



BURSA ULUDAĞ ÜNİVERSİTESİ

OCAK 2024

TASARIM PROJESİ

BİLGİSAYAR MÜHENDİSLİĞİ

ARAŞTIRMA KONUSU

YAZILIM TANIMLI AĞ TABANLI GERÇEK ZAMANLI
CİHAZ YÖNETİM UYGULAMASI

ANİL DURSUN İPEK - 031890131
BATUHAN ARSLANDAŞ - 032190097

PROJE DANIŞMANI: DOÇ. DR. MURTAZA CİCİOĞLU



SUNUM

AKİŞI

01. Giriş

13-14. Video Kalitesinin Ölçülmesi

02. Hedefler

15-21. Veri Analizi

03. Yazılım Tanımlı Ağ Nedir?

22-23. Sistem Akış Şeması

04-05. National Science Foundation Network

24-27. Önerilen Algoritmanın Başarımı

06-07. Kullanılan Araçlar

28-29. Sonuç

08. Metotlar

30. Tartışma

09-12. Veri Üretim Senaryoları

31. Teşekkür

Giriş

Yazılım tanımlı ağlarda kullanıcı deneyim kalitesini artttırmak için yapay zeka tabanlı yönlendirme algoritması önerilmektedir. Önerilen algoritma sanal simülasyon ortamında üretilen veriler ile eğitilen bir yapay zeka modeli kullanmaktadır. Yapay zeka entegrasyonu ile birlikte ağın daha optimize olması hedeflenmektedir.



Hedefler

- 1** Yazılım tanımlı ağlara yapay zeka entegrasyonu ve ağıın dinamik olarak yönetilmesi
- 2** Sanal simülasyon ortamında veri seti üretilmesi
- 3** Ağ üzerinde kaynakların daha verimli kullanılması
- 4** Kullanıcı deneyiminin iyileştirilmesi

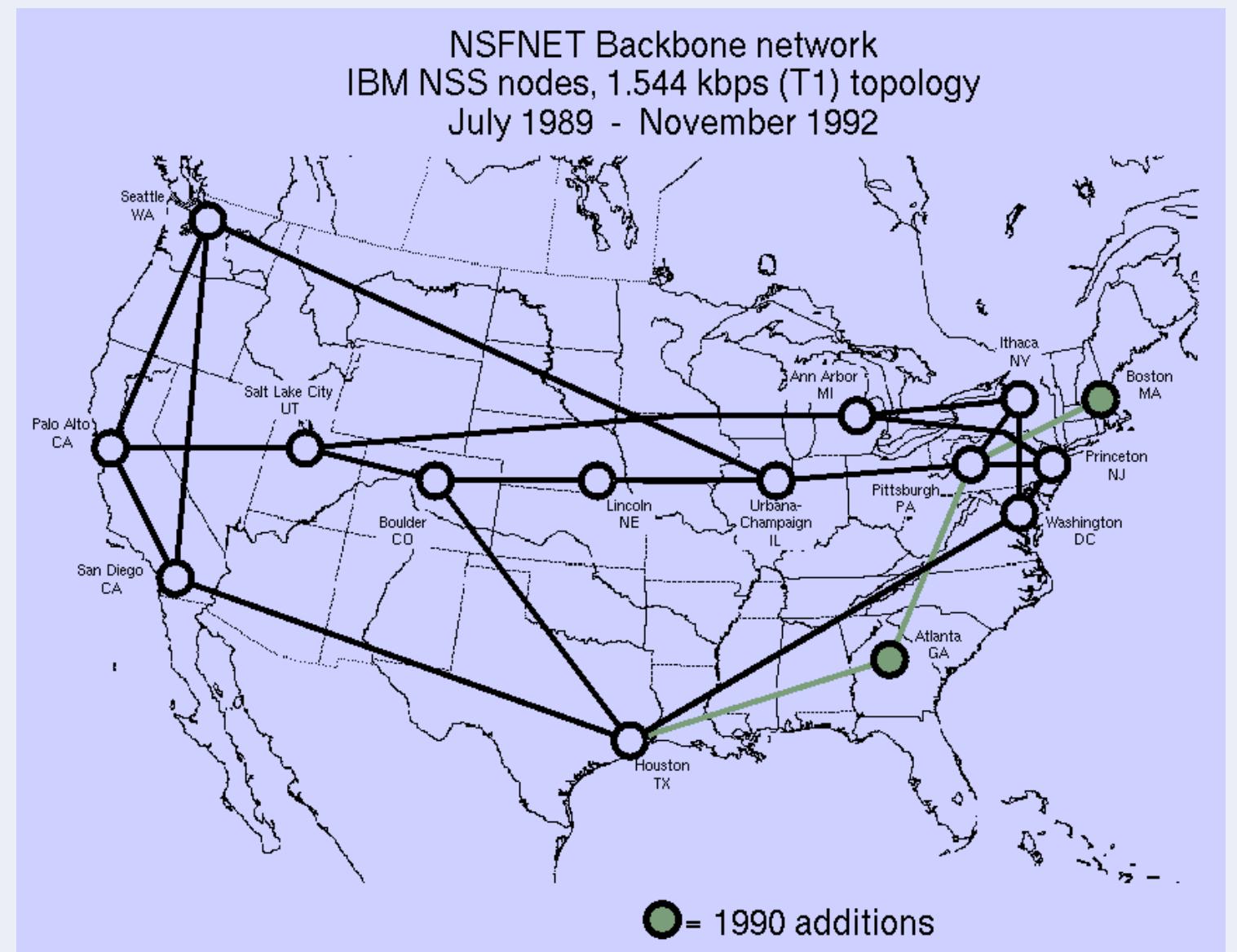


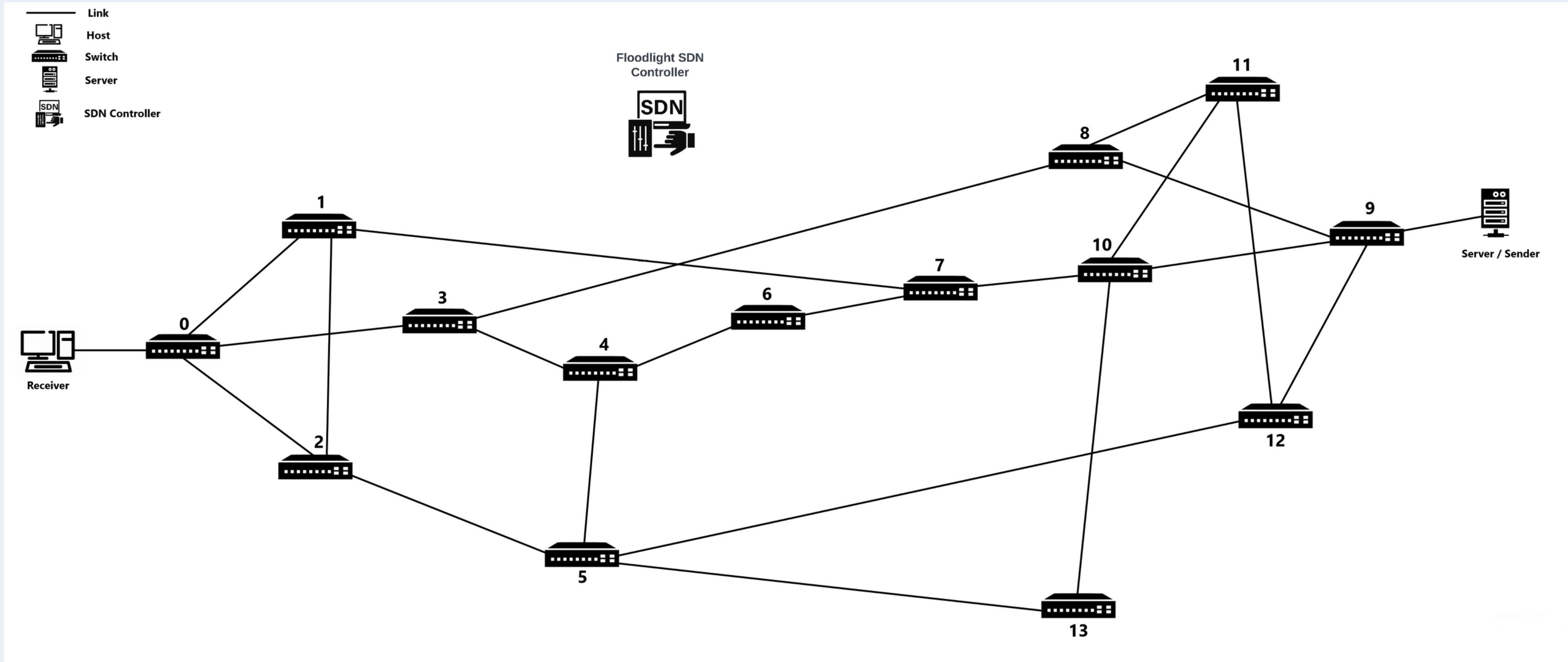
Yazılım Tanımlı Ağ Nedir?

Yazılım tanımlı ağlar geleneksel ağların mevcut sorunlarını çözmek için ortaya çıkmıştır. Merkeziyetçi, esnek, güncellenebilir bir ağ altyapısı yapısı sunmaktadır. Ağın merkezinde denetleyici adı verilen bir bilgisayar bulunmaktadır. Ağdaki tüm hesaplamaları ve kontrol işlemlerini denetleyici üstlenmektedir.

National Science Foundation Network

NSFNET, 1985-1995 yılları arasında Amerika'da bilimsel ve eğitimsel bilgi alışverişi için kullanılan bir ağdır. TCP/IP protokolü üzerinde çalışan NSFNET, o dönemdeki internetin temel protokol setini oluşturarak 1995'te geniş ve modern internet altyapısına yerini bırakmaktadır. Çalışmamızda literatürde önemli bir yeri olan NSFNET topolojisi kullanılmaktadır.





Şekil - Mininet simülasyon ortamında oluşturulan NSFNET ağının şeması

Kullanılan Araçlar - I

Mininet

Mininet, ağ protokollerini ve uygulamalarını test etmek için kullanılan açık kaynaklı bir emülsyon ve test platformudur. Mininet, gerçek ağ donanımını fiziksel olarak kullanmadan, sanal ağ topolojileri oluşturmamıza ve ağ protokollerini simüle etmemize olanak tanır.

Floodlight

Floodlight, açık kaynaklı bir SDN kontrol yazılımıdır. SDN, ağdaki yönetim ve kontrolün ayrılmasına olanak tanıyan bir yaklaşımı temsil etmektedir. Floodlight, bu ağ kontrolünü sağlamak ve yönetmek için kullanılan bir kontrol düzlemidir.

NFStream

NFStream ağ verileri ile çalışmayı kolay ve sezgisel hale getirmek için tasarlanmış bir Python çerçevesidir. Pratik bir şekilde ağ üzerinde veri analizi yapma imkanı sunmaktadır.

Kullanılan Araçlar - II

FFMpeg

FFmpeg, açık kaynaklı bir çoklu ortam çerçevesidir. Ses, video, altyazı ve diğer multimedya dosyalarını kodlama, kod çözme, dönüştürme ve yayınlama işlemleri için bir dizi araç ve kütüphane sağlamaktadır.

Iperf3

Iperf3, ağ bant genişliği ölçümü ve ağ performansı testleri yapmak için kullanılan bir araçtır. Sunucu-istemci modeli kullanarak ağ üzerinde veri iletimini simüle etmektedir ve bağlantı hızını, gecikmeyi ve paket kaybını ölçmek için kullanılmaktadır.

Ping

Ping, bağlantı, erişilebilirlik ve ad çözümleme sorunlarını gidermek için kullanılan birincil TCP/IP komutudur. Bu komut bir anasistemden ya da ağ geçidinden ICMP ECHO_RESPONSE almak için bir ICMP ECHO_REQUEST (Internet Denetim İletisi İletişim Kuralı) gönderir.



Metotlar

- **Simülasyon Ortamında Veri Üretimi**

Kurgulanan ağ ortamında veri üretimi gerçekleştirilmektedir. Veri üretimi için farklı araçlar kullanılmakta ve bu araçlar ile elde edilen veriler kayıt edilmektedir. Kayıt edilen veriler bir sonraki aşamada yapay zeka modelinin eğitimi için kullanılmaktadır.

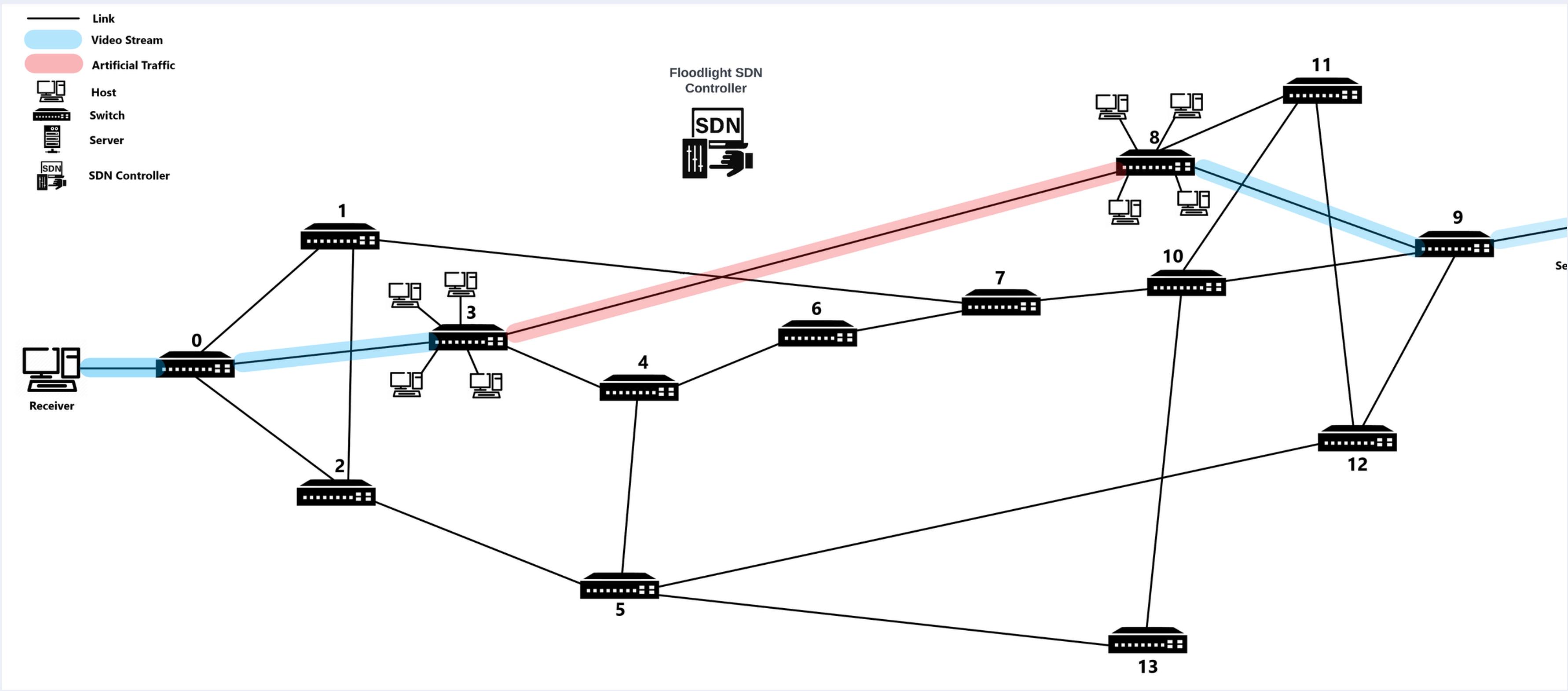
- **Yapay Zeka Tabanlı Yönlendirme**

Veri üretim aşamasından sonra elde edilen veriler ile ikili sınıflandırma gerçekleştirilmektedir. Girdi olarak istenen parametreler yapay zeka modeline verildiğinde trafik düzeyini çıktı olarak vermektedir. Model kullanarak bir yönlendirme algoritması tasarlannmaktadır.

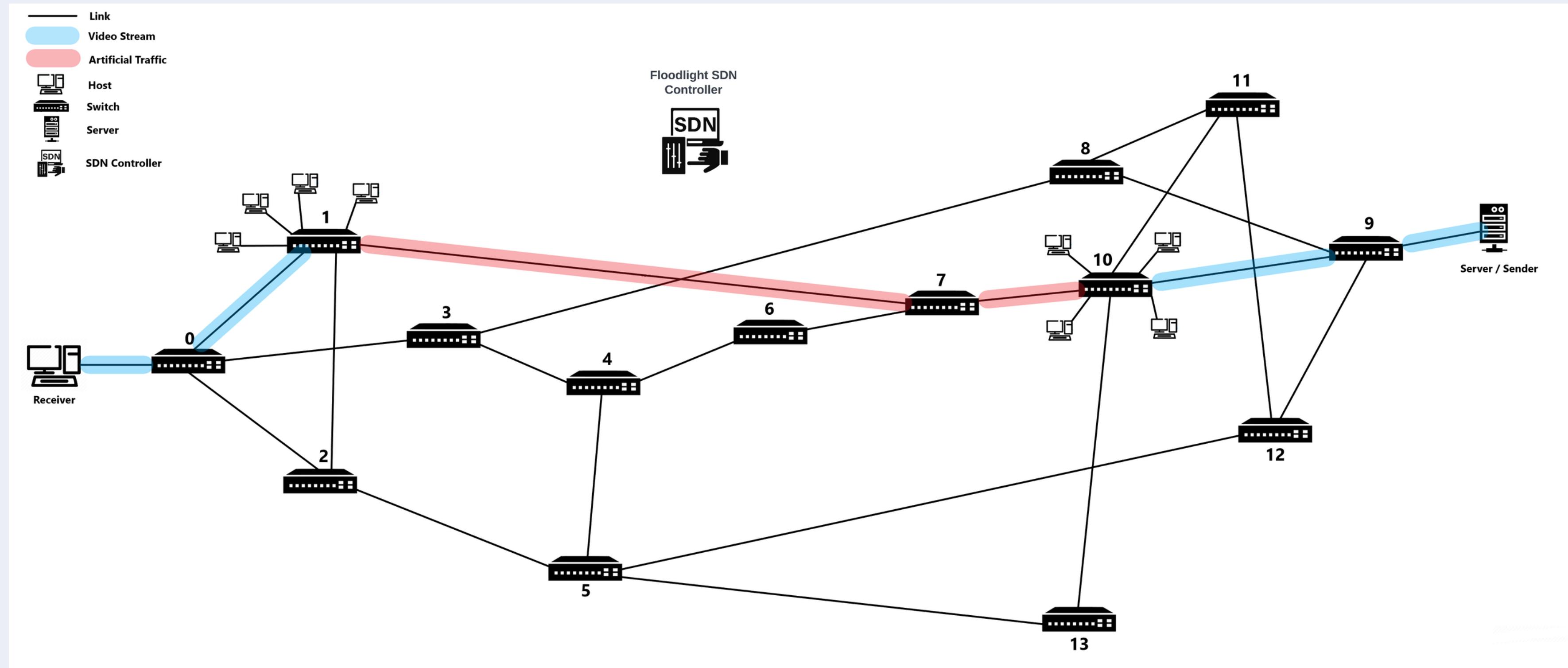
Veri Üretim Senaryoları

Mininet ortamında kurulan NSFNET ağı üzerinde veri elde edilmesi için belirli senaryolar oluşturulmuştur. Senaryolar temel olarak ağ üzerindeki 3 farklı yol üzerinde veri göndermeye dayanmaktadır. Bu yollar üzerinde ise 2 farklı trafik düzeyi oluşturulmuştur.

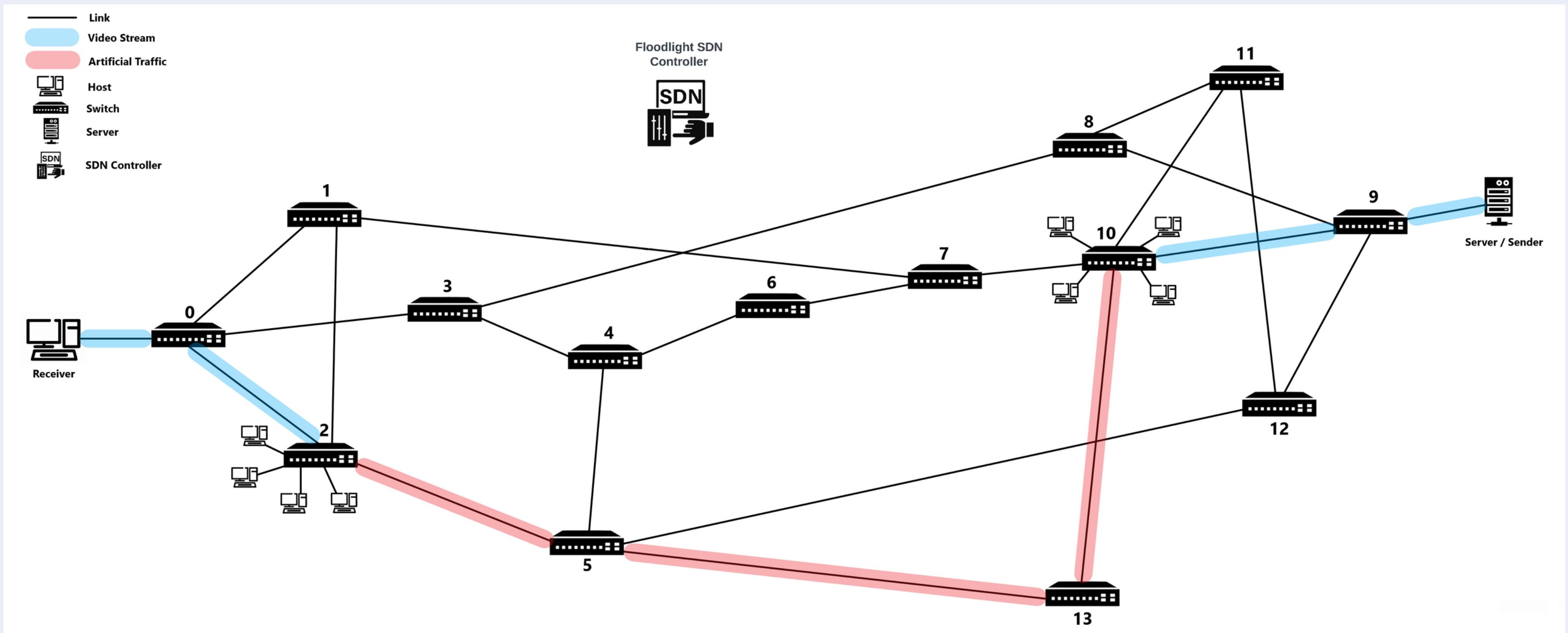
Ağ üzerinde toplam 14 adet anahtar, 1 adet istemci ve 1 adet sunucu bulunmaktadır. Ağ üzerindeki tüm anahtarları kontrol eden 1 adet de Floodlight SDN Denetleyici bulunmaktadır. Bu ağ cihazları kurgulanan senaryolarda temeli oluşturmaktadır. Senaryolar kurgulanırken ağa farklı istemciler eklenmekte ve aralarında trafik oluşturulmaktadır.



Şekil -Video Aktarımı Birinci Senaryo (3 hop)



Şekil - Video Aktarımı İkinci Senaryo (4 hop)



Şekil - Video Aktarımı Üçüncü Senaryo (5 hop)

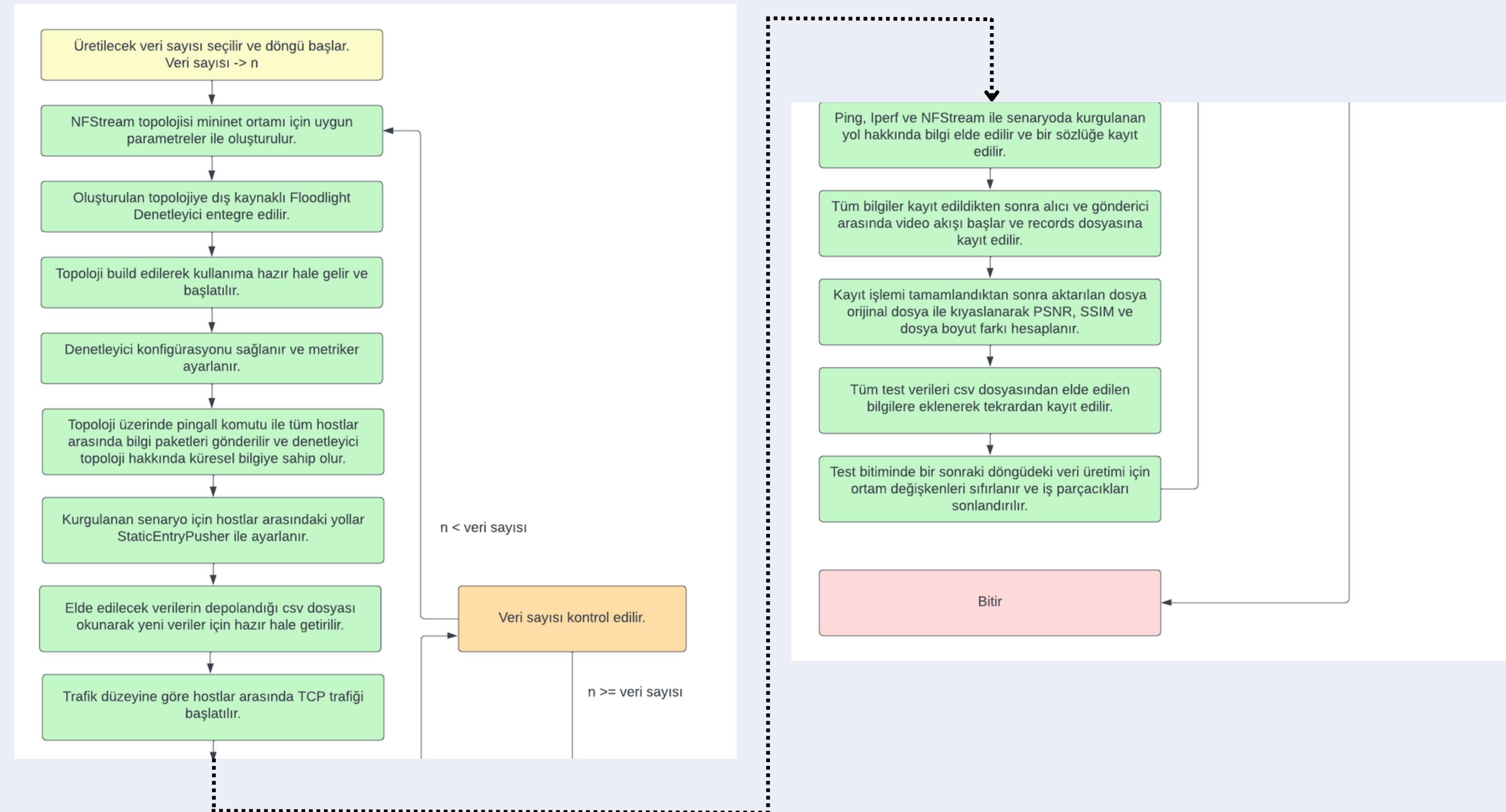
Video Kalitesinin Ölçülmesi

PSNR

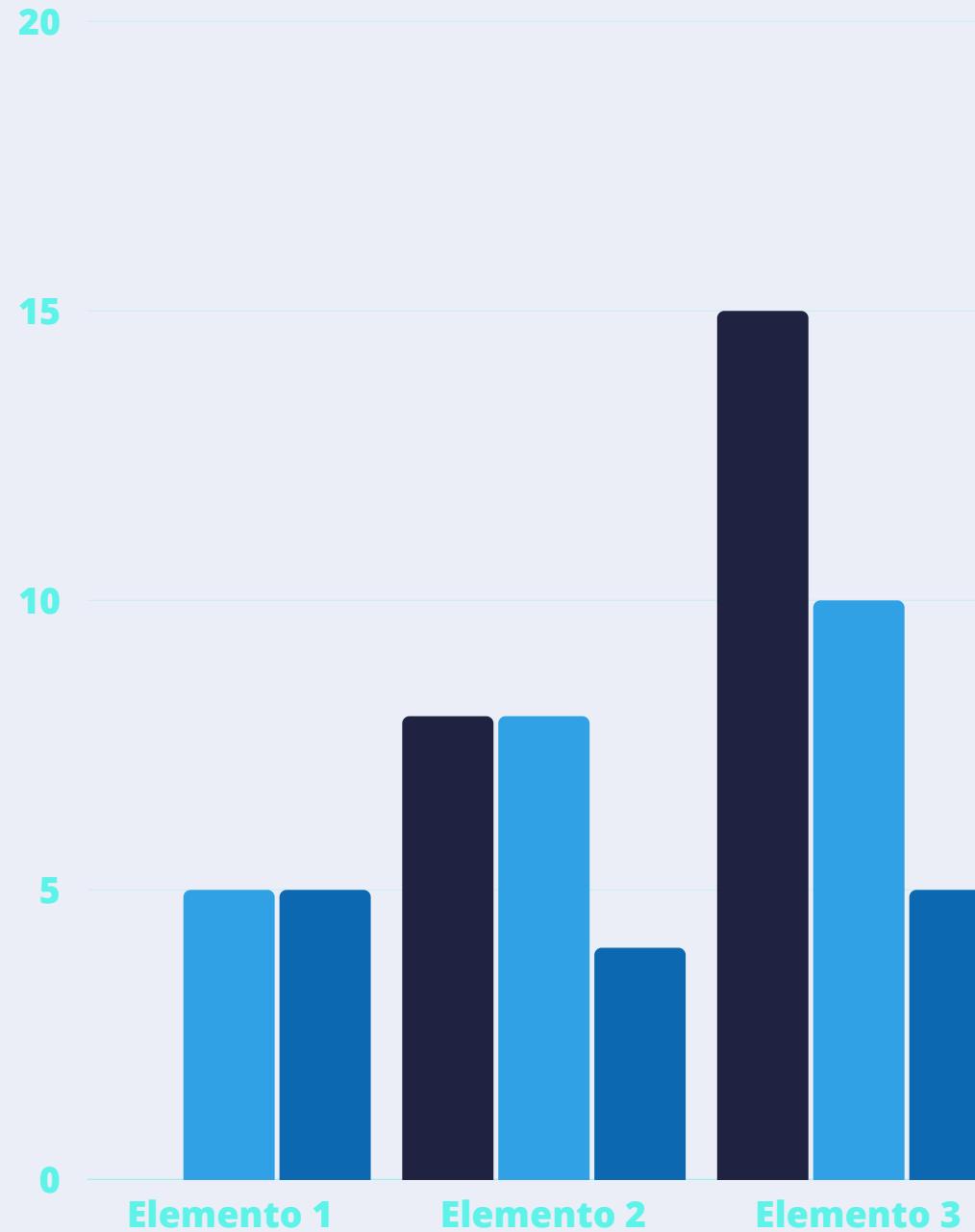
PSNR (Peak Signal-to-Noise Ratio) (Pik Sinyal Gürültü Oranı), video kalitesini değerlendirmek için yaygın olarak kullanılan bir ölçütür. Bu metrik, orijinal bir videonun üzerinde yapılan işlemler sonrasında elde edilen işlenmiş halinin, format değişiklikleri, sıkıştırma veya ağ üzerinde iletişim gibi süreçlere karşı ne kadar başarılı olduğunu ölçer.

SSIM

SSIM (Structural Similarity Index), iki görüntü arasındaki benzerliği karşılaştırın bir ölçümüdür. SSIM, insan görsel sistemine daha yakın sonuçlar elde etmeyi amaçlar ve görüntüler arasındaki yapısal, parlaklık ve kontrast benzerliği gibi ölçümelerini içerir.

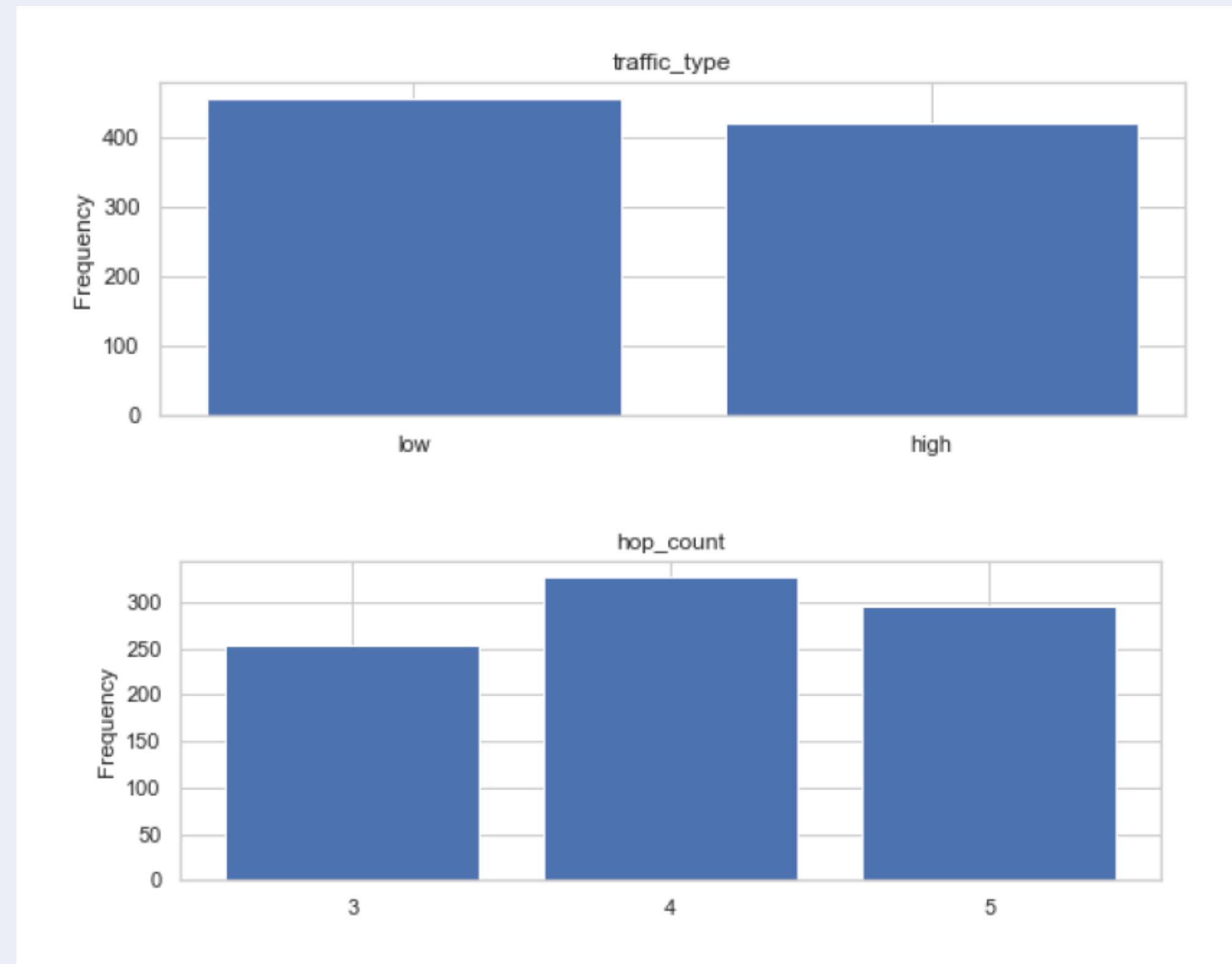


Şekil - Otomatik veri üreten programın akış şeması

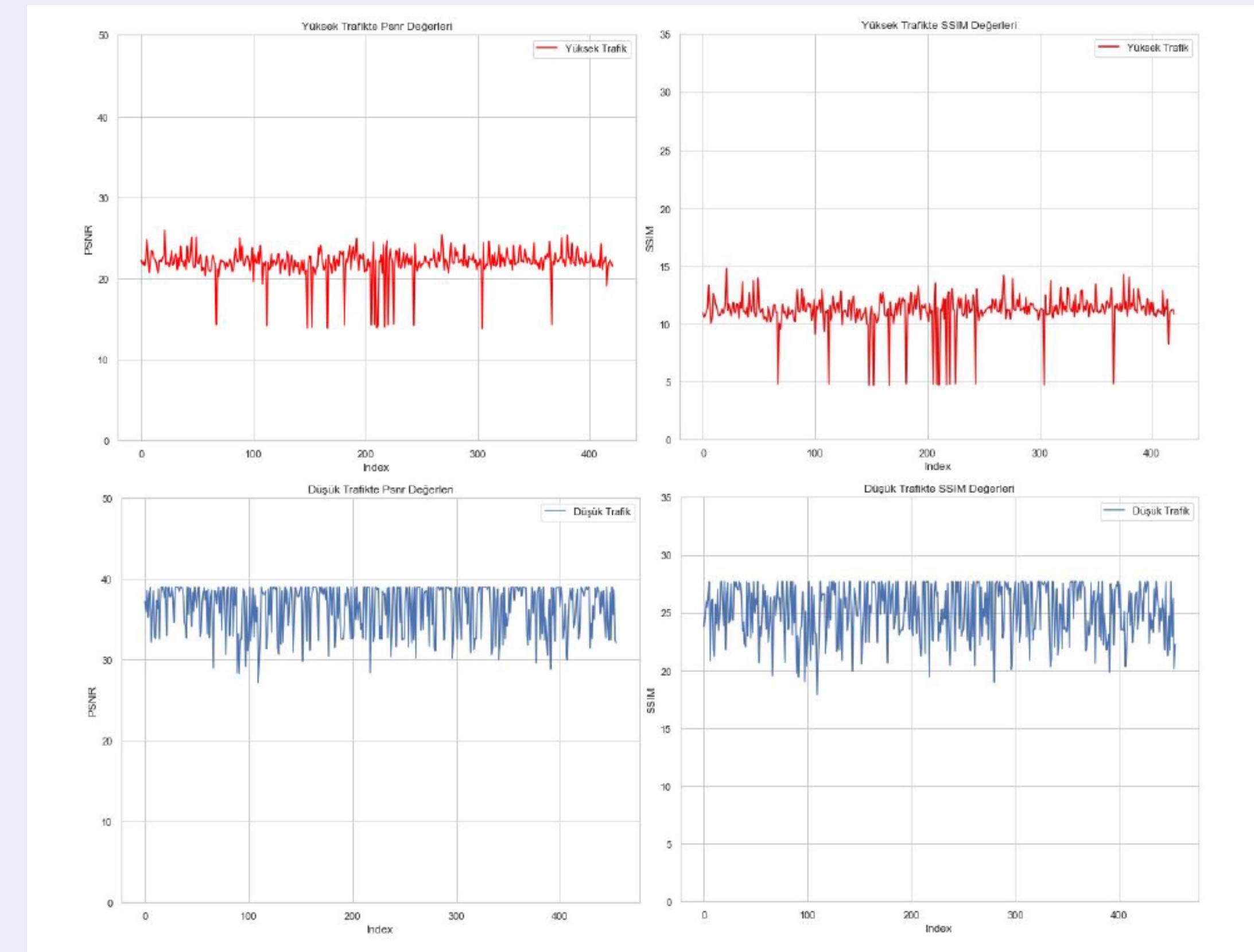


Veri Analizi

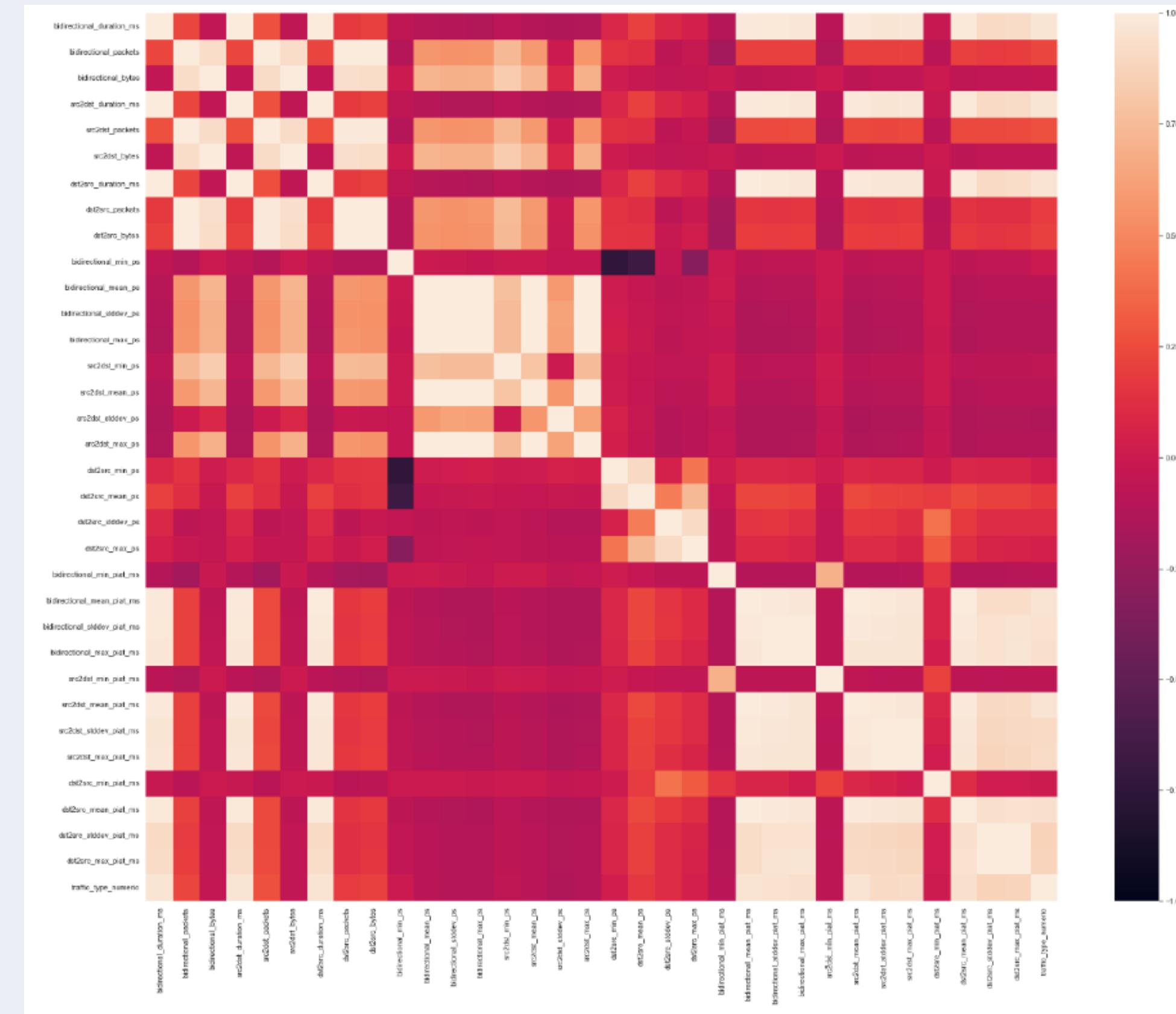
Kurgulanan 3 farklı senaryo ile toplamda 876 adet veri üretilmiştir. Üretilen her bir veri 51 adet değişkene sahip olmaktadır. Bunlardan 50 tanesi aktarım sırasında ve öncesindeki durumları temsil ederken 1 adet değişken ise ortamdaki trafik türünü ifade etmektedir.



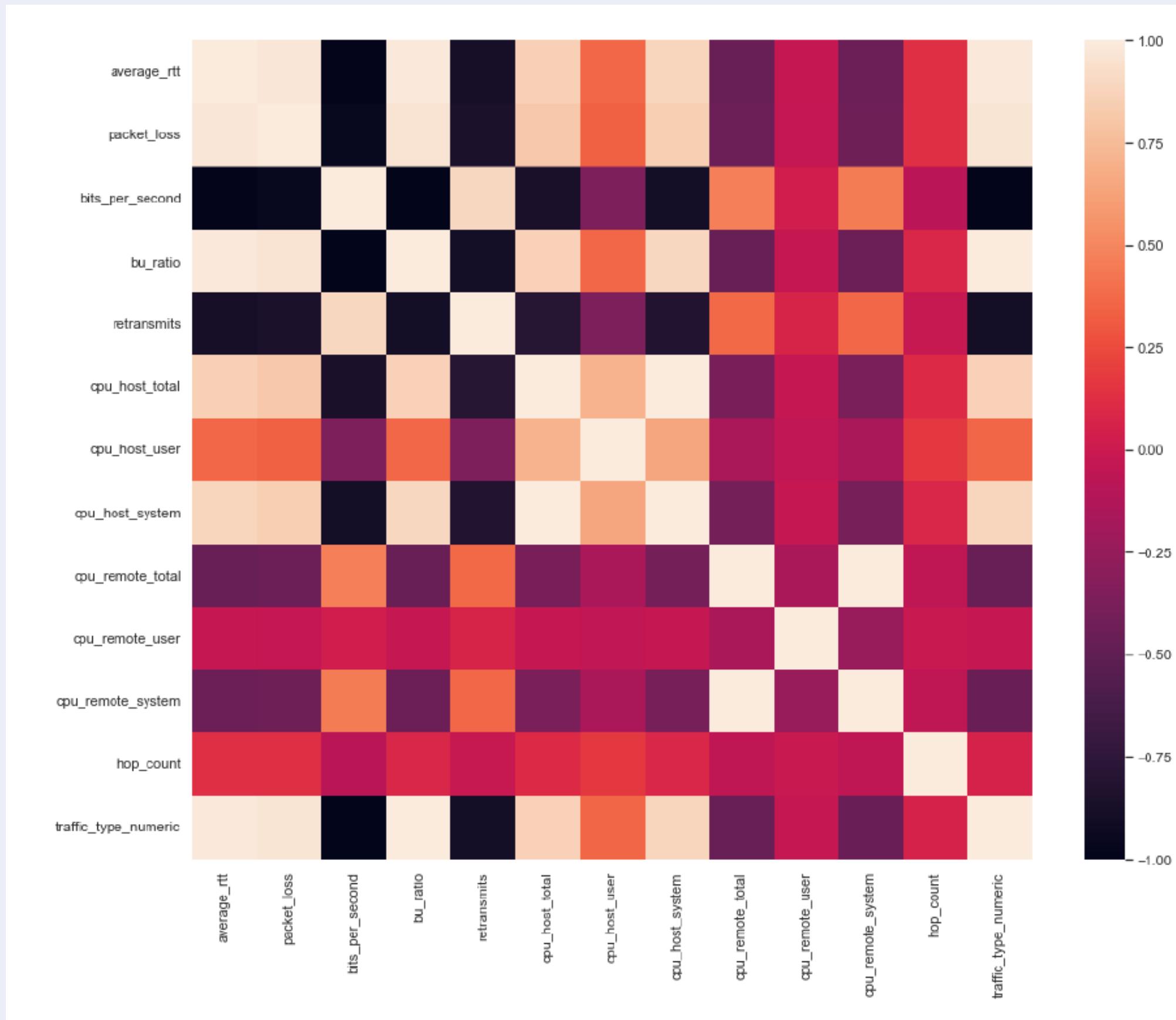
Şekil - Verilerin trafik düzeyine ve hop sayısına göre dağılımı



Şekil - Trafik düzeyine göre PSNR ve SSIM değerleri

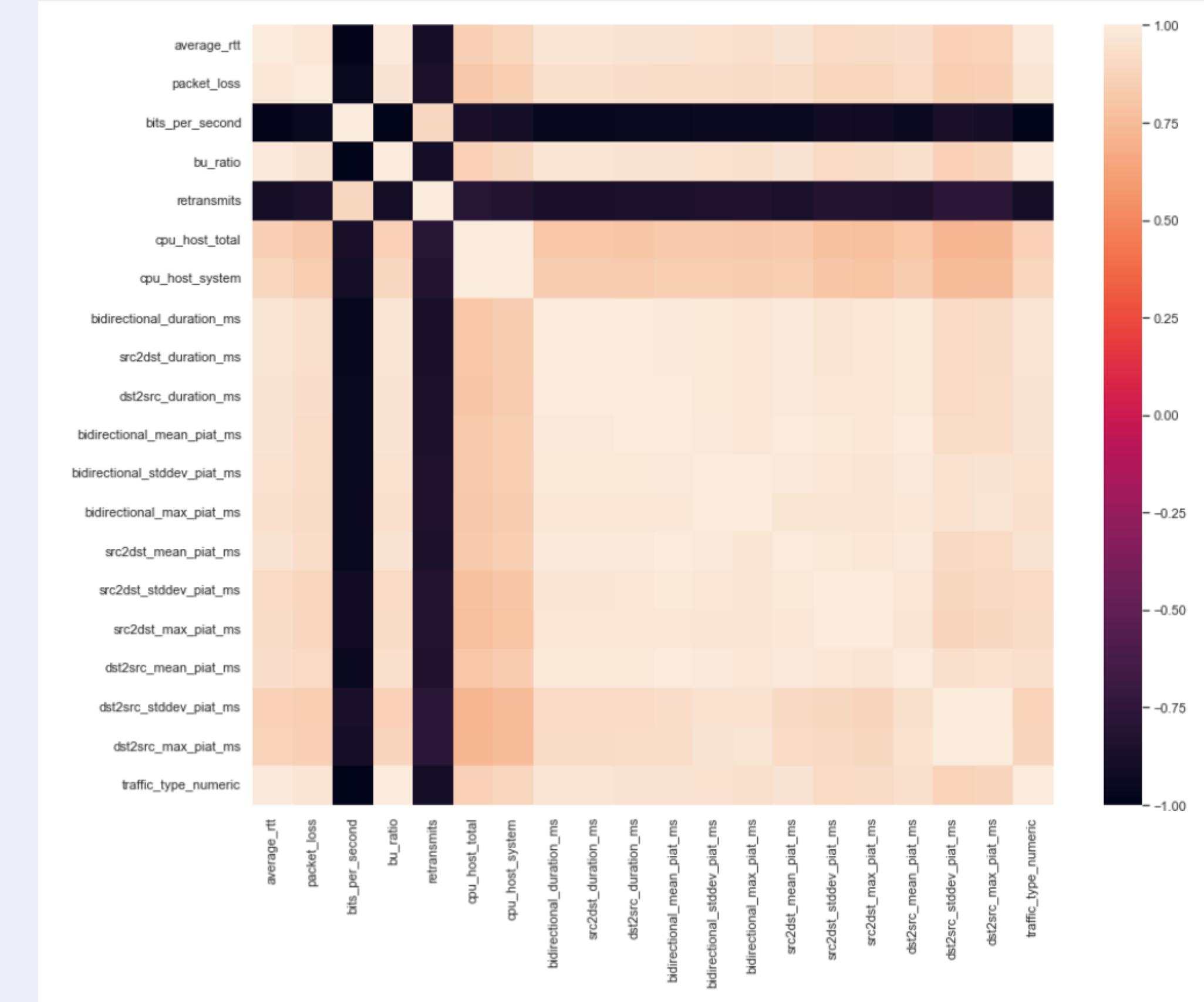


Şekil - NFStream verileri korelasyonunu gösteren ısı haritası



Şekil - Iperf3 ve ping değişkenleri korelasyonunu gösteren ısı haritası

Sonuç olarak başlangıçta 50 adet olan değişken sayısı veri analizi sonucu özellik çıkarımı yapılarak 19 adete düşürülmüştür. Sonuç veri setinin ısı haritası ve değişkenleri şekil de paylaşılmaktadır. Şekil incelendiğinde tüm değişkenler arasında yüksek miktarda korelasyon olduğu gözlemlenmektedir. Bu değişkenler trafik düzeyine doğrudan etki etmektedirler.



Şekil - Özellik çıkarımı sonrası veri seti ısı haritası

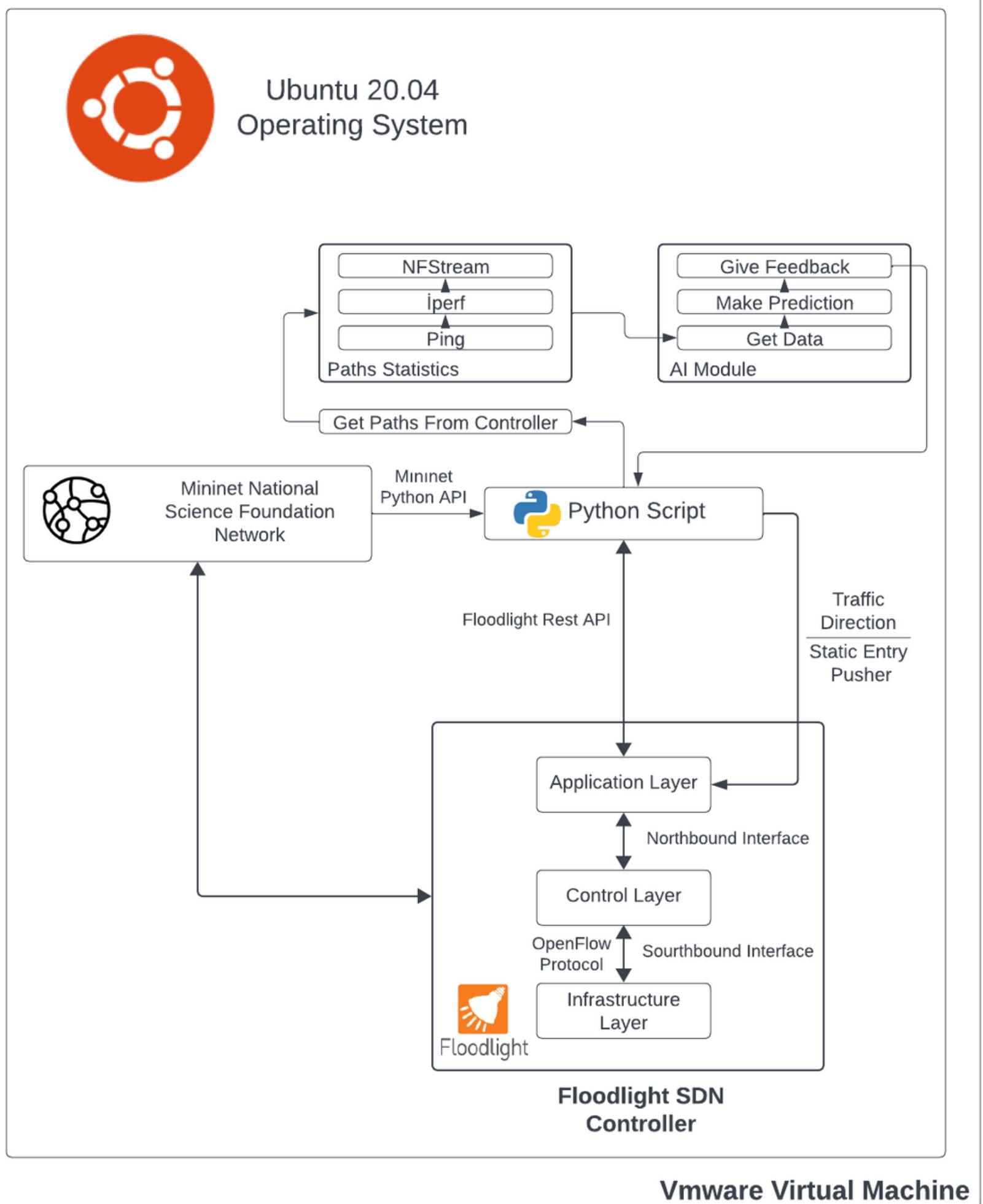
Analiz işlemi sonrasında sonuç veri seti farklı ikili sınıflandırıcı modeller ile eğitilmiştir. Eğitim esnasında veriler %85 eğitim ve %15 test için ayrılmıştır. Eğitim sonrasında modellerin başarıları veri setinin basit olması ve değişkenler arasında net bir ayırımın olması nedeniyle çok yüksek çıkmaktadır. 4 farklı model ile eğitim yapılmıştır. Bu modeller KNN, XGBoost, SVM ve Random Forest'dır.

Modellerin test verisi üzerinde precision, recall ve accuracy değerleri incelendiğinde değerlerin 1 olduğu gözlemlenmektedir. Yapay bir veri seti ile sanal bir ortamda veri üretildiği için veri seti içerisinde gürültü taşımamaktadır. Bu nedenle hatasız bir şekilde sınıflandırma yapabilmektedir. Veri toplama aşamasında verilere gürültü eklenerek yapay veri setinin gerçekçi bir grafik çizmesi sağlanabilir. Modeller eşit seviyede başarı göstermektedir.

MODEL	LOGARİTMİK KAYIP
KNN	9.99 x e-16
RANDOM FOREST	0.0015
SVM	0.0018
XGBOOST	0.0023

Şekil - İkili sınıflandırma modellerinin logaritmik kayıp değerleri

SİSTEM AKIŞ ŞEMASI



Algoritma: Yapay zeka tabanlı yönlendirme algoritması

Girdi: İstemci ve sunucu ipv4 adresleri

Cıktı: Trafik düzeyi en düşük olan yolun akış tablolarına eklenmesi

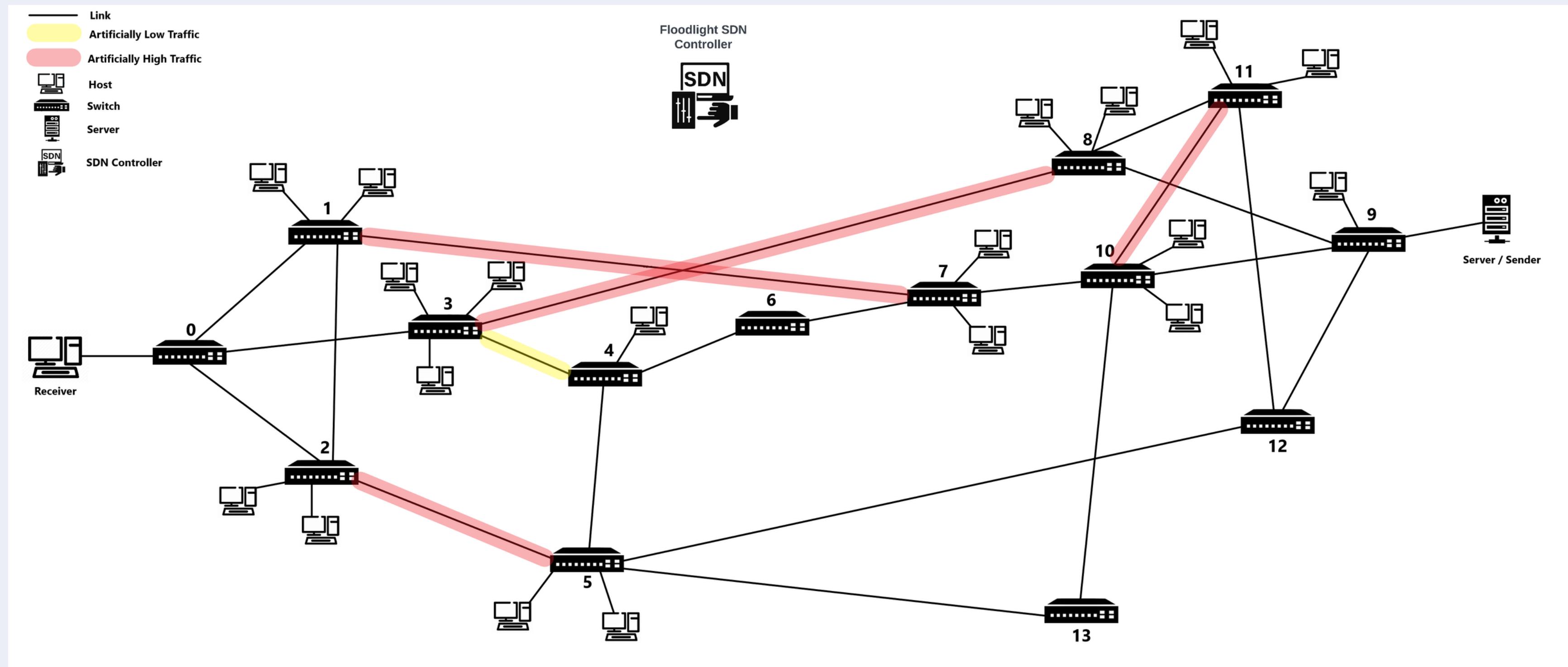
```
1: function YönlendirmeAlgoritması(istemci_ipv4, sunucu_ipv4):
2:     model ← YapayZekaModeliYükle()
3:     paths ← EnKısa5YoluGetir(istemci_ipv4, sunucu_ipv4)
4:     pathsTraffic ← Boş dizi tanımla
5:     for all paths do:
6:         AkışTablosunaEkle(path)
7:         path_statistics ← PathİstatistikleriniTopla()
8:         prediction ← model.prediction(path_statistics)
9:         pathsTraffic.append(prediction)
10:        deleteFlows()
11:    end for
12:    path ← TrafiğiEnAzOlanYoluBul(pathsTraffic)
13:    AkışTablosunaEkle(path)
```

Şekil - Yapay zeka tabanlı yönlendirme algoritması sözde kodu

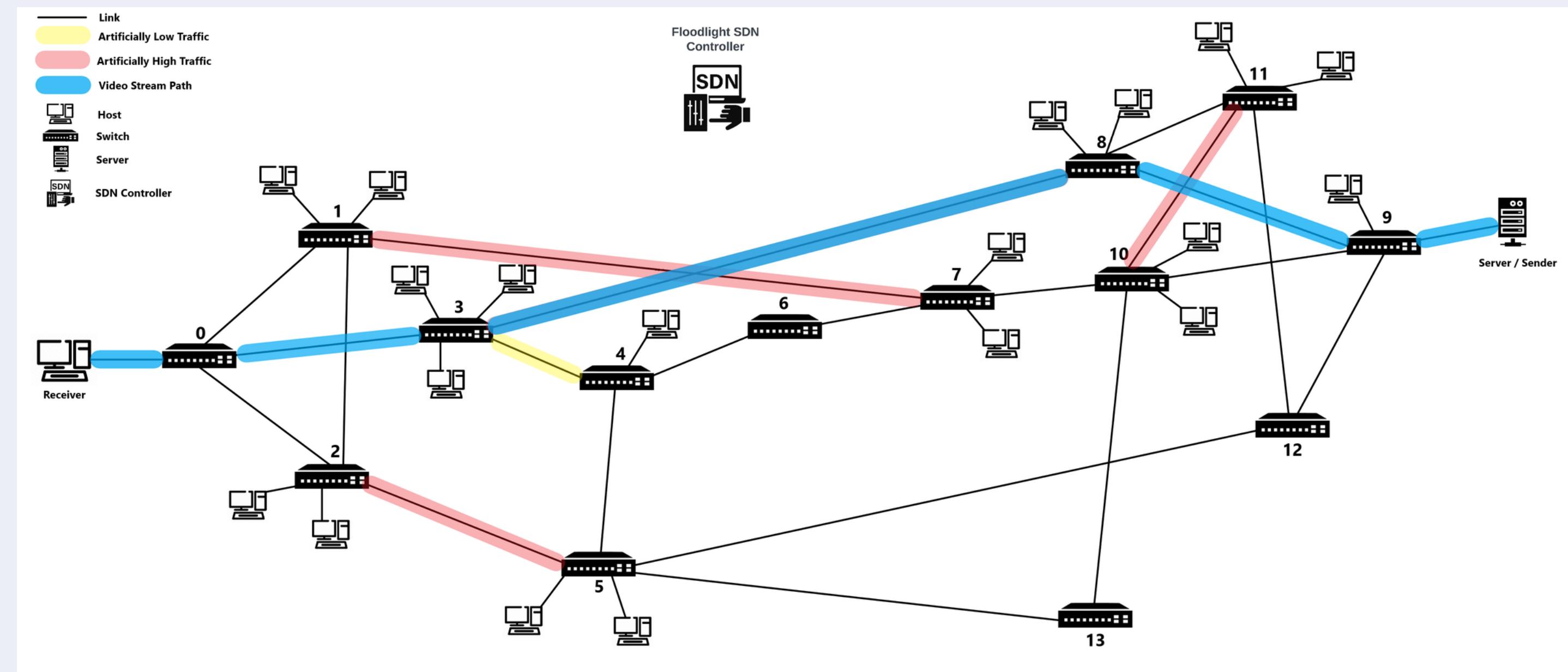
Önerilen Algoritmanın Başarımı

Bizim tarafımızdan önerilen yapay zeka tabanlı yönlendirme algoritması NSFNET topolojisi üzerinde kurgulanan senaryoda test edilmektedir. Senaryoda trafik oluşturan 19 adet istemci, 1 adet ana istemci (videoyu alan) ve 1 adet sunucu (videoyu gönderen) bulunmaktadır. Bunlara ek olarak ağır beyni olarak nitelendirdiğimiz Floodlight denetleyici metrik olarak hopcount değerine ayarlanmıştır. Şekilde hostlar arasındaki trafik akışları ve dereceleri gösterilmekte ve senaryonun genel bir görselleştirilmesi sunulmaktadır.

- ***Hopcount metrikli Floodlight denetleyici sonuçları***
- ***Önerilen algoritma sonuçları***

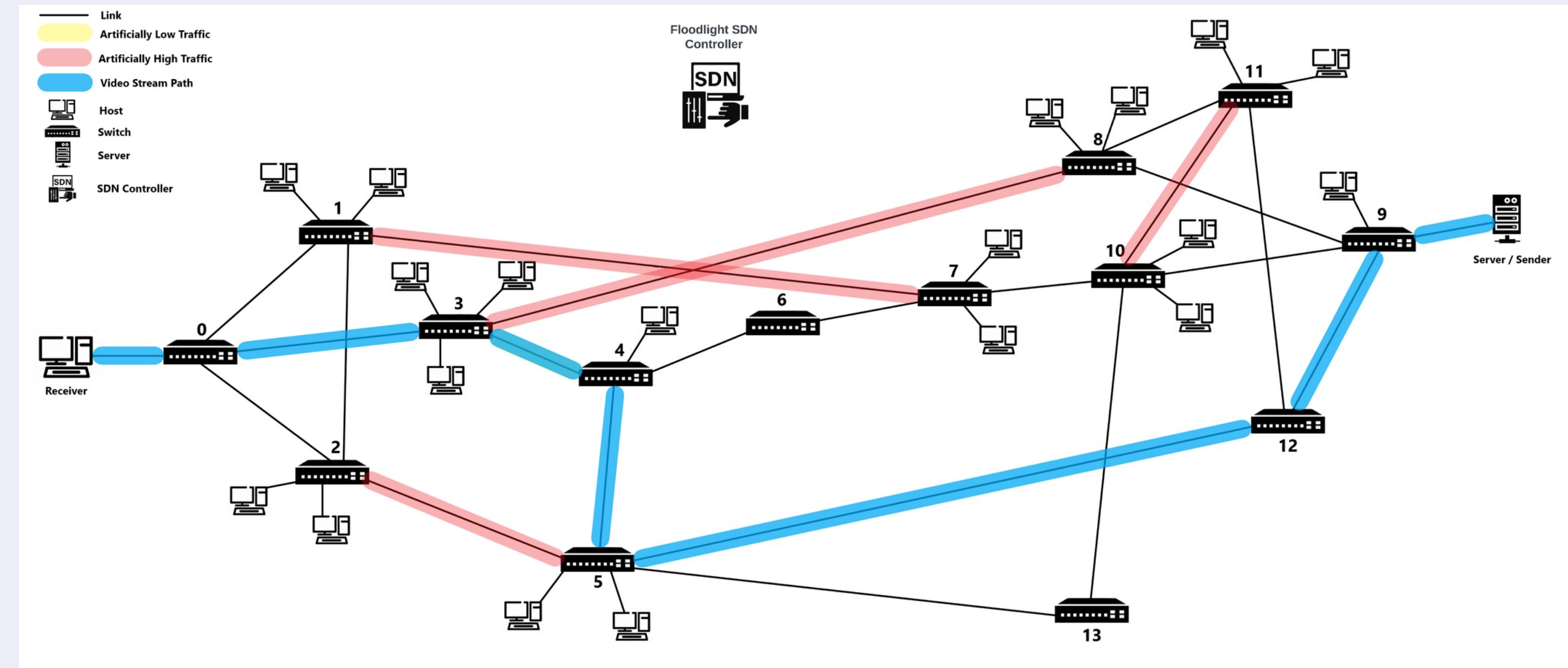


Şekil - Önerilen algoritmanın başarısını test etmek için kurgulanan senaryonun görselleştirilmesi



Şekil - Floodlight denetleyicinin hopcount metriği ile veri transferi sırasında takip ettiği yolun görselleştirilmesi

Test aşamasında ilk olarak ağa herhangi bir müdahale etmeden istemci ve sunucu arasında veri transferi yapılmaktadır. Floodlight denetleyici hopcount metriği ile en kısa mesafe üzerinden veri akışını gerçekleştirmekte ve 0-3-8-9 anahtarlarını takip etmektedir. Bu yol üzerinde ağır trafik olması veri akışını olumsuz etkilemeye ve veri kayıpları oluşturmaktadır. Daha önce yapılan testlerde denetleyicinin gitmiş olduğu yol üzerinde video gönderildiğinde kalite değerleri düşük çıkmaktadır.

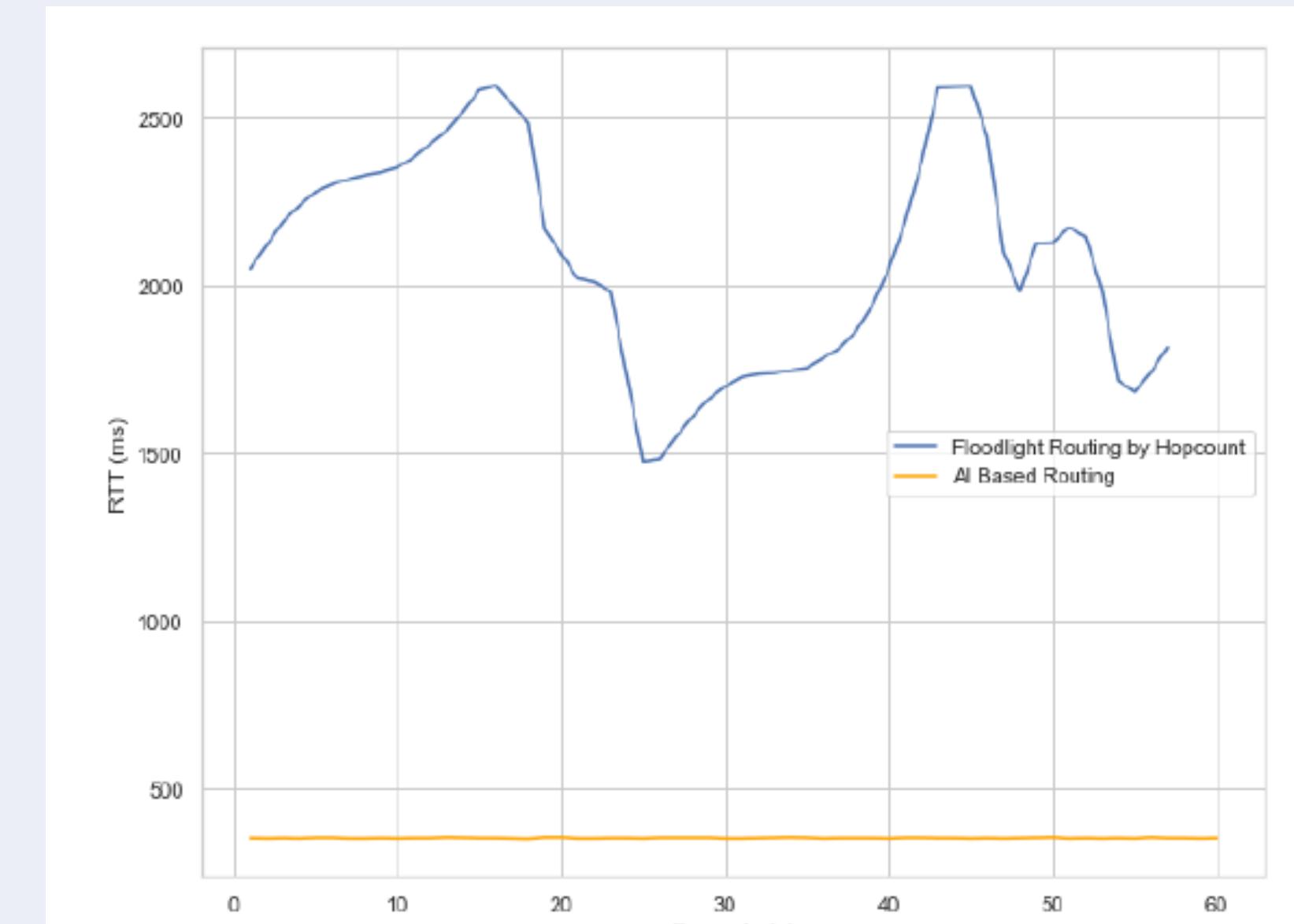


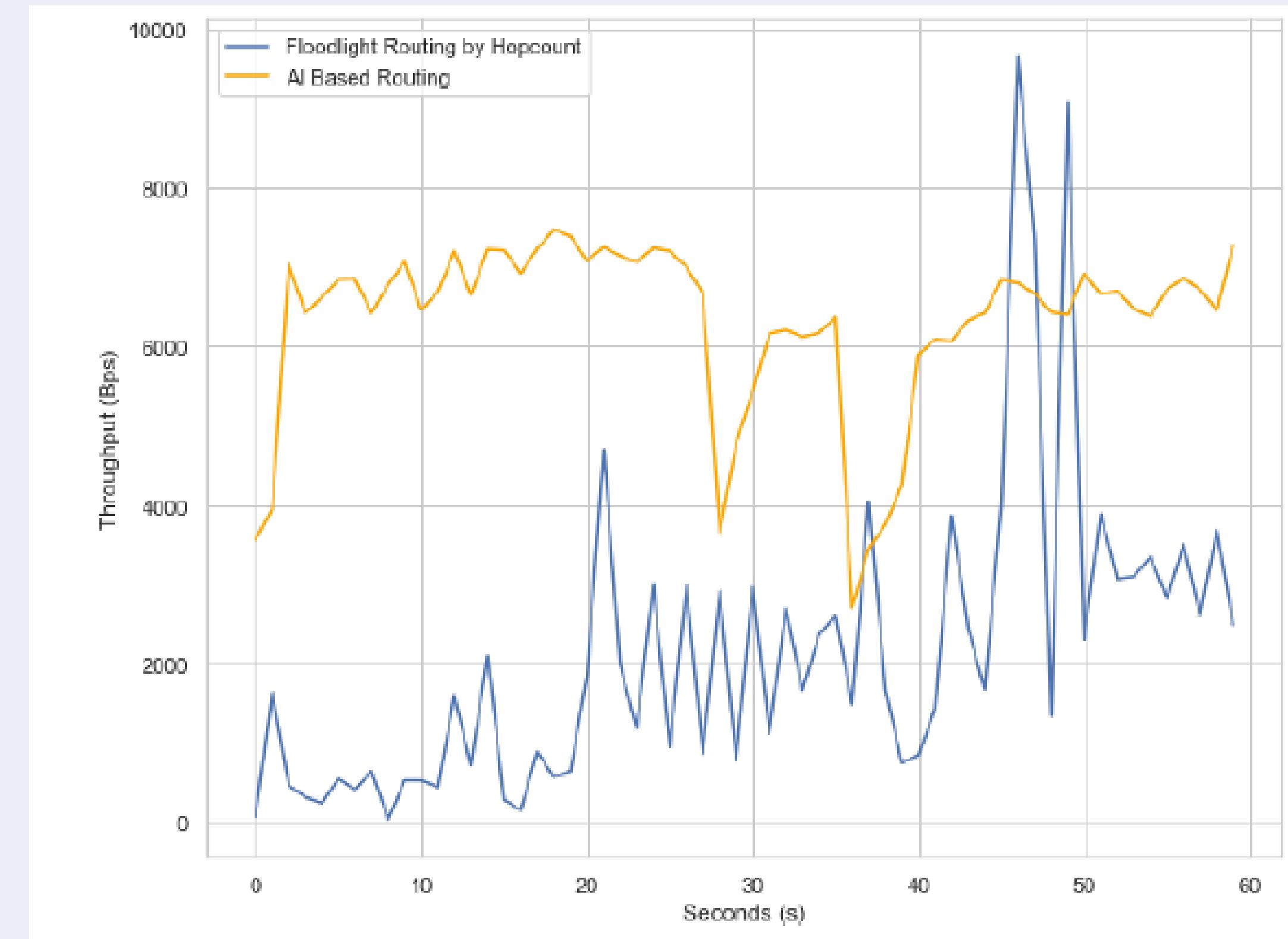
Şekil - Önerilen algoritmanın veri transferi sırasında takip ettiği yolun görselleştirilmesi

Test aşamasının ikinci kısmında önerilen yönlendirme algoritması test edilmektedir. Test için ilk aşama ile birebir aynı ortam değişkenleri kullanılmaktadır. İstemci ile sunucu arasındaki transfer öncesinde önerilen algoritma çalışmaktadır ve trafik düzeyi düşük olan yolu bulmaktadır. Aktarım sırasında 0-3-4-5-12-9 anahtarlarını takip etmektedir. Bu yol üzerinde trafiğin daha düşük olması daha başarılı bir veri aktarımı yapılmasına olanak tanımaktadır.

SONUÇ

İlk test aşamasında hopcount metrikli varsayılan yönlendirme algoritması sonucu ortalama 2063.14 ms RTT ve 2135.90 bps throughput değerleri elde edilmektedir. İkinci aşamada önerilen algoritması ile bu oranlarda iyileşme görülmekte 354.05 ms RTT ve 6313.50 bps throughput değeri elde edilmektedir. Karşılaştırma sonucu olarak önerdiğimiz algoritma başarılı bir şekilde trafik düzeyi az olan yolu tespit etmekte ve veri aktarımını bu yol üzerinden yapmaktadır.





Şekil - Floodlight hopcount metrikli yönlendirme algoritması ile önerilen algoritmanın Throughput değerlerinin karşılaştırılması

TARTIŞMA

- ***Tartışma - I***

Çalışma Mininet sanal simülasyon ortamında gerçekleştirilmekte ve gerçek bir ağ üzerinde test edilmemiştir. Sanal simülasyon ortamının maliyet, zaman ve test ortamı için kolaylık sağlama gibi birçok artısı bulunurken bazı dezavantajlı yönleri bulunmaktadır. Yapay zeka tabanlı yönlendirme algoritması için oluşturulan veri seti tamamen yapay ve bizim oluşturmuş olduğumuz senaryolar üzerinden üretildiği için daha basit ve gerçek zamanlı trafikten farklı olmaktadır. Bu çalışmanın daha verimli olabilmesi için gerçek mesafeler ve gerçek bir ağ ortamında test edilmesi daha doğru sonuçlar alınmasına olanak sağlayacaktır.

- ***Tartışma - II***

Önerilen algoritmanın tasarılanma aşamasında ağ üzerinde trafiği tespit etmek için birçok farklı parametre kullanılmaktadır. Bu parametreler ağ üzerindeki trafik durumunu daha net ve birçok farklı boyuttan tespit etmemize yardımcı olmaktadır. Burada temel amaç trafik düzeyini farklı boyutlardan izleyerek olağan ve olağan dışı durumları daha doğru tespit etmektir. Bu aşamada yapay zekanın ve mevcut bilgisayarların hesaplama kapasitesi önemli rol oynamaktadır.



BURSA ULUDAĞ ÜNİVERSİTESİ

TEŞEKKÜRLER...

CONTACT



anildursunipek@gmail.com



+90-531-517-7125



batuhanarslandass@hotmail.com



+90-539-951-5278

