FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ

MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

KENAR BİLİŞİM İÇİN SİBER SALDIRILARI TESPİT VE ÖNLEME YÖNTEMLERİ

YÜKSEK LİSANS TEZİ

Ebu Yusuf GÜVEN

(160221001)

Anabilim Dalı: Bilgisayar Mühendisliği

Tez Danışmanı: Prof. Dr. Ali Yılmaz ÇAMURCU

Tez Teslim Tarihi: 4 Temmuz 2018

FSMVÜ, Mühendislik ve Fen Bilimleri Enstitüsü'nün Bilgisayar Mühendisliği Ana Bilim Dalı Yüksek Lisans Programı **160221001** numaralı öğrencisi, "Ebu Yusuf GÜVEN", ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "Kenar Bilişim İçin Siber Saldırıları Tespit Ve Önleme Yöntemleri" başlıklı tezini, aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. Ali Yılmaz ÇAMURCU** **Fatih Sultan Mehmet Vakıf Üniversitesi**	**.....**
Jüri Üyeleri :	**Doç. Dr. Serhat ÖZEKES** **Üsküdar Üniversitesi**	**.....**
Jüri Üyeleri :	**Dr. Öğr. Üyesi Berna KİRAZ** **Fatih Sultan Mehmet Vakıf Üniversitesi**	**.....**

Teslim Tarihi: 4 Temmuz 2018

Savunma Tarihi: 31 Temmuz 2018

Aziz Şehitlerimizin Anısına...

ÖNSÖZ

Dünya'da yeni gelişen Kenar Bilişim teknolojisi hakkında Türkiye'de ilk çalışma olarak literatüre kazandırmak için çalışmalarımda beni yönlendiren ve her aşamasında takip edip yol gösteren Prof. Dr. Ali Yılmaz ÇAMURCU' ya teşekkürlerimi sunarım.

Yoğun çalışma süreçlerimde bana karşı destek ve motivasyonlarıyla yanımda olan aileme de teşekkür ederim.

Son olarak Fatih Sultan Mehmet Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü çalışanlarına, Bilgisayar Mühendisliği Bölümü öğretim görevlileri ve araştırma görevlisi kadrosuna teşekkürü borç bilirim.

Temmuz 2018

Ebu Yusuf GÜVEN

Bilgisayar Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

IoT: İnternet of Things (Nesnelerin İnterneti)

GCI: Cisco Global Cloud Index

ZB: Zettabyt

SPN: Substitution-Permutation Network (Yer değiştirme Permutasyon Ağı)

IDS: Intrusion Detection System (Saldırı Tespit Sistemi)

NIDS: Network Intrusion Detection System (Ağ Saldırı Tespit Sistemi)

HIDS: Host Intrusion Detection System (Ana Bilgisayar Saldırı Tespit Sistemi)

DIDS: Distributed Intrusion Detection System (Dağıtık Saldırı Tespit Sistemi)

MIM: Man In the Middle Attack (Aradaki Adam Saldırısı)

K-NN: K Nearest Neighbour (K-en Yakın Komşu)

SVM: Support Vector Machine (Destek Vektör Makinesi)

IDC: International Data Corporation

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

KENAR BİLİŞİM İÇİN SİBER SALDIRILARI TESPİT VE ÖNLEME YÖNTEMLERİ

ÖZET

Akıllı sistemler bilinen nesneleri değiştirdiği gibi, yaygın teknolojileri de hızla değiştirmektedir. Nesnelerin İnterneti dijital nesnelerle her alanda yayılmakta yeni ürün ve hizmetler tanıtılmaktadır. Dijital nesneler tarafından üretilen veri çoğunlukla mevcut İnternet alt yapısı üzerinden Bulut Bilişime taşınarak işlenmektedir. Bulut Bilişimle yapılan Nesnelerin İnterneti uygulamaları band genişliği ve ağ gecikmeleri nedeniyle yeterli servis kalitesi beklentisini karşılayamamaktadır. Bulut ile nesneler arasında yeni bir platform olarak tanıtılan Kenar Bilişim, gerçek zamanlı akıllı uygulamalarındaki servis kalitesi ihtiyaçları çözmeye odaklanmaktadır. Kenar Bilişim ile birlikte, Nesnelerin İnterneti yeni ürün ve hizmetlerine imkân sağlanacağı gibi önceki ürün ve hizmetlerin servis kalitesini artıran çalışmalar yapılabilecektir.

Akıllı nesnelerin yaygınlaşmasına güvenlik ve gizlilik kaygıları da önemli bir engeldir. Güvenliğin temeli olan gizlilik, bütünlük, inkâr edilemezlik, doğrulama gibi konular kaynakları kısıtlı cihazlar tarafından yeterince sağlanamamaktadır. Kenar Bilişim, gerek yeterli kaynağının bulunması gerek nesnelere yakınlığı sayesinde güvenlik uygulamalarının çalışması için ortam sunmaktadır.

Bu çalışmada Bulut Bilişim ile Nesnelerin İnterneti cihazları arasında çalışan Kenar Bilişim tanıtılmaktadır. Kenar Bilişim üzerinde çalışan ve Nesnelerin İnterneti cihazlarının güvenliğini sağlayan Kılıç Kenar Bilişim Güvenlik Uygulaması önerilmektedir. Tehdit seviyesine göre değişken güvenlik yaklaşımı sağlayan Kılıç ve modülleri ayrıntılı olarak açıklanmaktadır. Nesnelerin İnterneti cihazlarının maruz kaldığı saldırılar ve tehditler de incelenmektedir. Bu saldırı ve tehditlere karşı kılış uygulaması kural tabanlı ve makine öğrenmesi yöntemlerini kullanarak gerçek zamanlı bir koruma sağlamaktadır. Makine öğrenmesi yöntemlerinden Karar Ağacı, Destek Vektör Makinesi, K En Yakın Komşu, Derin Öğrenme ve Naive Bayes algoritmaları kullanılmaktadır. Ayrıca Kılıç'ın etkin kullanımını göstermek için gerçek zamanlı endüstri uygulaması Akıllı Fabrika bağlamında gösterilmiştir.

Tezin sonuç bölümünde proaktif Kılıç güvenlik yaklaşımının, veri iletişimindeki gecikmeye etkisi ve tehditlerin algılanmasında makine öğrenmesi yöntemlerinin doğruluk oranları olmak üzere iki konu üzerinde durulmuştur. Kılıç Kenar Bilişim güvenlik uygulamasının siber tehlike durumuna göre dinamik güvenlik seviyesi değiştirerek Nesnelerin İnterneti ekosisteminde, ağ gecikmesi ve şifreleme algoritmalarının çalışma süresinin performansa etkileri üzerine odaklanmaktadır. Yapılan test sonuçlarına göre Kılıç uygulamasının sistemin güvenliğini dinamik güvenlik seviyesi uygulaması sayesinde sabit güvenlik yöntemlerine kıyasla iki kata yakın daha iyi performans sağlamaktadır. Sonucun ikinci kısmı ise simülasyon ve test ortamında oluşturulan, iletişim ve saldırı verisi üzerinden Nesnelerin İnterneti uygulamalarına yönelik siber saldırıların tespitinde kullanılan makine öğrenmesi yöntemleri karşılaştırılmıştır. K en Yakın Komşu algoritmasının kullanılan veri setinde karşılaştırılan algoritmalara göre daha doğru sonuç verdiği sonucuna ulaşılmıştır.

CYBER ATTACK DETECTION AND PREVENTION METHODS FOR EDGE COMPUTING

SUMMARY

Intelligent systems change familiar objects as well as rapidly change common technologies. Internet of Things is spreading digital objects and introducing new products and services in every area. The data generated by digital objects is mostly processed through the current Internet infrastructure by moving to Cloud Computing. In applications using only Cloud Computing and Internet of Things technologies, there is not enough service quality expectation due to bandwidth and network delays. Introduced as a new platform between the cloud and the objects, Edge Computing focuses on solving these needs in real-time intelligent applications. Along with Edge Computing, Internet of Things will enable new products and services, as well as allowing the work done in advance to increase the quality of service. Security and confidentiality concerns are also a major obstacle to the widespread use of intelligent objects. The sources of cyber security such as confidentiality, integrity, irredeemability and verification are not adequately provided by restricted devices. Edge Computing provides the environment for the operation of security applications by virtue of the proximity to the objects, both of which must have sufficient resources. In this study, Edge Computing, which works between Bulut Computing and Internet of Things is introduced. Kılıç (Sword) Edge Information Security Framework which is working on Edge Computing is suggested.

A new security architecture is being introduced that regulates the powers of access from the outside according to the locations of the Cloud Computing, Edge Computing and IOT Devices on the network. Kılıç's modules for working on the Edge and IOT devices are explained in detail. The attacks and threats to which the Internet devices of objects are exposed are also examined. The hacking practice against these attacks and threats provides real-time protection using rule-based and machine learning methods. Decision Tree, Support Vector Machine and K Nearest Neighbor algorithms are used in machine learning methods. In addition, real-time industry practice has been shown in the context of Smart Factory to demonstrate effective use of Kılıç.

The conclusion of the thesis focuses on two topics. The first focuses on the effects of network latency and cryptography algorithms operational performance on Internet of Things ecosystem by changing the dynamic security level of Kılıç Edge Computing Security Framework according to the cyber threat situation. According to the test results, it is seen that Kılıç application performs better than the two methods thanks to dynamic security level application compared to fixed security methods. The second part of the study compared the machine learning methods used in the detection of the cyber attacks for the Internet of Things applications through communication and attack data generated in the simulation and test environment. The result is that K nearest neighbors algorithm yields better results than the compared algorithms in the data set used.

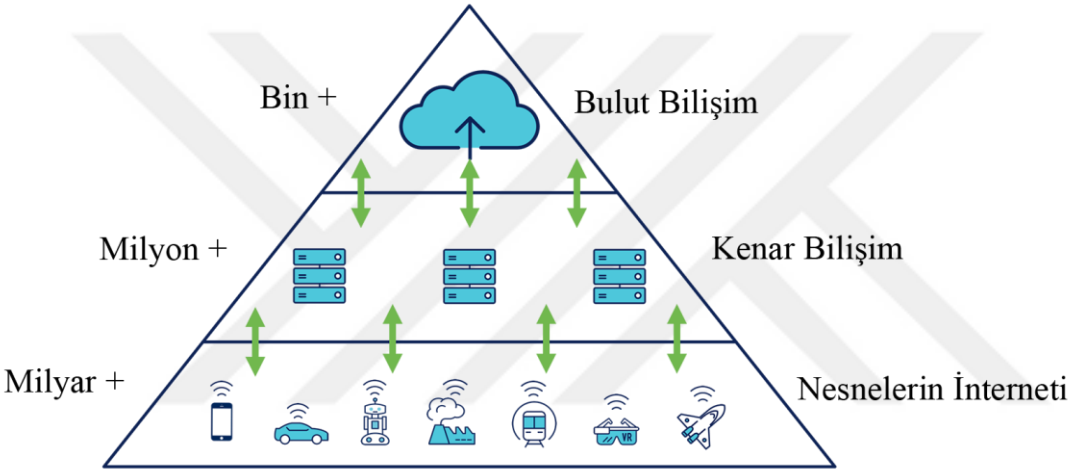
1. GİRİŞ

1.1 Kenar Bilişim Teknoloji ve Güvenlik

Teknolojik gelişmeler toplumun hayat şeklini değiştirmekte ve her geçen gün yeni teknolojik cihazlar piyasaya sürülmektedir. Nesnelerin İnterneti cihazlarıyla kurulan akıllı evler ve fabrikalar günlük kullanılan araçlardan değerli verileri toplayan sayısal cihazlara dönüştürmüştür [1]. Dördüncü endüstriyel devrimden sonra artık insanların hayatının her alanına giren Nesnelerin İnterneti (IoT) cihazlarıyla akıllı evler ve fabrikalar hizmete sunulmaktadır [2]. İnsanlık çeşitli özellik ve yeteneklerde bu cihazların ürettikleri veriler sayesinde akıllı bir dünya kurma vizyonu kazanmıştır [3]. Bununla birlikte İnternete bağlı cihaz sayısının insan sayısını geçtiği bir dönemde üretilen ham veriyi taşıma, depolama ve işleme maliyetlerinin nasıl düşürülebileceği sorunu akıllara getirmiştir.

Nesnelerin İnterneti cihazları kısıtlı kaynaklarına ve özelliklerine rağmen ürün ve hizmetlerinde niceliği sayesinde büyük veriler oluşturmaktadır. İnternet üzerinden Bulut Bilişim veya merkezi veri sistemlerine taşınarak bu cihazların ürettikleri veriler işlenir [4,5]. Mevcut alt yapının gelişimi göz önüne alındığında yapılacak Nesnelerin İnterneti uygulamaları için İnternet altyapısında yeterli band genişliği olmadığı ortaya çıkarmaktadır. Cisco Global Cloud Index'e (GCI) göre 2020 yılında bilişim sistemlerinin ürettikleri veriler 500 Zettabaytı (ZB) aşacaktır. İnternet altyapısı tarafında yalnızca 15,3 ZB verinin taşınmasına destek verebileceği yine GCI'ya göre düşünülmektedir [6]. Ayrıca Bulut Bilişim ile Nesnelerin İnterneti cihazları arasındaki İnternet alt yapısından kaynaklı ağ gecikmesi de kritik alt yapıların kurulmasına izin vermemektedir [3]. Kenar Bilişim, Nesnelerin İnterneti ile Bulut Bilişim teknolojileri arasında köprü olacağı için iki tarafında bir takım özelliklerini üzerinde bulunduracaktır. Kenar Bilişim merkezi depolama ve işlem birimlerine olan bağımlılığı azaltmak, ağdan daha hızlı geri dönüş sağlamak ve cihazlarla yakın haberleşme için geliştirilmiştir [5].

Nesnelerin İnterneti basit ve ucuz cihazlarıyla karmaşık ve yüksek maliyetli güvenlik ve gizlilik problemlerini de beraberinde getirmiştir. Düşük kaynakları yüzünden yeterli tedbir alınamayan Nesnelerin İnterneti cihazları saldırganların birincil hedefi haline gelmektedir [7]. Nesnelerin İnterneti uygulamaları dinleme, kopyalama, tekrarlama ve engelleme gibi birçok saldırı yöntemlerine maruz kalmaktadırlar [8,9]. Kenar Bilişim **Şekil 1.1** de görüldüğü gibi hem sayı olarak hem de yapı olarak Nesnelerin İnterneti ve Bulut Bilişimin arasında bulunmaktadır. Kenar Bilişim özellikle servis kalitesini artırma ve bant genişliği maliyet tasarrufu için gelişen bir teknoloji olsa da veri güvenliği ve gizlilik sorunlarını çözme potansiyeline sahiptir [4-9].



Şekil 1.1: Bulut Bilişim, Kenar Bilişim, Nesnelerin İnterneti cihaz sayıları karşılaştırması

1.2 Tezin Amacı ve Önemi

Kenar Bilişim'e kadar araştırmacılar Nesnelerin İnterneti ve Bulut Bilişim arasındaki iletişimin güvenliği için düşük cihaz kaynaklarıyla daha yüksek güvenlik seviyesi nasıl sağlanacağı üzerine çalışmışlardır [10-13]. Nesneler üzerinde kısıtlı kaynaklarla çalışabilecek hafif şifreleme yöntemlerinden yazılım tanımlı ağlara kadar birçok yöntemle gizlilik ve güvenlik sağlanma çalışmaları bulunmaktadır [10-18]. Fakat Kenar Bilişim güvenliğine ait literatürdeki üzerinde EdgeSec gibi güvenlik çerçeve (framework) çalışmaları yeni yapılmaya başlamıştır [2,3]. Gerek nesne üzerinde gerekse kenar sunucusu üzerinde yapılan güvenlik ve gizlilik çalışmaları biri diğerinin alternatifi değil birbirini destekleyen çalışmalardır.

Bu çalışmayla birlikte yeni güvenlik yaklaşımının sunulmasının yanında Kenar Bilişim için geliştirilen Kılıç Güvelik Sistemi açıklanmaktadır. Kaynak kısıtları performans ihtiyaçları nedeniyle IoT uygulamalarında yeterli güvenlik önlemi alınamamaktadır. Tez kapsamında tasarlanan Kılıç Güvenlik Sistemi kaynak kapasitesi düşük kenar sunucusu ve nesneler kullanarak değişken güvenlik seviyesi ile servis kalitesini düşürmeden proaktif yaklaşımla güvenlik sağlayacak bir sistem tasarlandı. Kılıç periyodik olarak ve siber saldırı tespit ettiği zamanlarda güvenlik seviyesini değiştirerek nesne ile kenar iletişiminin güvenliğini sağlamaktadır.

Bu tezde 2. Bölümde Nesnelerin İnterneti gizlilik ve güvenlik problemleri, IoT cihazlarının güvenliği için geliştirilen hafif (leightweight) şifreleme algoritmaları, IoT cihazları ve ağ üzerinden yapılan saldırıların tespiti ve önleme çalışmaları, Kenar Bilişim ve Sis Bilişim teknolojisi tanıtılmakta, gizlilik&güvenliğe sağladığı faydaları ve diğer benzer teknolojilerden farkları anlatılmaktadır. 3. Bölümde Kılıç, Kenar Bilişim güvenlik uygulamasının tasarımı ve önerilen proaktif yaklaşım anlatılmaktadır. Deneysel olarak Akıllı Fabrika örneğinde uygulanarak mevcut yöntemlere ağ gecikmeleri karşılaştırılmıştır. Ayrıca kendi önerdiğimiz yöntemler ve Kenar Bilişim için yaptığımız mimari tanıtımına da bu başlık altında değinilmektedir. 4 bölümde sonuçlar ve tartışmaya yer vermekte ve 5. Bölümde değerlendirme ve öneriler anlatılarak sonlanmaktadır.

2. LİTERATÜR ARAŞTIRMASI

2.1 Nesnelerin İnterneti Gizlilik Güvenlik

Bilişim sistemlerinde güvenliğin sağlanması gizlilik, aidiyet, bütünlük ve inkâr edilememe gibi temel ilkelerin korunmasına bağlıdır. Nesnelerin İnterneti kablosuz sensör ağları ve RFID gibi donanımlarıyla ev eşyalarından vücudumuza kadar birçok alanı kapsayan ürün ve hizmetlerin yanında pek çok güvenlik zafiyetini ortaya çıkarmaktadır. Nesnelerin İnternetini ürün ve hizmetlerini oluşturan birbirine bağlı cihazları, İnternet'te görülen güvenlik ve gizlilik sorunların doğrudan etkilemektedir. Nesnelerin İnterneti teknolojisi güvenliği için kullanıcı doğrulaması, cihaz doğrulaması ve erişim kontrolleri yapılmalı, üretilen veri filtre edilerek gönderilmelidir. Veriler içerisinde bulunan kişisel hayat verileri çıkartılarak, özellikle belirtilmediği sürece veriler gönderilmeden anonimleştirilmelidir [11]. Yerel ağ ve İnternet üzerinden her an gelebilecek tehditlerle karşı sistemin güvenli kabul edilmesi için kullanıcıların gerçek zamanlı güvenlik tehditlerine karşı kendi özel verilerinin güvenliğini sağlaması gerekir [19,20].

Farklı standart ve özelliklerde üretilen Nesnelerin İnterneti cihazları uyum ve güvenlik sorunlarını beraberinde getirmektedir. Nesnelerin İnterneti donanım ve yazılım kaynak kısıtları nedeniyle ağ güvenlik problemlerine karşı geliştirilen geleneksel çözümlerini uygulamaya imkân sağlamamaktadır. Bu yüzden yeni katmanlar ve bu katmanlara özgü kurgular geliştirmek gerekmektedir. Ayrıca katmanlara bölünen ve her bir alt katmanın kullandığı teknolojilere göre mimariler önermek gerekmektedir [10]. Kısıtlı kaynakları olan nesnelerin iletişimi için açık araştırma konularının yanı sıra mevcut protokolleri ve mekanizmaları analiz etmek gerekmektedir [12]. Mevcut yaklaşımların güvenlik gereksinimlerini nasıl sağladığını, iletişimlerini nasıl koruduğunu ve ne gibi çözülmemiş zorlukların olduğu analiz edilmelidir [13]. Yapılacak olan yeni çalışmalar çeşitli IoT iletişim protokolleri ile etkin ve uyumlu çalışabilir şekilde tasarlanmalıdır. Ayrıca protokol

birikimi, yeni çalışmaları heterojen ağları destekleyen protokoller olmaya zorlamaktadır.

IoT güvenlik problemleri ile ilgili olarak Tankard dışardan framework uygulayarak çözülemeyeceğini savunmaktadır [13]. Yerleşik bir güvenlik tasarımı gerekmektedir. Çünkü tehditler daha alt katmanlardan gelebilmektedir. Özellikle IoT alt yapısı güvenlik çerçevesi göz önüne alınarak tasarlanmadığında problemler oluşturabildiğini savunmaktadır. Ayrıca nesnelerin teknik güvenlik sistemi ile donatılmış olması önemli olmakla birlikte bütünleştirici ve yönetsel güvenlik personeli yetiştirmeye ihtiyaç vardır. Güvenlik yönetim sadece kendi çalışanları tarafından kaynaklanan bilgi ve teknoloji sızmasını önlemek değil, aynı zamanda bir güvenlik kültürü de oluşturulmalıdır [14].

Nesnelerin hareketli ve heterojen ağ yapısında şifreleme mekanizmaları bu artan gereksinimleri karşılamak için esnek, şeffaf ve güçlü olmanın yanında, kaynakları kısıtlı cihazlarda uygulanması için yeterince verimli olmalıdır [15]. Ancak aynı kriptoloji yönteminin farklı donanımsal kapasiteleri bulunan cihaza uygulandığında farklı performansla çalışmaktadır [15]. Yalnız algoritmaların değil cihazların da analiz edilerek birlikte güvenlik için planlanması gerektiği ürün ve hizmetlerin oluşturulurken heterojen ağlarda dikkat edilmesi gereken bir noktadır.

2.2 Hafif Şifreleme Yöntemleri

Nesnelerin İnterneti cihazlarının ürettiği hassas veri ve kaynak kısıtları nedeniyle güvenlik talepleri kriptoloji dünyası için de bir dönüm noktası olmuştur. Düşük enerji, işlem gücü ve hafıza üzerinde yüksek güvenlik sağlamak için kimi araştırmacılar donanımsal çözümlere yönelse de yeni kısıtlı kaynaklarla çalışan hafif (lightweight) kriptoloji algoritmaları olarak adlandırılan yöntemler de geliştirilmektedir [27]. Örneğin yüksek güvenlik sağlayan asimetrik kriptoloji yöntemlerinden olan 1024-bit RSA algoritması RFID etiketlerinde uygulanamaz [28]. Geleneksel kriptoloji yöntemlerine kıyasla bu hafif yöntemler **Çizelge 2.1** de görüldüğü gibi daha küçük anahtar uzunluğu ve algoritmalarında daha az döngü kullanarak şifreleme yapmaktadır. Hafif ağırlıklı yöntemlerde şifreleme bloklarını daha küçük tutarak daha düşük bellek kapasiteli cihazlarda çalışabilmesine imkân sağlamaktadır. Bu sayede hem zamandan hem de enerjiden bu sayede tasarruf sağlanmaktadır.

McKay'e göre, hafif kriptografi, yaygın olarak düşük güç tüketimin önemli olduğu kısıtlı cihazların kullanıldığı hızla büyüyen Nesnelerin İnterneti uygulamaları için çözümler sağlamayı amaçlayan bir kriptografinin alt kategorisidir [29]. Geleneksel bir kriptografi algoritması bilgisayarlarda, sunucularda ve bazı cep telefonlarında iyi performans gösterebilir. Ancak diğer yandan, ağın alt uçları RFID etiketleri, algılama cihazları ve sensör ağları ve gömülü sistem gibi cihazlar ve ağlar için hafif şifreleme platformları gerektirir.

Hafif blok kriptografi yöntemlerinin performans avantajlarını sağlamak ve maliyetten tasarruf etmek için blok boyutu küçük olmalıdır. Blok boyutu azaldığında, şifrelenecek metnin boyutunu sınırlanmaktadır [27]. Örneğin blok boyutu 128 bit AES yerine 64 bit blok boyutu kullanılmalıdır [34]. Sınırlı pil ömrüne sahip cihazlarda düşük güç tüketimini sağlamak için, daha küçük anahtar boyutu tercih edilmedir. Örneğin, PRESENT [30], anahtar boyutunda 80 bit kullanırken Kuzine [31] anahtar boyutunda 80 bit ya da 128 bit olarak seçim yapılabilmektedir.

Çizelge 2.1: Bazı hafif şifreleme algoritmaları ve özellikleri [30, 34-42]

Algoritma	Anahtar Uzunluğu	Blok Uzunluğu	Yapısı	Tur Sayısı
AES [34]	128/192/256	128	SPN	10/12/14
Klein [35]	64/80/96	64	SPN	12/16/20
PRESENT [30]	80/128	64	SPN	31
RC5 [36]	0–2040	32/64/128	Feistel	1–255
XTEA [37]	128	64	Feistel	64
LEA [38]	128,192,256	128	Feistel	24/28/32
DESLX [39]	64	184	Feistel	16
DES	54	64	Feistel	16
Seed	128	128	Feistel	16
Twine [40]	80/128	64	Feistel	32
DESL [42]	54	64	Feistel	16
3DES	56/112/168	64	Feistel	48
Hummingbird [32]	256	16	SPN	4
Hummingbird2	256	16	SPN	4
Iceberg [33]	128	64	SPN	16
Pride [41]	128	64	SPN	20

Şifreleme algoritması yapısı olarak, **Çizelge 2.1**'de adı geçen SPN (Substitution-Permutation Network) yerini alma (substitution) ve yer değiştirme (permutatiton) işlemlerinin art arda gerçeklenmesidir. Feistel şifreleme ve şifre çözmede, yer değiştirme, yerini alma ve özel veya (XOR) işlemlerini anahtar sırasını değiştirerek kullanılarak yapılmasıdır. Şifreleme ve şifre çözme işlemlerinin benzerdir bu sayede kod büyüklüğünü büyük ölçüde azaltmaktadır [30, 34-42]

Hafif blok şifreleme algoritmaları daha basit döngülerden oluşan düşük kaynak kısıtlı cihazları hedeflemektedir ve geleneksel blok şifreleme algoritmalarına kıyasla doğal olarak az tekrarlı basit hesaplama işlemlerine sahiptir. Örneğin, PRESENT tek bir S-

kutusu için geleneksel kriptografide 8-bitlik kutular yerine 4-bit S-kutusu hafif yöntemlerde kullanılmıştır [30]. Hummingbird2 [32] ve Iceberg [33] kriptografi algoritmaları sadece dört tur içerir.

Başlangıçta verilen anahtarı kullanarak farklı hesaplama turları için farklı anahtarlar üreten algoritmalar da bunmaktadır. Karmaşık anahtar listeleri, uygulamalar için daha fazla bellek ve enerji tüketir. Bu şekilde, hafif bir blok şifreleme alt anahtarları oluşturabilen daha basit anahtar listeleri kullanır. Örneğin, TEA'nın blok şifreleme algoritmasında, 128-bit bir anahtarı dört 32-bit anahtarlara ayırarak kullanılmaktadır.

2.3 Saldırı Tespit ve Önleme Teknikleri

İnternete bağlı cihazlar kullandıkları bağlantı protokollerinin açıkları, güvenliği sağlanmamış arayüzler ve yanlış yapılandırma ayarları gibi birçok zafiyetler nedeniyle siber saldırılara maruz kalmaktadır. Bu saldırılara karşı ağ güvenliği altyapılarındaki önemli bir araç olarak, ağ saldırı tespit sistemi (NIDS), ana bilgisayar tabanlı saldırı tespit sistemi (HIDS) ve Dağıtılmış saldırı tespit sistemleri (DIDS) bulunmaktadır [43]. Bu sistemler servis trafiği ihlalleri (DOS), port taramaları gibi kötü amaçlı etkinlikleri veya ağ trafiğini izleyerek ağa bağlı cihazlara saldırı girişimlerini tespit etmeyi amaçlamaktadır [44]. NIDS gelen ağ trafiğini denetlemenin yanı sıra, ayrıca giden veya yerel trafikten devam eden bir saldırı hakkında değerli bilgi edinebilmektedir. Şüpheli bağlantıları tespitte kural tabanlı, makine öğrenmesi yöntemleri [44-46], bulanık küme teorisi [47], yapay sinir ağları [48] yöntemleri kullanarak araştırmacılar çeşitli NIDS'ler tasarlanmıştır.

Nesnelerin İnterneti cihazları geleneksel ağlara dâhil olmaya başladığından beri saldırı tespit ve önleme çalışmalarında da kendine özgü problemleri beraberinde getirmiştir [49,50]. Kural tabanlı NIDS sistemlerinde IoT cihazları farklı işlev ve görevlerde çalışabildiği için görev ve işlev sınıfına göre kural tanımlanması gerekmektedir. Ayrıca sisteme eklenen her nesnede yeni analiz ve kuralların kontrol edilmesi gerekmektedir. Makine öğrenmesi ile çalışan NIDS'lerde ise önceden etiketlenmiş iletişim verisi ile öğrenme gerektirmektedir. IoT uygulamalarının etiketlenmiş öğrenme yapılacak iletişim verisinin verisi heterojen ağ yapısı yüzünden çeşitli ve az olmasının yanında genellenebilir olmayışı denetimli (supervised) yöntemlerin uygulanmasını zorlaştırmaktadır.

Saldırı tespiti genellikle bir ikili ya da çok-sınıflı bir sınıflandırma problemidir. Ağ trafiği davranışının normal mi yoksa anormal mi olduğunu anormal ise hangi saldırı yöntemi olduğunun tanımlanmasıdır [48]. Kısacası, saldırı tespitinin ana motivasyonu, sınıflandırıcıların etkili bir şekilde tanımlamadaki doğruluğunu artırmaktır. Birçok araştırma NIDS tasarımında veri madenciliği teknikleri uygulamaktadır. Destek Vektör Makinesi (SVM) bu tekniklerden birisidir. SVM, [46], [51,52] gibi önceki yapılan çalışmalardan görüleceği gibi verileri bir hiper düzlemde birçok sınıfa (en az iki) ayırır ve eşzamanlı olarak ampirik sınıflandırma hatasını en aza indirir ve geometrik marjı en üst düzeye çıkarır. Bu sayede iletişimden üretilen saldırı verisi normal veriden ayrılmasıyla oluşan sınıf kümeleri sayesinde saldırı tespit edilebilir [46].

Diğer bir makine öğrenmesi yöntemi de Karar Ağacıdır, büyük bir veri setinin öğrenilmesinde hızlı ve doğruluğu yüksektir. Bu nedenle, bir sistemde oluşturulan çeşitli verilerin ve IDS sistemi gibi ağa akan çok sayıda trafiğin analiz edilmesi için iyi sonuçlar verir [43]. Ayrıca, oluşturulan modeli karar ağacında anlamak kolaydır, bu yüzden bir güvenlik yöneticisi tarafından durum kolayca izlenebilir ve düzenlenebilir.

K-en Yakın Komşu (K-NN)'da IDS tasarlamak için kullanılan makine öğrenmesi yöntemlerinden biridir [50]. K-en yakın komşu (K-NN), özellik alanındaki en yakın eğitim örneklerine göre nesne sınıflandırması için basit ve etkili bir tekniktir [53]. Örnek kümedeki vektörlerin diğer vektörlere olan uzaklığı hesaplanarak en yakın komşular aynı sınıfta etiketlenir [54]. K-NN'de, Euclidean mesafesi genellikle iki vektör arasındaki benzerliği ölçmek için mesafe ölçüsü olarak kullanılır. K-NN sınıflandırıcılar çoklu-sınıf problemlerini çözmek için kullanılmaktadır [53]. Bu özelliği K-NN yöntemini saldırı çeşidi tespitinde diğer yöntemlere göre öne çıkmaktadır.

Derin öğrenme teorisini Profesör Hinton [55] 2006'da önerdikten sonra, derin öğrenme teorisi ve teknolojisi, makine öğrenimi alanında popüler yöntemler arasına girmiştir. Özellikle konuşma tanıma, görüntü tanıma [56] ve eylem tanıma [57-59] alanlarında başarılı sonuçlar üretmiştir. Derin öğrenme, üst düzey özelliklerin veya faktörlerin alt düzeydekilerden tanımlandığı gözlemsel verilerin hiyerarşik özelliklerini veya temsillerini hesaplamaktadır [60]. Derin öğrenme teknikleri, çok sayıda etiketlenmemiş veriden iyi bir özellik temsilini öğrenmeyi amaçlar, bu

nedenle model tamamen denetlenmemiş bir şekilde önceden eğitilebilir. Ancak temel olarak eğitim öncesi için derin öğrenme yöntemleri kullanır ve geleneksel denetleme modeli aracılığıyla sınıflandırmayı gerçekleştirilir. Sınıflandırmayı doğrudan gerçekleştirmek için derin öğrenme yöntemini uygulanması yaygın değildir ve çok-sınıflı sınıflandırmadaki performansı da düşüktür [60]. Derin öğrenmenin ağ saldırı tespitinde [61-63] ve ağ trafiği tanımlamasında [51] genellikle diğer sınıflandırma yöntemlerle birlikte kullanıldığında iyi sonuçlar verdiği görülmektedir.

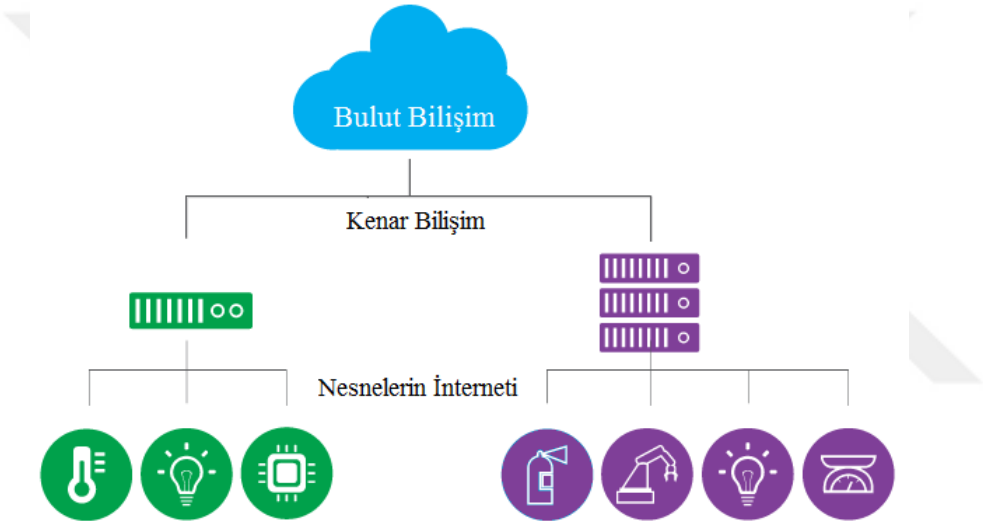
Naive Bayes sınıflandırıcı, Bayes teoreminin uygulanmasına dayalı denetimli bir öğrenme algoritmasıdır [64,65]. Naive Bayes, Bayes olasılığı modelinin sadeleştirilmiş halidir [66]. Naive Bayes sınıflandırıcı, güçlü bir bağımsızlık varsayımıyla çalışır [66]. Bu, bir öznitelik olasılığının diğerinin olasılığını etkilemediği anlamına gelir [67]. Naive Bayes başarılı olmasını eğitim verisi gürültüsü, sapma ve varyans faktörlerinden kaynaklandığını belirtir [68]. Eğitim verisi gürültüsü, yalnızca iyi eğitim verisi seçerek en aza indirgenebilir. Eğitim verileri, makine öğrenme algoritması tarafından çeşitli gruplara ayrılmalıdır. Sapma, eğitim verilerindeki yanlış gruplamaya neden olmaktadır. Varyans, bu gruplandırmaların çok küçük olmasından kaynaklanmaktadır [67].

Yapılan diğer araştırmalarda Syarif ve arkadaşları [69], doğruluk oranını artırmak ve yanlış pozitifi azaltmak için Naive Bayes, Decision Tree, JRip ve iBK kullanmışlardır. Bahri ve arkadaşları [70] Greedy-Boost adlı yeni bir topluluk yöntemine dayanan bir melez yaklaşım tanıtmışlar ve AdaBoost, C4.5 ve Greedy-Boost'un hassasiyetini deneysel olarak karşılaştırdılar. Bukhtoyarov ve arkadaşları [71], ağ saldırı tespit problemine, sinir ağ yapılarının olasılık temelli jeneratörü olarak adlandırılan temel sinir ağı sınıflandırıcılarını tasarlamak için olasılıksal bir yaklaşım uygulamıştır.

Tez çalışması kapsamında simülasyon ortamında hazırlan veri seti üzerinde kural tabanlı ve makine öğrenmesi yöntemlerinden Karar Ağacı, Destek Vektör Makinesi, K-en Yakın Komşu, Derin Öğrenme ve Naive Bayes teknikleri kullanılarak elde edilen sonuçlar karşılaştırılmıştır. Simülasyon üzerinde oluşturulan veriler bu yöntemlerle ayrı ayrı test edilmiş ve sonuç bölümünde karşılaştırılmıştır.

2.4 Kenar Bilişim

Nesnelerin İnterneti ve 5G ile birlikte aygıtlarının sayıları ve ürettikleri verilerin boyutu her geçen gün yükselen bir ivme göstermektedir [72]. İnternet altyapısı, bu hıza yetişememekte, Nesnelerin İnterneti uygulamaları band genişliği problemiyle karşı karşıya kalmaktadır. Akıllı ürün ve hizmetler için İnternet altyapısı, Nesnelerin İnterneti ve Bulut Bilişim arasında dar boğaz oluşturmaktadır. Bu durum, mevcut İnternet altyapısıyla veriyi Bulut Bilişim'e taşıyarak işlem ve depolama ihtiyaçlarının geleneksel bilgi işlem modeliyle çözülemeyeceğini göstermektedir. Bu problemleri aşmak için İnternet ağının kenarında verileri işleyip ve depolamasına imkân veren Kenar Bilişim'i ortaya çıkarmıştır.



Şekil 2.1 Bulut Bilişim, Kenar Bilişim ve Nesnelerin İnterneti

Kenar Bilişim doğuşu itibariyle Şekil 2.1 de görüldüğü gibi Bulut Bilişim ve Nesnelerin İnterneti cihazları arasında tampon vazifesi görmektedir. Nesnelerin İnterneti cihazlarına yakın olması nedeniyle verinin ağda taşınması sırasındaki gecikmeyi azaltan ve nispeten enerji ve işlem gücü problemi olmayan Kenar Bilişim sunucusu, Nesnelerin İnterneti cihazlarının ürettiği verileri işler [73]. Kenardan İnternet'e ham verinin değil işlenmiş verilerin Bulut Bilişim veya merkezi veri sistemlerine gönderilmesini sağlar. Bu sayede bant genişliği problemi çözülürken merkezi veri sistemlerinde işlem gücü ve hafızadan tasarruf sağlanmaktadır [4-8], [73-88].

Çeşitli donanım ve sensörlerden veri toplayan bilgi işlem altyapısına, örneğin endüstriyel makinelere ve kontrol cihazları Kenar Bilişim sunucusu olarak adlandırılabilir [75]. Genellikle Kenar Bilişim sunucuları veri merkezlerinden uzakta bulunmaktadır. Kenar Bilişim, International Data Corporation'a (IDC) göre, 100 metrekareden daha az bir alanda kritik verileri yerel olarak işleyen veya saklayan ve elde edilen verileri Bulut Bilişim'e gönderen bir mikro veri merkezleri ağıdır [76]. Kenar Bilişim ağ kenarlarında yalıtılmış bilgi işlem platformları olmaktan ziyade, Bulut Bilişimden nesnelere kesintisiz bir hizmet sağlamayı amaçlamaktadır. MobilData, Mikro Veri Merkezi, Mobil Bulut Bilişim, CloudLet ve Bulut-Deniz Bilişim gibi önceki çalışmalar band genişliği, Bulut Bilişimdeki depolama ve hesaplama yükünü azaltmak için tanıtılmış Kenar Bilişim uygulamalarıdır [61-70].

Gerçek zamanlı akıllı uygulamalar ağ gecikmesinden etkilenir ve servis kalitesi düşer. Nesnelerin İnterneti uygulamaları, geleneksel Bulut Bilişim ile sağlanamayan gerçek zamanlı gereksinimlere sahiptir. Kenar Bilişim, konum ve içerik duyarlılığı gibi farklı özellikleri sayesinde potansiyel bir çözüm olabilir [62]. Kenar Bilişim çözümleri kullanılan uygulamalarda Nesnelerin İnterneti cihazları veri merkezine veya Bulut Bilişimle doğrudan iletişim kurmak zorunda değildir. Sağlık uygulamaları, finansal hizmetler veya üretim gibi milisaniye gecikmelerin önemli olduğu sistemler için bu çözüm idealdir [72].

Akıllı nesneler kullanıldığı ortamın etkisi ve çalışma ortamındaki enerji değişimleri, ısı değişimleri dış faktörlerden kolaylıkla etkilenebilir. Bu yüzden bozulma veya hatalı veri üretme ihtimalleri diğer geleneksel cihazlara göre daha fazladır. Bu ihtimaller ağa bağlı bilgisayar tablet veya akıllı telefon gibi cihazlara göre daha fazladır. Kenar Bilişim teknolojisi kullanılarak Nesnelerin İnterneti cihazlarından alınan veya gönderilen verilerin daha doğruluğu kontrol edilebilir. Böylece veriler bulut sistemlerine temiz ve işlemeye hazır halde ulaşmaktadır.

Nesnelerin İnterneti uygulamaları genellikle ağ heterojen bir yapıya sahiptir [9]. Nesnelerin İnterneti düğümleri Bluetooth, ZegBee, Wi-Fi gibi birçok farklı iletişim protokolünü destekleyebilmektedir. Kenar Bilişim sunucusu üzerinden bu heterojen yapı bir noktada tekilleştirip Bulut Bilişimle iletişim kurulmaktadır. Bu sayede Bulut Bilişim sunucusu Nesnelerin İnterneti cihazlarının kullandıkları protokolden bağımsız olarak çalışabilmektedir.

Kenar Bilişimi kritik hale getiren özellikler şöyle sıralanabilir:

- İnternet alt yapısının bant genişliği kısıtları,
- Bulut Bilişim'e veri aktarmanın yüksek maliyeti,
- Ağ gecikmesinin servis kalitesini düşürmesi,
- Gerçek zamanlı bilgi işlem için düşük ağ gecikmesi ihtiyacı,
- Paralel işlem yetenekleri ve dağıtık mimari,
- Yazılım güncelleme ve siber güvenlik uygulamalarına ortam sağlaması,
- Nesneler arası uyumluluk ve heterojen ağların tekilleştirilmesidir.

Kenar Bilişim'in çözüm olarak kullanıldığı önemli alanlardan biri de endüstrinin binlerce sensör ve cihaz bulunan akıllı üretim merkezleridir. Akıllı üretim cihazları anlık olarak Bulut Sistemlerinden gelen değerlere göre hassas üretim süreçleri uygulanabilir. Sürecin işletilmesi için gecikme olmadan üretimin sadece ilgili kısmının verilerine ihtiyaç vardır. Kenar bilişim üretim cihazları için Bulut Bilişimden alınması gereken verileri önceden temin ederek çalışma zamanında yaşanacak İnternet alt yapısından kaynaklı gecikmelerini minimize etmektedir [73]. Günlük raporları veri merkezine veya uzun süreli depolama için Bulut Bilişim'e gönderir. İnternet üzerinden sadece sonuç verilerini göndererek düşük band genişliği kullanmanın yanında trafiğin düşük olduğu zamanlarda veri göndererek veri yükünü azaltır ve ağın yükünü de dengelemeye yardımcı olur.

Kenar Bilişimle içerik algısı, gerçek zamanlı bilgi işlem ve paralel işleme gibi etkili özellikleriyle öne çıkmaktadır. Endüstride üretilen verilerin yaklaşık %90'ı geleneksel veri merkezleri veya Bulut Bilişim sistemlerinde işlenmekte ve saklanmaktadır. Gartner'e göre 2022 yılına kadar merkezi veri depolama sistemleri %50 ye kadar gerileyeceği ve ağ üzerine dağıtılmış depolama sistemlerinin yaygınlaşacağı tahmin edilmektedir [77].

Kenar Bilişim başlangıçta Mikro Veri Merkezi, Mobil Bulut Bilişim, Sis Hesaplama, CloudLet ve Bulut-Deniz Bilgisayarı gibi önceki çalışmalar Bulut Bilişimde depolama ve hesaplama yükünü azaltmak için tanıtılmıştır [78-80]. Kenar Bilişimin tanınıp yaygınlaşmaya başlamasıyla da araştırmacılar, Mobil Kenar Hesaplama üzerine yoğunlaşarak, Bulut Bilişim hizmetlerini ağın kenarına kadar genişleten hesaplama işlemini desteklemek için yeni bir mimariler olarak sunulmuştur [87,88].

Tez bu kapsamında Kenar Bilişim'e Gerçek zamanlı IoT uygulamaları için güvenli veri iltimi için mimari önerisinde bulunulmuştur.

2.5 Kenar Bilişim ve Güvenlik

Nesnelerin İnterneti cihazları üzerinde enerji ve işlem gücü kısıtlarından dolayı güvenlik ve gizlilik işlemleri geçekleştirilmesi mümkün olmamaktadır. Nesnelerin İnterneti cihazlarının ürettikleri veri İnternete bağlanmadan önce, Kenar Bilişim sunucusunda verinin güvenliği sağlanabilir. Güvenlik seviyeleri artırılırken gerçek zamanlı uygulamaların çalışmasına engel olmayacak şekilde kurgulanması önemlidir. Ayrıca ihtiyaçlara göre özelleştirilebilir ve şeffaf bir güvenlik uygulaması kenar bilişim cihazları için önemli olacaktır.

Kenar Bilişim'den Bulut Bilişim'e veri gönderilmesi konusunda yalnızca İnternet üzerindeki gizlik sorunlarını değil bulut teknolojisi üzerindeki şüpheleri de dağıtacak çalışmalar yapılmaktadır. Veri kenar üzerinde çıkmadan genelleme anonimleştirme çalışmaları yapılarak bulutta ihtiyaç olmayacak kişisel verilerin gizlenmesi sağlandığı gibi hafıza ve işlem ihtiyacını azaltacağından Bulut Bilişim tarafındaki maliyetleri de düşürecektir. Örneğin saniyede alınan bir sıcaklık verisi dakika için ortalaması alınarak genelleştirilebilir. Kalp atış düzeni takip edilen hastanın kimlik bilgileri gizlenip benzersiz bir anahtar üzerinden buluta gönderilip çalışma yapılırsa bulut tarafında verinin kime ait olduğu bilgisi saklanmayarak güvenlik kaygıları azaltılabilir.

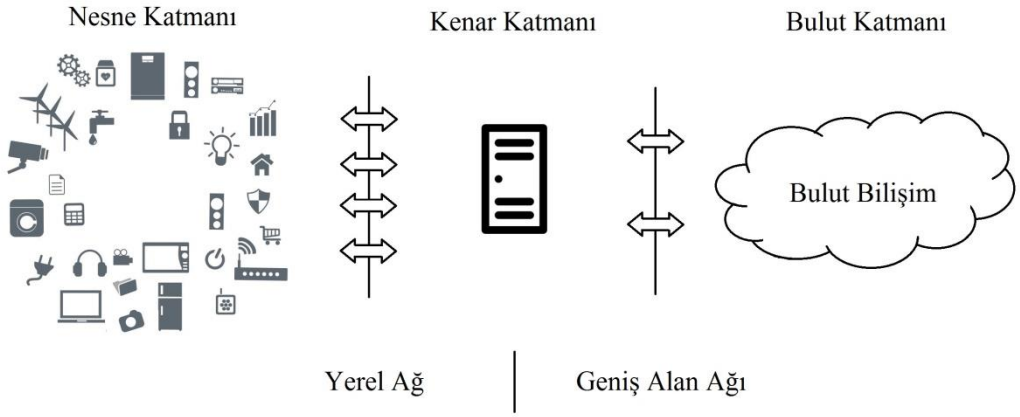
Güvensiz ağa çıkmadan verinin güvenlik ve gizlilik ihtiyaçlarını gidermek için Kenar Bilişim sunucusunun seçmesinin başlıca nedenleri şunlardır:

- Kenar Bilişim sunucularının işlem gücü ve pil ömrü gibi kaynak kısıtlarının olmayışı,
- IoT cihazlarına fiziksel yakınlığı,
- Yerel ağ üzerinde olması (Bütün Kenar Bilişim sunucuları için geçerli olmayabilir),
- Esnek uygulamalar geliştirme imkânı,
- Bulut sistemleriyle geleneksel güvenli bağlantıyla haberleşebilmesidir.

Nesnelerin İnterneti cihazlarına bulaştırılan zararlı yazılımlarla akıllı nesneler uzaktan kontrollü saldırı aracına dönüştürülmekte (IoTBotNet) ABD de yapılan DSN

DNS'e yapılan saldırısında olduğu gibi ciddi ekonomik zararlara neden olan DDos saldırıları yapılmaktadır [89]. Kenar Bilişim, Bulut Bilişim ve IoT cihazları arasında bir köprü görevi görmektedir. Kenar bilişime bağlı Nesnelerin İnterneti cihazları üzerinden yapılacak saldırıları da tespit ederek kurban cihazlara zarar vermeden engellenebilecektir [91,91]. Ayrıca dış ağ üzerinde herhangi bir kenarda oluşan saldırı yalnızca kenara zarar vereceğinden geri kalan Nesnelerin İnterneti cihazları bu saldırıdan etkilenmeyecektir [77-93].

Kenar Bilişim IoT uygulamalarındaki güvenlik ve gizlilik endişelerini, kaynak kısıtları olan IoT cihazları ile çözmek yerine yerel ağdaki Kenar Bilişim sunucusu üzerinden sağlaması yeni bir bakış açısı kazandırmaktadır [4-6]. Yönetilebilirlik geliştirme ve bakım için uçtan uca sistemin **Şekil 2.2** de görüldüğü gibi katmanlı bir yapıda ele alınması önemlidir [3,4] Nesne Katmanı, Kenar Katmanı ve Bulut Katmanı olarak üç ayrı katmana ayırmaktadır. Katmanlı yapı kullanılarak IoT uygulamasının gizlilik ve güvenlik endişeleri her katman için ayrı önlemler alınarak çözümler geliştirilmesine olanak sağlamaktadır.

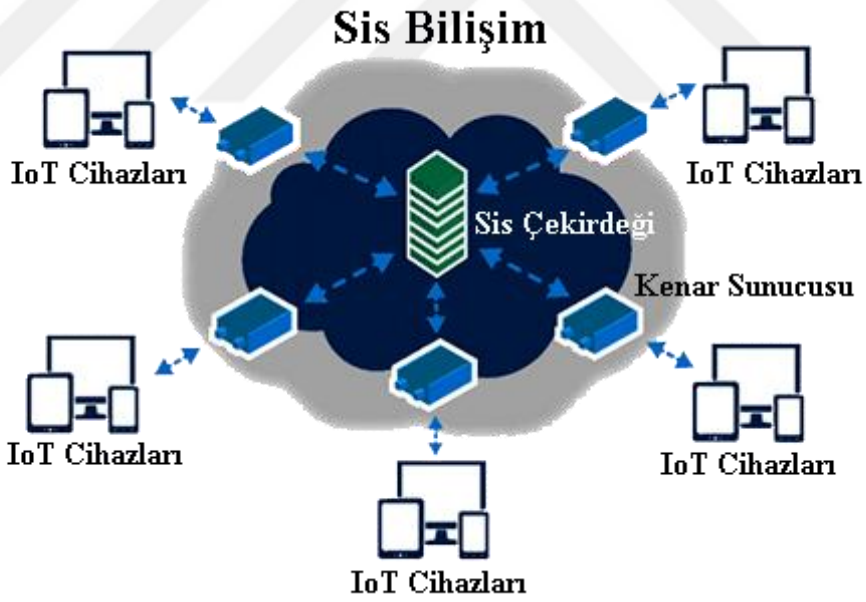


Şekil 2.2: Nesne, Kenar ve Bulut Katmanları

Bulut Bilişim kapsamında bazı mevcut güvenlik çözümleri, kenar sunucularındaki birçok güvenlik ve gizlilik sorununu için kullanılabilmektedir [9]. Ancak, hareketlilik desteği gibi farklı özellikleri nedeniyle yeni güvenlik ve gizlilik zorlukları getirebilir [9]. Bu nedenle Kenar bilişim güvenliği ayrıca ele alınması gereken önemli bir konudur. Güvenliği sağlanamayan Kenar Bilişim sunucuları güvenli ağlara giriş noktaları haline gelebilmektedir [9].

2.6 Sis Bilişim

Sis Bilişim veri işleme ve depolama hizmetlerinin uygulamanın veri kaynağı ile Bulut Bilişim arasına verimli şekilde çalışabileceği yerde dağıtıldığı merkezi olmayan dağıtık bilgi işlem altyapısıdır. Sis Bilişim, Bulut Bilişimin veri kaynağına yaklaştırıldığı metodolojidir. Kenar Bilişim ile Sis Bilişim veri işlemenin ve depolamanın kenarda yapılmasından dolayı literatürde birbirinin yerine kullanılsa da aralarında farklar bulunmaktadır. Kenar Bilişim veri üreten nesneleri ile Bulut Bilişim veri merkezleriyle arasında tampon (buffer) olmaktadır. Veriyi kenarda işleyip nesnelerin ağ gecikmesinden Bulut Bilişimin band genişliği problemlerini çözmektedir. Sis Bilişim ise birbirleriyle iletişim kuran Kenar sunucularına odaklanır. Bulut Bilişim'in eksiklerini tamamlayan değil dağıtık mimarisiyle gelişen alternatif bir teknolojidir [75]. Ancak Purdue Üniversitesinden Mung Chaing e göre Kenar, Sis ve Bulut bilişim birini domine eden değil destekleyen teknolojilerdir [88]. Sis Bilişim Kenar Bilişimi kapsar, ancak Sis Bilişim aynı zamanda işlenen verileri nihai hedefine almak için gereken ağı da içerir.



Şekil 2.3: Sis Bilişim mimarisi

Sis Bilişim, **Şekil 2.3** de görüldüğü üzere bulut, ağ, kenar, istemci ve nesnelerin tamamını içerir. Sis Bilişim mimarisi, bulutta dağıtılan kaynakların ve işlevlerin birleştirilmesini, düzenlenmesini, yönetilmesini ve güvenlik seviyesinin artırılmasını

da sağlayacaktır. Sis Bilişim, geleneksel telekomünikasyon hizmetleri dâhil birden fazla endüstri ve uygulama alanı için ortak kenar işlevlerini destekleyecek yatay bir platform öngörmektedir. Ayrıca Sis Bilişim mimarisiyle kablosuz ağların yanı sıra kablolu hat üzerinde çalışacak kadar esnek olacaktır [88].

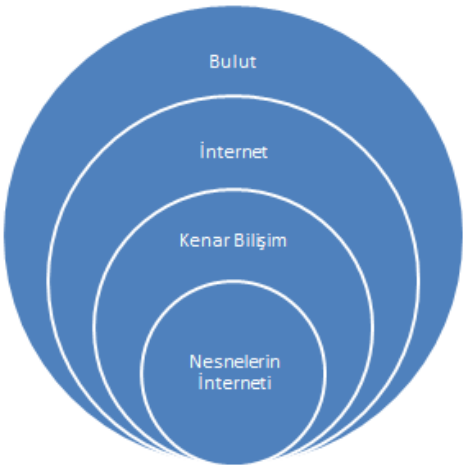
3. TASARLANAN KENAR BİLİŞİM GÜVENLİK SİSTEMİ ve MİMARİSİ

3.1 Tasarlanan Kenar Bilişim Mimarisi

Kenar bilişim doğuşu itibariyle güvenlik ve gizlilik problemlerini çözmek için geliştirilmemiştir. Gerçek zamanlı IoT uygulamalarındaki ağ gecikmesi ve band genişliği problemlerini aşmak için bir tampon (buffer) olarak tasarlanmıştır. IoT cihazlarının hem kenar hem bulut sistemleriyle çalıştığı çeşitli mimariler bulunmaktadır. IoT cihazlarının eş zamanlı çalışan Bulut ve Kenar Bilişim sunucularına erişebildiği mimari tasarımları bulunmaktadır [6-8]. Hareketli nesnelerin dolaşım yaparken gecikmesini engelleyecek Kenar Bilişim mimarileri de mevcuttur [61-68]. Önerilen mimarilerde IoT cihazları için güvenlik ve gizlilik problemlerini çözmediği gibi yeni açıklara neden olabilmektedirler [7].

Bu tez kapsamında güvenlik ve gizlilik problemlerine dikkate alan, tehditlere ve saldırılara karşı dirençli bir kenar bilişim mimarisi oluşturulması hedeflenmiştir. Önerilen mimaride **Şekil 3.1'de** görüldüğü gibi IoT cihazları doğrudan güvensiz ağa ve bulut sistemlerine ulaşması yerine ürettiği veriyi Kenar Bilişim sunucusu üzerinden göndermektedir. Bu sayede dış ağdan IoT cihazları izole edilmektedir. Verilerin Kenar Bilişim sunucusu üzerinde gizlilik ve güvenliği sağlandıktan sonra mevcut İnternet alt yapısı üzerinden Bulut Bilişim sunucularına ulaştırılacaktır. Bu mekanizma sayesinde kısıtlı kaynak nedeniyle yeterli güvenlik ve gizlilik tedbiri alamayan ağdaki nesnelerin ürettiği veri koruma altına alınmaktadır. Verilerin tamamı ön işlemden geçirilmekte bu sayede Bulut Bilişim'e kullanıma hazır olarak iletilmektedir.

IoT cihazları virüs bulaştırılarak IoT bot net düğümüne dönüşse bile dış ağda bir adrese saldırı yapması Kenar Bilişim sunucusu üzerinden engellenmektedir. Aynı şekilde Bulut Bilişim üzerinden gelen komutlar ve bilgi de kenar cihazı üzerinde değerlendirildikten sonra IoT cihazlarına iletilmesine fırsat sunmaktadır [2]. Bulut Bilişim ve diğer üçüncü platformlarında oluşabilecek herhangi bir zafiyet üzerinden gelen saldırının Kenar Bilişim üzerinde tespit edilip engellenmesine olanak sağlamaktadır. Kılıç uygulaması da nesnelerin yalnızca Kenar Bilişim sunucusu ile iletişim kurduğu izole edilmiş Güvenli Kenar Bilişim Mimarisi üzerinde çalışmaktadır.



Şekil 3.1: Güvenli Kenar Bilişim Mimarisi

3.2 Tasarlanan Sistemin Genel Yapısı

Bu tezde yapılan çalışmada "Kılıç" diye isimlendirilen güvenlik sistemi Kenar Bilişim sunucusu üzerinde çalışmaktadır. Kılıç, Kenar Katmanı ve Nesne Katmanı arasındaki iletişim güvenliği ve gizliliği üzerine odaklanmaktadır.

Bu tezde **Şekil 3.1** de görülen mimari esas alınarak tasarlanan ve geliştirilen Kılıç adını verdiğimiz Kenar Bilişim güvenlik sistemi kenar sunucusu ile nesneler arasında saldırı tehdidine göre değişken güvenlik seviyesi sağlamaktadır. Tasarlanan sistem değişken güvenlik seviyesi tehdit algılandığında yüksek güvenlik sağlayan yöntemlerin kullanılmasını tehdit olmadığı durumlarda düşük güvenlik sağlayan yöntemlerin ya da ham veri iletişiminin yapılmasını önermektedir. Kılıç üzerinde öncelikle cihazın ve ortamın güvenlik analizi yapılarak kaydedilmekte uygulama çalışma zamanında bu parametrelere göre çalışmaya başlamaktadır. Ortam ve nesne analizlerinin yanında sistem yöneticisinden nesnelerin hedef güvenlik seviyeleri de alınmaktadır. Kılıç Güvenlik Sistemi olası tehditler ve istenmeyen durumları tespit ederek güvenlik seviyelerini değiştirmektedir. "Kılıç hasma göre çekilir" ilkesiyle IoT uygulamalarının maruz kaldığı saldırı yöntemine ve hedef güvenlik seviyesine göre tedbir çalışmaları yapmaktadır. Uyguladığı çeşitli senaryolarla nesne ile kenar arasındaki iletişimin; güvenlik, gizlilik, doğruluk ve gerçeklik esaslarına uygun olarak yapılması sağlanmaktadır.

Kılıç Kenar Güvenlik Sistemi ağ ve daha önce anlatılan Kenar Katmanı ve Nesne Katmanı üzerine **Çizelge 3.1** de görüldüğü gibi yerleşmektedir. Kılıç'ın tamamı yerel

ağ üzerinde Kenar Bilişim sunucusu ve nesnelerin üzerinde çalışmaktadır. Kılıç nesne üzerinden gelen heterojen ağdaki farklı iletişim protokollerini Kenar sunucusu üzerinde tekilleştirerek, güvenli iletişim protokolleriyle Bulut sistemlerine veriyi ulaştırmaktadır. Bu işlemlerin nasıl gerçekleştiği ve işlemleri yapan modüller bölüm 3.4 de ayrıntılı olarak açıklanmaktadır.

Akıllı fabrika örneği temel alınarak Kılıç uygulaması test edilmiş ve saldırı senaryoları uygulanmıştır. Senaryolar için gerekli ham veri ve saldırı veri setleri oluşturulmuş, Kenar Bilişim alt yapısının IoT cihazlarının güvenlik seviyelerini artırmak için nasıl kullanılabileceği gösterilmiştir. Kılıç Kenar Bilişim Güvenlik Uygulaması IoT sistemlerinin güvenliğine ve performansına etkileri, yapılan çalışma kapsamında bölüm 3.5 de detaylı olarak incelenmiştir.

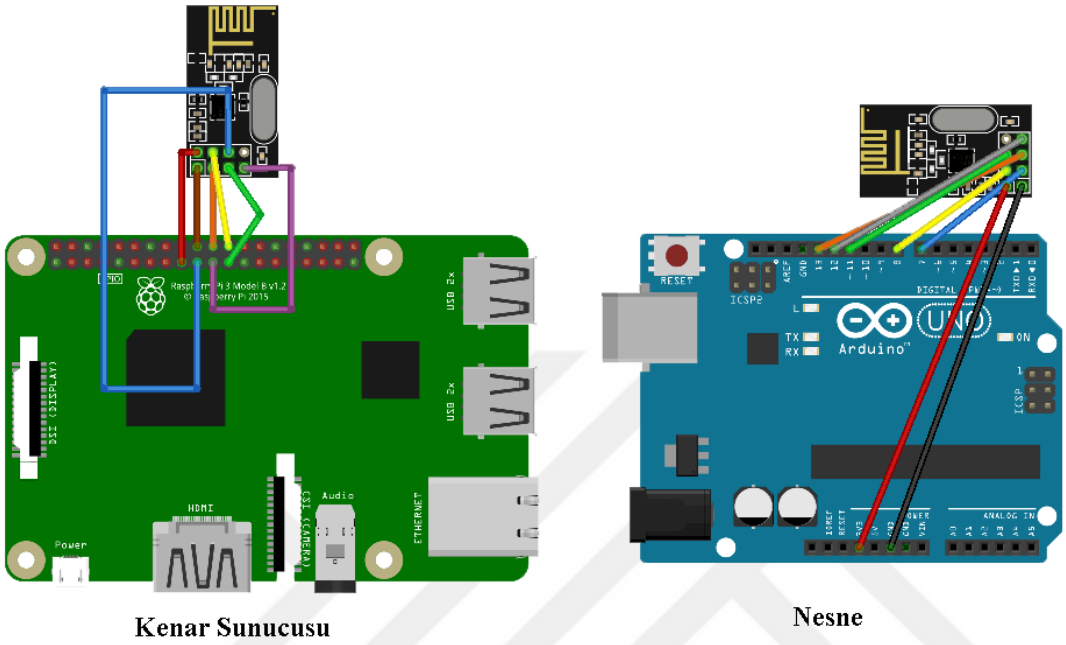
Çizelge 3.1: Kılıç Güvenlik Sistemi Modülleri

<table>
<tr><td>Bulut Katmanı</td><td colspan="5"></td></tr>
<tr><td>Geniş Alan Ağı</td><td></td><td colspan="2">TSL</td><td colspan="2">HTTPS</td><td></td></tr>
<tr><td rowspan="3">Kenar Katmanı</td><td rowspan="3">Anomali ve Saldırı Tespiti Modülü</td><td colspan="3">Bulut İletişim Arayüzü</td><td rowspan="3">Saldırı Önleme ve Veri Doğruluğu Sağlama Modülü</td></tr>
<tr><td>Ortam ve Cihaz Analizi Modülü</td><td colspan="2">Cihaz Kimlik Yönetimi Modülü</td></tr>
<tr><td colspan="3">Nesne İletişim Arayüzü</td></tr>
<tr><td>Yerel Ağ</td><td></td><td>ZigBee</td><td>Wi-Fi</td><td>Bluetooth</td><td>RF</td><td></td></tr>
<tr><td rowspan="2">Nesne Katmanı</td><td rowspan="2"></td><td colspan="3">Kenar İletişim Arayüzü</td><td rowspan="2"></td></tr>
<tr><td colspan="3"></td></tr>
</table>

3.3 Tasarlanan Sistemin Donanımsal Mimarisi

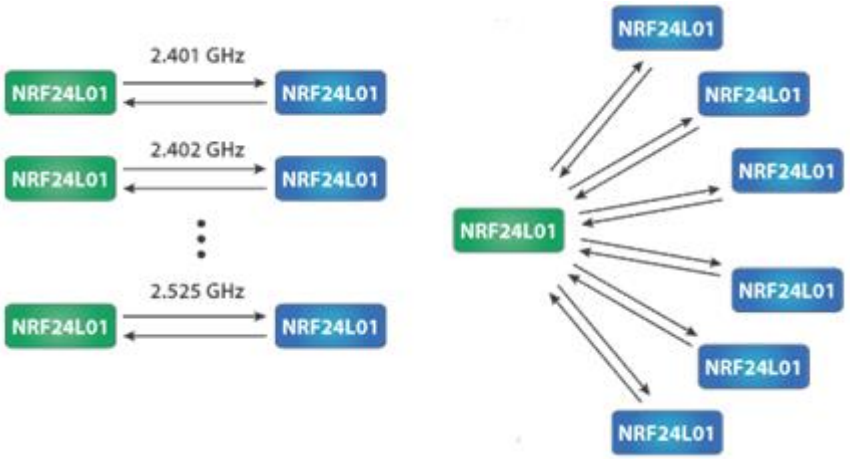
Kılıç Güvenlik Sistemi test edildiği laboratuvar ortamında yaygın kullanılan açık kaynak geliştirme kartlarından **Şekil 3.2** da görüldüğü gibi Raspberry Pi 3 kenar cihazı olarak ve nesne olarak da Arduino UNO cihazları kullanılmıştır. Raspberry Pi 3 üzerinde işlem gücü, çevre aygıtlarının ve sensör setlerinin kolay bağlanması, işletim sistemi bulundurması, Wi-Fi ve eternet gibi arayüzleri sayesinde İnternete kolay bağlanması gibi bir çok avantajlı özelliği nedeniyle tercih edilmiştir. Ayrıca

gelişmiş makine öğrenesi ve kriptoloji kütüphaneleriyle ile Kılıç Güvenlik Sistemi ana modüllerini tez sürecinde yetiştirmemize olanak sağlayan Python geliştirme diline destek verdiği için de Raspberry kullanılmıştır. Akıllı nesneler tarafında kolay implementasyon ve düşük maliyetleri avantajıyla Arduino UNO kullanılmıştır.



Şekil 3.2: Kenar Bilişim Sunucusu Raspberry Pi 3 Nesne Arduino UNO

Kenar ile nesnelerin iletişiminde NRF24L01 modülü çalışması **Şekil 3.3** de görüldüğü gibi 2400 MHz den 2525 MHz kadar 1 MHz ara ile 125 kanalı ile iki yönlü iletişime izin verdiği için tercih edilmiştir. NRF24L01 modülü aynı zamanda 1'e n iletişime izin verdiği için kenar sunucusunun aynı anda birden fazla nesne ile iletişimi mümkün kılmaktadır.



Şekil 3.3: NRF24L01 modülü

Yapılan bu çalışma sayesinde kısıtlı kaynakları olan cihazlar, düşük güç kapasiteli cihaz kenar sunucusu olarak rol verilip esnek, dinamik ve proaktif güvenlik sağlanabildiği gösterilmiştir.

3.4 Kılıç Güvenlik Sistemi Modülleri

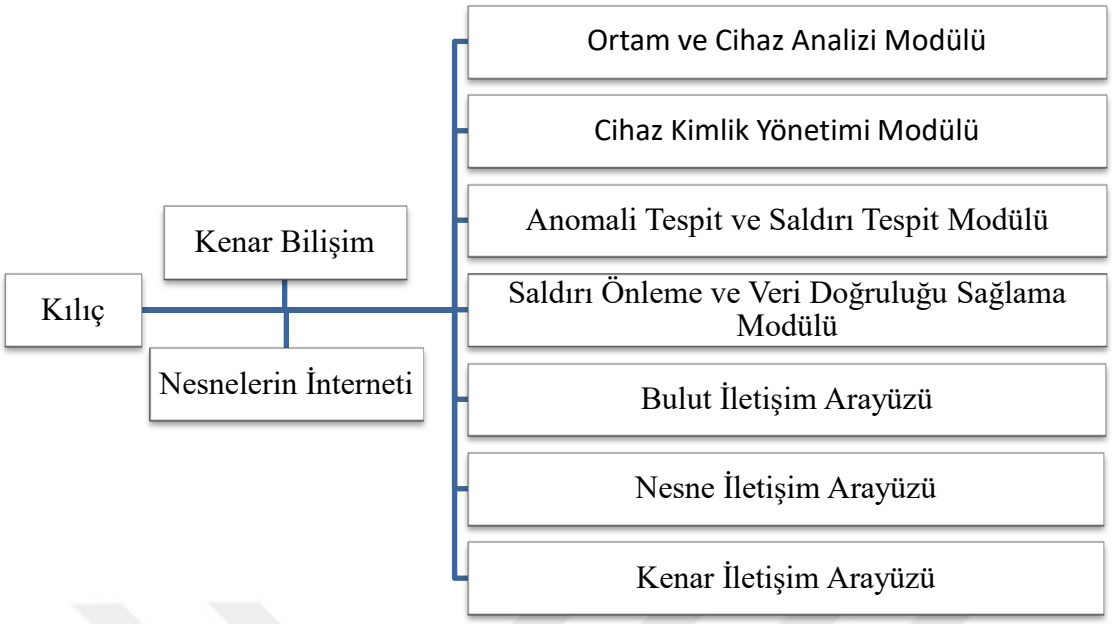
Kılıç uygulaması **Şekil 3.4** de görüleceği üzere yedi modülden oluşmaktadır. Bu modüllerden;

- Ortam ve Cihaz Analizi Modülü,
- Cihaz Kimlik Yönetimi Modülü,
- Anomali Tespit ve Saldırı Tespit Modülü,
- Saldırı Önleme ve Veri Doğruluğu Sağlama Modülü,
- Bulut İletişim Arayüzü,
- Nesne İletişim Arayüzü

olmak üzere altı modül Kenar Bilişim sunucusu üzerinde çalışmaktadır.

- Kenar İletişim Arayüzü

modülü ise nesneler üzerinde çalışmaktadır.



Şekil 3.4: Kılıç Kenar Bilişim Güvenlik Sistemi Modülleri

3.4.1 Ortam ve cihaz analizi modülü

Kılıç Güvenlik Sistemi çalışmaya başlamadan önce ortamı ve cihazın güvenlik analizinin yapılmasını gerekmektedir. Tez kapsamında hazırlanan projenin ilk aşaması cihazların ve çalışma ortamının güvenlik analizi verisinin girilmesiyle başlamaktadır. Bu analizin nasıl yapılacağı bu tez kapsamına alınmamıştır. Bu nedenle girilen analiz verisinin doğru olduğu kabul edilmiştir. Ayrıca Kılıç Güvenlik Sistemi, analiz verilerinde hata olsa bile, sistem çalışırken cihazların davranışlarına göre güvenlik seviyesini değiştirmektedir. Bu analizler, ortam güvenlik analizi ve cihaz güvenlik analizi olarak iki başlıkta incelenmektedir.

Nesnelerin İnterneti cihazlarının saldırılara karşı en belirgin özelliği kısıtlı kaynaklarla çalışmasıdır. Kısıtlı kaynakları verimli kullanmak için doğru ve yerinde analiz faaliyetleri yürütülmesi gerekmektedir. Yapılacak uygulamaların başarıya ulaşması için doğru analizler önemlidir. İyi analiz edilmeyen cihazlar ve ortamlarda yapılan IoT uygulamalarında ya saldırılara karşı önlem alınmamakta ya da yüksek gecikmeye sebep olan algoritma ve teknikler kullanılmaktadır [2].

3.4.1.1 Ortam güvenlik analizi

IoT cihazlarının hayatın içinde her hangi bir ortamda kullanılıyor olması muhtemeldir. IoT cihazının kullanıldığı her ortamın güvenlik ve gizlilik seviyesi farklıdır ve sistemin güvenliğini doğrudan etkilemektedir. Fiziki olarak IoT cihazının çalınması veya donanına zarar gelmesi gibi durumlar söz konusu olabilmektedir. Cihazlar manipülasyona açık olduğu için yangın sensörü yanında yakılacak bir çakmak yanlış alarm verilmesine neden olabilir. IoT cihazının bulunduğu ortam değerlendirilmeli ve geleneksel yöntemlerin dışında tedbirler alınması gerekmektedir.

Çizelge 3.2 de görüleceği üzere, Akıllı Fabrika Kenar Bilişim güvenlik sistemindeki IoT cihazlarının kullanıldığı ortamlar ve güvenlik seviyelerinin ait tanımlar yapılmıştır. Bu tanımlama yapılırken yapılabilecek saldırı yöntemlerine göre ortamların zafiyetleri göz önüne alınmıştır. Kenar sunucusuna gelen verilerin doğru ve gerçek olması önemlidir. Sistemi yazılımsal ve fiziksel manipülasyonlara karşı koruyabilmek için cihazların ortam bilgisi alınmaktadır. Bu bilgi sayesinde cihazların gönderdiği veri yakın konumdaki diğer düğümlerin ürettiği veri ile karşılaştırılarak doğruluk kontrolü yapılabilecektir.

Çizelge 3.2: Ortamlarının güvenlik analizleri

Ortam / Güvenlik Seviyesi	Düşük Tehdit	Orta Tehdit	Yüksek Tehdit
Bahçe			X
Depo		X	
Üretim Sahası	X		
Ofisler		X	

3.2.1.2 Cihaz güvenlik analizi

IoT uygulamalarında benzer işleri yapan yüzlerce hatta binlerce nesne bulunabilir. Her birisi için ayrı ayrı yapılacak olan analizler tekrar olacaktır. IoT cihazlarının güvenlik analizi yapılırken, cihazlar kullanılma amacına ve ürettikleri verinin önemine göre sınıflandırılır. Her cihaz için analiz yapılması yerine görev temelli güvenlik sınıfları oluşturularak güvenlik analizi hızlandırılır. Akıllı Fabrikada ortamın sıcaklık değerlerini ölçen IoT cihazları Ortam Sıcaklık Sensörleri diye

sınıflandırılırken, fabrikadaki kazanların sıcaklıklarını ölçen sensörlere Kazan Sıcaklık Sensörü diye sınıflandırılır. Sistem çalışırken güvenlik sınıfları ve bu sınıflara dâhil edilen cihazlarda esnek olarak değiştirilebilmektedir.

Her saniye yeni veri üreten bir cihazın anlık verisinin doğruluğu çok önemli olmamaktadır. Bunun yanında bazı IoT cihazlarının ürettiği verilerin doğru ve gerçek olması önemliyken, verinin gizliliği önemli olmayabilir. Kılıç Güvenlik Sistemi, bu belirlenen analiz verileri sayesinde, her cihaz için farklı güvenlik seviyeleri seçebilecektir. **Çizelge 3.3'**deki örnek kayıtlarda görüleceği gibi, cihaz sınıfları belirlendikten sonra; hedef güvenlik seviyeleri, veri doğruluk hassasiyeti ve veri gizliliği seviyesine ait yapılan analizler kenar sunucusu üzerindeki Kılıç Güvenlik Sistemi tanımlanmaktadır. Kılıç, IoT uygulamasına gelebilecek tehdit ve saldırı durumunda yine bu parametreler üzerinden güvenlik seviyelerini değiştirerek ortamın güvenliğini sağlamaya çalışır.

Çizelge 3.3: Cihaz sınıflarının güvenlik analizleri

Cihaz Sınıfı / Kriterler	Hedef Güvenlik Seviyesi	Veri Doğruluğu Hassasiyet	Veri Gizliği Seviyesi	Yayın Sonlandırma
Ortam Sıcaklık Sensörü	Orta	Düşük	Orta	Var
Döküm Sıcaklık Sensörü	Orta	Yüksek	Orta	Yok
Basınç Sensörleri	Düşük	Düşük	Düşük	Yok
Nem Sensörleri	Düşük	Düşük	Düşük	Var
Üretim cihazları	Yüksek	Yüksek	Yüksek	Yok
Güvenlik Kameraları	Orta	Orta	Orta	Yok

3.4.2 Cihaz kimlik yönetimi modülü

Kenar Bilişim sunucuları kendi üzerine yönlendirilen IoT cihazlarının veri trafiğini işlemektedir. Bu yöntem yerel ağda bile içerisinde birçok güvenlik zafiyetleri bulundurmaktadır. Sahte düğüm, tekrarlama ve aradaki adam gibi saldırı yöntemlerine karşı kenar bilişim savunmasızdır. Bu nedenle Kenar Bilişim sunucuları veri üreten cihazlarını benzersiz olarak tanımaları gerekmektedir.

Kenar Bilişim sunucularının, veri üreten IoT cihazlarını kimliklendirmek için benzersiz anahtar kullanmaları gerekmektedir. IoT cihazları yapılan uygulamalarda

benzersiz cihaz ID ve doğrulama anahtarıyla sisteme kaydedilmesi önemli bir temel güvenlik seviyesi oluşturmaktadır [91]. Yabancı düğümlere karşı benzersiz anahtar etkili olurken tekrarlama saldırılarına karşı zafiyetleri devam etmektedir. Zafiyetlerin yanında kullanıcı doğrulaması performans kayıplarına da yol açmaktadır [91]. Bu nedenle kullanıcı doğrulamasını kullanırken sisteme yük getirmeyecek şekilde esnek tekrarlama saldırılarına karşı güvenli şekilde yapılıdır. Kılıç Güvenlik Sistemi, kimlik doğrulamasını yayına başlarken, periyodik ve rastgele zaman aralıklarında olmak üzere üç farklı uygulamanın güvenlik ve doğrulama ihtiyacına göre yapmaktadır. Kenar Bilişim sunucusu cihaz doğrulama talep ettiğinde IoT cihazı doğrulama bilgilerini göndereceği şekilde programlanmıştır.

Akıllı Fabrika örneğinde IoT cihazları kimlik bilgisinin Kenar Bilişim sunucusuna veri gönderebilmesi için kaydedilmesi gerekmektedir. Kenar Sunucusuna IoT cihazları kaydedilirken benzersiz cihaz ID'si ve doğrulama anahtarının yanında **Çizelge 3.3** ve **Çizelge 3.4** de örnekleriyle verilen;

- Hedef Güvenlik Seviyesi,
- Bulunduğu Ortamın Tehdit Durumu,
- Veri Doğruluğu Hassasiyeti,
- Veri Gizlilik Seviyesi,
- Yayın Sonlandırma Durumu,
- IoT Cihazın Pil Ömrü ve İşlem Gücü,
- IoT Cihazın Ürettiği Verinin Önem Derecesi,
- İletişim Yöntemi,
- Cihaz Doğrulaması için Gizli Anahtarı

gibi özelliklerini **Çizelge 3.4** görüldüğü şekilde daha önceden belirlenen ortam ve cihaz sınıfları kullanılarak cihaz kimlikleri oluşturulmaktadır. Cihaz kimlikleri Kılıç Kenar Bilişim Güvenlik Sisteminin çalışma senaryolarını etkileyen önemli parametrelerdir.

Çizelge 3.4: Kaydedilen Cihaz kimlikleri

Cihaz ID	Cihaz Sınıf	Cihaz Ortam	Pil Ömrü	İşlem Gücü	İletişim	Özel Anahtar
25f25f8d5	Ortam Sıcaklık Sensörü	Depo	Limitsiz	Düşük	Wi-Fi	****
25f26f8d6	Ortam Sıcaklık Sensörü	Ofisler	Limitsiz	Düşük	Wi-Fi	****
26f27f9d7	Döküm Sıcaklık Sensörü	Üretim Sahası	Limitsiz	Düşük	Wi-Fi	****
27f28f0d8	Güvenlik Kameraları	Bahçe	Limitsiz	Orta	Kablolu	****

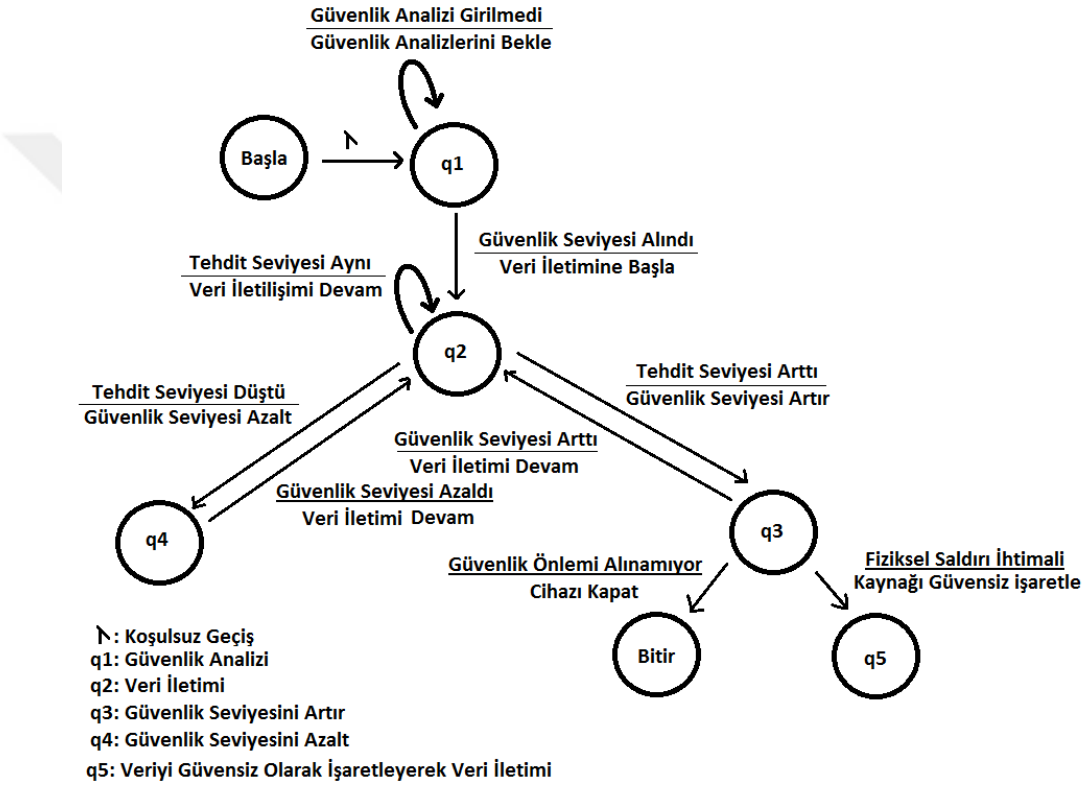
3.4.3 Anomali ve saldırı tespiti modülü

IoT cihazlarının ürettikleri veriler **Şekil 2.2** de görüldüğü gibi Kenar Bilişim sunucusu üzerinden Bulut bilişime iletilmektedir. IoT katmanı üzerindeki cihazların büyük çoğunluğu ürettikleri veriyi kısıtlı kaynak problemleri nedeniyle işleyemez [5]. Yeterli veri işleme ve depolama kaynağı bulunan ve IoT cihazlarına yakınlığı dolayısıyla anomali ve saldırı tespiti Kenar Bilişim sunucusu üzerinde yapılması daha uygundur [2].

Gerçek zamanlı uygulamalar için düşük gecikmeli veri iletişimi gerektiğinden, veri akışı sırasında uygulanan yöntemlerin performansı düşürmemesi gerekmektedir. Bu nedenle Anomali ve Saldırı Tespit Modülü Kılıç sisteminde arka planda veri akışından kopyalanan veriler üzerinde çalışmaktadır. Herhangi bir anormal durum ve saldırı tespit edildiğinde güvenlik bayrakları ile Saldırı Önleme ve Veri Doğruluğu Sağlama Modülü' ne iletmektedir.

Kılıç Kenar Bilişim Güvenlik Sistemi ilk olarak belirlenen cihaz güvenlik sınıflarına girilen analizlere göre çalışmaya başlar. Anomali ve Saldırı Tespiti Modülü gelen veriyi bir süre depolamakta ve analiz etmektedir. Kılıç içerisine tanımlanan yöntemler veri üzerinde bir anormallik tespit ederse **Şekil 3.5** tehdit seviyesini ve türüyle birlikte güvenlik seviyesini artırma bilgisini Saldırı Önleme ve Veri Doğruluğu Sağlama Modülü' e iletir. Alınacak önlemlerden sonra tespit edilen durum da iyileşme olursa bu seviyede çalışmaya devam eder. Eğer olmazsa tekrar güvenlik seviyesi artırımı ister. Ancak cihaz bütün güvenlik seviyelerine rağmen çözülemiyorsa Cihaz Kimliğindeki kapanma durumunu kontrol eder. Eğer kapanmasına izin verilmişse cihazın kapatılması durumuna geçer. Artırılan güvenlik seviyesi IoT cihazının ürettiği verinin gizliliğini ve güvenliğini sağladıktan sonra tehdit seviyesinin azaldığında güvenlik seviyesini düşürülmesine karar vermektedir.

Kılıç gelen verinin gerçek mi yoksa manipüle mi edilmiş olduğuna karar vermek için kural tabanlı ve makine öğrenmesi yöntemlerini kullanılmaktadır. Kural tabanlı yöntemler makine öğrenmesi yöntemlerine göre kolay ve hızlı kodlanmasının yanında daha hızı cevap vermektedir. Aşağıda uygulanan kural tabanlı yöntemler ve nasıl kontrol çalıştığı açıklanmaktadır. Bu yöntemlerin kullanımı da esnek olarak tasarlanması sayesinde yeni yöntem ve tekniklerin eklenmesi için esnek olarak bırakılmıştır.



Şekil 3.5: Anomali ve Saldırı Tespiti Modülü durum diagramı

Verinin kabul aralığı kontrolü: Özellikle sayısal veri üreten IoT cihazlarının ürettikleri veri için uygun olan bir yöntemdir. Kabul aralıkları analiz sırasında sistem yönetici tarafından belirlenerek Kılıç'a kaydedilmektedir. Gelen veriler bu aralıkları geçmesi ya da yaklaşması durumda Kılıç sistemi tehdit seviyesini artırır.

Önceki veri ile karşılaştırma: IoT cihazlarının ürettikleri verinin bir kısmı Kenar Bilişim Sunucusu üzerinde tutulduğu için Kılıç Güvenlik Sistemi önceki verileri anomali veri tespiti için kullanmaktadır. Ayrıca ayrıntılı çalışma yapmak için bulut sisteminden veri beslemesi yapılmaktadır.

Doğal Artış – Azalış kontrolü: IoT cihazları gerçek hayat verisi ürettikleri için doğal fonksiyonlar üzerinde değerlerin artış ya da azalış göstermesi beklenmektedir. Ani yükselmelerde ve düşmelerde Kılıç güvenlik mekanizmalarını devreye sokmaktadır.

Konum temelli veri kontrolü: Her IoT düğümü kenar sunucusuna bağlandığında cihaz bilgileri konum bilgisiyle birlikte Kılıç'a kaydedildiğini Cihaz Kimlik Yönetim Modülünde anlatılmıştı. Aynı ortamda bulunan IoT cihazlarının gönderdiği veriler karşılaştırılarak cihazın fiziksel ya da yazılımsal bir manipülasyona uğrayıp uğramadığı kontrol edilmektedir.

İletilen verinin özet kontrolü: Bir IoT uygulaması içerisindeki bütün düğümler aynı gizlilik seviyesine sahip değildir. Yüksek performans çalışma için bazı önem derecesi düşük verinin saldırganın eline geçmesi son kullanıcıları rahatsız etmemektedir. Ancak kullanıcılar daima doğru veri ile çalışmak istemektedirler. Bu nedenle veri doğruluğunu sağlamak için hash (özetleme) yöntemi kontrolü kullanılmaktadır. IoT cihazlarından ürettikleri veriyi kayıt aşamasında aldığı gizli anahtarıyla birleştirip özetleme fonksiyonundan geçirerek Kenar Bilişim sunucusuna gönderilmektedir. Kenar Bilişim Sunucusu üzerinde Kılıç hash'i tekrar oluşturup verinin doğruluğunu kontrol etmektedir. Hash eşleşmezse verinin yolda değiştirildiği anlaşılacak ve güvenlik seviyesi artırılacaktır.

Birçok farklı saldırı türü için ayrı geliştirme yapmak ve yöntemleri güncel tutmanın maliyetleri yüksek olduğu için makine öğrenmesi yöntemleriyle Kılıç desteklenmiştir. Makine öğrenmesi yöntemlerinden;

- Karar Ağacı,
- Destek Vektör Makinesi,
- K en Yakın Komşu,
- Derin Öğrenme,
- Naive Bayes

algoritmaları kullanılarak IoT cihaz iletişiminde herhangi bir saldırı tehlikesi olup olmadığına karar verilmektedir. Makine Öğrenmesi yöntemleri yüksek işlem gücü ve kaynak olmadığı durumlarda cevap süreleri uzadığı için Kenar Bilişim sunucusunun kaynak durumu göz önüne alınarak Kenar Bilişim üzerinde ya da Bulut Bilişim üzerinde bu işlemler gerçekleştirilebilir. Kılıç için yapılan bu çalışmada Karar Ağacı

algoritması Kenar Sunucusu üzerine çalıştırılmıştır. Destek Vektör Makinesi ve K en Yakın Komşu algoritması Bulut Bilişim üzerinde çalıştırılarak arayüz üzerinde sonuçların alınması uygun görülmüştür. Simülasyon ve test verisiyle uygulanan yöntemler aşağıda verilmiştir.

Karar Ağacı: Uygulanan veri setinde bu yöntem yapılan çalışmadaki performans testlerinde en iyi yöntem olmuştur. Performans testleri iyi olduğu için Kenar Sunucusu üzerinde çalıştırılmaktadır. Karar Ağaçları yüksek doğruluk ve performansıyla veri akışında saldırı tespitinde kullanılmak için öne çıkan bir yöntemdir.

Naive Bayes: Performans testlerinde Karar Ağacından sonra en iyi yöntem olduğu için Kenar üzerinde çalıştırılmaktadır.

Destek Vektör Makinesi (SVM): Ağ üzerinde saldırı tespiti için kullanılan yaygın yöntemlerden biridir. Ancak yapılan sınıflandırmalar için gerekli olan kaynak ihtiyacı Karar Ağacına göre daha fazladır. Bu nedenle Kılıç Güvenlik Sisteminde Bulut Bilişim üzerinde ya da Kenar Bilişim sunucusunun dışında bir kaynakta sonuçların değerlendirilip arayüz ile sisteme dâhil edilmesi uygun görülmüştür.

K en Yakın Komşu (K-NN): Saldırı tespiti yapılırken K En Yakın Komşu benzer kayıtların sınıflandırılması için etkili bir tekniktir. Çoklu sınıf problemlerini çözmek için etkindir. K-NN, yöntemi de dış kaynak üzerinde hesaplanıp Kılıç Güvenlik Sistemine arayüz ile sisteme dâhil edilmesi uygun görülmüştür.

Derin Öğrenme: Performans olarak yapılan testlerde veri seti için en yavaş çalışan yöntem olmuştur. Yöntemin Bulut Bilişim ya da merkezi veri sistemlerinde çalıştırılması uygun olduğu için sisteme arayüz üzerinden dâhil edilmiştir.

Kılıç Güvenlik Sistemi kural tabanlı ve makine öğrenmesi yöntemleriyle birlikte güçlü bir saldırı tespiti modülü ortaya koymuş olmasına karşın farklı yöntemlerden farklı sonuçların gelmesi karmaşıklığı artırmıştır. Saldırı olup olmadığına karar verirken ayrıca birde verilen kararlar arasında bir tercih yapmak durumunda kalmaktadır. Bu karmaşıklığı gidermek için yöntemler arasında klasik çoğulcu demokratik sistem uygulanarak her yöntemden gelen sonuç bir oy kabul edilerek hangi tarafın oyu yüksek çıkarsa o sonuç doğru kabul edilmektedir. Uygulanan yöntemlerin eklenip çıkarılması esnek bırakıldığından eşit sayıda çıkma ihtimalinde saldırı olduğu yöndeki karar geçerli sayılmaktadır. Modül arka planda çalışmasından dolayı Kılıç'ın performansını doğrudan etkilememesine karşın yöntemlerin birinden farklı sürelerde çalışması nedeniyle bayraklar bütün sonuçların gelmesi beklenmeden

anlık duruma göre çalışabilmektedir. Güvenlik seviyesindeki artışın bütün yöntemlerin sonuçları geldikten sonra mı çalışacağı yoksa anlık olarak gelen sonuçlara göre mi çalışacağı analiz modülündeki hedef güvenlik seçimlerine göre karar verilmektedir. Güvenlik ihtiyacı yüksek olan cihazlarda anlık karar verilmesi uygun görülürken güvenlik ihtiyacı düşük performans ihtiyacı yüksek cihazlarda bütün yöntemlerden gelen sonuç değerlendirilerek güvenlik bayrakları değiştirilmektedir.

3.4.4 Saldırı önleme ve veri doğruluğu sağlama modülü

IoT uygulamalarının çalıştıkları cihazlar ve ortamları hatalı veri üretmeye ve dışarıdan kötü niyetli kişilerin saldırılarına karşı zayıftır [5]. IoT cihazlarının büyük çoğunluğu, ürettikleri verileri geleneksel yöntemlerle gizliliklerinin sağlanması için yeterli kapasiteleri yoktur [5-9]. IoT cihazlarında çalışacak hafif ağırlıklı şifreleme yöntemleri geliştirilmiş olsa da IoT cihazları için yeni çalışma yöntemleri geliştirilmesi gerekmektedir [92].

IoT cihazlarının, çalıştığı ortam, tehdit unsurları ve tehdit süresi farklılık göstermektedir. Ortam ve Cihaz Analizi Modülünde girilen analiz verilerine göre Kılıç Kenar Bilişim Güvenlik Sistemi cihaza uygun gizlik ve güvenlik tedbirleri almaktadır. Gerçek zamanlı üretim hatlarında çalışan bir cihazın bütün güvenlik tedbirleriyle birlikte çalışması gerçek zamanlı IoT uygulaması performansını olumsuz etkileyecektir. Bu nedenle Kenar Sunucusu üzerinde Kılıç Güvenlik Sistemi tehdit seviyesine göre IoT cihazına uygun güvenlik tedbirlerini devreye sokulmaktadır.

Saldırı Önleme ve Veri Doğruluğu Sağlama Modülü Kenar Bilişim Sunucusu üzerinde IoT cihazlarının iletişimini etkilemeyecek şekilde arka planda çalışmaktadır. Tespit edilen tehdit durumu Anomali ve Saldırı Tespiti Modülü tarafından iletildikten sonra Kılıç üzerinde belirlenmiş **Çizelge 3.5** de görünen bayrakları değiştirmektedir. Bayrakların veri iletişimini nasıl etkilediği Nesne İletişim Arayüzü bölümünde anlatılacaktır. Kenar Sunucusuna birden fazla IoT cihaz sınıfı bağlı olacağı için tehdit hangi cihaz sınıfı üzerinde tespit edilmişse o sınıfın özellikleri göz önüne alınarak bayraklar belirlenmektedir.

Çizelge 3.5: Kılıç güvenlik sistemi güvenlik bayrakları

Bayrak	**Durum**
Kimlik Doğrulama	0,1
Şifreli Yayın	0,1
Şifreleme Anahtar Uzunluğu	0,32, 64, 80, 96, 128, 192, 168, 256
Şifreleme Yöntemi	-,AES, KLEIN, RC5
Hash Doğrulama	-,MD5, SHA-1, PHOTON-160
Şifresiz Yayın	0,1
Yayın Sonlandır	0,1

Cihaz sınıflarından birine gelen tehdit diğer cihazların güvenlik seviyelerini etkileyebilir ancak cihaz kapasitesine göre seçilecek yöntem **Çizelge 3.6**'daki örnek seçenekler arasında farklılık gösterebilir. Yayının dinlenme tehlikesi varsa Ortam sıcaklık sensörleri sınıfına RC5 yöntemi ve 32 bit anahtar uzunluğu şifreleyerek yayın yapması için bayrakları değiştirirken Kazan Sıcaklık Sensörü Sınıfına RC5 64 yöntemini seçmektedir.

Çizelge 3.6: Kılıç güvenlik sisteminde kullanılan gizlilik ve doğrulama yöntemleri

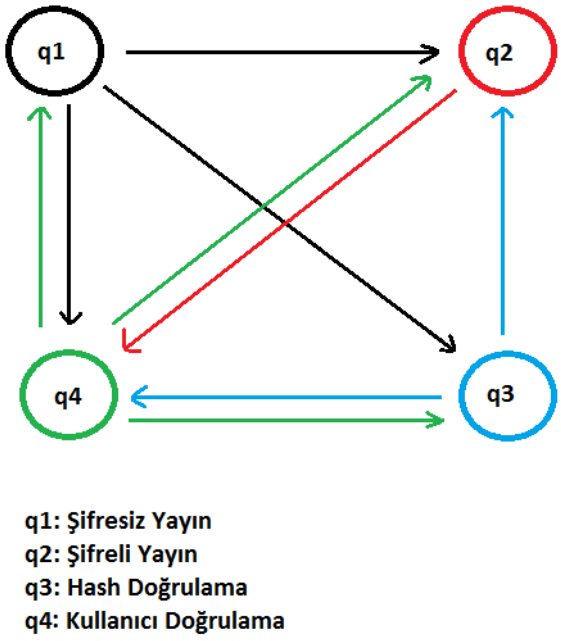
Kimlik Doğrulama	**Şifreli Yayın**	**Hash Doğrulama**
Yayın Başında	AES 128, 192, 256	MD5
Periyodik	KLEIN 64, 80, 96	SHA-1
Rast Gele	RC5 32, 64, 128	PHOTON-160

Kılıç güvenlik sisteminde, güvenlik senaryolarının uygulanma sırası, çalışma zamanında IoT cihazlarından gelen veriye göre belirlenir. Senaryolar arası geçiş yapılırken veri kaybı olamaması için veri akışı geçiş tamamlanana kadar Kılıç iki senaryoyla da veri kabulüne devam eder. Güvenlik, gizlilik ve doğrulama senaryoları arası geçiş kuralları **Şekil 3.6**'da gösterilmiştir. Bu kurallar sayesinde Saldırı Önleme ve Veri Doğruluğu Sağlama Modülü bayraklar arası geçiş yaparken güvenlik seviyesini belirli bir yol izleyerek artırıp azaltır. Örneğin şifreli yayından kullanıcı doğrulama güvenlik senaryosuna geçiş yapılabilir ancak hash kontrol durumuna

geçmesi için önce kullanıcı doğrulaması yapması gerekmektedir. Bu da aşamalı bir güvenlik seviyesi düşürme ve artırma imkânı sağlamıştır.

Kılıç, IoT cihazları üzerinde güvenlik senaryolarını uygularken önlemlerle tehditleri eşleştirmektedir. Her hangi bir tehdit algılandığında tehdidi bertaraf edecek önlemi almaktadır.

Koklama Saldırısı: Bu saldırı türü pasif bir atak çeşidi olup veriyi değiştirmeden yalnızca yayındaki paketleri okumayı hedeflemektedir [8]. Yerel ağda ya da kablosuz yayının ulaştığı mesafeye kadar saldırganın girmesi gerekir. Kılıç güvenlik sisteminin gizli kalması istenen veriyi şifreleyerek göndermesi gerekmektedir. Ancak şifreleme yöntemine ve anahtar uzunluğuna karar verirken verinin gizli kalması gerektiği süre boyunca gizli tutabilecek anahtar uzunluğu ve yöntemi seçilmelidir. Örneğin endüstri için bir sezon gizli kalması önemli olan veriyi bir asırda kırılamayacak bir yöntemle şifrelemek kaynak israfı olacaktır.



Şekil 3.6: Kılıç güvenlik ve gizlilik senaryoları arası geçiş kuralları

Aradaki adam Saldırısı: Aktif bir saldırı çeşidi olup bu saldırıda saldırgan, IoT düğümü ile Kenar sunucusu arasına girip bütün trafiği üzerinden geçirmeye çalışmaktadır [8]. Bu saldırıda başarılı olan saldırgan paketler üzerinden geçerken içeriğini okuyabilir, içeriğini değiştirebilir. Kılıç güvenlik sistemi bu saldırıya karşı

IoT düğümüne özel hash doğrulama ve şifreli yayın kullanılmaktadır. Kılıç güvenlik sisteminde girilen analiz verisine göre karar verilir. Hash doğrulama veri gizliliğinin değil doğruluğunun önemli olduğu durumlarda kullanılır. Şifreli yayın veri gizliliğinin önemli olduğu durumlarda kullanılır. Kılıç'ın doğru yöntemi seçmesi için analiz girişi yapılırken sistem yöneticisi tarafından verinin önemine göre cihazların güvenlik sınıfları doğru belirlenmelidir.

Tekrarlama Saldırısı: Aktif bir saldırı çeşidi olan Tekrarlama saldırısında saldırgan yayında olan paketleri bir süre dinler ve kaydeder. Daha sonra IoT düğümü veri göndermeyi durdurduğunda Kenar Bilişim sunucusuna bu orijinal paketleri kenara göndermeye başlar. Kılıç bu saldırıya karşı rasgele aralıklarla kullanıcı doğrulaması yaparak tespit etmeye çalışmaktadır.

Sahte Kimlik Saldırısı: Aktif bir saldırı çeşidi olan bu saldırı yönteminde saldırgan IoT cihazının kimliğini kopyalamaktadır. Böyle saldırıları tespit için Kılıç güvenlik sistemi kenar sunucusuna bağlı yakın konumdaki IoT düğümlerinin gönderdiği veri ile karşılaştırarak tespit edilir. Hatalı veri üreten IoT cihazı kimliği güvensiz olarak işaretlenir.

IoTBotNet Oluşturma Saldırısı: Bu Aktif saldırı yöntemleri IoT cihazlarına bulaştırılan virüs ile IoT cihazını uzaktan kumandalı bir saldırı botuna dönüştürme saldırısıdır. Bot'a dönüşen IoT düğümü dış ağda DDos Saldırısı yapmak için kullanılacak hale getirilmektedir [89]. Bu saldırıya karşı hem mimari olarak hem de Kılıç güvenlik sisteminde önlem alınmıştır. Mimaride IoT cihazları yalnızca kenara ulaşacak şekilde tasarlanmıştır. Ancak buna rağmen IoT cihazı dış ağa çıkacak şekilde çalıştığı Kılıç güvenlik sisteminde tespit edilirse IoT cihazı kapatma izini kontrol edilir. Kılıç'a izin verilmişse kapatılmakta ve kapatılma sebebi log'lara yazılmaktadır. Kapatılma izni verilmemişse sadece log kayıtlarına işlenmektedir. Sistem yöneticisi log kayıtlarından cihazın kapanma nedenini görüp müdahale etmesi beklenmektedir.

Fiziksel Saldırı: IoT cihazları insanların ulaşabilecekleri ortamlarda kullanıldığı için, fiziksel saldırılara maruz kalabilmektedirler [74]. Bu saldırı yöntemini yazılımsal olarak önlemek mümkün değildir. Anacak gelen verinin IoT uygulamasına zarar vermemesi için tedbir alınması gerekmektedir. Örneğin bir sıcaklık sensörünün yanında yakılan bir çakmakla halatı yangın uyarısı verdirilebilir.

Böyle bir durumda Kılıç güvenlik sistemi aldığı yazılımsal güvenlik önlemleri yeterli olmadığında fiziksel bir saldırı ihtimali değerlendirilir ve gelen veriyi güvensiz olarak işaretlenir.

3.4.5 Nesne iletişim arayüzü

IoT uygulamaları genellikle ağ heterojen bir yapıya sahiptir [9]. Cihazlar üretilirken farklı teknik özelliklere göre iletişim protokolleri seçilmektedir. Bu heterojen ağı tek evrensel bir kenar cihazına bağlanacağı gibi her iletişim protokolü ya da görev için bir kenar sunucusu tanımlanabilmektedir. Kılıç güvenlik sistemi tek görevli ve evrensel kenar sunucusunda çalışılabilecek şekilde esnek tasarlanmıştır. Akıllı Fabrika örneğinde evrensel Kenar Bilişim sunucusu üzerinde çalışılmıştır.

IoT düğümleri Bluetooth, ZegBee, Wi-Fi gibi birçok farklı iletişim protokolünü destekleyebilmektedir. Nesne İletişim Arayüzü sayesinde bu heterojen yapı bir noktada tekilleştirip Kenar Bilişim sunucusu üzerinden Bulut Bilişimle iletişim kurulmaktadır. Bu sayede Bulut Bilişim sunucusu IoT cihazlarının kullandıkları protokolden bağımsız olarak çalışabilmektedir.

Nesne İletişim Arayüzü, Kenar Katmanı üzerinde çalışan Kılıç güvenlik sisteminin IoT düğümleriyle iletişimini sağlar. Kılıç güvenlik sisteminden alınan çeşitli güvenlik tedbirleri Nesne İletişim Arayüzü üzerinden nesnelere iletilir. Saldırı Tespit Modülü ve Saldırı Önleme Modüllerinin arka planda çalışarak değiştirdikleri bayrak bilgilerine göre çalışır. **Çizelge 3.7**'de görüldüğü gibi bayrak bilgileriyle belirlenen güvenlik senaryoları Nesne iletişim Arayüzü üzerinde uygulanır.

Çizelge 3.7: Bayraklar ve Senaryolardan bazılarının örnek durumu

Tehdit Durumu	**Güvenlik Hedefi**	**Veri Gizlilik Seviyesi**	**Cihaz Doğrulama**	**Şifreli Yayın**	**Veri Doğrulama**	**Şifresiz Yayın**
Düşük	Düşük	Düşük	-	-	-	Var
Düşük	Orta	Düşük	Yayın Başında	-	Var	Var
Düşük	Yüksek	Düşük	Periyodik	KLEIN 64	-	Var
Orta	Düşük	Orta	Yayın Başında	-	Var	Var
Orta	Orta	Orta	Periyodik	80 bit RC5	-	-
Orta	Yüksek	Orta	Rastgele	128 bit AES	Var	-
Yüksek	Düşük	Yüksek	Periyodik	64 Bit KLEIN	Var	Var
Yüksek	Orta	Yüksek	Periyodik	196 bit AES	-	-
Yüksek	Yüksek	Yüksek	Rastgele	256 bit AES	Var	-

Öncelikle bağlanan IoT düğümüyle cihaz doğrulaması gerekli görüldüğünde gelen veriyi tampon (buffer) da bekleterek cihaz doğrulaması yapar. İlgili sınıf için cihaz doğrulaması bayrağı aktif ise doğrulama talebi sınıftaki her cihaz için tekrarlanır. İlk defa kullanıcı doğrulaması yapan cihazların verisi Bulut Bilişime gönderilmez. Önceden doğrulama yapılmışsa veri akışı veri kaybı olmaması için devam eder. IoT cihazından şifreli olarak gelen veri bu modülde başlık bilgisi kontrol edilerek çözülmektedir. Cevap gönderilirken cihazı güvenlik sınıfı şifreleme bayrağını kontrol edip, şifreleme yöntemini ve anahtar uzunluğu başlık bilgisine eklenerek gönderilir. Veri doğrulaması istenen veri akışları için Nesne İletişim Arayüzü veri doğrulaması yapmaz. Bu tür doğrulama işlemlerini arka planda çalışan Anomali ve Saldırı Tespit Modülü tarafından yapılmaktadır. Veri doğrulaması yapılmadığında ilgili cihazdan gelen veriyi güvensiz olarak işaretlenerek güvenlik seviyesi artırılmaktadır. Nesne İletişim Arayüzü nesnelerden gelen veri şifrelenmişse çözüp veriyi kenar sunucusunun diğer işleme mekanizmalarına iletilmesinde görevlidir. Yine kenar sunucusu üzerinden gelen veri, komutları ve Kılıç' bayraklarına göre iletişim şeklini nesneye iletilmesinden sorumludur.

3.4.6 Bulut iletişim arayüzü

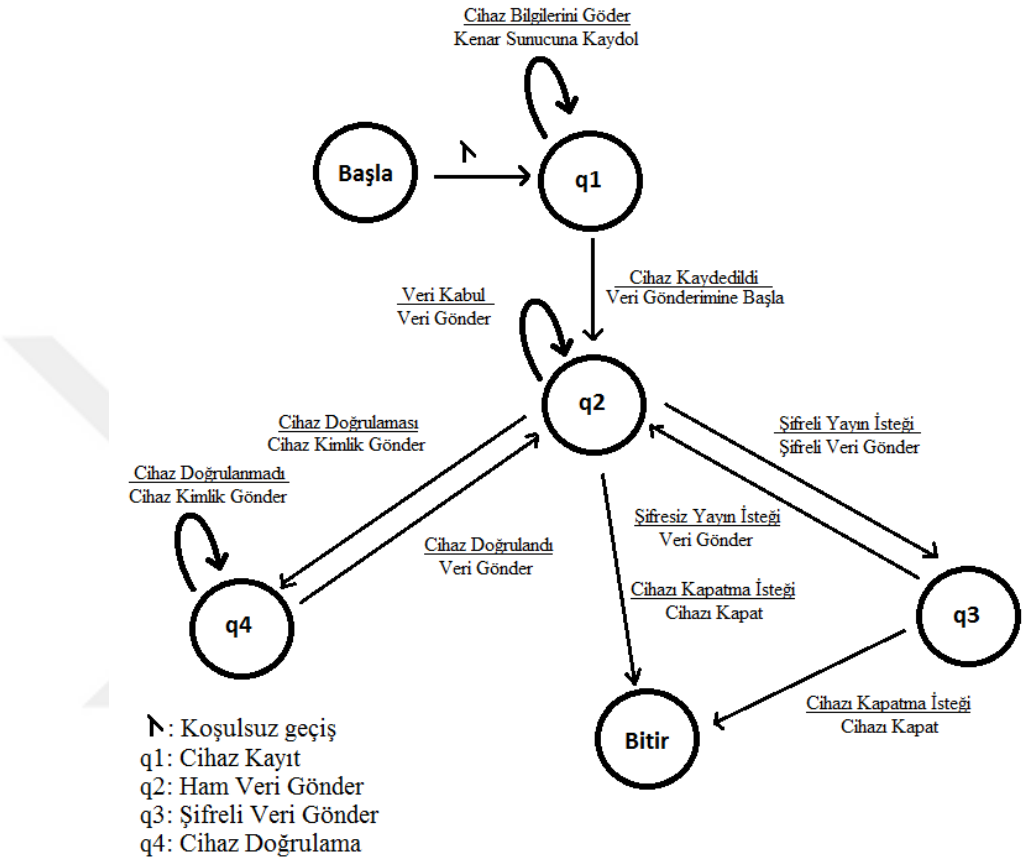
IoT cihazlarından Kenar Bilişim sunucusuna kadar gelen verinin güvenliği ve doğruluğu sağlanmış ve işlenerek Bulut sunucusuna iletilmesi gerekmektedir. Bulut sunucularına gönderilen veriler mevcut İnternet alt yapısını kullanacağı için gizli ve güvenli iletilmesi önemlidir. Kaynak kısıtları olmayan bu Kenar Katmanı ve Bulut Katmanı arasındaki iletişimin güvenli ve gizli olmasını Bulut İletişim Katmanı sağlamaktadır. Kenar sunucularıyla Bulut sistemleri arasındaki mevcut geleneksel doğrulama mekanizmaları kullanılarak bağlantı yapılmakta ve güvenli veri iletimi mevcut TSL ve HTTPS güvenli veri transfer protokolleri üzerinden yapılmaktadır.
Kenar Bilişim sunucusu tarafından, IoT cihazlarının çalışma zamanı ağ gecikmesinden etkilenmemeleri için ihtiyaç duyacakları veri Bulut Bilişim' den önceden talep edilecektir. Önceden hazırlanan veri ayrıca Kenar Bilişim sunucusunun güvenlik süreçlerinde kullanılmaktadır. Kenar üzerinde kara liste ya da beyaz liste gibi bilgilerin tutulması için yeterli hafıza bulunmadığında Bulut Bilişim tarafına kaydedilerek kontroller bulut üzerinden yapılmaktadır. Kenar üzerinde performans ve işlem gücü nedeniyle çalıştırılmayan makine öğrenmesi yöntemleri de bulut veya merkezi veri sistemlerinde çalıştırıldıktan sonra sonuçlar Bulut İletişim Arayüzü üzerinden Kenar sunucusuna geri besleme yapılmaktadır.

3.4.7 Kenar iletişim arayüzü

Kılıç, dinamik güvenlik seviyeleri oluşturmak için bütün katmanlarla birlikte iletişim halinde olması önemlidir. Ancak yalnızca veri üretmeye ve göndermeye programlanmış bir nesne ile bu talebin karşılanamayacağı anlaşılmıştır. Bu nedenle IoT cihazlarının Kenar Bilişim sunucusuyla iletişim kurmasını sağlayan istenen gizlilik ve güvenlik gerekliliklerine cevap veren Kenar İletişim Arayüzü geliştirilmiştir. Bu arayüz Kılıç güvenlik sisteminin nesne üzerinde çalışan tek modülüdür.

Kenar İletişim Arayüzü Kenar sunucusuyla iletişimi sağlarken yalnızca kendisinden istenen yöntemlere göre iletişim şeklini değiştirmektedir. **Şekil 3.7** da görüldüğü gibi Nesne İletişim Arayüzün'den gelen isteklere göre cihaz kimlik doğrulama bilgilerini göndermek, nesnenin ürettiği veriyi istenen yöntem ve anahtar uzunluğu ile şifreleyip iletmek ve özet üretmekle sorumludur. Kılıç nesneden tespit edilen güvenlik seviyesine göre şifreli veri iletişimi isteyebileceği gibi şifresiz yayın yapılması

talebinde de bulunabilir. Ayrıca daha öncede bahsedilen güvenlik önlemleri alınamayan durumlarda cihazı kapatma isteği de Kılıç tarafından gönderilmektedir. Kapatılma izni olmayan cihazlarda kılıç üzerinden kapatılma isteği gelse bile Kenar İletişim Arayüzü tarafından cihaz kapatılmamaktadır.

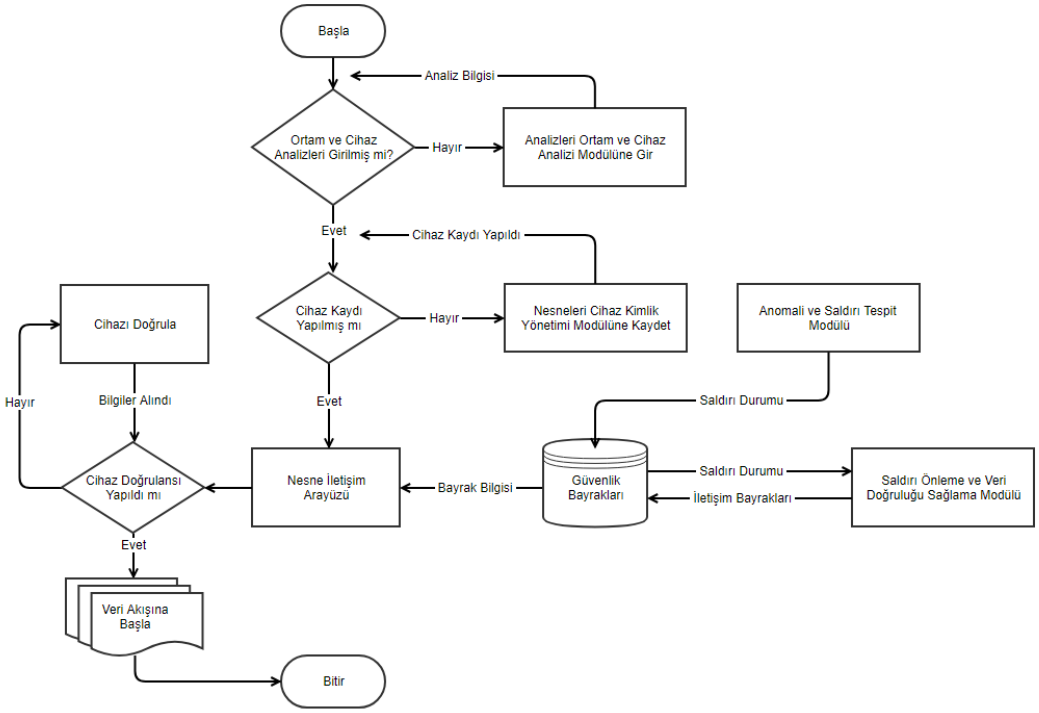


Şekil 3.7: Kenar İletişim Arayüzünün çalışma durumları

3.5 Kılıç Güvenlik Sistemi İşleyişi

Geliştirilen Kılıç güvenlik sistemi Akıllı Fabrika simüle edilerek laboratuvar ortamında test edilmiştir. Kılıç uygulaması üzerinde veri akışının sağlanması için **Şekil 3.8**'de görüldüğü gibi cihaz & ortam analizlerinin ve cihazın kimliğinin oluşturulması gerekmektedir. Kılıç Güvenlik Sistemi Ortam ve Cihaz Analizi Modülüne, ortamlar ve cihaz güvenlik sınıfları tanımlanarak analizleri eklenmektedir. Ortamlar değerlendirilirken, üretim hatları saldırı ihtimali daha düşük ortamlar olarak kabul edilirken bahçe ve toplantı salonlarının olduğu ofis ortamları saldırı ihtimali yüksek alanlar olarak kaydedilmiştir. Cihaz sınıflarının tanımlanması ve analizi yapılırken; pil ömürleri, işlem güçleri ve ürettikleri verinin hedef güvenlik

seviyeleri belirlenir. Örneğin üretim hattında kullanılan sensörlerin hedef gizliliği yüksek, ortam durumunu ölçen sensörlerin verilerin hedef gizliliği düşük olarak belirlenmektedir. Analiz verisi Kılıç'a girildikten sonra Cihaz Kimlik Yönetimi Modülüne cihazlar benzersiz ID, cihaz sınıfı, ortam sınıfı ve diğer bilgileri kaydedilmektedir.

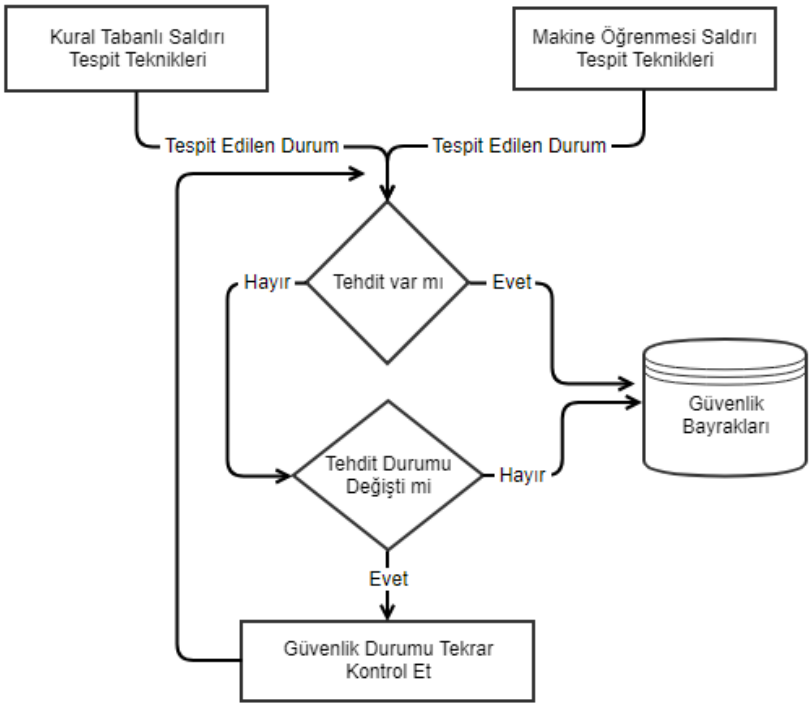


Şekil 3.8: Kılıç Modülleri arası veri akışı

Kılıç Güvenlik Sistemi **Şekil 3.8** deki akışa göre analiz durumlarını kontrol ederek sistemde veri akışını başlatmaktadır. Kılıç içerisinde şifresiz yayın, cihaz doğrulama, özet (hash) doğrulama ve verileri seçilen yöntem ve anahtar uzunluğunda şifreli yayın olmak üzere dört fonksiyonu vardır. Sistem yöneticisinin analizine ve hedef güvenlik seviyesine göre bu dört yöntemi Saldırı Önleme ve Veri Doğruluğu Sağlama Modülünün belirlediği senaryoları Kılıç otom olarak uygulamaktadır. Kılıç, arka planda çalışan bölüm 3.4.4 de anlatılan Anomali ve Saldırı Tespit Modülü tarafından tespit edilen durumlar güvenlik bayrakları değiştirilerek diğer modüllere bilgi verilmekte bölüm 3.4.5 de anlatılan Saldırı Önleme ve Veri Doğruluğu Sağlama Modülü saldırı bayraklarına göre **Çizelge 3.6'**de verilen gizlilik ve doğrulama yöntemleri bayraklarını değiştirerek hangi tehlike senaryolarında nasıl davranacağını

belirlemektedir. Güvenlik durumları ve uygulanan senaryolar **Çizelge 3.7** üzerindeki örnek kayıtlarda gösterilmektedir. Bölüm 3.4.5'de anlatılan Nesne İletişim Arayüzü bayraklarda olan değişiklikler nesnelere iletilerek belirlenen güvenlik seviyesinin icrası gerçekleştirilir. Bu sayede Kılıç üzerindeki güvenlik yöntemleri savunma araçları olarak tehdidi önlemek ve veri doğruluğunu sağlamak için kullanmaktadır.
Kılıç Güvenlik Sistemi yürürlükte olan senaryoya göre, veri akışı başladığında, periyodik olarak veya rastgele sürelerde olmak üzere üç farklı şekilde cihaz doğrulaması yapılmaktadır. Cihaz kaydında gizlilik seviyesine göre veri belirlenen yöntem ve anahtar uzunluğuyla haberleşme şifreli olarak yapılabilir. IoT cihazından Kenara gelen verinin doğruluğu hash kontrolleri ile sağlanabilir. Kılıç bu seçimleri analizlere ve çalışma zamanında algıladığı tehditlere göre yapmaktadır. Kılıç, Bulut Bilişim'e göndermesi ve alması gereken nesnelerin ve kendi süreçlerinde kullandığı veri iletişimi için Bulut İletişim Arayüzünü kullanmaktadır.

Anomali ve Saldırı Tespit Modülündeki kural tabanlı ve makine öğrenmesi yönetmeleriyle saldırı tespiti **Şekil 3.9** da görüldüğü gibi arka planda çalışarak gerçekleşmektedir. Saldırı tehdidi tespit edildiğinde güvenlik bayrakları üzerinden diğer modüllere iletilmektedir. Saldırı tespiti için kullanılan makine öğrenmesi yöntemlerinin Kılıç, kenar sunucusu ve Bulut Bilişim üzerinde eğitilmesi ve saldırı tespit sonuçların karşılaştırılmasını anlamak için Tekrarlama Saldırısı, Aykırı Değer Saldırısı ve Fiziksel Saldırı olmak üzere üç farklı saldırı senaryosu ele alınmaktadır. Bu saldırılara karşı, kural tabanlı yöntemler ve makine öğrenmesi yöntemlerinden; Karar Ağacı, Destek Vektör Makinesi, K en Yakın Komşu, Derin Öğrenme ve Naive Bayes yöntemlerin performans ve doğruluk karşılaştırılması Sonuç ve Tartışma bölümünde ele alınmaktadır.



Şekil 3.9: Anomali ve Saldırı Tespit Modülü

Öncelikle nesnelerden gelen iletişim verisi Kenar Bilişim sunucusu üzerinde Anomali ve Saldırı Tespit Modülü tarafından arka planda etiketlenmektedir. Bu etiketleme aşaması iletişim verisi için makine öğrenmesi yöntemlerinde daha kesin sonuç almak için yönteme girmeden önce ön işlem olarak yapılmaktadır. Bu aşamada makine öğrenmesi yöntemlerinde kullanılan kayıtların özellikleri **Çizelge 3.8** de tamamlanmaktadır.

Çizelge 3.8: Makine öğrenmesi yöntemlerinde kullanılan özellikler ve açıklamaları

Özellik Adı	**Açıklama**	**Tipi**
Sensör Değeri	Sayısal değer öğreten IoT Cihazlarının ürettikleri veri	Sayısal
Değişim Miktarı	Önceki gönderilen kayıtla arasında oluşan sayısal fark	Sayısal
Konum Ortalaması Sapması	Yakın konumlarda olan cihazların ürettikleri değerlerin ortalamasından gelen değerin sapması	Sayısal
Değişim Yönü	Sensör verisinin büyüme ve küçülme yönü	Ayrık
Ağ Gecikmesi	IoT cihazından çıktıktan sonra Kenar sunucusuna ulaşana kadar geçen süre	Sayısal
Özet Doğruluğu	IoT cihazından çıkarken üzerine eklenmiş özetin Kenara geldiğinde hesaplanıp karşılaştırıldığında eşleşme durumu	Ayrık
Cihaz Doğrulama	IoT cihazının veri gönderirken Kenar sunucusuna kullanıcı doğrulaması yapıp yapmadığı	Ayrık
Atak Türü	Yapılan siber atağın türü	Ayık

Kenar sunucusuna gelen iletişim verisinde bütün özellikleri içermez. Örneğin önceki kayıtla arasındaki değişim yönü IoT nesnesi üzerinde hesaplanabilir bir değer olmasına karşın kenar sunucusunda hesaplanması için bırakılmıştır. Bu hem IoT düğümünün işlem gücünden ve güç tüketiminden tasarruf sağlarken hem de iletilen verinin boyunu azaltmaktadır. Ağ gecikmesi ve özet doğruluğu gibi nesnenin hesaplaması mümkün olmayan verilerde saldırı tespitinde etkili özellikler olduğu için Kenar Bilişim sunucusunda iletişim verisine eklenmektedir. İletişim verisi **Çizelge 3.9** de görülen son halini aldıktan sonra uygulanan anomali ve saldırı tespit yöntemlerinde işlenmektedir. **Çizelge 3.9** de gösterilen Saldırı durumlarında Cihazın derece cinsinden ölçtüğü "Sensör Değeri", sensörden gelen önceki değerden farkını gösteren "Değişim Miktarı", benzer konumda bulunan diğer sensörlerle ortalama değeri "Konum Ortalaması Sapması", önceki gelen değere göre değişim artış/azalış olarak gösteren "Değişim Yönü", cihazın veriyi üretip şifreleme ve özet gecikmesi dâhil ağ üzerinde gecikmeyi gösteren "Ağ Gecikmesi", kenar üzerine veri ulaştıktan sonra özet tekrar doğruluğu bilgisi "Özet Doğruluğu", veri iletiminden önce cihazdan doğrulama isteği durumunu gösteren "Cihaz Doğrulama", saldırı yapılmışsa hangi tür olduğunu gösteren "Atak" özelliği öğrenme veri setlerinde etiketlenmiş olarak verilirken testlerde tespit ve tahmin edilmeye çalışılan özelliktir.

Çizelge 3.9: IoT cihaz etiketlenmiş iletişim veri örneği

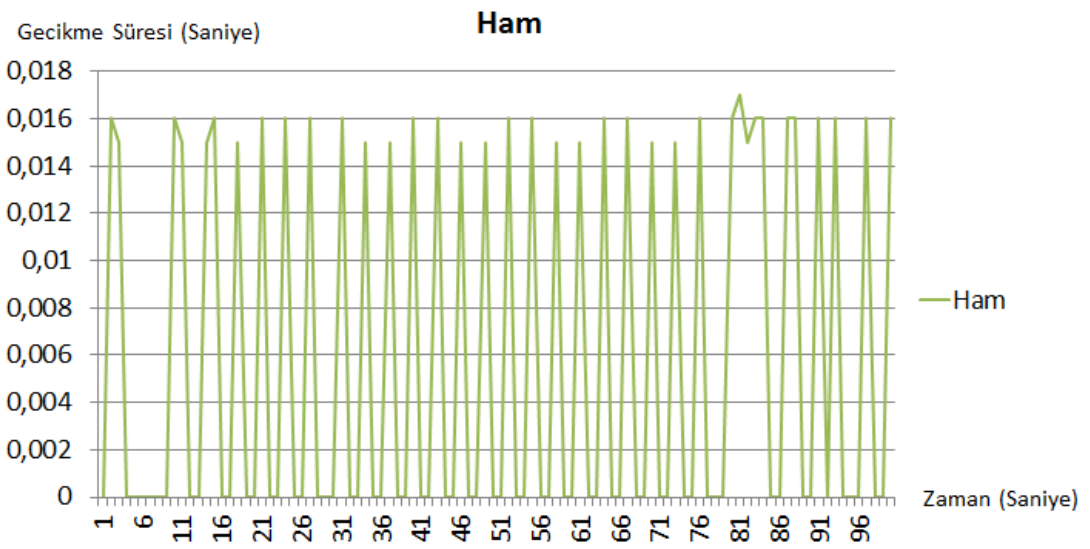
Saldırı Türleri / Özellikler	Sensör Değeri (Derece)	Değişim Miktarı	Konum Ortalaması Sapması	Değişim Yönü	Ağ Gecikmesi (Saniye)	Özet Doğruluğu	Cihaz Doğrulama	Atak Türü
Normal Kayıt	19.58453	0.913501	0.13050	Artı	0.002128	Doğru	Yapıldı	Normal
Aykırı Değer Saldırısı	7.502783	1.319656	0. 10956	Artı	0.058732	Yanlış	Yapıldı	MIM
Tekrarlama Saldırısı	0.088989	0.724539	0.09309	Eksi	0.3038773	Doğru	Yapıldı	Tekrar
Fiziksel Saldırı	21.26885	0.915598	0.49598	Stabil	0.0999529	Doğru	Yapıldı	Fiziksel

4. SONUÇLAR ve TARTIŞMA

Tez Kapsamında geliştirilen Kılıç Güvenlik Sistemi laboratuvar ortamında gerçekleştirilen Akıllı Fabrika örneği üzerinde bir çalışma yapılmıştır. Akıllı Fabrika ofis ortamlarında ve üretim hatlarında Döküm Sıcaklık Sensörü ve Ortam Sıcaklık Sensörü gibi birçok IoT cihazı bulunmaktadır. Akıllı Fabrikada bulunan üretim cihazları gerçek zamanlı IoT uygulamasının bir parçası olduğu için gecikme süreleri oldukça önemlidir. Bulut Bilişim ile haberleşirken kullanılan şifrelendirme algoritması ve İnternet'teki ağ gecikme süreleri, nesnelerin Kenar Bilişim sunucusu üzerinden haberleşmesi sayesinde çözülmektedir. Güvenlik ve gizlilik kaygıları için Kılıç Kenar Bilişim Güvenlik Sistemi tehdit durumuna göre değişken güvenlik seviyesi uygulandığı için Kılıç IoT uygulamasının çalışma performansına asgari etki ederek gelişmesi sağlanmaktadır.

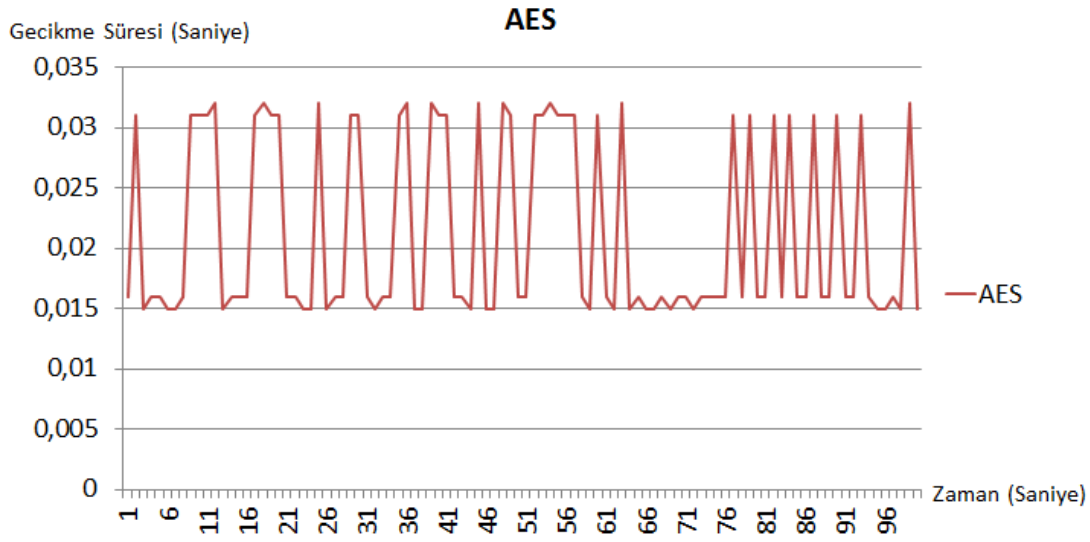
Geliştirilen Kılıç yapılan çalışmada yeni proaktif bir yaklaşımla değişken güvenlik seviyesi yaygın güvenlik algoritmaları üzerinden sağlanmaktadır. Bu yöntemler hafif şifreleme algoritmaları, cihaz doğrulama, hash doğrulama ve şifresiz yayındır. Kılıç'ın dinamik güvenlik yaklaşımı, üzerinde kullanılan şifreleme yöntemleri tek başına uygulandığındaki performans sonuçları karşılaştırılmıştır. Kılıç'ın nasıl çalıştığını daha iyi göstermek için birkaç tipik senaryo sunulmaktadır. Bu senaryolar, Akıllı Fabrikadaki güvenlik ve gizlilikle ilgili tipik bazı vakaları göstermektedir.

Normal İletişim Senaryosu: Sıcaklık sensörleri periyodik olarak veri üretip bu verileri Kenar Bilişim sunucusuna göndermektedir. Kenar Bilişim sunucusu üzerinde bu sıcaklıkların dakikalık ortalamaları alınıp bulut sistemlerine gönderilmektedir. Kılıç Güvenlik Sistemi **Çizelge 3.2** de görülen ortam sınıfları ve **Çizelge 3.3** de görülen cihaz sınıfları kullanılarak yapılan analizlerine göre çalışmaya başlamıştır. Veri iletimi boyunca herhangi bir saldırı yapılmamıştır. **Şekil 4.1** de görüleceği üzere ham veri iletimi yapıldığında sadece ağ gecikmesi sistemin çalışmasına etki etmektedir. Şifreleme ya da doğrulama gibi bir önlem alınmadığı için dinleme, araya girme ve tekrarlama saldırılarına karşı zayıftır.



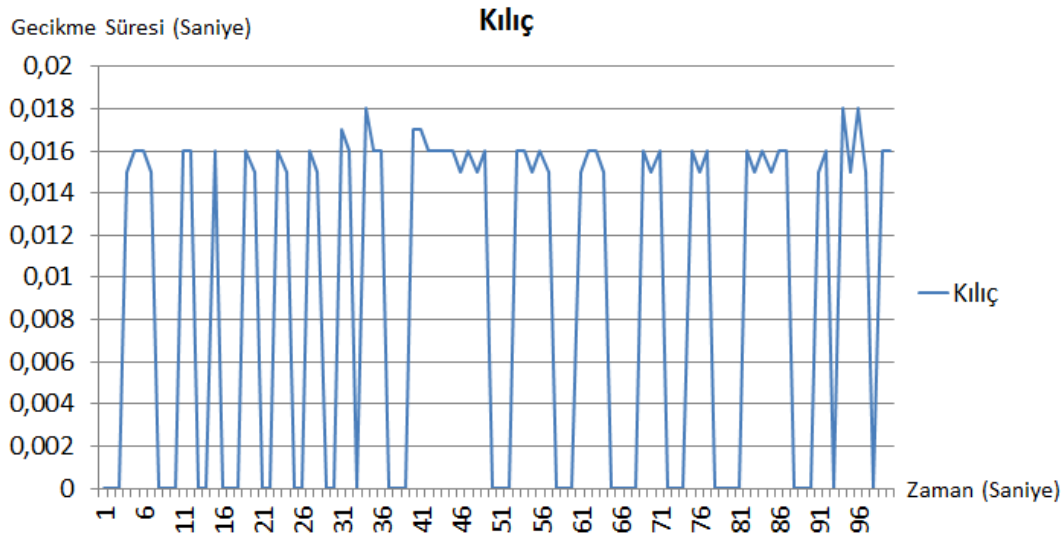
Şekil 4.1: Normal veri seti için şifreleme güvenlik işlemi yapılmadan ham veri iletişiminde ağ gecikme süreleri

Normal iletişim senaryosunda klasik güvenli şifreli iletişimi test etmek için AES hafif şifreleme metodu kullanıldığında **Şekil 4.2** de görüleceği üzere Kenar sunucusu ile IoT düğümünün haberleşmesindeki gecikmeye ağ gecikmesinin yanında algoritmanın neden olduğu şifreme ve şifre çözme zamanı iletişim gecikmesine eklendiği görülmektedir. Gerçek zamanlı (real-time) bir IoT uygulaması için servis kalitesi problemlerine neden olmaktadır. Bu yöntem ağ dinleme ve aradaki adam saldırısına karşı koruma sağlasa da tekrarlama saldırısına karşı zayıftır.



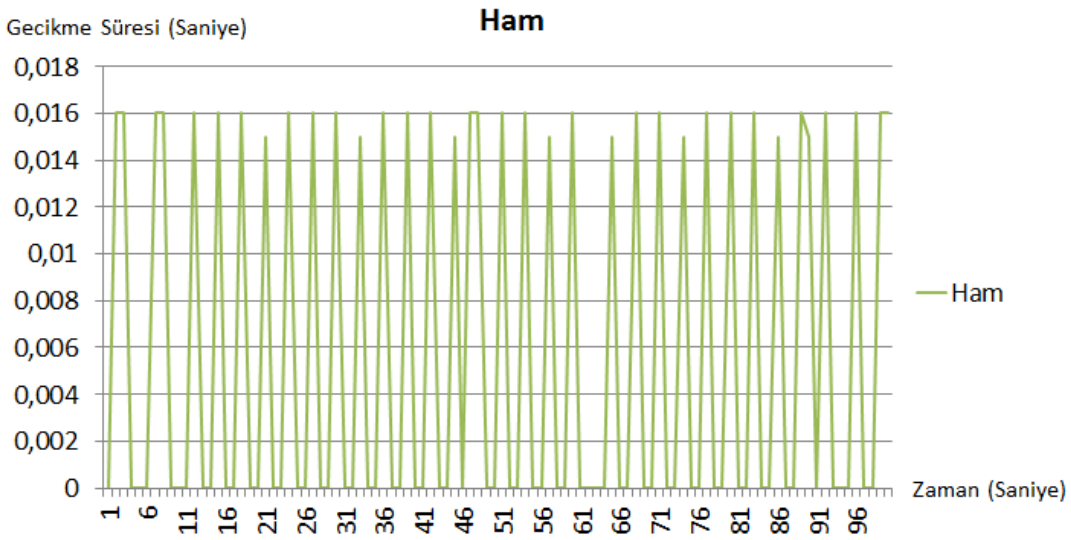
Şekil 4.2: Normal veri seti için hafif AES şifreleme işlemi yapıldığındaki ağ gecikme süreleri

Kılıç Güvenlik Sistemi aynı veri seti ile çalıştırıldığında **Şekil 4.3** görüldüğü gibi güvenlik kontrolleri yapıldığı anlarda iletişimde gecikmeye neden olsa da saldırı tespit edilmediğinden dolayı gecikmesi düşük yöntemlerle iletişimin devam etmesi sağlanmıştır. **Şekil 4.3'**de görülen iletişimdeki 5, 10, 20, 25 gibi saniyelerdeki gecikmeler periyodik cihaz doğrulamasından kaynaklanmaktadır ve 40 50, 80 saniyelerdeki gibi uzun gecikmeler ise periyodik şifreli yayın ve özet ile güvenlik kontrolleri yapıldığı için gecikmelere neden olmaktadır. Sadece şifreli yayın yapılmasına kıyasla daha dinamik güvenlik uygulanması Kılıç iletişim gecikmesinde daha iyi sonuçlar elde edilmiştir. Devamlı şifreleme ve cihaz doğrulaması yapmadığı için düşük gecikmeyle güvenli iletişim yapılmasını sağlamıştır. Kılıç, IoT uygulama çalışırken uyguladığı güvenlik metotlarıyla ağ dinleme, tekrarlama ve aradaki adam saldırılarına karşı önlem almaktadır.

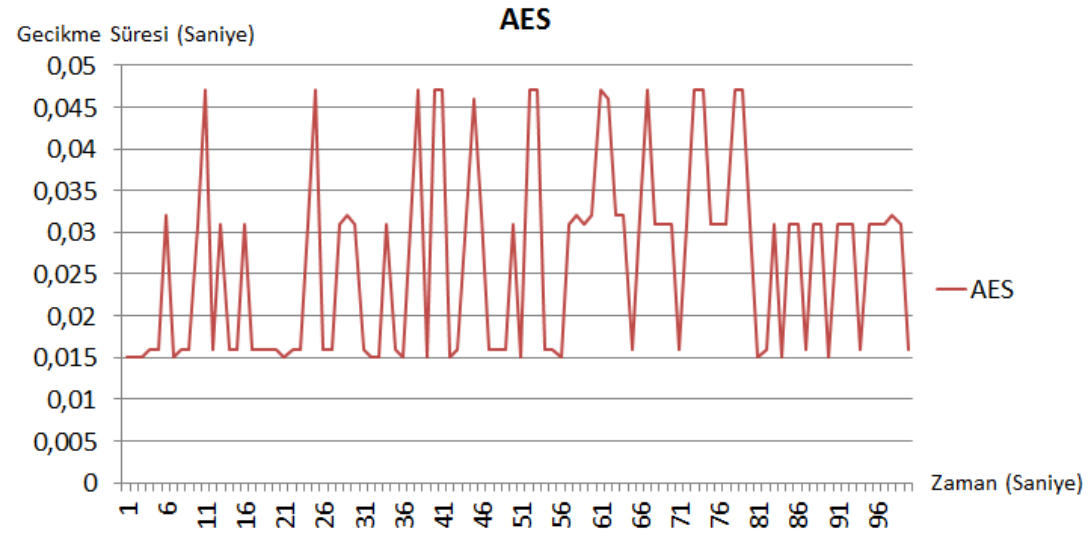


Şekil 4.3: Normal veri üzerinde Kılıç'ın iletişim gecikmesine etkisi

Saldırı altındaki İletişim Senaryosu: İkinci çalışmada hazırlanmış olan senaryoda IoT cihazları 50. Saniyeden sonra saldırı almışlardır. Saldırı tespiti yapılıncaya kadar Kılıç Güvenlik Sistemi hedef güvenlik seviyesini sağlamak için analizde belirlenmiş olan yöntemleri kullanmıştır. **Şekil 4.4** ham veri iletişimi ve **Şekil 4.5** şifreli veri iletişim yöntemi kullanıldığında saldırıya karşı proaktif bir yaklaşım olmadığı için küçük değişimlerin dışında mevcut gecikmeye bir etkisi olmamıştır.



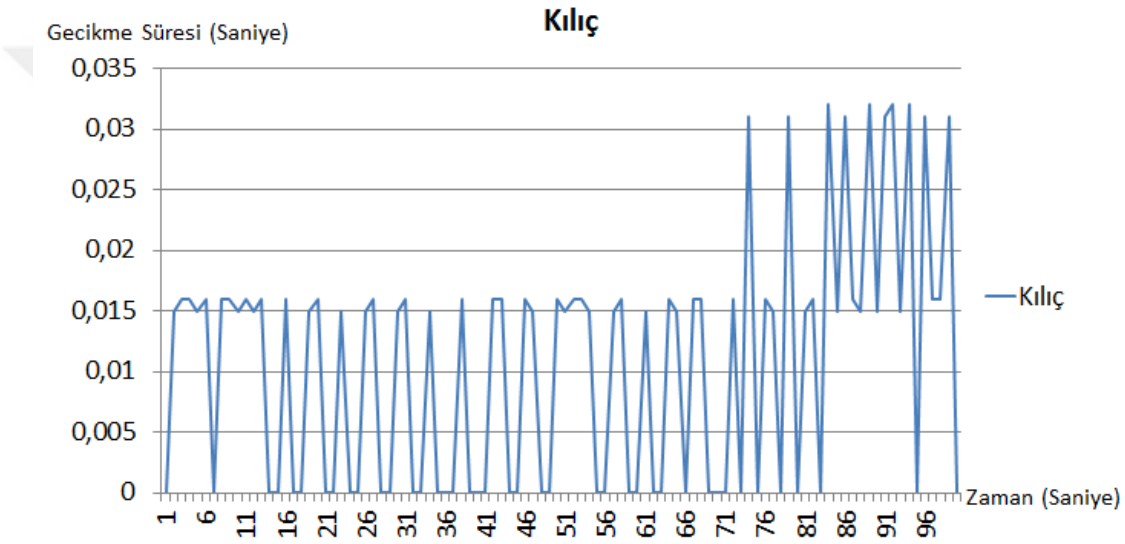
Şekil 4.4: Saldırı veri seti için şifreleme güvenlik işlemi yapılmadan ham veri iletişiminde ağ gecikme süreleri



Şekil 4.5: Saldırı veri seti için hafif AES şifreleme işlemi yapıldığındaki ağ gecikme süreleri

Kılıç Güvenlik Sistemi'nde ise **Şekil 4.6** de görüleceği gibi diğer yöntemlerin aksine görüleceği üzere 50. saniyeden önce periyodik kontroller yaparak 50. saniyeden sonra saldırı tespiti yapılıp kullanıcı doğrulaması ve şifreli yayına geçiş sağlanmıştır. Güvenlik seviyeleri geçişleri arasındaki **Şekil 3.6** da görülen kurallar sebebiyle aşamalı bir geçiş olduğu için veri akışında olan gecikme daha sonra yansımıştır. Önce cihaz doğrulaması sonra veri doğrulama daha sonra düşük anahtarlı şifreleme

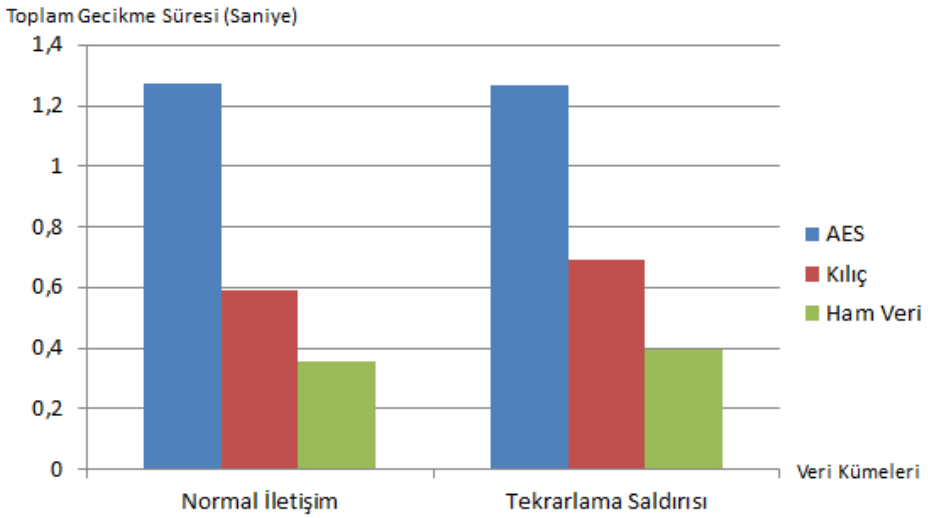
yöntemleri kullanılarak aşama aşama güvenlik seviyesi artırılmıştır. IoT uygulaması anormal veri üretmeye devam etmesi sağlandığı için Kılıç iletişimin güvenlik seviyesini devamlı artırılmıştır. 55. Saniyeden sonra güvenlik seviyesi artırıldığında bir süre o seviye çalışan Kılıç Güvenlik Sistemi verinin nesneden gelen gerçek veri olduğuna karar verdikten sonra bekleme sürelerini düşürmek için güvenlik seviyesini düşürmüştür. Ancak Kılıç gönderilen verinin makine öğrenmesi yöntemlerle saldırı olduğunu tespit edince sistemin tekrar hızlı bir güvenlik seviyesini artırması sağlanmıştır. Devamlı anormal veride bile verinin cihazdan kaynaklandığını tespit edilip sonra güvenlik seviyesini düşürmesi nedenle Kılıç geleneksel yöntemlere göre çok daha az gecikmeyle daha performanslı güvenlik sağlamaktadır.



Şekil 4.6: Normal veri seti üzerinde Kılıç'ın iletişim gecikmesine etkisi

Sonuç olarak yapılan iki farklı senaryoda üç farklı yöntem göz önüne alındığında başta sadece ham veri iletişimi ve AES şifreleme uygulanmıştır. Daha sonra Kılıç Güvenlik Sistemi aynı veri seti için çalıştırılarak sonuçları karşılaştırılmıştır. Normal veri setinde **Şekil 4.7** de görüleceği üzere Kılıç Güvenlik Sistemi, hiçbir güvenlik önlemi alınmayan ham veri trafiğine kıyasla %59,74 daha fazla gecikme ortaya çıkarır. Çünkü Kılıç sisteminin periyodik kontrolleri gecikmeye neden olur. Ancak sürekli şifreli yayının yaklaşık yarısı kadar %46,39'u kadar gecikmeye neden olmuştur. Çünkü Kılıç periyodik kontrollerde güvenlik tehlikesi bulunmadığında güvenlik seviyesini düşürterek performansı iyileştirir. Saldırı veri seti uygulandığında ise ham veri trafiğine kıyasla ortalama %75.69'u kadar fazla gecikmeye neden olmuştur önceki kıyastan kötü sonuç vermesinin nedeni Kılıç

tehlike tespit ettiği için güvenliği artırmış ve daha fazla gecikmeye sebep olan güvenlik algoritmalarını kullanmıştır. Şifreli yayına göre kıyaslandığında Kılıç yalnızca tehlike olan durumda güvenlik önlemlerini kullandığı için şifreli yayında oluşan gecikenin %54,81'i kadar gecikmeye neden olmaktadır. Kılıç Güvenlik Sistemi test sonuçlara göre gecikme sürelerinin önemli olduğu sistemlerde servis kalitesini artırmak için güvenlik sistemi olarak kullanılabileceğini göstermektedir.

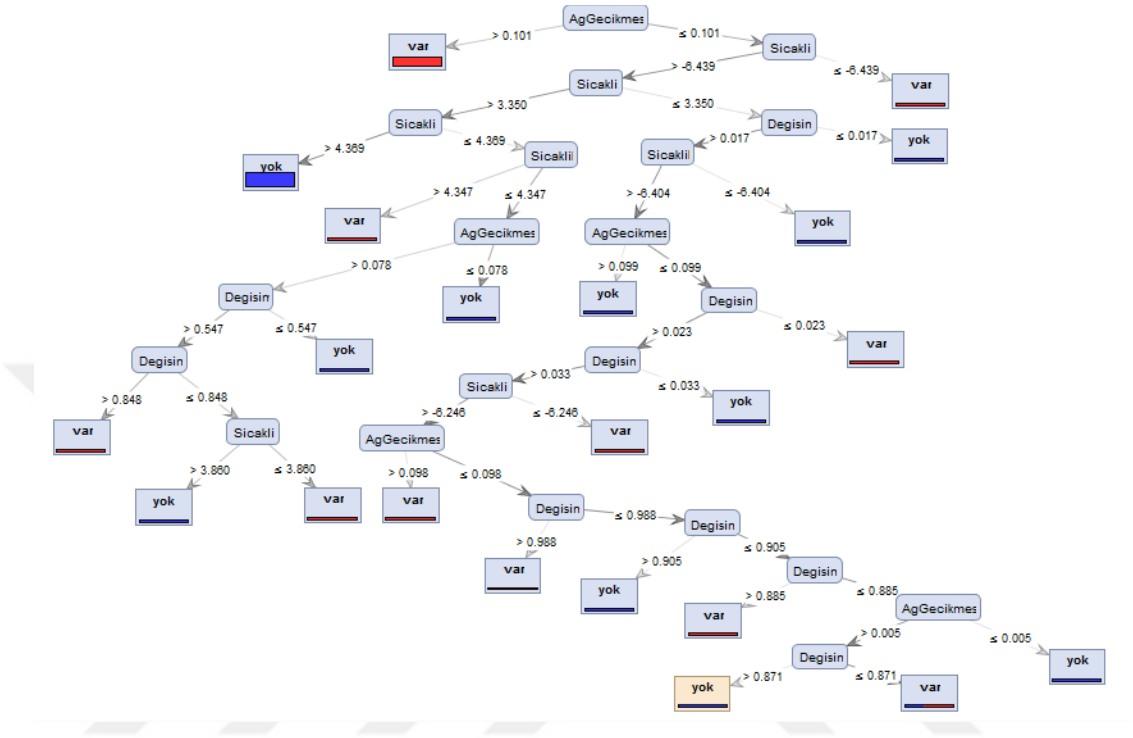


Şekil 4.7: Kılıç'ın AES ve Ham veriyle ağ gecikmesi karşılaştırması

Sonucun ikinci kısmında geliştirilen Kılıç Güvenlik Sisteminde makine öğrenmesi yöntemleri ile saldırı tespiti yapan yöntemlerin doğrulukları ve performansları karşılaştırılmaktadır. Saldırı yöntemleri olarak Tekrarlama, Aykırı değer ve Fiziksel saldırı senaryoları uygulanmıştır. Makine öğrenmesi yönetmelerinden Karar Ağacı, Destek Vektör Makinesi, K-en Yakın Komşu, Derin Öğrenme ve Naive Bayes ayrı ayrı öğrenme ve test olarak çalıştırılıp sonuçları karşılaştırılmıştır.

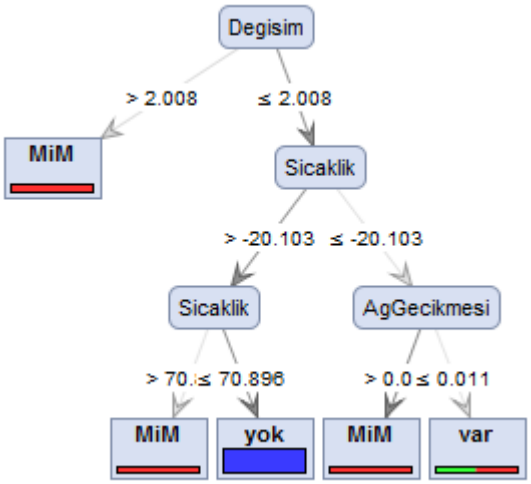
Tekrarlama Saldırısı: Bu saldırı yönteminin veri seti oluşturulurken pasif olarak dinleme işleminde nesnenin gönderdiği paketleri yakalayarak nesnenin veri iletimi yapmadığı anlarda Kenar Bilişim sunucusuna gönderen bir saldırgan simüle edilmiştir. Kenar sunucusunun bu tekrarlanan paketleri gerçeklerinden ayırarak işlemlerini yürütmelidir. Eğitim veri seti Karar Ağacı algoritmasına uygulandığında **Şekil 4.8** ki ağaç oluşmaktadır. Bu oluşan karar ağacının doğruluğu %96,90 olmuştur. Destek Vektör Makinesi ile aynı veri seti öğrenme yaptığında %95,10 ile daha düşük bir sonuç elde edilmiştir. K en Yakın Komşu algoritması bu veri seti uygulandığında %98.94 ile en iyi tekrarlama saldırısı tespiti sağlayan doğruluk elde

edilmiştir. Derin Öğrenme algoritması uygulandığında %94.63 ile en düşük doğruluk oranı elde edilmiştir. Naive Bayes yöntemi aynı saldırı verisinin tespiti için öğrenme yapıldığında %95.37 doğruluk sonucunu vermiştir.



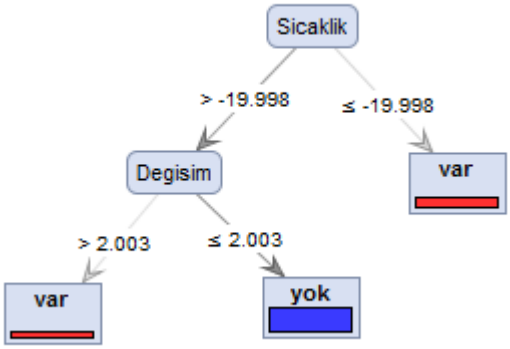
Şekil 4.8: Tekrarlama saldırısı Karar Ağacına uygulandığında oluşan ağaç

Aykırı Değer Saldırısı: Bu saldırı da ise Aradaki Adam (Man in the Middle) Saldırısı simüle edilirken araya giren saldırganın sensörden gelen verileri değiştirerek Kenar Sunucusuna aykırı değerler gönderdiği düşünülmüştür. Ani artış azalışları kural tabanlı sistem tarafından zor algılanması için daha parabolik bir artış yapması sağlanmıştır. Bu değerlerin araya giren bir saldırgandan mı yoksa sensörden mi geldiğini tespit için gerek periyodik cihaz doğrulama ve özet kontrolü yapılsa da bu kontrolleri aşarak saldırı yaptığı kabul edilmiştir. Eğitim veri seti Karar Ağacı algoritmasına uygulandığında **Şekil 4.9** ki ağaç oluşmaktadır. Bu oluşan karar ağacının doğruluğu %99,60 ile en iyi sonuç olmuştur. K en Yakın Komşu algoritması aynı veri seti öğrenme yaptığında %97.66 ile en iyi ikinci sonuç elde edilmiştir. Destek Vektör Makinesi ile bu veri seti uygulandığında %94,52 ile doğruluk yüzdesi sıralamasında 3. olmuştur. Derin öğrenme %95.09 doğruluk verirken Naive Bayes ise %90.52 ile yöntemler arasındaki en başarısız sonuç olarak aykırı değer saldırısı tespiti sağlayan doğruluk elde edilmiştir.



Şekil 4.9: Aykırı değer saldırısı tespiti Karar Ağacı

Fiziksel Saldırı: Bu saldırı yöntemi ulusal ve uluslararası literatürde göz ardı edilen bir yöntemdir. IoT cihazlarının insanların rahat erişebileceği yerlerde olduğu göz önüne alındığında en kolay ve en tespit edilmesi zor saldırı yöntemi olarak karşımıza çıkmaktadır. Bu saldırı yöntemine karşı mücadelede en etkili yol nesnelerin konum bilinciyle yerleştirilmesidir. Bu sayede yakın konumdaki nesnelerin ürettikleri değerler kolaylıkla karşılaştırılarak sapmaya göre ortamdaki değişikliğin gerçek mi ya da manipülasyon mu olduğu değerlendirebilir. Eğitim veri seti Karar Ağacı algoritmasına uygulandığında **Şekil 4.10** ki ağaç oluşmaktadır. Bu oluşan karar ağacının doğruluğu %99,99 ile en iyi sonuç olmuştur. Destek Vektör Makinesi ile aynı veri setiyle öğrenme yaptığında %95,83 ile en kötü 2. sonuç elde edilmiştir. K en Yakın Komşu algoritması bu veri seti uygulandığında 99.81 ile en iyi 2. fiziksel saldırısı tespiti sağlayan doğruluk elde edilmiştir. Derin Öğrenme yöntemi %95,87 olarak 3. en iyi sonucu vermiştir. Naive Bayes ise %83,73 doğruluk oranıyla en kötü sonuç elde edilmiştir.



Şekil 4.10: Fiziksel saldırı veri seti için oluşan Karar Ağacı

Sonuç olarak saldırı tespitinde kullanılan makine öğrenmesi yöntemlerinin saldırı tespitindeki doğruluk oranları karşılaştırılmıştır. Makine öğrenmesi yöntemleri kullanılarak saldırı tespiti yapılırken kullanılan Karar Ağacı, Destek Vektör Makinesi, K en Yakın Komşu, Derin Öğrenme ve Naive Bayes algoritmalarının saldırı yöntemlerine göre vermiş oldukları doğrulukları **Çizelge 4.1** de kıyaslanmıştır.

Çizelge 4.1: Makine öğrenmesi yöntemlerinin farklı saldırı yöntemlerinin tespitinde doğruluk (accuracy) oranları

Saldırı Türleri / Uygulanan Yöntem	**Karar Ağacı**	**Destek Vektör Makinesi**	**K en Yakın Komşu**	**Derin Öğrenme**	**Naive Bayes**
Tekrarlama Saldırısı	% 96,90	% 95,10	**% 98,94**	% 94,63	% 95,37
Aykırı Değer Saldırısı	**% 99,60**	% 94,52	% 97,66	% 95,09	% 90,52
Fiziksel Saldırı	**% 99,99**	% 95,83	% 99,81	% 95,87	% 83,73
3 Saldırı Yöntemi Karışık	% 98,27	% 92,76	**% 98,42**	% 81,52	% 88,55

Simülasyon ve testlerden elde edilen veri kümesi üzerinde yapılan öğrenme ve tahmin çalışmasından ortalama en iyi değeri K en Yakın Komşu algoritması vermiştir. Üretilen veri kümelerinde tekrarlama saldırısı için K en yakın komşu, aykırı değer saldırısı ve fiziksel saldırı için Karar Ağacı en yüksek doğruluk oranı sağlamıştır. Kullanılan metodların çalışma performansı olarak değerlendirildiğinde

ise en hızlı sonuç Karar Ağacı en yavaş sonuç Derin Öğrenme yöntemi vermiştir. Saldırı verileri kümeleri saldırı senaryolarında üretilen veri ile ayrı ayrı öğrenme ve test işlemleri yapılmasının yanında yöntemlerin saldırı tespiti doğrulukları karşılaştırılırmıştır. Ayrıca saldırı yöntemleri karışık olarak tek bir veri seti olarak makine öğrenmesi yöntemlerine uygulandığında saldırının varlığı ve türünün doğru tespitinin oranları **Çizelge 4.1**'nin "3 Saldırı Yöntemi Karışık" kaydında gösterilmekte ve en iyi saldırı çeşidini K en Yakın Komşu algoritması tespit etmiştir. **Çizelge 4.2** de farklı yöntemler karşısında duyarlılık oranları da gösterilmiştir. Simülasyonda oluşturulan bu veri setine göre tekrarlama saldırısında Naive Bayes %98,48 ile en yüksek duyarlılık sonucu vermiştir. Diğer iki aykırı değer saldırısında %100,00 ve fiziksel saldırıda %99,97 ile Karar Ağacı en iyi duyarlılık sonucu vermiştir. Saldırı yönteminin tespitinde en başarılı duyarlılık sonucunu 98,60 ile Karar Ağacı algoritmasında sağlanmıştır.

Çizelge 4.2: Makine öğrenmesi yöntemlerinin farklı saldırı yöntemlerinin tespitinde duyarlılık (Sensitivity) oranları

Saldırı Türleri / Uygulanan Yöntem	**Karar Ağacı**	**Destek Vektör Makinesi**	**K en Yakın Komşu**	**Derin Öğrenme**	**Naive Bayes**
Tekrarlama Saldırısı	% 95,20	% 97,79	% 98,33	% 94,63	**% 98,48**
Aykırı Değer Saldırısı	**% 100,00**	% 98,06	% 99,88	% 99,77	% 92,16
Fiziksel Saldırı	**% 99,97**	% 97,84	% 99,94	% 99,51	% 83,73
3 Saldırı Yöntemi Karışık	**% 98,60**	% 91,77	% 97,45	% 84,50	% 88,55

Sonuç olarak Kılıç uygulamasında kullanılan kural tabanlı ve makine öğrenmesi yöntemleri ile saldırı tespiti yapan, proaktif ve değişken seviyeli anlayışı sayesinde kısıtlı kaynakları olan ve güvenlik önlemleri servis kalitesini düşürmeden Kenar Bilişim sunucusu üzerinde esnek, şeffaf ve yönetilebilir bir güvenlik perspektifi geliştirilmiştir.

5. DEĞERLENDİRME VE ÖNERİLER

Yapılan bu tez çalışmasında Kenar Bilişim'in Nesnelerin İnterneti ve Bulut Bilişimin sahada karşılaştığı band genişliği, ağ gecikmesi gizlilik ve güvenlik gibi problemlere çözümler sağlayan Kılıç adını verdiğimiz tasarım ve geliştirilen sistem açıklanmaktadır.

Geliştirilen Kılıç Güvenlik Sistemi nesnelere özgü, esnek, dinamik ve proaktif bir güvenlik sağlarken servis kalitesini olabildiğince yüksek tutmaya çalışmaktadır. Modüler tasarımı sayesinde farklı çalışmalar ve konulardaki ilerleyen dönemdeki gelişmelerin içerisine kolaylıkla dâhil edilmesine imkân sağlamaktadır. Az sayıda güvenlik yöntemi içerisinde bulunmasına karşın bir güvenlik sistemi çerçevesi gibi tasarlanarak yeni yöntem ve metotlar eklenebilmesi için esnek olarak tasarlanmıştır.

Nesnelerin ihtiyacı kadar güvenlik ve gizlilik sağlayarak nesnelerin kaynaklarından tasarruf sağlanır. Ayrıca sistemde saldırı tespit edildiğinde kenar-nesne iletişiminin güvenliğini artırması gizlilik ve güvenliğe yeni dinamik güvenlik bakış açısı katmıştır. Saldırı tespiti için yaygın çalışılan saldırı yöntemlerinin yanında göz ardı edilen fiziksel saldırılara karşı da tespit ve önleme yöntemleri geliştirilmiştir. İletişim şeklinde yeni bir model ortaya koyarak dışarıdan gelecek ve dışarıya nesneler üzerinden gidebilecek tehditlere karşı temelinden önlem almıştır. Ayrıca, Kılıç Akıllı Fabrika uygulamasında çeşitli kullanım durumları bağlamında test ve simüle edilmiştir.

Bu çalışma kapsamında kenar sunucusu ile nesne düğümleri arasındaki veri iletişimine odaklanılmış olduğundan Bulut Bilişim ve Kenar Bilişim iletişimindeki gizlilik ve güvenlik çalışmaları geri planda kalmıştır. Kılıç, kenar-nesne iletişiminde kullanılan dinamik güvenlik önlemleri Bulut iletişim ara yüzünde kullanılmamış, geleneksel güvenli iletişim protokolleri tercih edilmiştir. Nesnelerin İnterneti uygulamalarına yapılan güvenlik analizlerinde üçüncü parti mobil, web ve bulut ara yüzlerinden gelen tehditler çalışmanın kapsamı dışarısında bırakılmıştır.

Başka bir çalışma alanı olduğu için nesnelerin ve ortamları güvenlik analizleri hangi başlıklarda ve nasıl yapılması gerektiği konusu tez kapsamında olmadığı için yüzeysel olarak bahsedilmiş olup sistem yöneticisi tarafından eklendiği belirtilmiştir. Ayrıca Kılıç üzerindeki kullanıcılar ve rolleri ayrıntılı olarak belirtilmemiş olup,

sisteme eklenebilecek manuel müdahale ve raporlama imkânlarına girilmemiştir. Bu manada çalıştır unut otonom bir güvenlik sistemi profili çizilmiştir.

Kılıç, kenar-nesne iletişimde saldırı ve anomali tespiti için kural tabanlı yöntemlerin yanında makine öğrenmesi yöntemleri kullanmıştır. Az sayıda tehdit göz önüne alınarak test yapılmasının yanında iletişim verisinin simülasyon ortamında hazırlanmış olması Kılıç Güvenlik Sisteminin başarısının sorgulanmasına neden olmaktadır. Ayrıca ortaya konulan yeni proaktif ve değişken güvenlik seviyesi yaklaşımının yeni güvenlik açıklarına sebebiyet verebileceği için bu alandaki test ve geliştirme faaliyetlerinin devam etmesi gerekmektedir. Özellikle cihaz kapatma gibi servisi sonlandırmaya özelliği ve yüksek güvenlik için yüksek gecikmeye neden olan özellikler saldırıyı engellese de servis kalitesindeki düşüşe neden olacağı unutulmamalıdır. Kılıç servis kalitesi ve güvenlik arasındaki dengeyi çözmekten ziyade şeffaf ve esnek bir şekilde yeni güvenlik yaklaşımlarını göz önüne alarak tehdit riskini yönetmeye odaklanmaktadır.

Kılıç üzerinde yer alan saldırı tespit ve saldırı önleme görevindeki modüller ağda yer alan benzer görevdeki cihazlarla geri besleme ve haberleşmesi bulunduğundan tez kapsamında bahsedilmemektedir. Kurulacak geri besleme mekanizmaları sistemin daha bütünsel ve güvenli çalışmasına yardımcı olacaktır.

Nesnelerin İnterneti cihazı üzerinde bulunan Kenar İletişim Arayüzü kaynak kısıtları gerekçe gösterilerek saldırı tespiti ve önleme süreçlerinde kullanılmamıştır. Nesne üzerinden gelecek güvenlik ve gizlilik durumu geri beslemesiyle daha hızlı saldırı tespiti yapılması sağlanabilir. Ayrıca nesnenin donanımsal ve yazılımsal durumunu bu modül üzerinden analiz edilip kenar cihazına bilgi verilebilir. Bu yaklaşım Kenar Bilişim'in yakın istihbarat olarak adlandırılan özelliğine yeni bir perspektif katılmış olacaktır.

Yapılan çalışma genel olarak değerlendirildiğinde nesne, kenar ve bulut sistemleri ile ilgili süreçlerin güvenliği sağlandığı görülse de kenar sunucularının hiyerarşisinden, kenarların birbirleriyle iletişiminden bahsedilmemiştir. Ayrıca nesnelerin de aralarında bulunan hiyerarşi erişim yetkileri ve veri paylaşımı konusuna değinilmemiştir.

Bu değerlendirmede önerilen noktalar gerek bu çalışmanın devamı olarak tarafımızca yapılabileceği gibi, Dünya'daki diğer araştırmacılar tarafından ilerleyen dönemde

gerçekleştirilebilir. Bu çalışmanın ışığında Kenar Bilişim Nesnelerin İnterneti ve Bulut Bilişim'in kaçınılmaz geleceği olarak görülmektedir. Gelecekte yeni Nesne, Kenar ve Bulut Bilişim üçlemesiyle yeni hizmet ve ürünlerin hayatımıza gireceğini göstermektedir.

KAYNAKLAR

[1] H. Rahman, R. Rahmani. “Enabling Distributed Intelligence Assisted Future İnternet of Things Controller (FITC).” under review at Elsevier journal of Applied Computing and Informatics 2017

[2] Ranadheer Errabelly, Kewei Sha, Wei Wei, T. Andrew Yang “EdgeSec: Design of an Edge Layer Security Service to Enhance IoT Security” Mayıs 2017 1. ICFEC

[3] Ruei-Hau Hsu, Jemin Lee, Tony Q. S. Quek, Jyh-Cheng Chen "Reconfigurable Security: Edge Computing-based Framework for IoT" arXiv:1709.06223v1 19 Eylül 2017

[4] Hasibur Rahman, Rahim Rahmani, Theo Kanter “Multi-Modal Context-Aware reasoNer (CAN) at the Edge of IoT” Procedia Computer Science 109C (2017) 335–342 8

[5] M. Vardhana, N. Arunkumar, Enas Abdulhay, P. V. Vishnuprasad "IoT based real time trafic control using cloud computing" Future Generation Computer Systems. Jan 2018, Vol. 78, p964, 12 p. Elsevier B.V. Ocak 2018

[6] GC Idex, ‘‘Cisco global cloud index: Forecast and methodology, 2016–2021,’’ Cisco, San Jose, CA, USA, White Paper C11-738085-02,Şubat 2018.

[7] H.Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, Lanyu Xu “Edge Computing: Vision and Challenges” Ekim 2016

[8] Pavan Pongle, Gurunath Chavan "A Survey: Attacks on RPL and 6LoWPAN in IoT" 2015 International Conference on Pervasive Computing (ICPC) 2015 IEEE

[9] Arwa Alrawais, Abdulrahman Alhothaily, Chunqiang Hu, Xiuzhen Cheng "Fog Computing for the İnternet of Things: Security and Privacy Issues" IEEE İnternet Computıng IEEE Nisan 2017

[10] Qi Jing, Athanasios V. Vasilakos , Jiafu Wan , Jingwei Lu , Dechao Qiu, “Security of the İnternet of Things: perspectives and challenges”, Springer Science+Business Media 2014.

[11] Shancang Li Theo Tryfonas Honglei Li, “The İnternet of Things: a security point of view”, İnternet Research 2016.

[12] Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva, "Security for the İnternet of Things: A Survey of Existing Protocols and Open Research Issues", IEEE Communıcatıon Surveys & Tutorıals 2015.

[13] Colin Tankard, Digital Pathways, "The security issues of the İnternet of Things" Computer Fraud & Security 2015.

[14] Minkyung Kang, Onechul Na, Hangbae Chang, "Security experts’ capability design for future İnternet of things platform" Springer Science+Business Media 2015.

[15] Lukas Malina, Jan Hajny, Radek Fujdiak, Jiri Hosek, 2016, "On perspective of security and privacy-preserving solutions in the İnternet of things" Computer Networks 102 (2016) 83–95

[16] Wenbo Shi, Neeraj Kumar, Peng Gong, Naveen Chilamkurti, Hangbae Chang, 2014, "On the security of a certificateless online/offline signcryption for İnternet of Things" Peer-to-Peer Netw. Appl. (2015) 8:881–885

[17] Sabrina Sicari,Alessandra Rizzardi, Daniele Miorandi, Cinzia Cappiello, Alberto Coen-Porisini, 2016 "A secure and quality-aware prototypical architecture for the İnternet of Things", Information Systems 58 (2016) 43–55

[18] Ricardo Neisse, Gary Steri, Igor Nai Fovino, Gianmarco Baldini, 2015,"SecKit: A Model-based Security Toolkit for the İnternet of Things", Computers & Security 54 (2015) 60 e7 6
[19] L. Atzori, A. Iera, and G. Morabito, "The İnternet of things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[20] C. P. Mayer, "Security and privacy challenges in the İnternet of things," in Proc. Electron. Commun. EASST, 2009, vol. 17, pp. 1–12.
[21] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes", Advances in CryptologyEUROCRYPT99, Springer, Berlin Heidelberg, 1999, pp. 223–238.
[22] D. Boneh, X. Boyen, H. Shacham, "Short group signatures", Advances in Cryptology–CRYPTO 2004, Springer, Berlin Heidelberg, 2004, pp. 41–55.
[23] C. Delerablée, D. Pointcheval, "Dynamic fully anonymous short group signatures", Progress in Cryptology-VIETCRYPT 2006, Springer, 2006, pp. 193–210.
[24] F. Li, Z. Zheng, C. Jin, "Secure and efficient data transmission in the İnternet of things", Telecomm. Syst. (2015) 1–12.
[25] J. Hajny, L. Malina, "Unlinkable attribute-based credentials with practical revocation on smart-cards", Springer, Berlin Heidelberg, 2013.
[26] V. Goyal, O. Pandey, A. Sahai, B. Waters, "Attribute-based encryption for fine–grained access control of encrypted data", Proceedings of the 13th ACM Conference on Computer and Communications Security, Acm, 2006, pp. 89–98.
[27] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Jong Hyuk Park "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions" Springer-Verlag Berlin Heidelberg 2017
[28] Padmavathi B, Kumari SR (2013) A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution. Int J Sci Res 2(4):170–174
[29] McKay KA, Bassham L, Turan M S, Mouha N (2016) Report on lightweight cryptography. NIST DRAFT NISTIR, pp 1–29
[30] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., ... & Vikkelsoe, C. (2007). "PRESENT: An ultra-lightweight block cipher." Cryptographic Hardware and Embedded Systems-CHES 2007 (pp. 450-466). Springer Berlin Heidelberg.
[31] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., ... & Vikkelsoe, C. (2007). "PRESENT: An ultra-lightweight block cipher." Cryptographic Hardware and Embedded Systems-CHES 2007 (pp. 450-466). Springer Berlin Heidelberg.
[32] Hosseinzadeh J, Hosseinzadeh M (2016) A comprehensive survey on evaluation of lightweight symmetric ciphers: hardware and software implementation. Adv Comput Sci Int J 5(4):31–41
[33] Mohd BJ, Hayajneh T, Vasilakos AV,"A survey on lightweight block ciphers for low-resource devices: comparative study and open issues." J Netw Comput Appl 58:73–93 2015
[34] Standaert FX, Piret G, Rouvroy G, Quisquater JJ, Legat JD "ICEBERG: an involutional cipher efficient for block encryption in reconfigurable hardware." Proceeding of International Workshop on Fast Software Encryption, Springer, Berlin, pp 279–298 2004
[35] Daemen, J., & Rijmen, V. (1998, June). "AES proposal: Rijndael", First Advanced Encryption Standard (AES) Conference

[36] Gong, Z., Nikova, S., & Law, Y. W. (2012). “KLEIN: a new family of lightweight block ciphers.” RFID. Security and Privacy (pp. 1-18). Springer Berlin Heidelberg.
[37] Rivest, R. L. (1995, January). The RC5 encryption algorithm. In Fast Software Encryption (pp. 86-96). Springer Berlin Heidelberg.
[38] Lu, J. (2009). Related-key rectangle attack on 36 rounds of the XTEA block cipher. International Journal of Information Security, 8(1), 1-11.
[39] Hong, D., Lee, J. K., Kim, D. C., Kwon, D., Ryu, K. H., & Lee, D. G. (2014). LEA: A 128-bit block cipher for fast encryption on common processors. In Information Security Applications (pp. 3-27). Springer International Publishing.
[40] Leander, G., Paar, C., Poschmann, A., & Schramm, K. (2007, January). New lightweight DES variants. In Fast Software Encryption (pp. 196-210). Springer Berlin Heidelberg.
[41] Suzaki, T., Minematsu, K., Morioka, S., & Kobayashi, E. (2011). Twine: A lightweight, versatile block cipher. In ECRYPT Workshop on Lightweight Cryptography (pp. 146-169).
[42] Martin R. Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar and Tolga Yalcın (2014). Block Ciphers -- Focus On the Linear Layer (feat. PRIDE), Full Version, IACR Cryptology ePrint Archive, 2014, 453.
[43] Leander G, Paar C, Poschmann A, Schramm K (2007) New lightweight DES variants. In: Proceeding of International Workshop on Fast Software Encryption, Springer, Berlin, pp 196–210
[44] Daesung Moon, Hyungjin Im, Ikkyun Kim, Jong Hyuk Park "DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks" J Supercomput (2017) 73:2881–2895
[45] Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann Kao, Rong-Jian Chen, Jui-Lin Lai "Citra Dwi Perkasa aA novel intrusion detection system based on hierarchical clustering and support vector machines" Expert Systems with Applications 38 (2011) 306–313
[46] Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. The International Journal on Very Large Data Bases, 16(4), 507–521.
[47] Toosi, A. N., & Kahani, M. (2007). A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Computer Communications, 30, 2201–2212
[48] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, Xinzheng He "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks" IEEE Access 2017
[49] Sajid A, Abbas H, Saleem K (2016) Cloud-assisted IoT-based SCADA systems security: a review of the state of the art and future challenges. IEEE Acc 4:1375–1384
[50] Zhou J, Cao Z, Dong X, Vasilakos, AV (2017) Security and privacy for cloud-based IoT: challenges. IEEE Commun Mag 55(1):26–33
[51] W. Zhanyi "The Applications Of Deep Learning On Traffic Identification" Blackhat 2015.
[52] R. R. Reddy, Y. Ramadevi, K. V. N. Sunitha, "Effective discriminant function for intrusion detection using SVM", Proc. Int. Conf. Adv. Comput. Commun. Inform. (ICACCI), pp. 1148-1153, Sep. 2016.

[53] Abdulla Amin Aburomman, Mamun Bin Ibne Reaz "A novel SVM-kNN-PSO ensemble method for intrusion detection system" Applied Soft Computing 38 (2016) 360–372
[54] W. Li, P. Yi, Y. Wu, L. Pan, J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network", J. Elect. Comput. Eng., Jun. 2014.
[55] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning", Nature, vol. 521, pp. 436-444, May 2015.
[56] J. Schmidhuber, "Deep learning in neural networks: An overview", Neural Netw., vol. 61, pp. 85-117, Jan. 2015.
[57] L. Liu, L. Shao, X. Li, K. Lu, "Learning spatio-temporal representations for action recognition: A genetic programming approach", IEEE Trans. Cybern., vol. 46, no. 1, pp. 158-170, Jan. 2016.
[58] A.-A. Liu, Y.-T. Su, W.-Z. Nie, M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition", IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 1, pp. 102-114, Jan. 2017.
[59] J. Wu, Y. Zhang, W. Lin, "Good practices for learning to recognize actions using FV and VLAD", IEEE Trans. Cybern., vol. 46, no. 12, pp. 2978-2990, Dec. 2016.
[60] Nguyen Thanh Van, Tran Ngoc Thinh, Le Thanh Sach "An anomaly-based network intrusion detection system using Deep learning" 2017 International Conference on System Science and Engineering (ICSSE)
[61] U. Fiore F. Palmieri A. Castiglione A. Santis "Network an1omaly detection with the RBM. Neurocomputing" in Neurocomputing Elsevier B.V pp. 11 2013.
[62] Md. Zahangir Alom V. Bontupalli Tarek M. Taha "Intrusion Detection using DBN" National Aerospace and Electronics Conference (NAECON) 2015.
[63] Q. Niyaz W. Sun A. Javaid M. Alam "A Deep Learning Approach for NIDS" Bio-inspired Information and Communications Technologies (BIONETICS) 2014.
[64] Palmer J (2011) Naive Bayes classification for intrusion detection using live packet capture. In: Palmer J (ed) Data mining in bioinformatics. Springer, Berlin
[65] Mehmood A, Umar MM, Song H (2017) ICMDS: secure inter-cluster multiple-key distribution scheme for wireless sensor networks. Ad Hoc Netw 55:97–106
[66] Wafa' S. Al-Sharafat, and Reyadh Naoum "Development of Genetic-based Machine Learning for Network Intrusion Detection" World Academy of Science, Engineering and Technology 55, 2009.
[67] Saurabh Mukherjee, Neelam Sharma "Intrusion Detection using Naive Bayes Classifier with Feature Reduction" Procedia Technology 4 (2012) 119 – 128
[68] Ms.Nivedita Naidu, Dr.R.V. Dharaskar "An effective approach to network intrusion detection system using genetic algorithm", International Journal of Computer Applications (0975-8887) 1 – No. 2, 2010.
[69] W. Wang, X. Zhang, S. Gombault "Constructing attribute weights from computer audit data for effective intrusion detection" J. Syst. Softw., 82 (12) (2009), pp. 1974-1981
[70] I. Syarif, E. Zaluska, A. Prugel-Bennett, G. Wills "Application of bagging, boosting and stacking to intrusion detection" Machine Learning and Data Mining Pattern Recognition, Springer (2012), pp. 593-602

[71] E. Bahri, N. Harbi, H.N. Huu "Approach based ensemble methods for better and faster intrusion detection" Computational Intelligence in Security for Information Systems, Springer (2011), pp. 17-24
[72] V. Bukhtoyarov, V. Zhukov "Ensemble-distributed approach in classification problem solution for intrusion detection systems" Intelligent Data Engineering and Automated Learning-IDEAL 2014, Springer (2014), pp. 255-265
[73] Jıale Zhang, Bıng Chen, Yanchao Zhao, Xıang Cheng, Feng Hu "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues" Special Section On Mobile Edge Computing 2169-3536 2018 IEEE. Mart 2018
[74] TU Yaofeng, DONG Zhenjiang, and YANG Hongzhang "Key Technologies and Application of Edge Computing" DOI: 10.3969/j. issn. 1673-5188. 2017. 02. 004 http://kns.cnki.net/kcms/detail/34.1294.TN.20170419.1031.002.html, [Çevrimiçi] Nisan 19, 2017
[75] Kasey Panetta "Gartner Top 10 Strategic Technology Trends for 2018." Çevrimiçi https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/ [Erişim 13 May 2018]. [07 GE]
[76] Brandon Butler "What is edge computing and how it's changing the network." Çevrimiçi https://www.networkworld.com/article/3224893/İnternet-of-things/what-is-edge-computing-and-how-it-s-changing-the-network.html [Erişim 13 Mayıs 2018].
[77] Rob Meulen "What Edge Computing Means for Infrastructure and Operations Leaders" Çevrimiçi https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders/ [Erişim 13 Mayıs 2018].
[78] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, ''The cost of a cloud: Research problems in data center networks,'' ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 68–73, 2009.
[79] M. Armbrust et al., "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.
[80] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," J. Netw. Comput. Appl., vol. 59, pp. 46–54, Ocak 2016.
[81] A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," IEEE Commun. Surveys Tuts. vol. 16, no. 1, pp. 393–413, 1st Quart., 2014.
[82] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," J. Netw. Comput. Appl., vol. 59, pp. 46–54, Ocak 2016.
[83] Z.-W. Xu, ''Cloud-sea computing systems: Towards thousand-fold improvement in performance per watt for the coming zettabyte era,'' J. Comput. Sci. Technol., vol. 29, no. 2, pp. 177–181, Ocak 2014.
[84] N. Fernando, S. W. Loke, and W. Rahayu, ''Mobile cloud computing: A survey,'' Future Generat. Comput. Syst. vol. 29, no. 1, pp. 84–106, 2013
[85] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, ''Fog computing and its role in the İnternet of Things,'' presented at the 1st Ed. MCC Workshop Mobile Cloud Comput., Helsinki, Finland, Ağustos. 2012, pp. 13–16. [(04-1) 15]
[86] S. Yi, C. Li, and Q. Li, ''A survey of fog computing: Concepts, applications and issues,'' in Proc. Workshop Mobile Big Data (Mobidata), Hangzhou, China, Haziran 2015, pp. 37–42.

[87] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, ‘‘Mobile edge computing—A key technology towards 5G,’’ ETSI, Sophia Antipolis, France, White Paper 11, Eylül 2015, pp. 1–16.
[88] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, ‘‘Mobile edge computing: A survey,’’ IEEE İnternet Things J., vol. 5, no. 1, pp. 450–465, Şubat 2018.
[89] Mung Chiang, Sangtae Ha, Chih-Lin I, Fulvio Risso, Tao Zhang "Clarifying Fog Computing and Networking: 10 Questions and Answers" IEEE Communications Magazine Nisan 2017
[90] Tasnuva Mahjabin, Yang Xiao, Guang Sun, Wangdong Jiang "A survey of distributed denial-of-service attack, prevention, and mitigation techniques" International Journal of Distributed Sensor Networks 2017 Kasım 2017
[91] Yuan Ai, Mugen Peng, Kecheng Zhang "Edge computing technologies for İnternet of things: a primer" Digital Communications and Networks Nisan 2017
[92] Buyuk Oguzhan Oktay, Camurcu Ali Yilmaz. “A Novel Actual Time Cyber Security Approach to Smart Grids” , 6th International Istanbul Smart Grid and Cities Congress and Fair 25-26 Nisan 2018
[93] Amith Seth. “İnternet of Things to Smart IoT Through Semantic, Cognitive, and Perceptual Computing.” in IEEE Intelligent Systems, vol. 31, No. 2, pp. 108-112, 2016

ÖZGEÇMİŞ

Ebu Yusuf GÜVEN 1993 yılında Erzincan'da doğmuştur. İstanbul Üniversitesi Bilgisayar Mühendisliği Bölümüne 2010 yılında başlamış 2015 de mezun olmuştur. 2015-2017 yılları arası CPM Yazılım A.Ş. de yazılım geliştirici olarak çalışmıştır. 2016 yılında Fatih Sultan Mehmet Vakıf Üniversitesinde Bilgisayar Mühendisliği dalında tezli yüksek lisans programında eğitimine başlamıştır. 2018 Nisan Ayından beri İstanbul Üniversitesi Bilgisayar Mühendisliği Siber Güvenlik Ana Bilim Dalı'nda Araştırma Görevlisi olarak çalışmaktadır.

İletişim Bilgileri

eyguven@istanbul.edu.tr

Turgut Özal Mahallesi 37. Sokak No 17-19 Esenyurt / İstanbul

Ödüller

- Mansiyon Ödülü TÜBİTAK Liseler Arası Proje Yarışması 2010
- Bronz Madalya e-Devlet Yenilikçi Proje Yarışması 2014

Tezden Türetilen Yayınlar

1. Ebu Yusuf GÜVEN, Ali Yılmaz ÇAMURCU "Kenar Bilişim Güvenlik Uygulaması: Kılıç" UBMK Eylül 2018 (Kabul Edildi)
2. Ebu Yusuf GÜVEN, Ali Yılmaz ÇAMURCU "Akıllı Nesneler için Fiziksel Saldırı Tespiti" IDAP Eylül 2018 (İncelemede)