



BURSA TEKNİK ÜNİVERSİTESİ

BTÜ-İMEP SONUÇ RAPORU

ÖĞRENCİNİN

Adı Soyadı	: Mehmet DAMKACI
Numarası	: 20332629030
Bölümü	: Elektrik Elektronik Mühendisliği
E-posta adresi	: mehmetdamkaci02@gmail.com
BTÜ-İMEP akademik danışmanı	: Salih Mehmed BOSTAN
BTÜ-İMEP sektör danışmanı	: Melih AKKAYA
Çalışma dönemi	: 25.09.2023 – 05.01.2024

BTÜ-İMEP Kurumunun:

Adı	: BAYKAR Teknoloji
Adresi	: Hadımköy, Başakşehir/İstanbul
İnternet sitesi	: baykartech.com
Görev alınan birim	: Yapay Zeka Yazılım Geliştirme

İçindekiler

ŞEKİLLER DİZİNİ	3
1. GİRİŞ	4
2. BAYKAR Teknoloji	5
2.1. Yapay Zeka Yazılım Geliştirme Birimi.....	6
3. ÇALIŞMALAR	7
3.1. Çalışmanın Hedefi.....	7
3.2. Yöntem	8
3.2.1. Soket Bağlantısı	8
3.2.2. Enum Eşleştirme	9
3.2.3. Ayarlar	11
3.2.4. Ana Tablo.....	12
3.2.5. Grafikler	15
3.2.6. Verilerin Kaydedilmesi.....	20
3.2.7. Dışarı Aktarma	21
3.2.8. Geçmiş Verileri Simüle Etme	25
3.3. Sonuçlar	26
4. ÇALIŞMA DÖNEMİNİN DEĞERLENDİRMESİ	27
5. KAYNAKLAR	28

ŞEKİLLER DİZİNİ

Şekil 2.1 Baykar firmasının geliştirdiği bazı ürünler	5
Şekil 3.1 Programın ana sayfası	7
Şekil 3.2 Soket bağlantı paneli	8
Şekil 3.3 Enum yapısının Dictionary'de tutulması	9
Şekil 3.4 Enum eşleştirme paneli	10
Şekil 3.5 Enum eşleştirmelerini gösteren tablo	11
Şekil 3.6 Ayarlar penceresi.....	12
Şekil 3.7 Ana tablo	13
Şekil 3.8 Ana tablo değerlerinin Dictionary kaynağı	14
Şekil 3.9 Paket Dağılım Grafiği	15
Şekil 3.10 Grafik eşleştirilmesi için oluşturulan algoritma.....	17
Şekil 3.11 Örnek frekans grafiği	17
Şekil 3.12 Örnek boyut grafiği.....	18
Şekil 3.13 Paketlerin proje dağılım grafiklerini temsil eden butonlar	19
Şekil 3.14 Örnek paket proje dağılım grafiği.....	19
Şekil 3.15 Kaydedilen örnek boyut verileri	20
Şekil 3.16 Program kapatılırken ekrana gelen uyarı paneli	21
Şekil 3.17 Genel Tablo sayfası.....	23
Şekil 3.18 Frekans Tablosu sayfası	23
Şekil 3.19 Frekans Grafikleri sayfası.....	24
Şekil 3.20 Boyut Tablosu sayfası	24
Şekil 3.21 Boyut Grafikleri	25
Şekil 3.22 Geçmiş verileri yükleme ve silme paneli	26

1. GİRİŞ

Yaklaşık 4 ay süren staj sürecim boyunca alanında uzman mühendisler ile çalıştım, çalışma ortamlarını yakından görerek bu ortama dahil oldum. Bu sebeple mezun olduğumda başlayacak olan çalışma hayatıma kolay uyum sağlayabileceğimi söyleyebilirim. Staj sürecim boyunca bir proje üzerinde çalıştım ve bu sayede bir projenin başlangıcından bitimine kadar olan sürecin nasıl ilerlediği noktasında önemli tecrübeler edindim. Proje üzerinde çalışırken araştırma yeteneğimin geliştiğini, etkili doküman okuma noktasında kendimi ileriye taşıdığımı, algoritma kurma ve geliştirme yeteneğimin geliştiğini ve bu konulardaki bilgi birikimimin arttığını söyleyebilirim. Bana mentorluk yapan mühendise günlük olarak projenin o anki durumunu, karşılaştığım sorunları raporlamam ve sözlü olarak bunları anlatmam gerekiyordu. Bu görüşmenin sonucunda ise karşılaştığım problemlere çözüm önerisi veya proje için yeni görevler alıyordum. İş durumunun yoğunluğuna göre bu görüşmelerin bazı zamanlar kısa sürmesi gerekiyordu bu sebeple gereksiz detayları gizleyerek ve ana konuya odaklanarak yaşadığım problemi kısa zamanda etkili bir şekilde anlatma noktasında kişisel anlamda kendimi geliştirdiğimi düşünüyorum.

Çalışma sürecim boyunca birçok opsiyonu olan bir masaüstü uygulaması geliştirdim. Planladığım kariyerimde yazılım üzerine çalışmak istediğim için gelecek kariyerime önemli faydalar sağlayacak bir staj süreci geçirdim. İleride savunma sektöründe çalışmak istiyorum, 4 ay boyunca stajyer olarak bu sektörü deneyimlemek benim için motive kaynağı ve gelecek kariyerim için önemli bir fırsat olduğunu düşünüyorum.

Staj sürecim boyunca C# ile WPF (Windows Presentation Foundation) uygulaması geliştirdim. Bu program sokete gelen paketlerin önce şifresini çözer ve sonrasında anlamlandırır. Geliştirdiğim masaüstü uygulaması kullanıcıya paketlerin kapsamlı analizini sunar. Bu sayede erken anormallik tespiti yapılabilir. Ayrıca program tüm anlamlandırdığı verileri kaydeder ve geçmiş verileri simüle edebilir. Bu sayede geçmiş veriler üzerinde işlem yapılabilir. Bu raporda yapmış olduğum proje ana başlıklara bölünmüştür. Her başlık altında teknik detaylardan, programın kullanılabilirliğinden ve kullanılan yöntemlerden bahsedilmiştir. Bahsedilen bölümün görseli de rapora eklenmiştir.

2. BAYKAR Teknoloji

Baykar, Türkiye merkezli bir makine imalat ve savunma şirkettir. Baykar Makina ismiyle 1986 yılında Özdemir Bayraktar tarafından otomotiv sanayine yerleştirmeye tabi hassas motor, pompa ve dişli kutusu parçalarının imalatı konusunda hizmet etme amacıyla kurulmuştur. Şirketin adı Bayraktar Kardeşler'in kısaltılmış hâlidir. 2000 yılından itibaren ailenin ikinci nesil mühendislerinin katılımı ile (Haluk ve Selçuk) Baykar bünyesinde savunma ve havacılık dünyasında insansız hava aracı (İHA) teknolojileri ile bu alanda Ar-Ge çalışmalarına başlanılmıştır. Şirket aynı zamanda Teknofest'in paydaşlarından biridir.[1]

Ar-Ge merkezi ve üretim tesisleri bünyesinde farklı mühendislik disiplinlerinden 3500+ kişilik (mühendis, tekniker ve teknisyen) Ar-Ge ekibiyle farklı sınıflarda yer alan sabit kanat ve döner kanat insansız hava araçları için Uçuş Kontrol Sistemleri, Görev Bilgisayar Sistemleri, Otomatik İniş ve Kalkış Sistemleri, Komuta Kontrol ve Arayüz Sistemleri, elektromekanik Servo Aktüatör, Elektrik Güç Birimleri vb. sistemler elektronik donanım, yazılım ve yapısal özellikleri ile tasarlanıp, prototip ve test aşamaları neticesinde butik olarak üretilmektedir. [1]



Şekil 2.1 Baykar firmasının geliştirdiği bazı ürünler

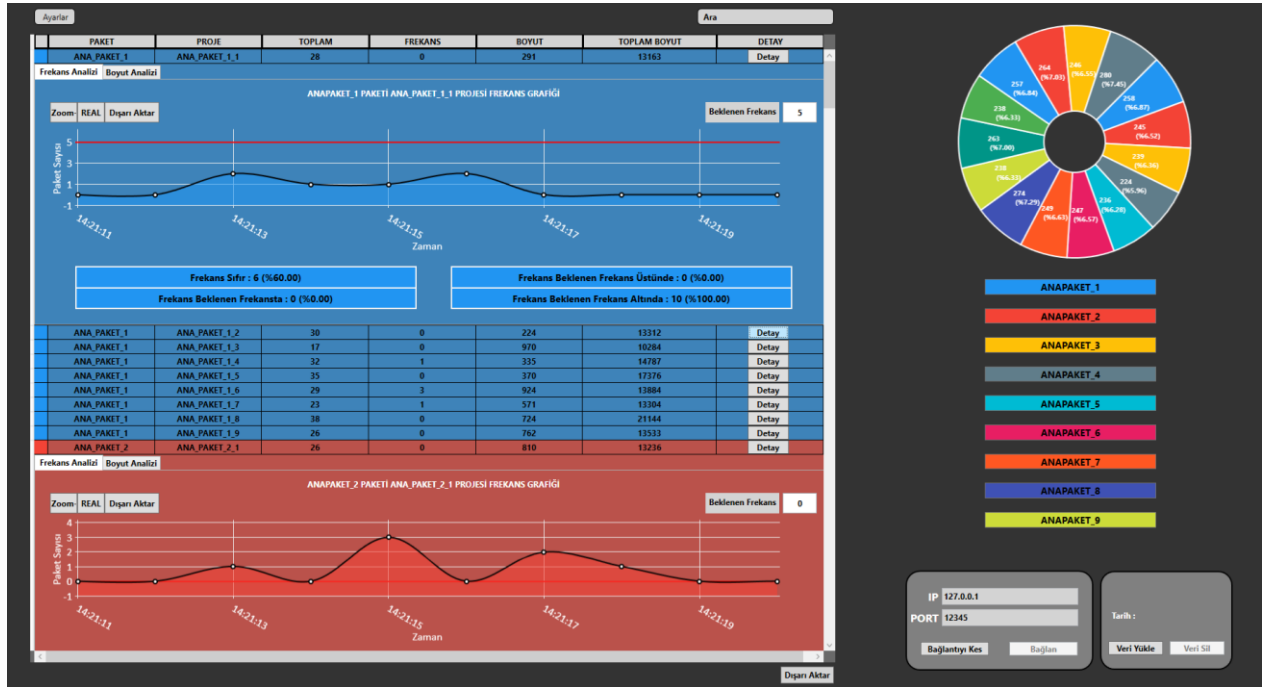
2.1. Yapay Zeka Yazılım Geliştirme Birimi

Yapay Zeka Yazılım Geliştirme Biriminde, Yapay zeka uygulamaları, akıllı otonomi algoritmaları, rota planlama, üst düzey yazılım mimarileri geliştirme, makine öğrenmesi, sinyal işleme ve big data gibi konularda çalışmalar yürütür. Birim kendi içerisinde Akıllı Sistemler, İleri Otonomi, Derin Öğrenme ve Dağıtık Sistemler olmak üzere 4 ekibe ayrılır. Birimin faaliyet alanları aşağıda sıralanmıştır.

- Hava aracı akıllı uçuş sistemleri
- Makina öğrenmeye dayalı algoritmalar
- Kural tabanlı uzman sistemler
- Derin öğrenmeye dayalı sistemler
- Sinir ağlarına dayalı mimari tasarımlar
- Reinforcement Learning (Pekiştirmeli Öğrenme) Sistemleri
- CNN ve görüntü tabanlı yapay zeka sistemleri
- Yapay zeka algoritmaları için veri bilimi yönetimi araçları
- Veri yönetim algoritmaları
- Yapay zeka sistemlerinin çalışacağı işletim sistemi mimarisi
- Doğal dil işleme kütüphaneleri
- Grup temelli rota takipçi algoritmaları
- Bilgisayarlı görü sistemleri
- Sinyal işlemeye dayalı algoritmalar
- Big data süreçleri

3. ÇALIŞMALAR

4 aylık staj sürecim boyunca bir proje üzerinde çalıştım. Proje sokete gelen paketlerin şifresini çözer ve daha sonra gelen paketleri anlamlandırarak kullanıcıya paketlerin kapsamlı analizlerini sunar. Proje bir çok bölümden oluşmaktadır. İçerisinde birçok thread ve task barındırır. Birçok fonksiyonu asenkron çalıştırarak performansı artırmak amaçlanmıştır. Yöntem kısmında proje birçok parçaya ayrılmıştır ve bu parçalar detaylı bir şekilde teknik detaylarıyla anlatılmıştır.



Şekil 3.1 Programın ana sayfası

3.1. Çalışmanın Hedefi

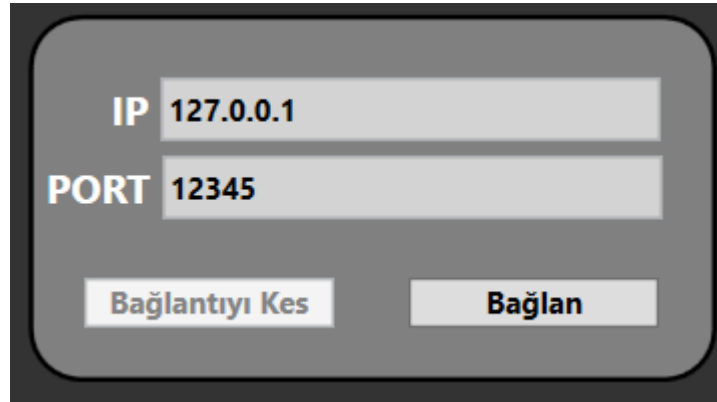
Çalışmanın hedefi, sokete gelen paketlerin anlamlandırılması ve çeşitli analizlerini kullanıcıya sunmaktır. Program, veriler üzerinde çeşitli filtreler yapabilir bu sayede analiz süresi kısaltılır. Program çalıştığı sürece gelen tüm paketler ve paketler üzerinde yapılan tüm analizleri kaydedilir. Bu sayede geçmiş veriler kaybedilmez. Kullanıcı kaydettiği verileri programa yükleyerek o zamanki veri akışını simüle edebilir. Simülasyon sürecinde süreci ileriye veya geriye sarma seçenekleri mevcuttur.

3.2. Yöntem

Projenin tüm bölümleri C# ile geliştirilmiştir. Program çeşitli parçalara bölünüp incelenebilir. Bu sebeple bu raporda proje 8 parçaya bölünüp, her parça detaylı bir şekilde ve görselleştirilerek anlatılmıştır.

3.2.1. Soket Bağlantısı

Bu panelde kullanıcıdan bağlanacağı IP adresi ve port numarası girmesi beklenir. Eğer kullanıcı başarılı bir şekilde girmiş olduğu adrese bağlanırsa ekran büyür ve tablo, grafikler ekrana gelir. Eğer geçersiz bir IP adresi veya port numarası girilmiş ise kullanıcıya kontrol mesajı gönderilir. Bağlan butonuna tıklayarak bağlantı yapılabilir ve bağlantıyı kes butonu ile de bağlantı kesilebilir.



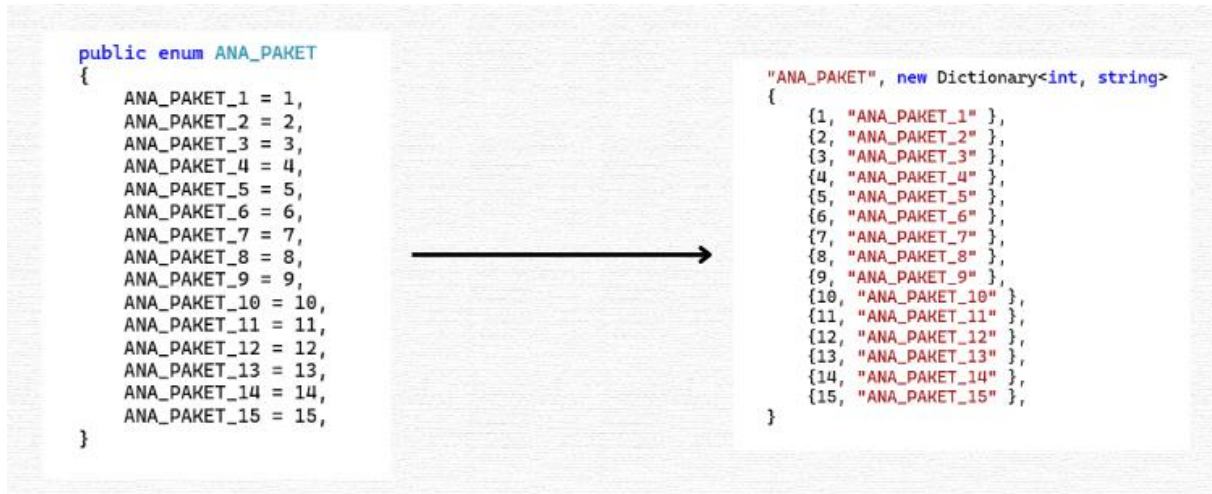
Şekil 3.2 Soket bağlantı paneli

Bağlantı kesildiğinde soket kapatılır. Soket paketleri bir thread içerisinde while döngüsü ile alır, bağlantısı kesildiği zaman bu thread kapatılır (Abort edilir). Bağlan butonuna tekrar tıklandığında yeni soket oluşturulur ve başarılı bir şekilde IP adresine bağlanırsa thread aktif edilir ve paketler tekrar alınmaya başlar. Eğer kullanıcı geçersiz bir IP adresi girerse uyarı paneli çıkar ve geçerli bir IP adresi girmesi beklenir. Sokete bağlı değilken “Bağlantıyı Kes” butonu aktif değildir. Bu durumda “Bağlan” butonu aktif durumdadır. Eğer sokete bağlıysak “Bağlantıyı Kes” butonu aktif edilir ve “Bağlan” butonu deaktif edilir.

3.2.2. Enum Eşleştirme

Enum yapısı, belirli bir türdeki sabit değer kümesini tanımlamak için kullanılır. Geliştirmiş olduğumuz programda paketler, içerisinde enumların bulunduğu .cs uzantılı bir dosyaya göre anlamlandırılır. Bu dosyada bir ana paket enumu bulunur. Bu enumun içerisinde paketlerin isimleri bulunur. Ana paket enumunun içerisindeki her bir elaman ise içerisinde proje isimleri olan bir enuma denk gelir. Paketlerin doğru anlamlandırılması için ana paket enumunun doğru belirlenmesi, paket ve proje enumlarının doğru eşleştirilmesi gerekir. Bu sebeple programın ayarlar sayfasında enum dosyasını değiştir kısmından enum dosyası eklenebilir ve gerekli eşleştirmeler yapılabilir.

Enum eşleştirilmesi yapılabilmesi için yüklenen .cs uzantılı dosyanın derlenmesi gerekir. Kullanıcı .cs uzantılı dosyayı yüklediğinde program bu dosyayı derler. İlk olarak kullanıcının yüklediği enum dosyası text olarak okunur. Metin olarak okunan enum dosyası C# syntax'ına parse edilir. Enum dosyası derlenir ve sonuçlar bir bellek akışına yazılır. Eğer derleme başarılı olmazsa bir uyarı paneli ile hata kullanıcıya belirtilir. Eğer derleme başarılı ise derlemede oluşan tüm türler bir listeye atılır. Enum türünde olanlar bu listeden çekilir ve Dictionary<string, Dictionary<int, string>> yapısında tutulur. Dictionary yapısının Key'i enum ismini Value kısmı ise enum elemanlarının isimlerini ve nümerik karşılıklarını ifade eder.



Şekil 3.3 Enum yapısının Dictionary'de tutulması

Eşleştirmelerin yapıldığı ekranda ana paket enumunun tüm elemanları listelenir ve bu elemanların karşısında seçim kutuları bulunur. Seçim kutusunun seçeneklerinde ana paket enumı hariç dosyadaki tüm enumlar bulunur. Seçim yapıldıktan sonra seçilen enum seçeneklerden kaldırılır.

Paket Enumı Olarak ANAPAKET Seçildi. Enum Eşleştirmelerini Yapınız.

ANAPAKET_1	PAKET_1
ANAPAKET_2	PAKET_2
ANAPAKET_3	PAKET_3
ANAPAKET_4	PAKET_4
ANAPAKET_5	PAKET_5
ANAPAKET_6	PAKET_6
ANAPAKET_7	
ANAPAKET_8	PAKET_7
ANAPAKET_9	PAKET_8
ANAPAKET_10	PAKET_9
ANAPAKET_11	PAKET_10
ANAPAKET_12	PAKET_11
ANAPAKET_13	PAKET_12
ANAPAKET_14	PAKET_13
ANAPAKET_15	PAKET_14

Kaydet

Şekil 3.4 Enum eşleştirme paneli

Kullanıcı her bir enumı eşleştirdiğinde eşleşmiş tüm enumlar bir tabloada gösterilir eğer kullanıcı bu eşleştirmeleri onaylarsa appData/Roaming dizinine, enum eşleşmelerine göre yeniden oluşturulan .cs uzantılı enum dosyası kaydedilir. Artık program, bu dosyaya göre gelen paketleri anlamlandırır. Şekil 3.4.'te bulunan “kaydet” butonuna tıklandığında tüm program güncellenir. Grafikler gibi paketlere göre düzenlenen her şey yeniden oluşturulur.

Program enum eşleştirmesi sonucunda bir konfigürasyon dosyası oluşturur. Bu dosyadaki bilgilere göre açılır program. Bu sayede en son yaptığımız enum eşleştirmesini değiştirmedığımız sürece program her zaman en son kullanılan enum dosyasına göre açılır.

ENUM İSMİ	EŞLEŞTİĞİ ENUM
ANA_PAKET.ANA_PAKET_1	PAKET_1
ANA_PAKET.ANA_PAKET_2	PAKET_2
ANA_PAKET.ANA_PAKET_3	PAKET_3
ANA_PAKET.ANA_PAKET_4	PAKET_4
ANA_PAKET.ANA_PAKET_5	PAKET_5
ANA_PAKET.ANA_PAKET_6	PAKET_6
ANA_PAKET.ANA_PAKET_7	PAKET_7
ANA_PAKET.ANA_PAKET_8	PAKET_8
ANA_PAKET.ANA_PAKET_9	PAKET_9
ANA_PAKET.ANA_PAKET_10	PAKET_10
ANA_PAKET.ANA_PAKET_11	PAKET_11
ANA_PAKET.ANA_PAKET_12	PAKET_12
ANA_PAKET.ANA_PAKET_13	PAKET_13
ANA_PAKET.ANA_PAKET_14	PAKET_14
ANA_PAKET.ANA_PAKET_15	PAKET_15

Kaydet

Şekil 3.5 Enum eşleştirmelerini gösteren tablo

3.2.3. Ayarlar

Ayarlar penceresi programın sol üst kısmında “ayarlar” butonuna tıklandığında açılır. Bu pencerenin sol üst kısmında ana paket enumının ismi yazar. Sol orta kısımda ise paketler listelenir. Bu bölümün altında ise enum dosyasını değiştirmek için tıklamamız gereken buton bulunur. Pencerenin orta kısmında bir tablo bulunur. Bu tabloda paketler projeleri ile listelenir. Beklenen frekans ve beklenen boyut değerleri bu tablo ile ayarlanır. Güncelle sütununda düzenleme butonu bulunur. Düzenleme butonuna tıklanmazsa beklenen boyut ve frekans değerleri salt okunur şekildedir. Düzenleme butonuna tıklanırsa eğer tıklanan satırın beklenen frekans ve boyut değerleri değiştirilebilir formata gelir. Bu sayede beklenen frekans ve boyut değerleri bu sayfa üzerinden güncellenebilir.

Pencerenin sağ kısmında grafik ayarları bulunur. Grafik uzunluğu gerçek zamanlı grafikte toplam kaç değer gösterileceğini belirten değerdir. Kaydetme uzunluğu ise program verileri kaç veride bir kaydedeceğini belirten değerdir. Örneğin grafik uzunluğu 20 olarak belirtilirse gelen son 20 veri grafikte gösterilir. Kaydetme uzunluğu 40 olarak belirtilirse her 40 veri geldiğinde veriler kaydedilir ve buffer boşaltılır bu sayede bellek şişmesi de önlenmiş olur.

Şekil 3.5.'te ayarlar penceresi görülmektedir. Her paket farklı renkte ifade edilir. Bu renkler ana sayfada belirlenir. Sağda paketler kısmında üzerine tıklanan paket, pencerenin ortasındaki tabloda filtrelenir. Sadece o pakete ait değerler tabloda gösterilir. Beklenen değerler güncellenirken pozitif bir tamsayı ile güncellenmelidir aksi takdirde program bir uyarı panelinde kullanıcıya bu durumu belirtir. Beklenen değerler güncellendiğinde programın konfigürasyon dosyasında da güncellenir. Bu sayede yapılan değişiklikler kaybedilmemiş olur.



The screenshot shows a software interface titled "SettingsWindow". On the left, there is a sidebar with a "Paket Enumu" section containing a list of packages: ANA_PAKET_1, ANA_PAKET_2, ANA_PAKET_3, ANA_PAKET_4, ANA_PAKET_5, ANA_PAKET_6, ANA_PAKET_7, and ANA_PAKET_8. Below this list is a button labeled "Enum Dosyasını Değiştir". The main area of the window displays a table with the following columns: "Paket Adı", "Proje Adı", "Beklenen Frekans", "Beklenen Boyut", and "Güncelle". The table contains 16 rows, grouped by package. The first 9 rows are for ANA_PAKET_1 (labeled "PAKET_1_1" to "PAKET_1_9"), the next 5 rows are for ANA_PAKET_2 (labeled "PAKET_2_1" to "PAKET_2_5"), and the last 2 rows are for ANA_PAKET_3 (labeled "PAKET_3_1" and "PAKET_3_2"). Each row has a value of 0 in the "Beklenen Frekans" and "Beklenen Boyut" columns, and a pencil icon in the "Güncelle" column. On the right side of the window, there are input fields for "Grafik Uzunluğu" (set to 10) and "Kaydetme Uzunluğu" (set to 3), and a "Kaydet" button.

Şekil 3.6 Ayarlar penceresi

3.2.4. Ana Tablo

Ana tablo paketlerin anlık olarak frekansını, boyutunu, toplam alınan paket sayısı ve alınan paketlerin toplam boyutunun ne olduğunu gösterir. Tablonun son sütununda detay butonları bulunur. Bu butona tıklandığında tablonun o satırı aşağıya doğru açılır. Açılan detay bölümünde frekans analizi ve boyut analizi olmak üzere iki sekme bulunur. İki sekmede de aynı opsiyonlar vardır. Grafikler, [Grafikler](#) bölümünde detaylı olarak anlatılmıştır.

BURSA TEKNİK ÜNİVERSİTESİ
PROJE TABANLI İŞLETMEDE MESLEKİ EĞİTİM
BTÜ-İMEP ÖĞRENCİ FİNAL RAPOR FORMU F4

PAKET	PROJE	TOPLAM	FREKANS	BOYUT	TOPLAM BOYUT	DETAY
ANA_PAKET_1	ANA_PAKET_1_1	3	0	379	1384	Detay
ANA_PAKET_1	ANA_PAKET_1_2	2	0	945	1469	Detay
ANA_PAKET_1	ANA_PAKET_1_3	3	0	213	1926	Detay
ANA_PAKET_1	ANA_PAKET_1_4	7	0	725	2651	Detay
ANA_PAKET_1	ANA_PAKET_1_5	4	0	425	1773	Detay
ANA_PAKET_1	ANA_PAKET_1_6	5	0	503	3527	Detay
ANA_PAKET_1	ANA_PAKET_1_7	1	0	128	128	Detay
ANA_PAKET_1	ANA_PAKET_1_8	8	3	176	4222	Detay
ANA_PAKET_1	ANA_PAKET_1_9	5	0	788	3122	Detay
ANA_PAKET_2	ANA_PAKET_2_1	8	0	292	4569	Detay
ANA_PAKET_2	ANA_PAKET_2_2	8	1	637	3466	Detay
ANA_PAKET_2	ANA_PAKET_2_3	3	0	690	1746	Detay
ANA_PAKET_2	ANA_PAKET_2_4	5	0	237	2642	Detay
ANA_PAKET_2	ANA_PAKET_2_5	4	1	201	2699	Detay
ANA_PAKET_2	ANA_PAKET_2_6	4	0	131	2111	Detay
ANA_PAKET_2	ANA_PAKET_2_7	6	2	605	3674	Detay
ANA_PAKET_2	ANA_PAKET_2_8	6	1	816	3258	Detay
ANA_PAKET_2	ANA_PAKET_2_9	3	1	991	2467	Detay
ANA_PAKET_2	ANA_PAKET_2_10	4	0	836	2321	Detay
ANA_PAKET_3	ANA_PAKET_3_1	5	0	615	2671	Detay
ANA_PAKET_3	ANA_PAKET_3_2	7	0	523	4011	Detay
ANA_PAKET_3	ANA_PAKET_3_3	8	0	273	3117	Detay
ANA_PAKET_3	ANA_PAKET_3_4	6	1	953	2461	Detay
ANA_PAKET_3	ANA_PAKET_3_5	6	1	716	3093	Detay
ANA_PAKET_4	ANA_PAKET_4_1	11	0	771	5996	Detay
ANA_PAKET_4	ANA_PAKET_4_2	9	1	939	6606	Detay
ANA_PAKET_4	ANA_PAKET_4_3	4	0	104	1437	Detay
ANA_PAKET_4	ANA_PAKET_4_4	13	3	992	8077	Detay
ANA_PAKET_4	ANA_PAKET_4_5	9	0	529	3613	Detay
ANA_PAKET_5	ANA_PAKET_5_1	14	0	245	6949	Detay
ANA_PAKET_5	ANA_PAKET_5_2	7	1	928	4670	Detay
ANA_PAKET_5	ANA_PAKET_5_3	15	1	734	8482	Detay
ANA_PAKET_5	ANA_PAKET_5_4	6	0	418	3913	Detay
ANA_PAKET_5	ANA_PAKET_5_5	7	1	271	4328	Detay
ANA_PAKET_6	ANA_PAKET_6_1	4	1	402	2698	Detay
ANA_PAKET_6	ANA_PAKET_6_2	6	1	435	2653	Detay
ANA_PAKET_6	ANA_PAKET_6_3	2	0	783	1405	Detay
ANA_PAKET_6	ANA_PAKET_6_4	0	0	0	0	Detay
ANA_PAKET_6	ANA_PAKET_6_5	6	1	439	3830	Detay
ANA_PAKET_6	ANA_PAKET_6_6	2	1	627	1385	Detay
ANA_PAKET_6	ANA_PAKET_6_7	2	0	278	593	Detay
ANA_PAKET_6	ANA_PAKET_6_8	6	1	280	2925	Detay
ANA_PAKET_6	ANA_PAKET_6_9	1	0	743	743	Detay

Şekil 3.7 Ana tablo

Sokete saniyede 500-1000 paket gelir. Bu paketler bir thread içerisinde alınır ve çözülür. Her paket geldiğinde tablo güncellenir. Yani tablo saniyede 1000 kez güncellenebilir. Ana paket enumına göre tüm paketler ve paketlerin anlık frekansı, anlık boyutu, toplam alınan paket sayısı ve toplam gelen boyut bir Dictionary yapısında tutulur. Bu yapı ile, program içerisinde çeşitli algoritmalar, diğer yapılar karşılaştırılır. String dizi yapıları nedeniyle doğrudan karşılaştırma yapılamaz. Bu sebeple hash kodu üzerinden karşılaştırma yapılır. Hash kodu , değişken uzunluklu veri kümelerini, sabit uzunluklu veri kümelerine haritalamak amacıyla kullanılan hash algoritması sonucu ortaya çıkan koddur [3]. Dizilerin hash kodunun hesaplanması için programda StringArrayComparer adında bir sınıf oluşturulmuştur. Bu sınıf için IEqualityComparer<T> arabirimi kullanılmıştır.

```
Dictionary<string[], int[]> packetStruct = new Dictionary<string[], int[]>
{
    { new[] { "ANA_PAKET_1", "ANA_PAKET_1_1" }, new[] { 4, 150, 540, 684655 } },
    { new[] { "ANA_PAKET_1", "ANA_PAKET_1_2" }, new[] { 3, 100, 440, 456151 } },
    { new[] { "ANA_PAKET_1", "ANA_PAKET_1_3" }, new[] { 6, 140, 290, 154690 } }
};
```

Şekil 3.8 Ana tablo değerlerinin Dictionary kaynağı

Şekil 3.7.'de örnek olarak bu Dictionary yapısı gösterilmiştir. Örneğin ANA_PAKET_1 paketinin ANA_PAKET_1_1 projesinin anlık frekansı 4, toplam bu paketten 150 adet gelmiş, anlık boyut değeri 540 byte ve gelen paketlerin toplam boyutu 684655 byte'dır. Bu yapı tabloya bağlıdır yani bu yapı içerisindeki değerler değiştiğinde bu değişim tablo üzerinde de gözlemlenir. Bu yapı enum eşleştirilmesinden sonra oluşturulur ve programdaki tüm grafikler ve tablolar bu yapıyı referans alarak güncellenir.

Grafiklerin bulunduğu detay kısmında grafik haricinde beklenen frekans ve boyut değerlerine göre analizler kullanıcıya sunulur. Bu analizler aşağıdaki gibidir:

- Gelen paketlerin kaç tanesinin beklenen değerde geldiği ve bu paketler toplam gelen paketlerin yüzde kaçını oluşturduğu
- Gelen paketlerin kaç tanesinin beklenen değer altında geldiği ve bu paketler toplam gelen paketlerin yüzde kaçını oluşturduğu
- Gelen paketlerin kaç tanesinin beklenen değer üstünde geldiği ve bu paketler toplam gelen paketlerin yüzde kaçını oluşturduğu
- Gelen paketlerin kaç tanesi sıfır ve bu paketler toplam gelen paketlerin yüzde kaçını oluşturduğu

Bu analizler sayesinde örneğin, kullanıcı frekans değerlerine baktığında bazı paketlerin beklenen değerden çok farklı geldiğini ve bu toplam gelen paketlerin önemli bir kısmını oluşturduğunu gördüğünde

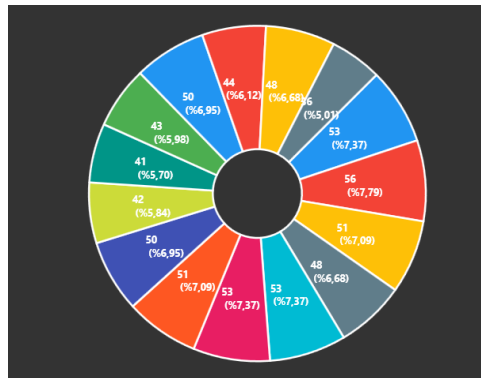
paketin geldiği kaynakta bir problem olduğunu anlayacak ve bu problemi erken düzeltebilecek. Yapılan analizler sonucunda ve verilerin görselleştirilmesi ile anormal durumlar kolay bir şekilde tespit edilebilir.

3.2.5. Grafikler

Programda verilerin görselleştirilmesi ve analizleri kolaylaştırmak adına pasta dilimi, bar grafiği ve çizgi grafiği olmak üzere toplamda 3 çeşit grafik bulunur. Çizgi grafikleri frekans ve boyut analizleri için kullanılır ve her paket için ayrı üretilir. Bar grafikleri paketlerin proje dağılımları gösterir ve her paket ayrı üretilir. Pasta dilimi grafiği her paket için toplam kaç tane geldiğini ve toplam paket sayısının yüzde kaçını oluşturduğunu gösterir. Grafikler için LiveChart.WPF kütüphanesi kullanılmıştır.

Paket Dağılımı Grafiği

Bu grafik kullanıcıya anlık olarak paketlerin dağılımını gösterir. Paketlerin tablolarda ve diğer grafiklerdeki rengi bu grafiğe göre belirlenir. Dilimlerin üzerinde paketten kaç adet geldiği ve tüm paketlerin yüzde kaçını oluşturduğu bilgisi yazar. Paketlerin anlık akışını bu grafikten gözlemleyebiliriz. Grafikte üzerine tıkladığımız dilim hangi pakete denk geliyorsa tabloda sadece o pakete ait veriler gösterilir ve bu paketin proje dağılımını ifade eden bar grafiği açılır. Bu şekilde bu grafik üzerinden kolay ve anlaşılır bir şekilde filtreleme yapılabilir. Enum dosyası değiştirildiğinde grafik sıfırlanır ve yeniden oluşturulur.



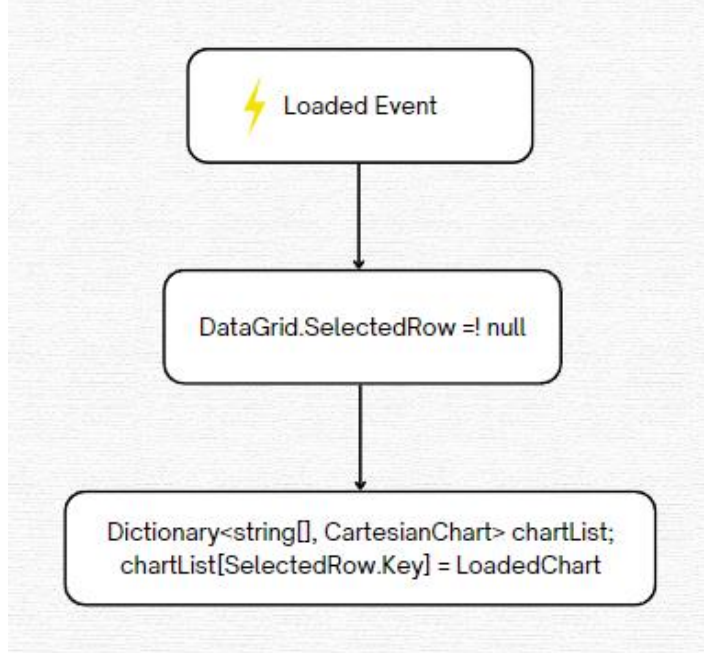
Şekil 3.9 Paket Dağılım Grafiği

Frekans ve Boyut Grafikleri

Frekans ve boyut grafikleri çizgi grafikleridir. X eksenleri zamanı ifade eder Y eksenini ise frekans grafiğinde frekansı, boyut grafiğinde boyutu ifade eder. Y eksenindeki değer çizgi grafiklerinde daire ile işaretlenmiştir. Kullanıcı grafik üzerinde kaydırma, yakınlaştırma ve uzaklaştırma yapabilir. Bu sayede çeşitli analizlerin yapılmasında kolaylık sağlanmıştır. Grafiklerde beklenen değer kırmızı bir çizgi ile çizilmiştir. Kullanıcı gelen paketlerin beklenen değerinin altında veya üstünde olduğunu kolayca gözlemleyebilir. Grafiklerin başlığı grafik türünü, paket adı ve proje adından oluşmaktadır. Grafiklerin animasyon özellikleri, performansı artırmak adına devre dışı bırakılmıştır. Bu şekilde programın kaynak tüketimini azaltmak amaçlanmıştır.

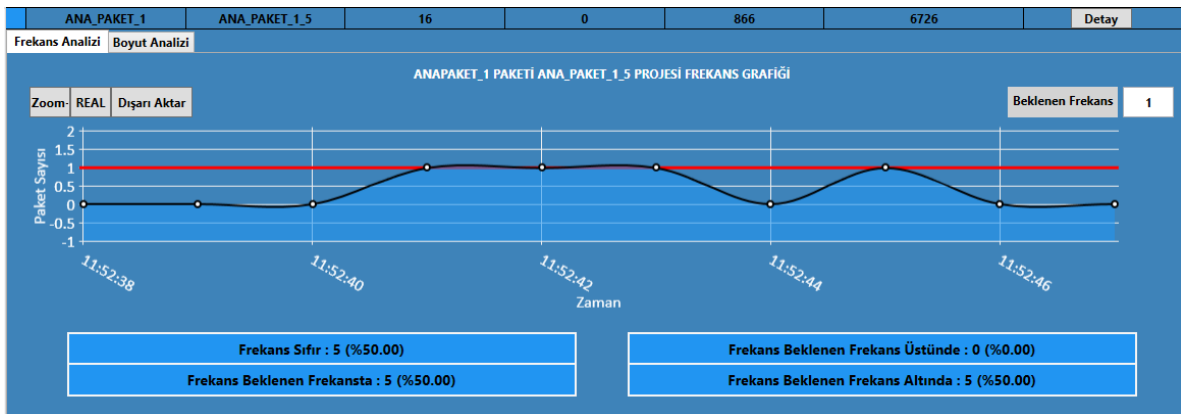
Grafiklerin üzerinde bulunan “ZOOM-” butonu ile tüm veriler grafiğe sığdırılır, kullanıcının tüm verileri tek bir çerçevede görmesine imkan tanır. “REAL” butonu verilerin gerçek zamanlı olarak akmasını görselleştirir. Grafikte her zaman 20 veri gösterilir. Her yeni veri geldiğinde grafik bir veri kadar sola kaydırılır. Grafikler üzerinden bulunan bir diğer buton ise “Dışarı Aktar” butonudur. Bu butona tıklandığında grafik excel dosyasında, kullanıcının istediği konuma, kaydedilir. Dışarı aktarma işlemleri [Dışarı Aktarma](#) bölümünde daha detaylı bir şekilde anlatılmıştır.

Grafikler `Dictionary<string[], CartesianChart>` yapısında tutulur. Sözlüğün Key değerinde bulunan string dizisi iki elemandan oluşmaktadır. Birinci elemanı paket ismini ikinci elemanı ise proje ismini ifade eder. Her paket ve proje ikilisi farklı grafikler üretilir ve bu yapıda tutulur. Programın XAML kısmında tablonun detay bloğunda grafik tanımlanmıştır. Detay butonuna tıklandığında grafik yüklendiği için Loaded olayı tetiklenir. Bu olay tetiklendiğinde seçilen satır tespit edilir. Bu satırdan paket ve proje ismi alınır ve `Dictionary<string[], CartesianChart>` yapısında eşleşen anahtar değerine karşılık gelen grafiğe yüklenen grafik atanır. Paket ve proje isimleri ile sözlük yapısının anahtar değerinin eşleştirilmesi hash kodu üzerinden, özel olarak oluşturulan `StringArrayComparer` sınıfı tarafından yapılır.



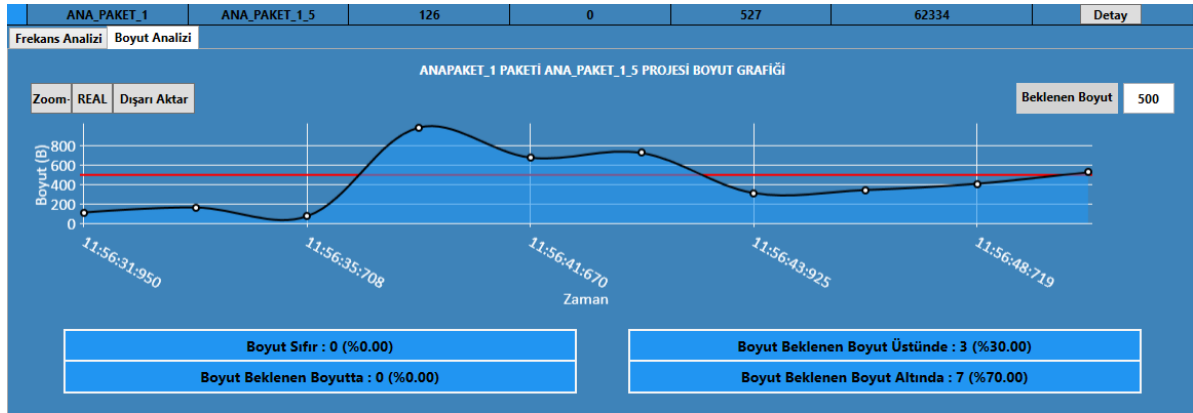
Şekil 3.10 Grafik eşleştirilmesi için oluşturulan algoritma

Frekans grafiği, paketten bir saniyede kaç adet geldiğini ifade eder. Bu grafik saniyede bir güncellenir. Tüm paketler için X eksenı aynı değerlere sahiptir. Zaman “saat:dakika:saniye” olarak ifade edilir. Program başladığından itibaren her bir saniye bir event (olay) oluşturulur. Bu event içerisinde her paket için frekans değeri hesaplanır ve ana tablo bu değerler ile güncellenir.



Şekil 3.11 Örnek frekans grafiği

Boyut grafiği, gelen paketlerin byte cinsinden uzunluğunu ifade eder. Bu grafik her yeni paket geldiğinde güncellenir. Her paket için X eksenini değişiklik gösterir. X eksenini zamanı “saat:dakika:saniye:milisaniye” olarak ifade eder. Bu grafik soketten paketlerin alındığı thread içerisinde güncellenir.



Şekil 3.12 Örnek boyut grafiği

Paket Proje Dağılım Grafiği

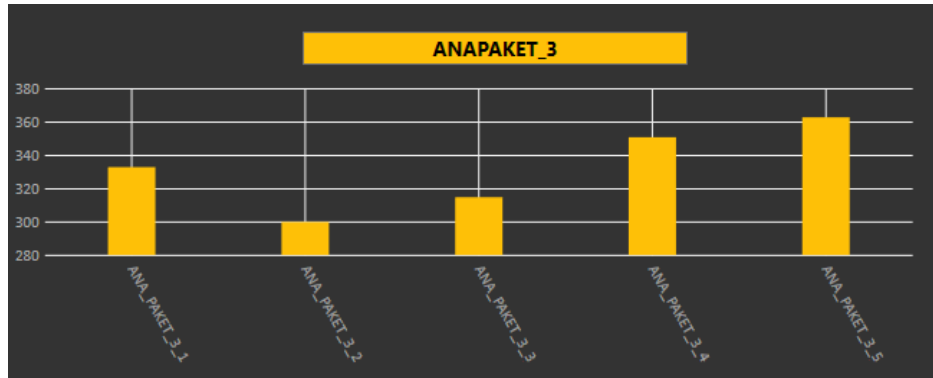
Bu grafik bir bar grafiğidir. Her paket için ayrı olarak üretilir. Paketlerin içerdiği projelerin dağılımını gösterir. Y eksenini proje sayısını, X eksenini proje ismini ifade eder. Her yeni paket geldiğinde bu grafikler güncellenir. Ana sayfanın sağ orta kısmında her paketin ismini içeren butonlar bulunur. Bu butonların rengi tablodaki paket renkleri ile aynıdır. Her paket farklı bir rengi temsil eder. Bu butonlardan birine tıklandığında sadece sayfada üzerine tıklanan buton kalır ve bu butonun temsil ettiği paketin proje dağılım grafiği ekrana gelir. Buton üzerine tekrar tıklanıldığında grafik kapanır ve bütün butonlar tekrar ekrana gelir.

Dışarı aktarma işlemlerinde bu grafik de excel dosyasında dışarı aktarılır. Bu sayede kullanıcı paket proje dağılım grafikleri üzerinde excel uygulamasının sunduğu çeşitli imkanlarla da analizler yapabilir.



Şekil 3.13 Paketlerin proje dağılım grafiklerini temsil eden butonlar

Şekil 3.14.'te ANAPAKET_3 butonuna tıklandığında ekrana gelen grafik gösterilmiştir. X ekseninde bu paketin içerdiği 5 adet proje gözlemlenmektedir. Örneğin ANA_Paket_3_2 projesini içeren toplam 300 adet paket alınmıştır. Bu grafik sayesinde pakete ait projelerin hepsi tek bir grafik üzerinde gözlemlenebilir ve bazı analizler yapılabilir.



Şekil 3.14 Örnek paket proje dağılım grafiği

3.2.6. Verilerin Kaydedilmesi

Programda kısa süre içerisinde çok fazla paket akışı olur. Grafiklerin oluşturulması için ve program içerisinde birçok algorithmada List, Dictionary, Array yapıları kullanılır. Program çalıştığı sürece bu yapılara sürekli olarak veri eklenir. Bu sebeple kullanılan bilgisayara bağlı olarak kısa veya uzun sürede bellek dolar ve şişer. Bu durumu önlemek adına veriler sürekli olarak kaydedilir.

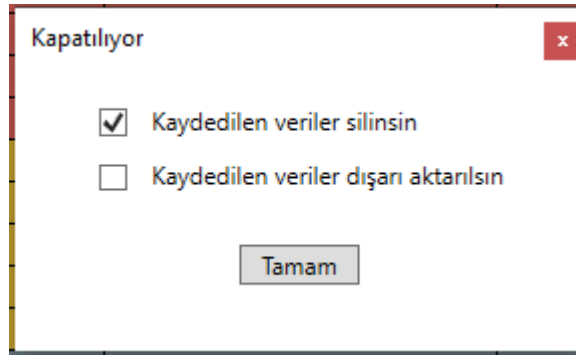
Program çalışmaya başladığında appData/Roaming dizininde program adına açılan klasörün içerisinde DATA klasörü oluşturulur. Bu klasörün içerisine ana paket isminde klasör oluşturulur örneğin; ana paket enumının ismi ANA_PAKET ise belirtilen dizinde ANA_PAKET adında klasör oluşturulur. Bu klasörün içerisine programın çalıştırıldığı zamanı isim olarak klasör eklenir. Bu klasör isminin formatı “gün-ay-yıl—saat-dakika” şeklindedir. Zaman klasörünün içerisinde FREKANS ve BOYUT olmak üzere iki klasör oluşturulur. FREKANS klasörünün içerisinde tüm paketlerin adında metin dosyaları oluşturulur bu dosyaların ismi “paket ismi_proje ismi” şeklindedir. Örneğin ANA_PAKET_1 paket adına ve ANA_PAKET_1_5 projesine sahip paketin metin dosyasının ismi ANA_PAKET_1_ANA_PAKET_1_5 şeklindedir. Bu dosyaların içerisine frekans ve boyut bilgileri “zaman, değer” şeklinde yazılır. Boyut verilerinin kaydının tutulduğu örnek bir metin dosyasının içeriği Şekil 9’da gösterilmiştir.

```
ZAMAN,BOYUT
11:52:10:996, 674
11:52:11:722, 523
11:52:12:073, 520
11:52:13:712, 973
11:52:13:803, 718
11:52:16:285, 199
11:52:16:627, 376
11:52:17:145, 231
11:52:19:631, 508
11:52:19:770, 556
11:52:23:551, 842
11:52:25:935, 526
11:52:26:824, 148
11:52:27:823, 883
11:52:28:541, 684
11:52:31:225, 470
```

Şekil 3.15 Kaydedilen örnek boyut verileri

Ayarlar penceresinin sağ kısmında yer alan grafik uzunluğu ve kaydetme uzunluğu bellek şişmesini önlemek için kullanıcı tarafından belirlenir. Grafik uzunluğu grafikte ne kadar verinin gösterileceğini ifade eder. Grafik verisinin tutulduğu yapının uzunluğu kullanıcının belirlediği grafi uzunluğuna eşitlendiğinde grafiğin en başındaki veri X ve Y değeri olmak üzere bir buffera atılır. Kaydetme uzunluğu ise bu bufferın uzunluğunu belirtir. Eğer bufferın uzunluğu kaydetme uzunluğuna eşit olursa yukarıda belirtilen dizindeki kaydetme dosyasına X ve Y eksenı değerlerini sıralı bir şekilde yazar. Bu sayede buffer temizlenmiş olur ve bellek şişmesi önlenir. Dosyaya yazma işlemleri Task içerisinde gerçekleşir. Bu sayede bu işlem yapılırken programın diğer fonksiyonlarında aksama veya veri kaybı olmaz.

Program kapatılırken kullanıcının karşısına bir uyarı paneli çıkar. Bu paneldeki seçeneklerden bir tanesi “kaydedilen veriler silinsin” seçeneğidir. Eğer bu seçenek seçilirse programın çalıştığından itibaren kaydedilen tüm veriler silinir ve program sonlandırılır. Bu işlemi yaparken “Windows_Closing” olayı kullanılır.



Şekil 3.16 Program kapatılırken ekrana gelen uyarı paneli

3.2.7. Dışarı Aktarma

Ana sayfanın orta sağ alt kısmında dışarı aktarma butonu bulunur. Bu butona tıklandığında kullanıcıdan dışarı aktarılacak verilerin kaydedileceği konumu seçmesi beklenir. Konum başarılı bir şekilde seçildiyse “Dışarı Aktar” butonu ortadan kalkar ve yerine hareket eden toplar ve “Veriler dışarı

aktarılıyor” yazısı gelir. Veriler başarılı bir şekilde dışarı aktarıldıysa hareket eden toplar ve pencerenin en altındaki “Veriler dışarı aktarılıyor” yazısı ekrandan kaybolur, “Dışarı Aktar” butonu eski yerine geri gelir.

Veriler dışarıya excel dosyası olarak aktarılır. Dışarı aktarma işlemi başlamadan önce tüm bufferlar boşaltılır ve metin dosyasına veriler yazılır de daha sonra dışarı aktarma işlemleri başlar. Program içerisinde verilerin dışarı aktarılması için oluşturulan sınıf içerisindeki metotlar kullanılarak dışarı aktarma işlemi gerçekleştirilir.

[Verilerin Kaydedilmesi](#) bölümünde belirtilen dosya dizinindeki tüm metin dosyaları sırasıyla bir while döngüsü içerisinde okunur. Okunan değerler excel işlemleri için kullanılan bir C# kütüphanesi fonksiyonları ile excel dosyasına yazılır. Bu sayede kullanıcı geçmiş verileri kaybetmez ve excel üzerinden istediği analizleri yapabilir. Dışarı aktarılan excel dosyası 5 sayfadan oluşur:

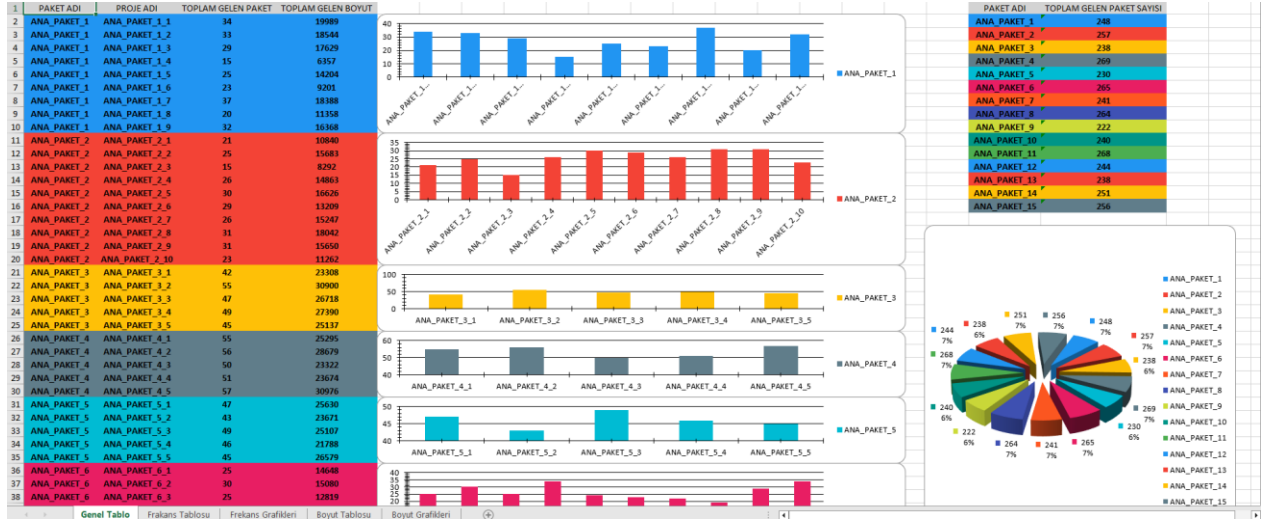
1. Genel Tablo
2. Frekans Tablosu
3. Frekans Grafikleri
4. Boyut Tablosu
5. Boyut Grafikleri

1. Genel Tablo

Genel tabloda ana tablodaki veriler, paket dağılım grafiği ve paket proje dağılımı grafikleri bulunur. Genel tablo üzerindeki renkler programdaki paket renkleri ile aynıdır. Kullanıcı bu sayfaya bakarak paketler üzerinde genel bir bilgi elde edebilir.

Bu sayfadaki tablo Paket Adı, Proje Adı, Toplam Gelen Paket, Toplam Gelen Boyut olmak üzere dört sütundan oluşur. Aynı paket adına sahip satırların karşısında paket proje dağılım grafikleri bulunur. Sağ üstte hangi paketten ne kadar geldiğini gösteren bir tablo bulunur. Bu tablonun altında ise paket dağılımını gösteren 3 boyutlu pasta dilimi grafiği bulunur. Genel olarak bu sayfa üzerinden bazı analizler yapılabilir. Bu sayfadaki amaç programın ana sayfasında bulunan tüm verileri kullanıcıya excel formatında sunmaktır.

BURSA TEKNİK ÜNİVERSİTESİ
PROJE TABANLI İŞLETMEDE MESLEKİ EĞİTİM
BTÜ-İMEP ÖĞRENCİ FİNAL RAPOR FORMU F4



Şekil 3.17 Genel Tablo sayfası

2. Frekans Tablosu

Bu tabloda tüm paketlere ait zaman ve frekans değerleri bulunur. Tabloda paketler programdaki paket renklerine göre renklendirilir. Frekans sıfır olanlar zamanlar kırmızıya boyanarak kullanıcının analizlerini kolaylaştırmak amaçlanmıştır.

ANA_PAKET_9	ANA_PAKET_9_3	ANA_PAKET_9_4	ANA_PAKET_9_5	ANA_PAKET_9_6	ANA_PAKET_10	ANA_PAKET_10_1	ANA_PAKET_10_2
4	0	3	3	7	2		
1	0	0	0	0	1		
2	0	0	0	0	1		
0	0	1	0	0	1		
2	0	3	0	0	1		
1	1	0	0	0	0		
2	2	1	1	0	2		
0	0	0	1	0	0		
2	0	0	0	0	0		
0	0	2	0	2	0		
0	0	0	0	0	0		
1	1	0	0	0	0		
2	0	0	0	1	2		
1	0	2	0	0	0		
3	0	1	1	1	0		
1	1	0	2	0	0		
0	1	1	0	0	0		
1	0	1	2	0	0		
0	1	0	0	0	0		
0	2	1	3	1	0		
0	0	0	0	0	0		
1	1	2	0	1	3		
0	1	0	0	0	1		
0	0	0	0	0	0		
0	1	0	0	0	1		
0	1	0	1	1	1		
0	0	2	0	0	1		
0	1	1	0	0	0		
0	0	1	0	0	0		
0	0	0	0	0	0		
0	1	0	0	0	0		
0	1	1	2	0	1		
0	0	0	1	0	0		
1	1	0	0	2	0		
0	3	0	0	0	0		
0	1	1	1	1	3		
2	0	2	0	0	1		
1	0	1	1	2	2		

Şekil 3.18 Frekans Tablosu sayfası

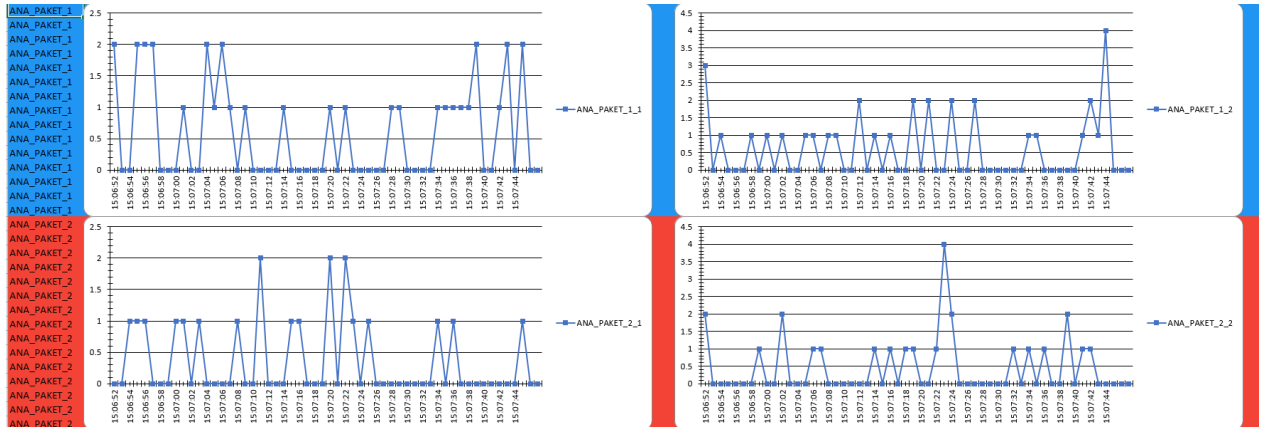
Bu form BTÜ-İMEP Koordinatörlüğü tarafından hazırlanmıştır.

İzinsiz kopyalanması, çoğaltılması kullanılması ve değiştirilmesi yasaktır.

BURSA TEKNİK ÜNİVERSİTESİ
PROJE TABANLI İŞLETMEDE MESLEKİ EĞİTİM
BTÜ-İMEP ÖĞRENCİ FİNAL RAPOR FORMU F4

3. Frekans Grafikleri

Bu sayfada programdaki tüm frekans grafikleri bulunur. Program çalıştığından itibaren kaydedilen tüm frekans değerleri bu grafikte gözlemlenir. Kullanıcı bu grafikler üzerinden istediği analizleri yapabilir.



Şekil 3.19 Frekans Grafikleri sayfası

4. Boyut Tablosu

Tüm paketlerin boyut değerleri ve zaman karşılıkları ayrı ayrı olarak bu sayfada gözlemlenebilir. Her paket programda olduğu gibi farklı renktedir ve renkler programda kullanılan renkler ile aynıdır. Kullanıcı buradan paket boyutları üzerinde istediği analizleri yapabilir.

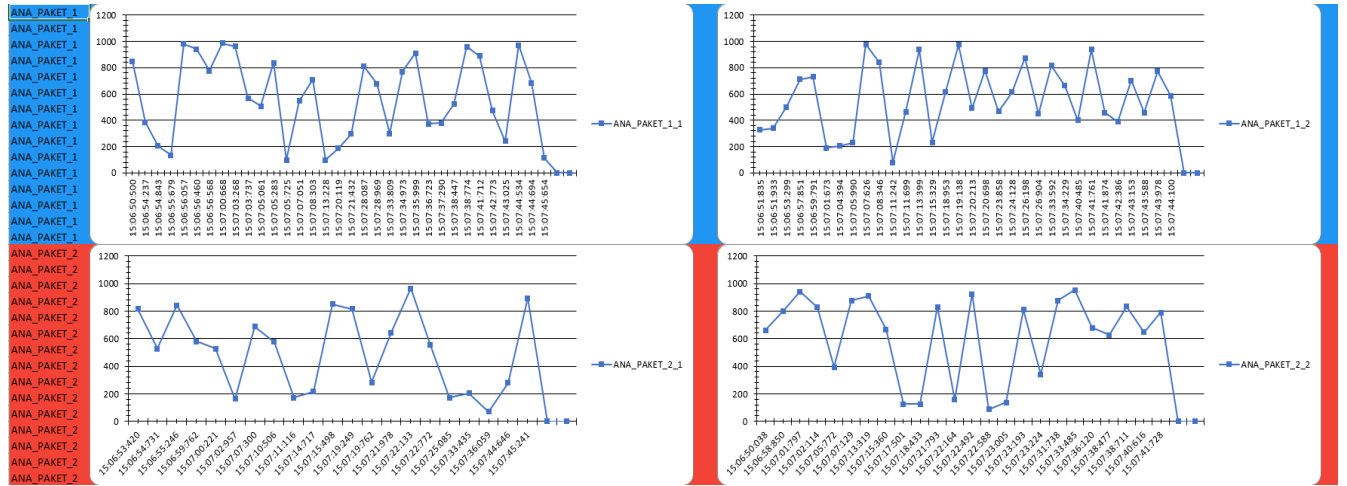
ANA_PAKET_1 ANA_PAKET_1_8		ANA_PAKET_1 ANA_PAKET_1_9		ANA_PAKET_2 ANA_PAKET_2_1		ANA_PAKET_2 ANA_PAKET_2_2	
15:06:49:770	809	15:06:49:781	897	15:06:53:420	817	15:06:50:038	661
15:06:51:550	605	15:06:49:822	408	15:06:54:731	526	15:06:58:850	798
15:06:55:497	979	15:06:50:127	625	15:06:55:246	842	15:07:01:797	942
15:06:55:561	780	15:06:52:208	89	15:06:59:762	579	15:07:02:114	828
15:06:56:654	393	15:06:52:520	890	15:07:00:221	527	15:07:05:772	393
15:06:59:550	522	15:06:53:392	559	15:07:02:957	162	15:07:07:129	877
15:07:06:662	401	15:06:55:715	169	15:07:07:300	688	15:07:13:319	910
15:07:07:376	210	15:06:56:383	504	15:07:10:506	576	15:07:15:360	668
15:07:14:528	183	15:07:01:425	578	15:07:11:116	173	15:07:17:501	126
15:07:18:062	487	15:07:06:619	559	15:07:14:717	217	15:07:18:433	126
15:07:19:512	536	15:07:06:775	710	15:07:15:498	850	15:07:21:791	830
15:07:20:566	480	15:07:07:516	297	15:07:18:249	816	15:07:22:164	158
15:07:22:805	234	15:07:11:385	775	15:07:19:762	284	15:07:22:492	923
15:07:25:721	686	15:07:13:834	318	15:07:21:978	643	15:07:22:588	88
15:07:29:605	841	15:07:16:506	380	15:07:22:133	964	15:07:23:005	136
15:07:30:923	469	15:07:16:866	428	15:07:22:772	553	15:07:23:193	811
15:07:31:245	808	15:07:18:286	633	15:07:25:085	172	15:07:23:224	341
15:07:33:001	440	15:07:18:458	422	15:07:33:435	204	15:07:31:738	878
15:07:36:089	686	15:07:19:685	378	15:07:36:059	71	15:07:33:485	953
		15:07:22:662	492	15:07:44:646	281	15:07:36:120	679
		15:07:22:679	147	15:07:45:241	895	15:07:38:477	627
		15:07:24:545	252			15:07:38:711	832
		15:07:24:979	601			15:07:40:616	648
		15:07:25:271	80			15:07:41:728	789
		15:07:29:454	660				
		15:07:30:597	649				
		15:07:30:797	301				
		15:07:36:812	934				
		15:07:38:509	486				
		15:07:38:555	414				
		15:07:40:815	836				

Şekil 3.20 Boyut Tablosu sayfası

BURSA TEKNİK ÜNİVERSİTESİ
PROJE TABANLI İŞLETMEDE MESLEKİ EĞİTİM
BTÜ-İMEP ÖĞRENCİ FİNAL RAPOR FORMU F4

5. Boyut Grafikleri

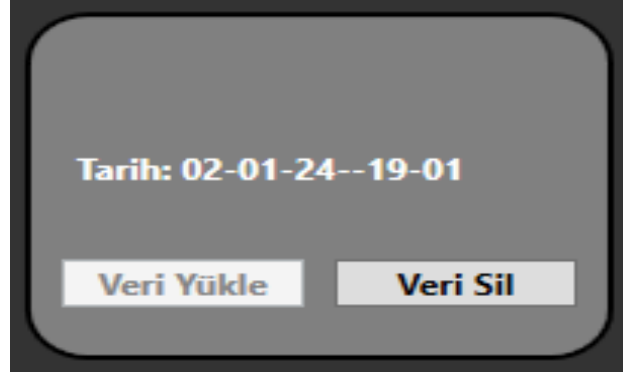
Kaydedilen tüm boyut değerleri bu grafikler üzerinde gözlemlenebilir. Bir pakete ait tüm grafikler bir satırda bulunur ve bu satırların rengi her paket için farklıdır. Programda kullanılan paket renkleri aynı olarak bu sayfada da kullanılır. Kullanıcı bu grafiklerin üzerinden tüm değerlere ilişkin çeşitli analizler yapabilir ve excel uygulamasının hazır sunduğu analizlerden de faydalanabilir.



Şekil 3.21 Boyut Grafikleri

3.2.8. Geçmiş Verileri Simüle Etme

Kullanıcı kaydetmiş olduğu verileri, istediği zaman simüle edebilir. Ana sayfasının sağ alt kısmında bulunan “Veri Yükle” butonuna tıklayarak [Verilerin Kaydedilmesi](#) bölümünde anlatılan zaman klasörünü seçmesi gerekir. Seçilen zaman klasörünün içindeki tüm veriler programa yüklenir. Bu verilere göre grafikler ve tablolar doldurulur. Programda “Tarih:” kısmına seçilen verilerin kaydetme zamanı yazılır. Sayfanın en altına oynatma çubuğu ve oynatma butonu gelir. Kullanıcı oynatma çubuğundan süreci ileriye veya geriye sarabilir veya oynatmayı durdurabilir. Soket panelde “Bağlan” ve “Bağlantıyı Kes” butonlarının aktifliği kesilir. Bu butonların kullanılabilmesi için kullanıcının “Veri Sil” butonuna tıklayarak yüklediği verileri programdan kaldırması gerekir.



Şekil 3.22 Geçmiş verileri yükleme ve silme paneli

“Veri Sil” butonuna tıklayarak kullanıcı yüklediği verileri silebilir. Veriler silindiğinde programda hangi konfig dosyası yüklü ise o dosyadaki paket bilgileri tabloya gelir. Veriler silindikten sonra gerçek zamanlı işlemlere devam edilebilir. Veri silme ve yükleme işlemlerinde soketten bağlantı kesilir ve soket otomatik olarak kapatılır.

3.3. Sonuçlar

Geniş fonksiyonlara sahip, birçok özelliği olan bir masaüstü uygulaması geliştirilmiştir. Geliştirme aşamasında sürekli yeni özellikler eklenmiştir. Bu sebeple temelde kurulan yapı sağlam inşa edilmiştir. Üzerine eklemeler yapıldıkça programın hiçbir fonksiyonu çalışabilirliğini yitirmemiş olup tüm bir sistem halinde tüm fonksiyonlar başarılı bir şekilde işlev göstermektedir. Kullanıcının yüklediği enum dosyasına göre özel konfigürasyon dosyaları oluşturarak tamamen dinamik bir yapı kurulmuştur. Program içerisinde birçok thread ve task bulunduğu için fonksiyonların bellek erişimleri esnasında çeşitli problemler yaşanmış olup birbiri ile iletişim halinde olan threadler oluşturarak bu problemler aşılmıştır. Proje başlangıcında amaçlanan tüm hedefler gerçekleştirilmiştir fakat program üzerinde bazı optimizasyon işlemleri yapılabilir ve yeni özellikler eklenebilir. Geliştirilmeye açık olan bu proje yazılım alanında benim için önemli bir tecrübe olmuştur.

4. ÇALIŞMA DÖNEMİNİN DEĞERLENDİRMESİ

BTÜ-İMEP programının henüz öğrenciyken sektörü tanımak adına büyük fayda sağladığını söyleyebilirim. Alanında uzman mühendisler ile çalışmak, onların çalışma ortamını deneyimlemek ve okulda öğrenilen teorik bilgiyi bu program ile pratiğe dökme şansı bulmak gelecek kariyerim için büyük fayda sağlayacaktır. Bu program süreci boyunca baştan sona bir projeyi geliştirmek benim için önemli bir tecrübe olmuştur.

Gelecekte Baykar firmasında çalışmak isterim. Gelecek planım savunma sektöründe çalışmak olduğu için ve Baykar firması bu alanda en önemli isimlerden biri olduğu için gelecek kariyerime burada devam etmek isterim. Ayrıca üretim departmanında çalışmaktan ziyade araştırma ve geliştirme departmanında çalışmak istiyorum. Baykar'ın güçlü bir Ar-Ge ekibi olduğu için de burada mühendis olarak çalışmak isterim.

5. KAYNAKLAR

- [1] **Baykar Teknoloji.** (t.y.). Erişim: 31 Aralık 2023, https://tr.wikipedia.org/wiki/Baykar_Teknoloji
- [2] **Baykar Bayraktar Kızılelma.** (t.y.). Erişim: 31 Aralık 2023, https://tr.wikipedia.org/wiki/Baykar_Bayraktar_Kızılelma
- [3] **C# ile GetHashCode ve GetHashCode().** (t.y.). Erişim: 1 Ocak 2024, <https://medium.com/@sergenkahraman/c-ile-hashcode-ve-gethashcode-46e943876970#>