düzenlenen ve görselleştirilen veriler sonuç elde etme noktasında geliştiriciyi hızlandırmakta ve kolay analiz yapmasını sağlamaktadır.

Pandas, verileri analiz için uygun hale getirmek için kullanılan bir kütüphanedir. Verileri filtreleme, birleştirme, temizleme gibi imkanlar sunmaktadır. Verilerde bu işlemlerin yapılabilmesi için verileri yükleyebilmekte ve eğitime hazır hale getirilmiş veriyi farklı formatlarda dışarıya aktarabilmektedir.

Numpy, çeşitli matematiksel işlemler yapılabilmesini sağlayan bir kütüphanedir. Çok boyutlu dizeler ve matrislerle çalışabilmekte, büyük verilerde de hızlı işlem yapılabilir (Sapre & Vartak, 2020).

Matplotlib kütüphanesi veri görselleştirmede kullanılmaktadır. Sütun grafiği, alan grafiği, pasta ve nokta grafiği gibi birçok grafik türünü oluşturabilmektedir. Oluşturulan grafikleri belli bir düzende ekranda gösterebilme ve grafikleri özelleştirme gibi imkanlar sunmaktadır.

3.15 Java Programlama Dili

Java, yüksek seviyeli ve nesne yönelimli bir programlama dilidir. Mobil, web, masaüstü uygulamaları ve oyun gibi farklı alanlarda kullanımları vardır. Beraberinde gelen kütüphaneler, hızlı ve kolay bir şekilde uygulama geliştirmeye olanak sağlamaktadır.

JVM üzerinde çalışmaktadır. JVM sayesinde Java ile platformdan bağımsız uygulamalar geliştirilebilmektedir. İşletim sisteminden ayrı olarak sanal makine üzerinde çalıştırılması doğrudan erişimi engelleyerek güvenlik açısından avantaj sağlamaktadır.

3.16 Java Spring Boot ve PostgreSQL

Spring Boot, bir web uygulaması geliştirme çerçevesidir. Spring tabanlıdır ve Java diliyle uygulamalar yazılmaktadır. Web uygulamalarının geliştirilirken her şeyi en baştan yapmak yerine çerçevenin sunmuş olduğu imkanlar kullanarak hızlı ve kolay yazılmasına olanak sağlamaktadır.

PostgreSQL, açık kaynak kodludur ve ilişkisel bir veri tabanı yönetim sistemidir. Veri tabanının oluşturulması, güncelleme yapılması ve değişiklik yapılabilmesi gibi işlemlerin yönetilmesini sağlamaktadır.

3.17 Java Script Programlama Dili

Javascript, günümüzde birçok kişi tarafında web geliştirme için kullanılmaktadır. Sadece web geliştirme değil aynı zamanda mobil uygulama, masaüstü uygulaması geliştirme gibi farklı alanlarda da kullanılmaktadır. Tarayıcılarda direk çalışabilmektedir. Platform bağımsız bir dildir. Birçok kütüphanesi bulunmakta ve her geçen gün kullanıcı sayısı artmaktadır.

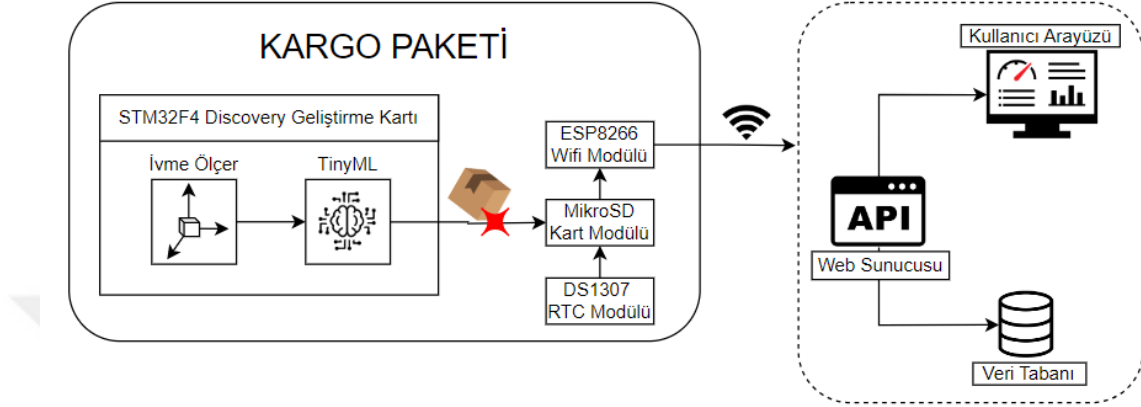
3.18 Kalman Filtresi

Kalman filtresi, bir sistemdeki eski verilere dayanarak sistemin güncel durumunun tespit edilmesinde kullanılır. Yeni gelen veriler eski veriler ile birleştirilmeye devam ederek filtrenin güncelliği sağlanmaktadır. Filtre olarak isimlendirilmekte fakat durum tespiti yaptığı verideki gürültüyü engellediği için o şekilde isimlendirilmektedir (Küçükateş, 2020). Otomasyon sistemlerinin kontrolü, sensör verilerinin okunması, hava tahminleri, robotik hareketler gibi birçok farklı alanda kullanımları bulunmaktadır.



4. BULGULAR VE TARTIŞMA

Kargo üzerindeki cihaz ile ürünlerin sevkiyat sürecinde yaşanan olumsuz durumların tespiti ve izlenmesi için kurulan sistem sıralaması Şekil4.1’de gösterilmiştir.



Şekil 4.1. Çalışmanın genel sistem mimarisi

Öncelikle kargo üzerindeki taşıma verilerinin analizi için bir cihaz üretilmesi gerekmektedir. Bu projede verilerin toplanabilmesi için geliştirme kartı üzerindeki ivme ölçer kullanılmıştır. Verilerin kayıt zamanları için RTC modül de eklenmiştir. Bu sayede kargonun düşme saati tam olarak bilinebilmektedir. Kaydedilen veriler gönderilirken ESP8266 modülü kullanılarak internet bağlantısı sağlanmıştır.

Kullanılan ivme ölçer verileri ilk ölçümlerde sabit durumda belli bir bantta sürekli değişkenlik göstermektedir. Sensör verilerindeki bu yanıltıcı oynamaları filtrelenmesi gerekmektedir. Filtreleme için sensör verileri kalman filtresinden geçirilmiştir. Geçirilen bu veriler sonrasında eğitim verileri olarak kullanılması için kaydedilmiştir. Kayıt esnasında kargo düzgün bir şekilde taşınarak, olması gereken durum ve kargo yere atılarak düşme durumları kaydedilmiştir. Kaydedilen veriler eğitim için temizlenerek veri setleri haline getirilmiştir.

Veri setinin oluşturulması adımı tamamlandıktan sonra, oluşturulan veri seti kullanılarak kargonun düzgün taşıma durumu ve düşme durumunun doğru bir şekilde sınıflandırılması hedeflenmiştir. Bu kısımda yapay sinir ağları kullanılmıştır. İvme verileri belli adette toplanıp sıralı olarak model girdisi için kullanılmıştır. Çıktı değerleri de sonucu göstermektedir. Girdide kullanılacak ivme miktarının doğru tespit edilmesi önemlidir. Eksik veya fazla olma durumunda davranış tipi tespit edilemeyebilir. Bunun için veri setleri incelenerek optimum sayı seçilmiştir. Diğer düzeltme işlemleri de gerçekleştirildikten sonra model eğitimi tamamlanmıştır. Fakat bu eğitim modeli Python

üzerinde kodlanmıştır. Bu kodun gömülü sistemde çalışması için eğitilmiş model kaydedilip dönüştürülerek gömülü sistemlerde kullanıma hazır hale getirilmiştir.

Buraya kadar olan kısımda gerçek zamanlı olarak tespitin yapılabilmesi için çalışmalar yapılmıştır. Cihazın gerçek zamanlı çalışması sırasında ivme verileri kalman filtresinden geçirilerek dönüştürülmüş olan modelden geçirilerek sonucu eğer düşüş ise SD karta kaydedilmiştir. Sonrasında cihaz interneti bulduğu noktada verileri sunucuya göndermektedir. Kargo sahipleri kargo cihaz numaralarını girerek arayüzden durumlarını görebilmektedir. Sistem genel olarak bu şekilde çalışmaktadır. Anlatılan bu kısımlar daha ayrıntılı olarak devam eden kısımda anlatılacaktır.

4.1 Veri Seti Oluşturma

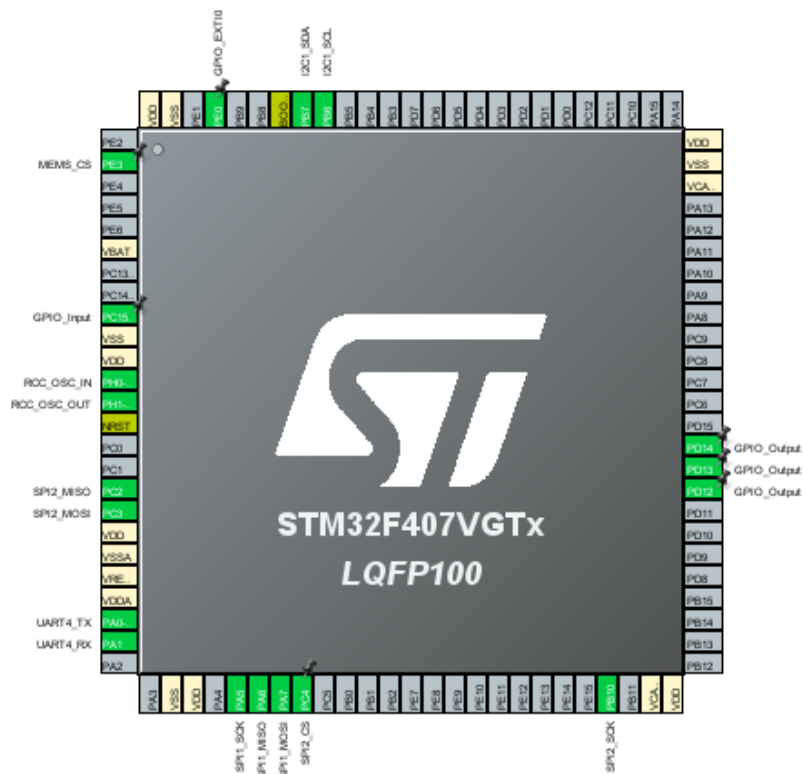
Makine öğreniminde veri setinin düzgün olması eğitilecek olan modelin başarısı için önemli bir rol oynamaktadır. Bu projede eğitim için kullanılacak veriler, veri toplaması için geliştirilen kart ile toplanmıştır. Şekil4.2’de veri seti elde etme aşamaları gösterilmektedir. İlk olarak cihazın tasarımı ve verilerin toplanması ile başlamaktadır.



Şekil 4.2. Veri setinin elde edilme aşamaları

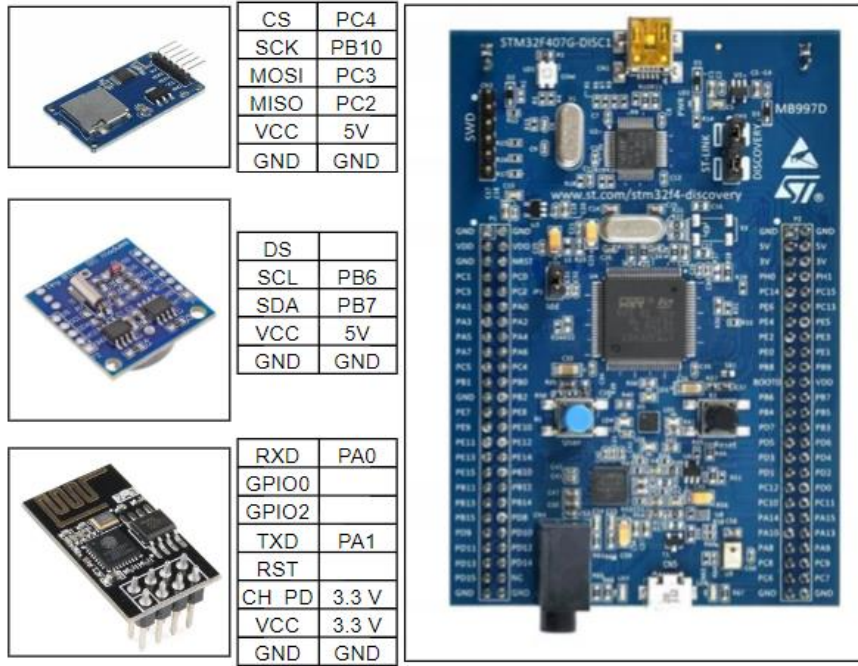
4.1.2 Cihazın tasarımı ve verilerin kayıt edilmesi

Bu projede gömülü sistem için STM32F4-Discovery geliştirme kartı kullanılmıştır. Üzerindeki ST-MEMS dijital ivme ölçer ile ivme verileri okunabilmektedir. Bu sensörün yanı sıra harici olarak RTC, ESP8266 ve SD kart okuyucu modülleri sisteme kullanılması gerekmektedir. Bağlantıları yapmadan önce kart üzerindeki modüller için gerekli olan pinlerin belirlenmesi gerekmektedir. Şekil4.3’te tanımlamalar görülmektedir. Yeşil ile işaretlenmiş kısımlar kullanılacak olan pinlerdir.



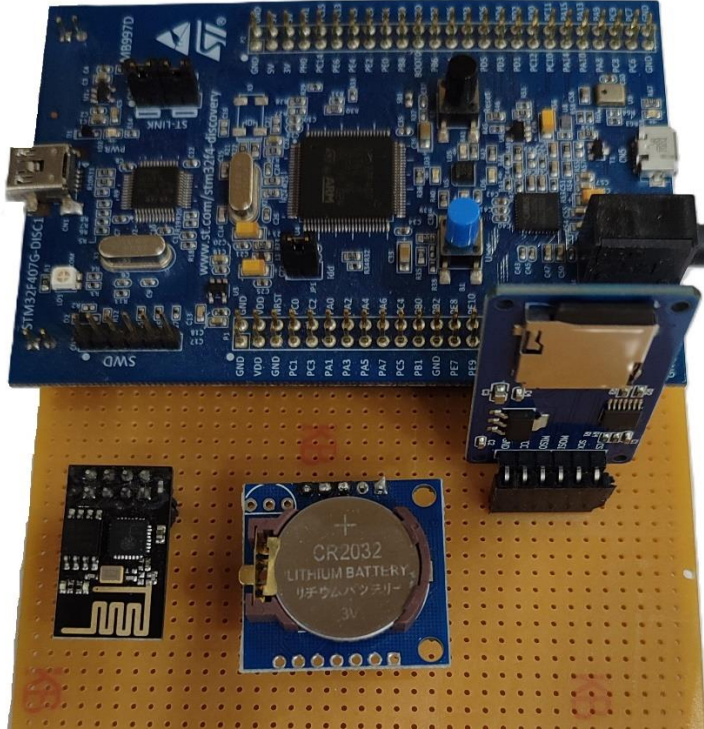
Şekil 4.3. Bağlantı için pinlerin tanımlanması

Geliştirme kartının üzerinde hazır bağlantılı bulunan ivme ölçer kullanılacağı için dokümandan bağlandığı pinlerin öğrenilmesi gerekmektedir ve bağlantının yapıldığı pinler ivme ölçer için tanımlanmalıdır. Diğer modüller için de kullanılacak haberleşme protokolüne göre pinlerin seçilmesi gerekmektedir. Pin yerleşimleri kesinleştirildikten sonra modüllerin ve kartın bağlantıları delikli levha üzerinde yapılmıştır. Şekil4.4'te bağlantı şeması gösterilmektedir.



Şekil 4.4. Tasarlanan cihazdaki pin bağlantı şeması

Delikli levha üzerine bağlantılar yapıldıktan sonra eklenen devre elemanlarının bağlantı kontrolleri yapılmıştır. Şekil4.5'te devrenin tamamlanmış hali gösterilmiştir.



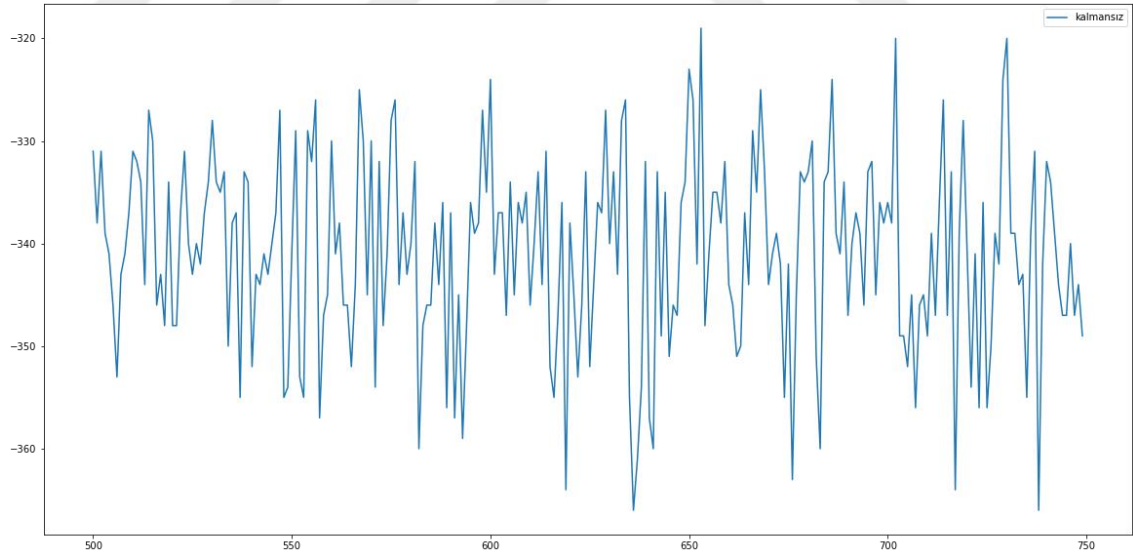
Şekil 4.5. Cihazın tamamlanmış halinin görseli

Öncelikle ivme üzerindeki verilerin zaman bilgileri ile kaydedilmesi için mikro SD kart ve RTC modülünün kodları yazılmıştır. İvme ölçerden alınan verilerin x, y, z verilerinin karışmaması ve kaydedildikten sonra analiz sırasında ayrıştırmayı kolaylaştırmak için belirli bir formatta kayıt işlemi yapılmıştır. RTC modülünden doğru verileri alabilmek için ilk kullanımda zaman bilgileri ile kurulum işlemleri gerçekleştirildi. Sırasıyla yıl, ay, gün, saat, dakika, saniye şeklinde kaydedilmiştir. Sonrasında 6'şar karakterlerle veriler başlarında x, y, z olacak şekilde kaydedilmiştir.

Kaydedilen veriler analizi yapılması için bilgisayar ortamına aktarılmıştır. Aktarılan veriler .txt formatında ve tek bir satır halinde olduğundan kullanıma hazır değildir. Verilerin görselleştirilmede ve eğitimde kullanılması için bir Python kod parçası yazılmış ve .csv formatına dönüştürülmüştür.

4.1.3 Verilerin filtreden geçirilmesi

Elde edilen .csv formatındaki veriler Python kütüphanesi yardımı ile görselleştirilmiştir. Şekil4.6'da durağan konumdaki ivme verisinin x değeri çizdirilmiş fakat durağan konumdaki veriler incelendiğinde çok fazla oynamanın olduğu gözlemlenmiştir. X değerindeki oynaklık eğitimde yanlış sonuçların doğmasına ve zorluklara yol açabilmektedir. Bu nedenle veri düzeltme yöntemi olarak kalman filtresi seçilmiştir.



Şekil 4.6. Durağan cismin filtresiz ivme verileri

Her bir değer ayrı kalman filtresinden geçirilmelidir. Yazılım içerisinde kalman için fonksiyon oluşturulmuştur. Fonksiyon oluşturulurken öncelikle kalman kazancı hesaplanmıştır. Kalman kazancının (K_k) hesabı sırasında önceki hata kovaryansı (P_k^-), hata miktarı (R) ve eski hata kovaryansının toplamına bölünerek bulunmaktadır (4.1). Bulunan kalman kazancı tahmin değerinin ($\hat{x}_k$) hesaplanmasında kullanılacaktır.

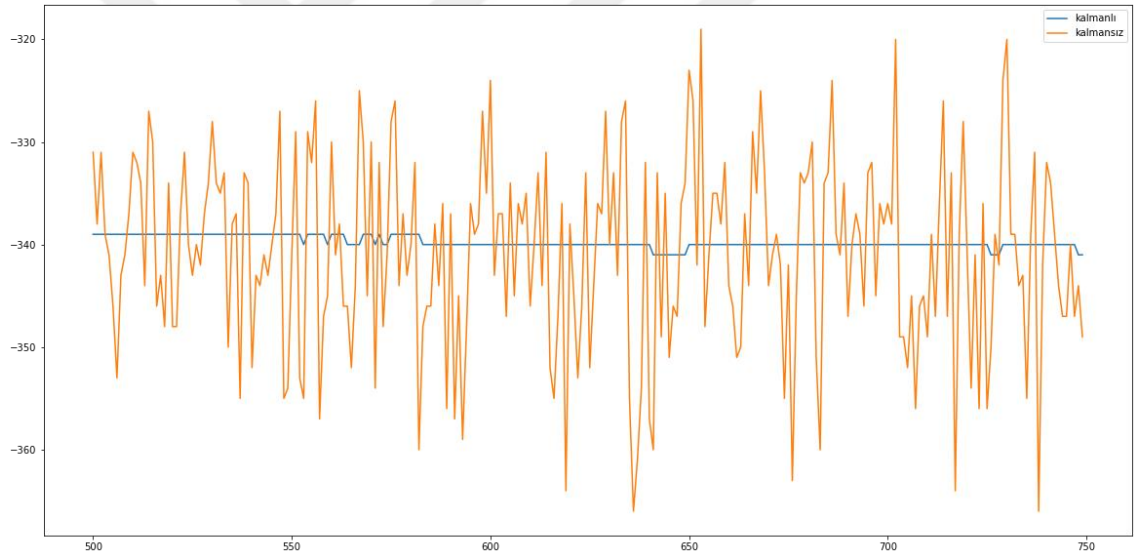
Hesaplama sırasında sensörlerden ölçülen değer (z_k) ile önceki tahmin değeri çıkarılıp kalman kazancı ile çarpılmaktadır. Çıkan sonuç eski tahmin değeriyle toplanarak yeni tahmin değeri bulunmaktadır (4.2). Bu sayede eski fiyatla ilişkisini korumuş olmaktadır. Bir sonraki aşama için yeni hata kovaryansı (P_k) hesaplanmaktadır (4.3).

$$K_k = \frac{P_k^-}{P_k^- + R} \quad (4.1)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-) \quad (4.2)$$

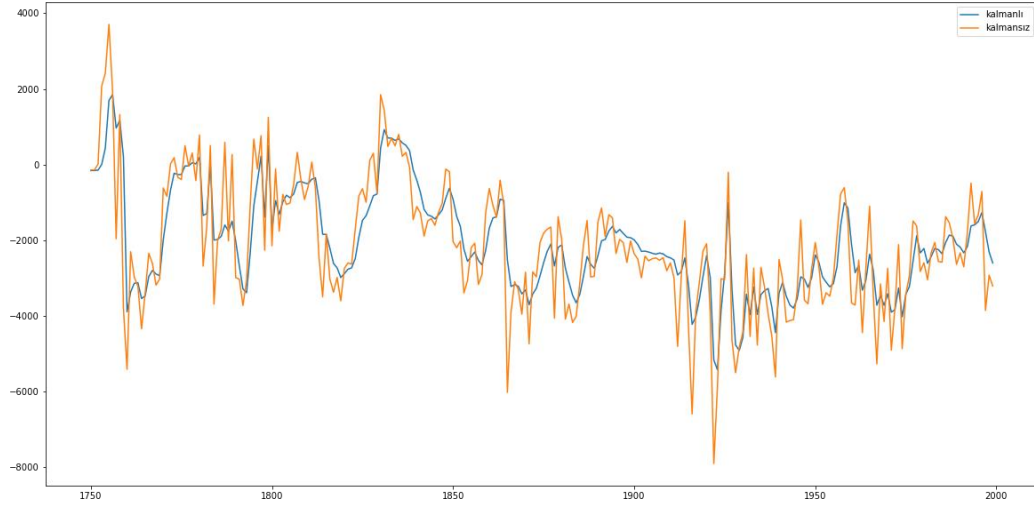
$$P_k = (1 - K_k)P_k^- \quad (4.3)$$

Bu işlem her ölçümden sonra yapılmaktadır. Ölçüm sonrası kalmanlı değerler ile kaydedilmiştir. Şekil4.6'nın kalman ile filtrelenmiş hali Şekil4.7'deki gibidir. Görüldüğü üzere ölçülen veriler kalman sonrası daha anlamlı bir hal almıştır ve verideki oynaklık temizlenmiştir.



Şekil 4.7. Durağan ivme verisinin kalman filtreli ve filtresiz hali

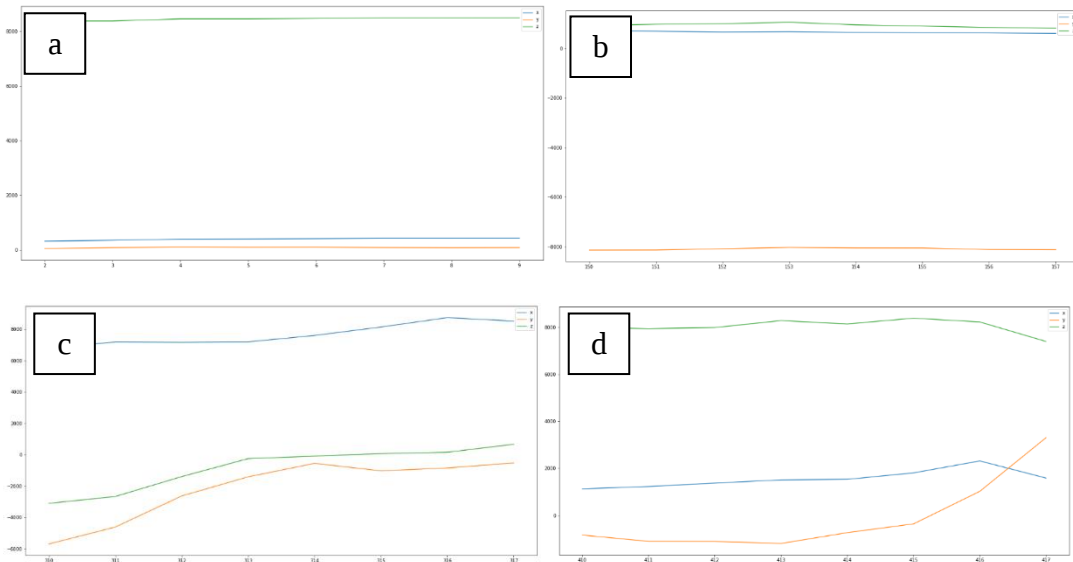
Veriler sabit dururken bu oynaklığı gösterdiği gibi hareket durumlarında da göstermektedir. Şekil4.8'de de görüldüğü gibi hareket değişim noktalarında ekstradan atımlar meydana gelmiştir. Bu da çarpma gibi davranış tespitlerinde yanıltıcı olabilmektedir. Kalmanlı veri daha akışkan bir şekilde ilerlemektedir.



Şekil 4.8. Hareketli ivme verisinin filtreli ve filtresiz hali

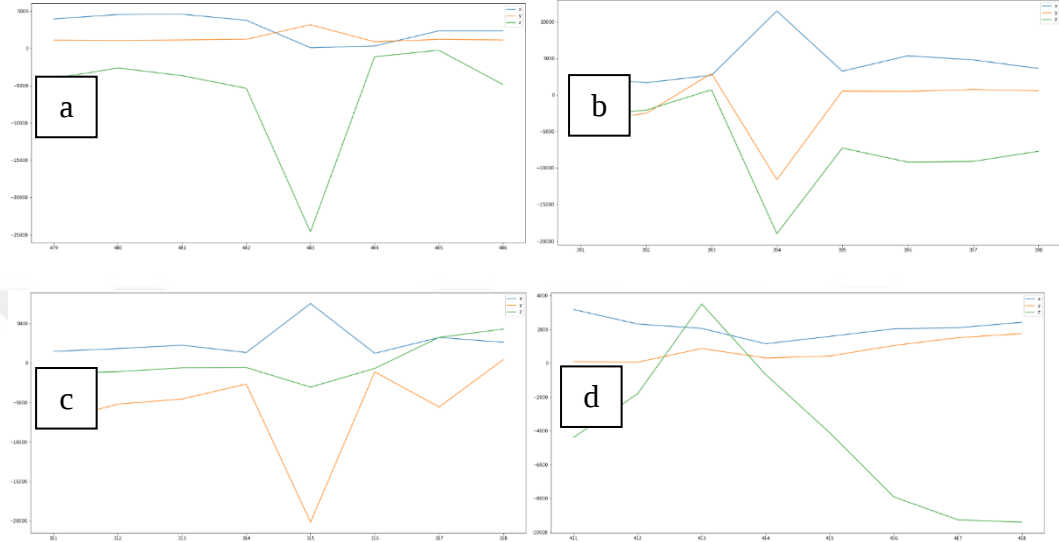
4.1.4 Veri setini oluşturulması

Veri toplama cihazı geliştirildikten sonra, ivme verilerinin kalman filtresinden geçirilmesi sonucunda cihaz eğitim verilerini toplaması için hazır hale gelmiştir. Veri toplamak için ürün bir kutu üzerine sabitlenmiştir. Kutu burada kargo olarak kullanılmıştır. Kargo ilk olarak farklı yönlerde normal taşıma olarak hareket ettirilmiştir. Hareket sırasında kalman filtresinden geçirilmiş ivme verileri saat bilgileri ile kaydedilmiştir. Örnek olarak Şekil 4.9’da normal taşıma sırasındaki x, y ve z değerleri gösterilmiştir.



Şekil 4.9. Normal taşımada x, y ve z eksenleri ivme ölçer verileri grafikleri; **a); b)** sabit durumdaki veri grafiği **c); d)** hareket halindeki veri grafiği

Normal taşıma verileri bu şekilde oluşturulduktan sonra düşüş verileri toplanmaya başlanmıştır. Düşüş verileri için kargo kutusu yere atılmış ve farklı düşüş durumları simüle edilmiştir. Düşüş verileri incelendiğinde 8 adet örnekleme miktarının düşüşün tespit edilmesinde yeterli olduğu tespit edilmiştir. Örnekleme miktarının doğru tespit edilmesi eğitim sırasında eğitimin istenilen şekilde çıkması için önemlidir. Tüm düşüş verileri bu şekilde tek tek tespit edilmiş Örnek düşüş verileri Şekil4.10'daki gibidir.

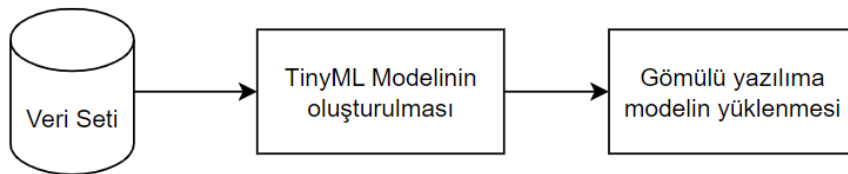


Şekil 4.10. Düşüş anında x, y ve z eksenleri ivme ölçer verileri; a); b); c); d) kargo kutusunun farklı yönlerdeki düşüş ivme verileri

Kargonun normal durumu ve düşüş durumları birlikte kaydedildikten sonra eğitim veri seti oluşturulmuştur. Bu veriler yapay sinir ağına sokularak eğitim gerçekleştirilecektir.

4.2 Model Eğitimi ve Kullanımı

Bu kısımda çıktı değerleri ile birlikte ayıklanmış veri setiyle model eğitimi gerçekleştirilmiştir. Eğitilen model gömülü yazılıma entegre edilecek hale getirilmiştir. Şekil4.11'de aşamalar gösterilmiştir.

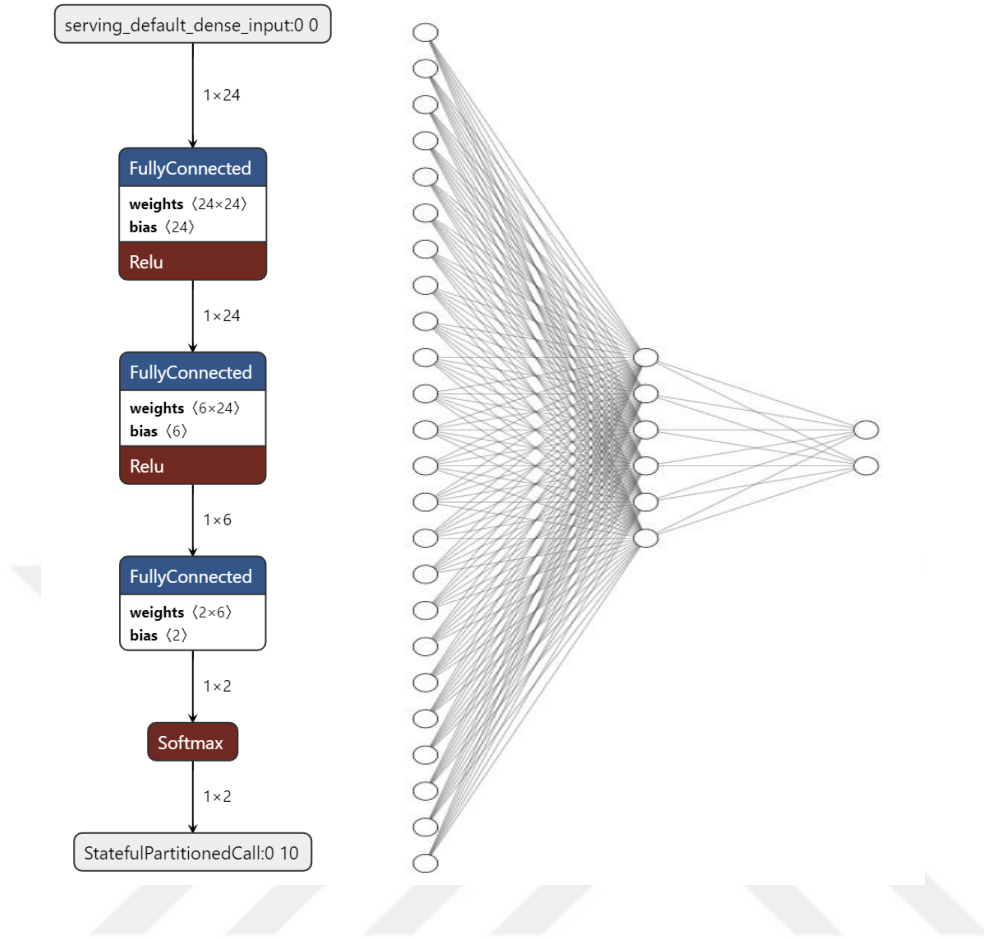


Şekil 4.11. Model eğitim ve kullanım aşamaları

4.2.2 Verilerin bölünmesi ve yapay sinir ağı modeli

Toplanan veriler sıralı bir şekilde gelmektedir. Önceki kısımda bahsedildiği gibi veriler art arda 8 örneklemeden oluşmaktadır. İlk olarak veriler 8’li şekilde ayrılmıştır. Veri setinde aynı durumların art arda eğitime sokulması eğitimin başarı yüzdesini düşürebilmektedir. Bu yüzden veriler 8’erli grup haline getirildikten sonra rastgele bir şekilde karıştırılmıştır. Tüm veriler sonrasında eğitim, test ve doğrulama olarak sırasıyla %60, %20, %20 olacak şekilde veri kümelerine bölünmüştür. Eğitim veri seti, modelin eğitiminde kullanılabilecek veri setidir. Eğitim sırasında doğrulama veri setiyle eğitimin optimizasyonu sağlanır. Model eğitimi tamamlandıktan sonra modelin performansı eğitim sırasında kullanılmamış olan test veri seti ile sağlanmaktadır.

Verilerin kümeleme işlemi yapıldıktan sonra kullanılacak olan yapay sinir ağının hazırlanması gerekmektedir. Bu uygulamada TensorFlow kütüphanesinden Keras ardışık, çok katmanlı yapay sinir ağı modeli kullanılacaktır. Yapay sinir ağı modeli 24 nöronlu giriş katmanı ile başlamaktadır. 24 nöronun olma nedeni x,y,z bilgilerinden 8 örnekleme ile girişin yapılmasındandır. Yapay sinir ağı giriş katmanından ara katmanlara geçiş yapmaktadır. Yapılan çalışmada 6 nöronlu 1 adet ara katman sonuç çıktılarında iyi sonuç vermiştir. Giriş ve ara katmanda aktivasyon fonksiyonu için ReLU kullanılmıştır. Son olarak çıkış katmanı oluşturulmuştur. Çıkış olarak 2 durum olduğundan 2 nöronlu olmaktadır. Aktivasyon fonksiyonu olarak Softmax fonksiyonu kullanılmıştır. Bu fonksiyon ile çıktılar 0 ve 1 arasında olasılık döndürmektedir. Şekil4.12’de yapay sinir ağı görselleştirilmiştir.

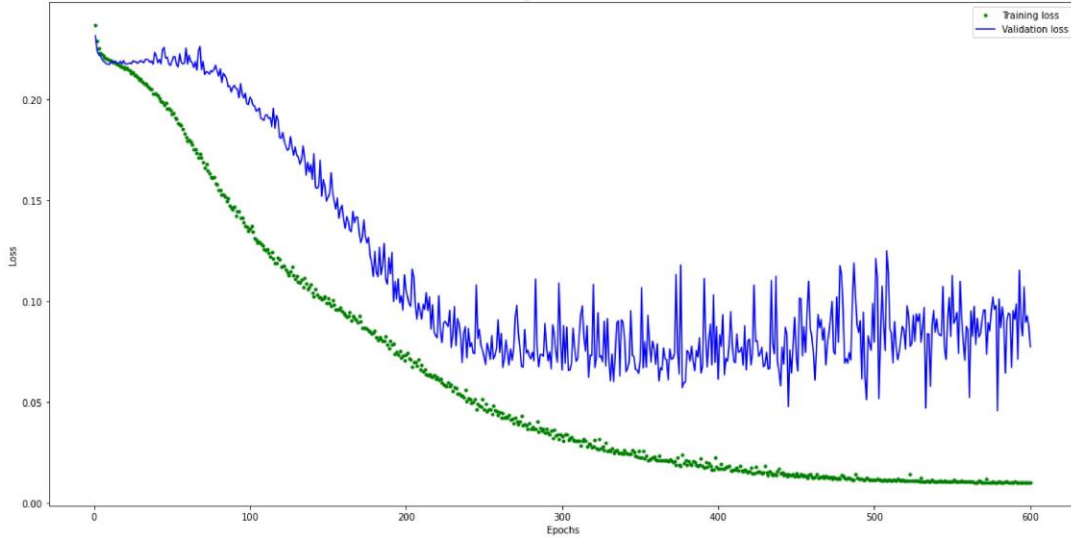


Şekil 4.12. Kullanılan yapay sinir ağı temsili şeması

Yapay sinir ağıyla model eğitimi bu şekilde gerçekleştirilmektedir. Model oluştuktan sonra girdi değerleri verilip çıktı sonuçları tahmin edilebilmektedir. Örnek tahmin sonuçları Şekil4.13'teki gibidir. Tahmin sonuçları ve gerçek değerler incelendiğinde eğitilen modelin doğruluk oranı %97'dir. Şekil4.14'te eğitilen modeldeki eğitim ve doğrulama kayıpları grafiği verilmiştir.

```
[0.983 0.017]
[0.999 0.001]
[0.999 0.001]
[0.995 0.005]
[0.018 0.982]
[0.996 0.004]
[0.994 0.006]
[0.995 0.005]
[0.995 0.005]
```

Şekil 4.13. Çıktı sonuçları örneği



Şekil 4.14. Eğitim ve doğrulama kayıpları grafiği

4.2.3 Modelin dönüştürülmesi ve gömülü sisteme aktarılması

Önceki bölümde oluşturulmuş bu model küçük cihazlarda kullanılmak için TensorFlow Lite formatına dönüştürülmesi gerekmektedir. Dönüştürmek için kütüphanedeki fonksiyon kullanılmış ve .tflite formatında kayıt edilmiştir. Kaydedilen bu dosyanın gömülü yazılım içine aktarılması için STM32CubeIDE içerisinde bulunan X-CUBE-AI paketi kullanılmıştır. Dosyanın içeri aktarılması için yazılım paketi yüklendikten sonra yeni ağ ekleme kısmından dosyayı seçmek yeterlidir. İçeri aktarma işlemi bittikten sonra proje dosyalarının arasında modelin .h ve .c dosyaları oluşmaktadır. Bu kısımdaki kodlar ile eğitilmiş olan yapay sinir ağı modeli kullanılabilmektedir. Şekil4.15'te yapay sinir ağındaki ağırlıkların nasıl aktarıldığını gösteren kodlardan örnek bir kesit bulunmaktadır.

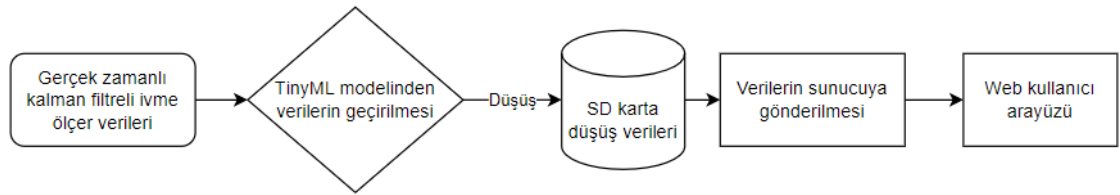
```
0xbe9a73833e6e7252U, 0x3de8eefcbdcf465cU, 0xbe45a10c3e190ffeU, 0xbe31c98c3cdd1910U,
0x3e98c3c7be489c4eU, 0x3e6310163d114b50U, 0xbdf998bc3d1b3d18U, 0x3d2c0138bd306f58U,
0xbe6738703d06f810U, 0x3d91e90cbdb49458U, 0xbef2920ebe97ae67U, 0xbdf36e8e3e9f0fd1U,
0xbe43e2c6be800080U, 0xbe9a8f983e02dafcU, 0xbea2d9f43e90ca23U, 0x3ea73291be53efe2U,
0xbe4970593d6669f8U, 0xbe81f24a3e85b72bU, 0x3dec1fc83d16bcc0U, 0xbd7ce768bc270ee0U,
0xbd96a79d3d6190a3U, 0x3e0915bcbd576e74U, 0x3f0bfe5abe7f8b96U, 0x3f2f683e3e40b6ffU,
0xbe2f0b6abd90e8adU, 0xbda21c82be652660U, 0xbe681ab4bee444c3U, 0x3c3a3efebd9d1e15U,
0xbe9c00913d379126U, 0xbeba8f5f5e57f40eU, 0x3e37a366bd622d18U, 0x3d1f572c3c51bae2U,
0x3dcd42b73eda4ae3U, 0xbc6ac92fdbc1709U, 0xbe991d333e8fd5deU, 0x3e03221d3d4c1c06U,
0xbe5478a13e92166fU, 0x3e4abb9d3e2de6e3U, 0x3e126d50be670843U, 0x3ea7b3e53e748ce1U,
0x3deb68983e35c292U, 0xbe8022a03df58a59U, 0x3e77a88abe66cd0fU, 0xbe85588cbe90b71eU,
0x3dc736cd3cf1ede3U, 0x3d24de1fbd5b4560U, 0x3ea51c7abc1e2960U, 0x3e654ae83eeb30cbU,
0x3d5ce5363e3e184cU, 0x3ec28368beb1de3bU, 0xbeedef8cbea9179cU, 0xbdcbc265bca9c105U,
0xbef4660c3e9a95a2U, 0xbe81e5863e57cce4U, 0x3e4de075bd8f839aU, 0xbe844d33bd2deea2U,
0x3dcd2849be778a6fU, 0x3ed70f40bd4c4634U, 0xbefaf3713eb8255fU, 0xbfd85f18bee8cf32U,
0xbda068653f053f9dU, 0xbe0e4fa13f0a344cU, 0xbdaba617be1c64baU, 0xbdf4ad45beae5649U,
0x3ea769843ea38632U, 0xbe9364013d6f32cfU, 0x3ed53c2d3e521ed9U, 0xbdde1f0b3e51c771U,
0x3e423a12be8db33bU, 0x3d73cb463d7f8a74U, 0x3eb49b4e3e1a3b49U, 0x3e33ed19bddd8825U,
0xbe286bed3ca8cb1cU, 0xbd86ff7f5e0f9bcfU, 0xbd7d4b183de9819bU, 0xbe1b5ff8bec6b679U,
0xbcd317693d105cccU, 0x3e8581053d8d6534U, 0x3e53947f5e15f5dcU, 0x3ea8d3dc3d2d7d84U,
0x3d77e934bce026f5U, 0x3d7d089d3d3a0425U, 0x3dbd0cb53dca771cU, 0x3f3e4aab3e8af5d9U,
```

Şekil 4.15. Yapay sinir ağı ağırlık katsayıları

Yapay sinir ağının dönüştürülmüş kodları main.c içerisine aktarıldıktan sonra gerekli başlatma ayarlamaları yapılması gerekmektedir. Girdi ve çıktı bilgilerinin atanabilmesi için girdi boyutunda ve çıktı boyutunda dizi oluşturulmuştur. Cihaz ilk açılıp girdi miktarı kadar veri toplandıktan sonra tahmin işlemleri başlamaktadır. Paketin çıkartmış olduğu fonksiyonlar sayesinde yapay sinir ağında tahmin işlemleri yapılmaktadır. Her döngüde kalmanlı ivme verileri dizi kaydırılarak eklenir ve yeni girdi dizisi oluşturularak çıktı durumu kontrol edilmektedir. Böylece eğitilmiş olan model ile sürekli olarak durum tespiti yapılabilmektedir. Eğer düşüş verisi tespit edilirse SD kart içerisine saat bilgileri ile birlikte kaydedilmektedir.

4.3 Sonuçların Sunucuya Gönderilmesi ve Kullanıcı Arayüzü

Bu aşamaya kadar olan kısımda ivme verileri başarı ile alındı ve veriler ile eğitilen model gömülü sisteme aktarıldı. Yapay sinir ağı modeli ile gerçek zamanlı durum tespiti yapıldı ve düşme durumlarında SD kart içerisinde bu bilgi kayıt edildi. Bu bilgiye kullanıcının erişmesi için bir kullanıcı arayüzü gerekmektedir. Bu projede kullanıcı arayüzü olarak web arayüzü seçilmiştir. Bunun nedeni interneti olduğu herhangi bir konumda cihazın verilerini sunucu ortamına göndererek gerçek zamanlı durum takibinin yapılabilmesini sağlamaktır. Bu kısımda verilerin sunucuya gönderilmesi ve kullanıcı arayüzü kısımları ele alınacaktır. Şekil4.16'te işleyiş aşamaları gösterilmiştir.



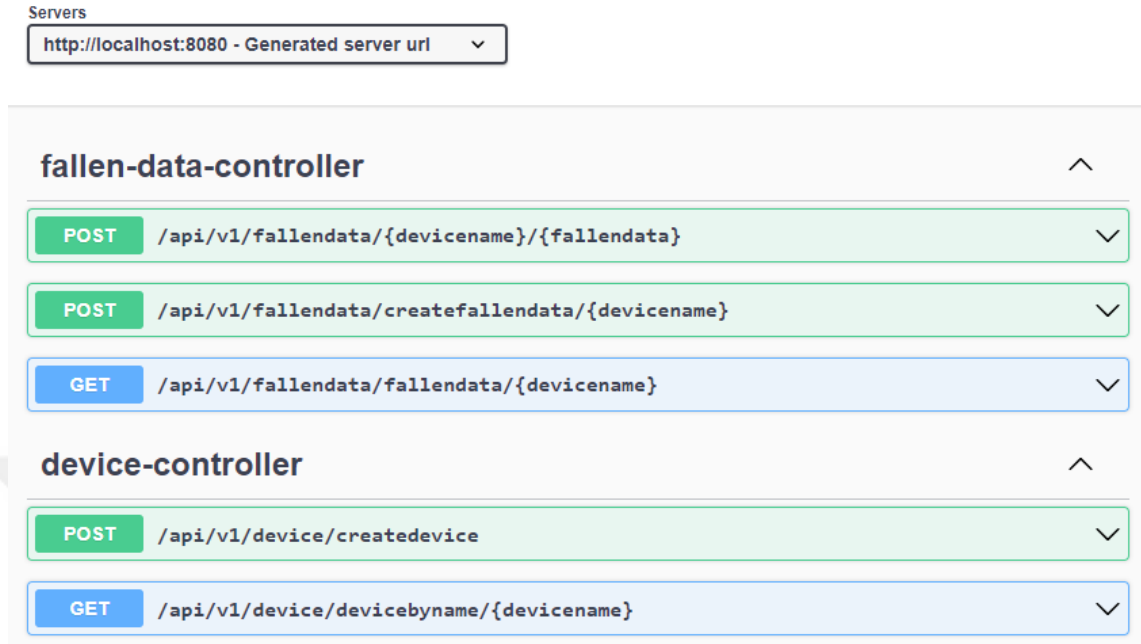
Şekil 4.16. Verilerin web arayüzünde gösterilme aşamaları

4.3.2 Sunucunun oluşturulması

Düşüş verilerinin gönderilmesi ve depolaması için bir sunucu gerekmektedir. Web uygulamasının arka kısımdaki işlemlerin yapılması için Java Spring Boot ile bir uygulama geliştirilmiştir. Bu uygulamaya yapılan HTTP istekler ile düşüş verileri Postgresql veri tabanına kayıt edilmektedir. Veri tabanındaki veriler arayüzde görselleştirilmektedir. Arayüz verileri almak için yine bu sunucuyu kullanmaktadır. Tasarlanan sistemde her cihazın bir cihaz numarası bulunmaktadır. Bu cihaz numaraları üzerinden istenilen cihazın verileri çekilebilmektedir.

Bu uygulama yazılırken öncelikle düşüş verisi ve cihaz bilgisi için sınıf dosyaları oluşturulmuştur. Sonradan eklenecek veriler bu sınıflardan türetilerek kullanılmaktadır. Veri tabanı ile bağlantısı yapılabilmesi için gerekli konfigürasyonlar yapılmıştır. Arayüz ve cihazlardan ulaşabilmek için noktalar oluşturulmuştur. Gerekli olan fonksiyonlar da

yazılarak veri tabanına kayıt işlemleri gerçekleştirilmiştir. Şekil4.17’da istek adreslerinin yapıları gösterilmiştir.



Şekil 4.17. Server istek adresleri

4.3.3 Sunucuya düşüş verilerinin gönderilmesi

Sunucu ortamına verilerin gönderilmesi için cihazın internete bağlanması gerekmektedir. Bunun için bu projede ESP8266 wifi modülü kullanılmıştır. ESP8266 modülü ile işlemler AT komutları ile yapılmaktadır. Cihaz içerisine wifi ve şifresi (4.1) bağlantısındaki komut UART üzerinde gönderilerek tanımlanmıştır. Böylece belirli bölgelerde aynı wifi adı ve şifresi ile modül sunucuya veri gönderebilecektir.

$$AT + CWJAB = "wifi_ismi", "wifi_sifresi" \quad (4.1)$$

Program içerisinde her döngüde SD kart içerisine düşüş verisi yazılıp yazılmama durumu kontrol edilmektedir. Eğer düşüş verisi ve wifi bağlantısı varsa sunucu ile TCP bağlantı başlatılmaktadır. Bu bağlantı (4.2) bağlantısındaki komut ile başlatılmaktadır.

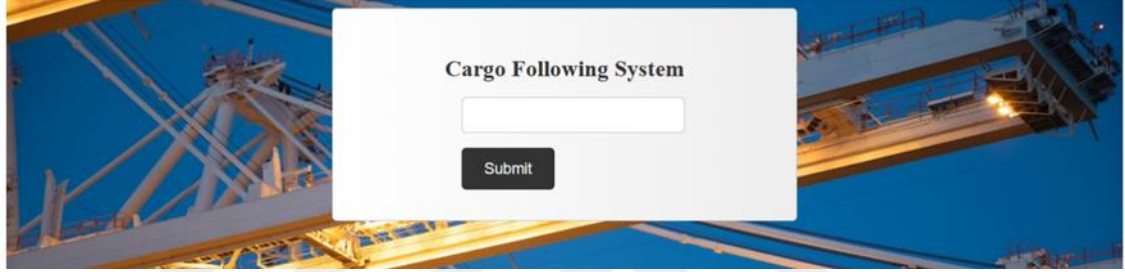
$$AT + CIPSTART = "TCP", "192.168.43.241", 8080 \quad (4.2)$$

Bağlantı kurulabilmiş ise veriler POST isteği olarak sunucuya gönderilmektedir. Öncelikle “AT+CIPSEND” komutu ile gönderilecek verinin boyutu belirtilmektedir. Sonrasında POST isteği ile “/api/v1/fallendata/{cihaz ismi}/{düşüş verisi}” adresine url

ile veri gönderilmektedir. Cihaz ismi her cihaz için özeldir. Bu sayede verinin hangi cihaza ait olduğu belirtilir ve sonraki kısımda da düşüş verisinin zamanı bulunmaktadır. Gönderilen düşüş verileri kayıt edilerek sonradan tekrar gönderilmez. Bu işlem her döngüde tekrarlanır, gönderilmeyen veri var mı diye kontrol edilir ve varsa bu şekilde gönderilir.

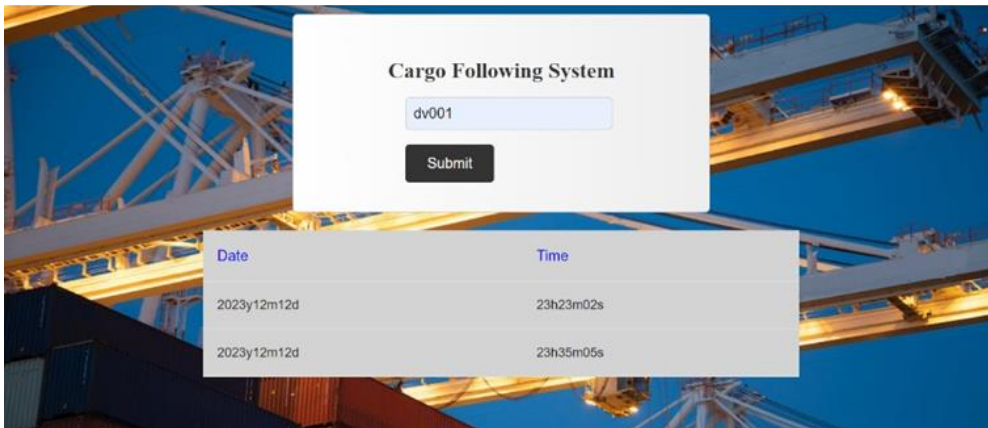
4.3.4 Kullanıcı arayüzünün oluşturulması

Tespit edilen düşüş verileri sunucuya gönderildikten ve kayıt edildikten sonra bunun son kullanıcıya aktarılması için bir arayüz gerekmektedir. Arayüz bilgileri oluşturulmuş olan adreslerden sorgular ile çekilmektedir. Şekil4.18’de arayüz gösterilmektedir.



Şekil 4.18. Arayüz giriş ekranı

Boş olan metin alanına cihaz numarası girilerek düşme durumları öğrenilebilmektedir. Eğer durumda bir değişiklik olursa tüm sayfayı yenilemeden durum bilgileri ekranda gösterilebilmektedir. Düşüş verilerinin olduğu durum Şekil4.19’deki gibi görülebilmektedir.



Şekil 4.19. Düşüş verilerinin gösterildiği ekran

6. SONUÇLAR

Gün geçtikçe hayatımızdaki teknoloji büyümekte ve gelişmektedir. Büyümenin kontrol altında ve bilinçli olması ilerleyen süreçteki yaşanacak sıkıntıların önüne geçecektir. Yapılan çalışmada da gözlemlendiği gibi yapay zeka günümüzde farklı cihazlara entegre edilebilir konuma gelmiştir. Entegre edilen cihazlar, fazla performans gerektirmeyen, yapılan çalışmadaki gibi küçük cihazlar da olabilmektedir. Bu durum uç noktalarda zorlu işlemleri eğitilmiş modeller ile yapabilme imkanı sunmuştur. Büyüyen veriler ile baş etmenin en önemli yollarından birisi de yüklerin dağıtılmasıdır. Dağıtılmış cihaz yükleri sayesinde bir noktada oluşabilecek yük azaltılabilmektedir.

Çalışmada TinyML ve uç bilgi işleme yöntemi kargo sistemlerine uyarlanmıştır. Bu yapı sensörden verinin alındığı her alana entegre edilebilir bir sistemdir. Uç noktalarda gerçekleştirilebilen bu teknoloji ile birlikte çevredeki yapay zekanın bulunduğu cihazları artırarak endüstrideki gelişime katkı sağlanabilir. Matematiksel çözümlerin zor olduğu durumlar için kolay bir çözüm yolu sunulabilmektedir.

%97 başarı oranına sahip olan eğitilmiş model, düşüşleri doğru bir şekilde tahmin etmektedir. Çıktı durumunda iki farklı senaryo bulunmaktadır. Veri setinin zenginleştirilmesiyle birlikte düşüşlerin yanı sıra, normal taşıma dışındaki farklı durumlar da tespit edilebilir. Ayrıca model üst kısmı her zaman yukarıya bakması gereken kargolar için özel olarak eğitilerek, düşüşleri ve kargonun istenen yönde kalma durumunu tespit etme yeteneği kazandırılabilir.

IoT cihazları ile toplanan veriler her geçen gün daha da artmaktadır. Bu verilerin analizi yapılmadıkça anlamsız bir veri yığınının başka bir şey olmayacaktır. Toplanan verinin büyümesi eğitimler için artı olarak düşünülsede bilinmeden çok sayıda veri de depolanmaktadır. Tezdeki yaklaşım ile birlikte anlamlı verinin biriktirilmesi iyileştirilebilir. Temiz verinin biriktirilmesi sonradan o verilerin kullanılmasını kolaylaştırılacak ve kullanım öncesi gereken veri temizleme, gruplama süreçleri gibi süreçleri hızlandıracaktır. Yığın ile birlikte gelen depolama sorununu da önemli ölçüde azaltarak kurulum, bakım maliyetleri gibi harcamaları da düşürecektir.

Uç noktadaki yapay zekanın çalıştırılmasıyla yapılabilecek farklı projelere bakacak olursak insan hareketlerinin tespit edilmesi örnek verilebilir. Yapılması gereken işlem çalışmadaki sıralamayla önce veri toplamak olacaktır. Toplanan verilerin temizlenmesi, eğitilmesi ve eğitilmiş modelin bir cihaza aktarılması ile tamamlanacaktır. Bir başka örnek verecek olursak konum bilgisinin çekmediği durumlarda ivme, yön verileri ile güncel konum bilgisi tahmini yapılabilir. Yapılması gereken gerçek konum bilgileri varken ,konum verilerinin ve ivme , yön verilerinin kayıt edilmesidir. Sonrasında bu veriler ile hareket verilerinin girdi olarak verildiği, çıktı olarak da konum bilgisinin değişim durumuyla yeni konumun tespit edilmesi olabilir. Bu gibi örnekler daha da arttırılabilir. Gündelik yaşantımızdaki birçok şey bu yapıya uygundur.

7. KAYNAKLAR

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., ... Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, 265-283. <https://arxiv.org/abs/1605.08695v2>
- Chen, Z., Gao, Y., & Liang, J. (2023). LOPdM: A Low-Power On-Device Predictive Maintenance System Based on Self-Powered Sensing and TinyML. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 72, 2023. <https://doi.org/10.1109/TIM.2023.3308251>
- Durgun, Y. (2023). Fall Detection Systems Supported by TinyML and Accelerometer Sensors: An Approach for Ensuring the Safety and Quality of Life of the Elderly. *International Scientific and Vocational Studies Journal*, 7(1), 55-61. <https://doi.org/10.47897/BILMES.1299289>
- Erdoğan, A. T. (2022). *Detection of cargo package drop*. <https://katalog.marmara.edu.tr/veriler/cokluortam/cokluortam/A/F/C/A/F/63b7e90621b3b.pdf>
- Espressif Systems IOT Team. (2016). *ESP8266 AT Instruction Set*.
- Gupta, P., Young, A., & Rao, A. (2022). *Investigating Cargo Loss in Logistics Systems using Low-Cost Impact Sensors*. 197-206. <https://doi.org/10.5121/csit.2022.120618>
- Hemmanur, K. (2009). *Inter-Integrated Circuit (I2C)*.
- Kim, J., Kim, E., Kyung, Y., & Ko, H. (2022). A Study on ECG Monitoring Embedded Systems. *International Conference on ICT Convergence, 2022-October*, 1671-1673. <https://doi.org/10.1109/ICTC55196.2022.9952978>
- Kocer, S., Dunder, O., & Butuner, R. (2021). *Programmable Smart Microcontroller Cards EDITORS*. www.isres.org
- Küçükateş, B. (2020). *İnsansız hava aracının boylamasına durum vektörlerinin kalman filtresi ve luenberger gözleyici ile kestirilmesi*. <https://gcris.etu.edu.tr/handle/20.500.11851/3930>

- Leens, F. (2009). An introduction to I2C and SPI protocols. *IEEE Instrumentation and Measurement Magazine*, 12(1), 8-13.
<https://doi.org/10.1109/MIM.2009.4762946>
- Lorenc, A., & Kuźnar, M. (2016). *Risk and costs evaluation in intermodal transport-variant analysis of using cargo positioning and parameters monitoring systems)-System for proactive management and analysis of resources View project Risk and costs evaluation in intermodal transport-variant analysis of using cargo positioning and parameters monitoring systems*. <https://www.researchgate.net/publication/306353604>
- Peña, E., & Legaspi, M. G. (2020). *UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter*. 54(4).
- Pimentel, J. F., Murta, L., Braganholo, V., & Freire, J. (2019). A large-scale study about quality and reproducibility of jupyter notebooks. *IEEE International Working Conference on Mining Software Repositories, 2019-May*, 507-517.
<https://doi.org/10.1109/MSR.2019.00077>
- Recoupex. (2021). *Recoupex Claim Statistics 2021*. <https://recoupex.com/reasons-and-statistics-for-cargo-claims/>
- Sapre, A., & Vartak, S. (2020). Scientific Computing and Data Analysis using NumPy and Pandas. *International Research Journal of Engineering and Technology*. www.irjet.net
- Setio Pambudi, D., Handayani, R., & Hidayah, L. (2018). *Template Matching Algorithm For Noise Detection in Cargo Container*.
- STMicroelectronics. (2020). *Discovery kit with STM32F407VG MCU*. www.st.com.
- STMicroelectronics. (2023). *STM32 C code of the optimized model and AI runtime (.a) Load an own trained neural network model Generate optimized code for STM32 Optimize and validate the NN model STM32Cube.AI for desktop STM32Cube ecosystem Command-line interface Neural network model converter for STM32 MCU X-CUBE-AI*. www.st.com
- UTİKAD Sektörel İlişkiler Departmanı. (2021). *LOJİSTİK SEKTÖRÜ RAPORU 2021*.
- Zhou, S., Du, Y., Chen, B., Li, Y., & Luan, X. (2022). An Intelligent IoT Sensing System for Rail Vehicle Running States Based on TinyML. *IEEE Access*, 10, 98860-98871. <https://doi.org/10.1109/ACCESS.2022.3206954>

ÖZGEÇMİŞ

HÜSEYİN OĞUZALP AKGÜL

ÖĞRENİM BİLGİLERİ

Yüksek Lisans	Akdeniz Üniversitesi
2020-2024	Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Antalya
Lisans	Akdeniz Üniversitesi
2014-2019	Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Antalya