



POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)

URL: <http://dergipark.org.tr/politeknik>



Heterojen medikal IoT verilerinin depolanmasında ilişkisel olmayan veritabanına dayalı bir yaklaşım

An approach to non-relational database-based in the storing of heterogeneous medical IoT data

Yazar(lar) (Author(s)): Hüseyin POLAT¹, Saadin OYUCU²

ORCID¹: 0000-0003-4128-2625

ORCID²: 0000-0003-3830-3039

Bu makaleye şu şekilde atıfta bulunabilirsiniz (To cite to this article): Polat H. ve Oyucu S., “Heterojen medikal IoT verilerinin depolanmasında ilişkisel olmayan veritabanına dayalı bir yaklaşım”, *Politeknik Dergisi*, 22(4): 989-998, (2019).

Erişim linki (To link to this article): <http://dergipark.org.tr/politeknik/archive>

DOI: 10.2339/politeknik.477311

Heterojen Medikal IoT Verilerinin Depolanmasında İlişkisel Olmayan Veritabanına Dayalı Bir Yaklaşım

Araştırma Makalesi / Research Article

Hüseyin POLAT*, **Saadin OYUCU**

Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, Gazi Üniversitesi, Türkiye

(Geliş/Received : 01.11.2018 ; Kabul/Accepted : 22.01.2019)

ÖZ

Modern medikal cihazlar, her türlü temel medikal veriyi üretme ve iletebilme kabiliyetine sahip olmuşlardır. Bu cihazlar, birbirleriyle veri paylaşabilir veya bulutta merkezi bir platforma veri gönderebilir. Sağlık endüstrisinde yeni trend, her zaman ve her yerden erişilebilecek şekilde, buluttan sunulan elektronik medikal kayıtlarla entegre bir tıbbi izleme sisteminin oluşturulmasıdır. Hacmi gittikçe artan heterojen medikal verilerin düşük maliyetle, hızlı ve güvenli bir şekilde veritabanı sisteminde depolanması, verilerin aktarılması, paylaşılması ve görselleştirilmesi esastır. Bu çalışmada heterojen medikal verileri algılayıcılardan toplamak, verileri görselleştirmek ve depolamak için farklı veritabanı sistemlerini kullanabilecek şekilde bir medikal Nesnelerin İnterneti (medical Internet of Things – mIoT) platformu gerçekleştirilmiştir. mIoT platformu üzerinde dört farklı veritabanı modeli dört farklı senaryo ile test edilmiştir. Bu senaryolarda mIoT platformunda kullanılan veritabanı modellerinin performansları; sorgu süresi, veri hazırlığı, esneklik, güvenlik ve ölçeklenebilirlik parametreleri göz önüne alınarak karşılaştırılmıştır. mIoT platformunda kullanılan ilişkisel olmayan veritabanı modelinin (NoSQL: Not only Structured Query Language) okuma/yazma işlemlerinde ilişkisel veritabanı modellerine göre daha verimli çalıştığı, performansının, esnekliğinin ve ölçeklenebilirliğinin ilişkisel veritabanı sistemlerine göre daha iyi olduğu gözlemlenmiştir.

Anahtar Kelimeler: Medikal IoT, veritabanı performansı, SQL, NoSQL veri tabanları elemanlar analizi.

An Approach to Non-Relational Database-Based in the Storing of Heterogeneous Medical IoT Data

ABSTRACT

Modern medical devices have the ability to producing and transmitting all kinds of basic medical data. These devices can share data with each other or send data to a central platform in the cloud. The new trend in the healthcare industry is the creation of a medical monitoring system integrated with electronic medical records that presented from the cloud, that can be accessed anytime and anywhere. It is essential to store, transfer, share and visualize large volume heterogeneous medical data in a low-cost, fast and secure database system. In this study, a medical Internet of Things (mIoT) platform that can use different database systems was realized in order to acquisition heterogeneous medical data from sensors to visualize and store data. Four different database models on mIoT platform were tested with four different scenarios. The performance of the database models used in the mIoT platform in these scenarios; query time, data preparation, flexibility, security and scalability parameters were compared. It was observed that the non-relational database model (NoSQL: Not only Structured Query Language) used on the mIoT platform was more efficient in reading/writing operations than relational database models, and its performance, flexibility and scalability were better than relational database systems.

Keywords: Medical IoT, database performance, SQL, NoSQL databases.

1. GİRİŞ (INTRODUCTION)

Nesnelerin İnterneti (Internet of Things - IoT), birbirine bağlı nesnelere oluşan, internet vasıtasıyla birbirleriyle haberleşen ve bilgi paylaşımı yapan bir ağ sistemi olarak tanımlanmaktadır. IoT, medikal tabanlı fonksiyonel uygulamaların geliştirilmesine uygundur. IoT sayesinde sağlık hizmetleri daha düşük maliyetle daha kaliteli olarak sunulabilmektedir [1]. Özellikle giyilebilir algılayıcı ve IoT teknolojilerinin kombinasyonu sağlık alanında devrim yapacak güce sahiptir. Giyilebilir algılayıcılar sayesinde doktorlar bir hastanın sağlık durumunu gerçek zamanlı olarak uzaktan

izleyebilmektedir[2]. Nesnelerin interneti uygulamalarının sağlık hizmetleri alanında kullanımı medikal IoT (medical Internet of Things - mIoT) olarak ifade edilmektedir.

mIoT uygulamalarının en önemli bileşenleri, algılayıcılar ile donatılmış ve İletim Denetim Protokolü/İnternet Protokolü (Transmission Control Protocol/İnternet Protocol - TCP/IP) tabanlı iletişim yeteneği kazandırılmış medikal cihazlardır. mIoT uygulamaları sayesinde vücut ısısı, kan basıncı, kan şekeri, kandaki oksijen (SPO2), hava akışı (nefes alma durumu), elektrokardiyogram (EKG) ve hasta konumu gibi medikal veriler toplanabilmektedir [3]. Uzmanlar, hasta tedavisine başlamadan önce hastanın hastalığına ilişkin geçmişten günümüze bütün bilgileri ayrıntılı olarak

*Sorumlu Yazar (Corresponding Author)
e-posta : polath@gazi.edu.tr

öğrenmek istemektedir. Bu bilgiler sayesinde uzmanlar, hastalık ve hastalığın ilerlemesi hakkında bilgi edinmektedir. Bu nedenle çoğu medikal veriler kişilerin sağlık durumunun uzun dönemli takibi ve yaşam kalitesinin artırılması için saklanmalıdır. Kişisel tıbbi kayıtlar, radyoloji görüntüleri, klinik test verileri, insan genetiği, nüfus bilgileri vb. çeşitli bilgiler sürekli olarak kayıt altına alınmaktadır. Kayıt altına alınan medikal verilerin hacmi sürekli artış göstermektedir. Dünya çapındaki medikal verilerin toplam hacminin 2012 yılında 500 petabayt 2020 yılında ise 25.000 petabayt olması beklenmektedir [4].

mIoT uygulamalarında heterojen medikal veriler birden fazla mIoT cihazından sürekli olarak üretilebilir. Bu verilerin görselleştirilmesi, inceleme ve analizi için bulut platformu üzerinde bir veritabanında depolanması gerekmektedir. mIoT uygulamalarında kullanıcıların istenilen medikal veriye kesintisiz, güvenilir ve hızlı bir şekilde ulaşılabilmesi esastır. Bu konudaki gereklilikleri yerine getirebilmek için heterojen ve hacimli medikal verilerin yüksek dağıtım kabiliyetine sahip uygun veritabanı modelleri kullanılmalıdır [4].

Birçok medikal uygulama, verileri depolamak için geleneksel ilişkisel veri tabanı modeli kullanmaktadır. Günümüzde ilişkisel veri tabanlarının büyük öneme sahip olması ve büyük projelerde kullanılması arka planda desteğinin çok iyi olmasındandır. Özellikle Oracle, IBM ve Microsoft gibi dev teknoloji firmalarının desteklemesi ile bu veri tabanları piyasada oldukça fazla yer tutmuştur. İlişkisel veritabanları şu ana kadar basitlik, sağlamlık, esneklik, ölçeklenebilirlik, performans ile birlikte tüm kullanıcı gereksinimlerini desteklemiştir. Fakat büyük hacimli ve heterojen veriler için ilişkisel veritabanlarının etkin bir şekilde veri işleme, ölçeklendirilme, paralel işlem ve maliyet bakımından bazı dezavantajları vardır. Özellikle veri miktarındaki önemli artış ilişkisel veri tabanlarının performansını düşürmektedir [5]. İlişkisel veri tabanlarında performansı arttırmak için yüksek maliyetli sunucu donanımı gerekmektedir. Bir başka husus ise ilişkisel veritabanlarının uygun olmayan statik şemaları, heterojen medikal veriler için en temel sınırlamadır.

İlişkisel veritabanı sistemlerine alternatif bir çözüm olarak ilişkisel olmayan NoSQL (Not only Structured Query Language) veritabanı kavramı ortaya çıkmıştır. NoSQL veritabanı modelleri veri formatından bağımsız ve yatay ölçeklenebilir olmasından dolayı büyük hacimli verileri depolamak için yüksek performans, düşük maliyet ve tek bir sunucu için daha fazla kullanıcı desteği sunmaktadır. Ayrıca NoSQL veritabanı modelleri, önceden tanımlanmış şema ve veriler arasında ilişki olmadan da çalışabilmektedir. Dahası kullanıcıların yapısal sorgu dili (Structured Query Language –SQL) hakkında bilgi sahibi olmaları da gerekmez. HBase, MongoDB, Cassandra ve CouchDB gibi birçok ilişkisel olmayan NoSQL veritabanı modelleri bulunmaktadır [5].

Yukarıda bahsedilen sebeplerden dolayı ilişkisel olmayan veri tabanlarının heterojen mIoT verilerinin

depolanması için maliyet/performans açısından ilişkisel veri tabanlarına göre avantaj sağlayabileceği düşünülmüştür.

Bu çalışmada, mIoT cihazlarının ürettiği heterojen medikal verilerin temel gereksinimleri göz önüne alınarak bir prototip mIoT platformu geliştirilmiştir. mIoT platformu ilişkisel ve ilişkisel olmayan farklı veritabanı modellerini kullanabilecek şekilde tasarlanmıştır. mIoT platformu üzerinde heterojen mIoT verileri kullanılarak ilişkisel ve ilişkisel olmayan dört veritabanı modeli farklı senaryolar ile test edilmiştir. Veritabanı modellerinin sorgu süresi, veri hazırlığı, esneklik, güvenlik ve ölçeklenebilirlikler göz önüne alınarak veritabanları karşılaştırılmıştır. Tüm test senaryolarında ilişkisel SQL ve ilişkisel olmayan NoSQL veritabanlarının heterojen mIoT verileri üzerinden performansları değerlendirilmiştir. Bu sayede mIoT uygulamalarında verimlilik ve performans açısından ilişkisel olmayan veri tabanlarının ilişkisel veritabanlarına göre ne tür avantajlar sağlayabileceği konusuna açıklık getirilmiştir.

1.1. Literatür özeti (Literature review)

Medikal verilerin kapsamlı analizi hasta veya hastalık hakkında karar verme sürecinin kalitesini arttırmaktadır. Ayrıca verilerin uzaktan ve sürekli takibi hastaya daha kolay müdahale edilmesini sağlamaktadır. mIoT ile ilgili yapılan çalışmalar incelendiğinde farklı hasta profiline hizmet eden birçok uzaktan sağlık izleme sisteminin geliştirildiği görülmüştür. Bu sistemlerin çoğunluğu hızlı veri elde etmek ve düşük maliyetli sistem tasarımı geliştirmeye yoğunlaşmaktadır [6].

mIoT temelli sistemler üzerine yapılan çalışmalar algılayıcılar sayesinde hasta üzerinden çeşitli verileri almaya ve bu verileri merkezi bir sunucu üzerinde saklamaya odaklanmıştır. Cruz J. ve arkadaşlarının yapmış oldukları çalışmada, Kronik Obstrüktif Akciğer Hastalığından (KOA) muzdarip hastaların uzaktan sağlık durumlarını izleme ve uzaktan müdahalelerinde kullanılan metotların kapsamlı bir tanımı sunulmuştur. KOA hastalarının farklı ortamlara ve hava şartlarına olan bağlılığı araştırılmıştır [7]. Kalp yetmezliği hastaları üzerine yapılan bir çalışma da ise hastaların uzaktan izlenmesinde kullanılan elde taşınabilir bir aygıt geliştirilmiştir Bu aygıt sayesinde hasta üzerinden her 15 saniyede bir çeşitli veriler bluetooth ile alınarak, internet bağlantısı olan bir cihaz sayesinde merkezi bir veritabanı sunucusuna gönderilmektedir [8]. Yunus Msayib ve arkadaşları yapmış oldukları bir çalışmada diz ağrıları olan hastaların durumlarını ve egzersiz çalışmalarını izlemek için bir uygulama geliştirmişlerdir. Hasta üzerinde bulunan ivmeölçer ve jiroskop algılayıcılarından alınan veriler internet yardımı ile merkezi bir sunucuya aktarılmaktadır. Diz bükülme açıları ve günde kaç defa büküldüğü bilgisi alınarak doktorlar tarafından değerlendirilmiştir [9].

Yeni doğan bebeklerin takibi [10], kalp rahatsızlıklarının izlenmesi ve tespiti [11-12], uzaktan kan şekeri seviyesinin izlenmesi ve erken uyarı sistemleri gibi

birçok mIoT tabanlı uygulama geliştirilmiştir [13]. Daha karmaşık bir çalışmada ise vücut hareketlerinin hayati bulguları arasında olan kalp hızı, solunum hızı, cilt sıcaklığı ve ivmeölçer ile elde edilen hareket verileri alınarak hasta sürekli olarak izlenmiştir. Bu verilerin yüksek veya düşük eşik değerlerine göre hasta ve ilgili kişilere uyarılar da bulunulmuştur [14].

Yapılan çalışmalar göz önüne alındığında uzaktan sağlık izleme sistemlerinin IoT uygulamalarının yaygınlaşması ile ivme kazandığı görülmektedir. IoT sayesinde uzaktan hasta izleme sistemlerinin geliştirilmesi kolaylaştırılmış, fonksiyonelliği ve uygulama sayısı artmıştır. Örneğin yüksek tansiyondan kaynaklanan kalp sorunları için IoT teknolojisine dayanan düşük maliyetli ve kullanımı kolay bir uzaktan izleme sistemi önerilmiştir. Bu çalışmada veriler Yapısal Sorgulama Dili (SQL: Structured Query Language) veritabanı olarak ta bilinen merkezi bir ilişkisel veritabanı sunucusunda saklanmıştır [15]. Diğer bir çalışmada ise yaşlı bir kişinin günlük faaliyetlerini kendi yaşantısını engellemeyecek şekilde tamamlamasına yardımcı olunması, aynı zamanda aile bireylerinin ve yaşlı bakım görevlilerinin yaşlı kişiyi takip etmesi için bir uygulama geliştirilmiştir [16].

mIoT uygulamaları sağladığı faydaların yanı sıra beraberinde bazı sorunları da beraberinde getirmiştir. Örneğin hasta üzerinden alınan ve büyük hacimlere ulaşan verilerin depolanması, entegrasyonu, yorumlanması ve yönetilmesi önemli bir sorun olarak karşımıza çıkmaktadır [17]. Medikal alanda geliştirilen uygulamalar incelendiğinde birçok çalışma da merkezi bir veritabanı sunucusunun yer aldığı ve bu verilerin merkezi bir veritabanına kaydedildiği görülmektedir. Ayrıca her uygulamanın farklı bir veritabanı yönetim sistemini kullanıldığı görülmüştür.

Birçok çalışmada kolay kullanımı ve ücretsiz olması nedeni ile MySQL ilişkisel veritabanı tercih edilmiştir [15-20]. Aynı zamanda, Microsoft SQL Server ilişkisel veritabanı [21-22], PostgreSQL ilişkisel veritabanı [23-24] ve SQLite ilişkisel veritabanını kullanan uygulamalara da rastlanılmıştır [25]. SQL'in hız ve performans etkisi üzerine çalışan bazı araştırmacılar ise dosya tabanlı yerel sunucular [6] ve nesne ilişkisel veritabanı üzerine çalışmalar yapmışlardır [26]. SQL, dosya tabanlı sunucular ve nesne ilişkisel veritabanlarının performansını araştıran bazı araştırmacılar ise NoSQL üzerine çalışmalar yapmış ve MongoDB kullanarak bir uygulama geliştirip MySQL ile performans karşılaştırması yapmışlardır [27-28]. Paethong ve arkadaşlarının gerçekleştirdiği çalışmada veritabanlarının karşılaştırılması Raspberry Pi üzerinde çalışan sistemler üzerinden gerçekleştirilmiştir. Veri ekleme, silme ve veri seçimi için MySQL ve MongoDB performansı karşılaştırılmıştır [27]. Çalışmada 1 milyon satır veri kullanılmış ve belirtilen test işlemleri zaman ekseninde değerlendirilmiştir. Poulter ve arkadaşlarının gerçekleştirdiği diğer bir çalışma da ise IoT

uygulamalarında kullanılan web servislerin JSON yapısını destekleyen MongoDB ile hız açısından daha performanslı olacağı belirtilmiştir [28]. Ayrıca MongoDB kullanan birçok uzaktan hasta izleme çalışması da yapılmıştır [5, 29-30].

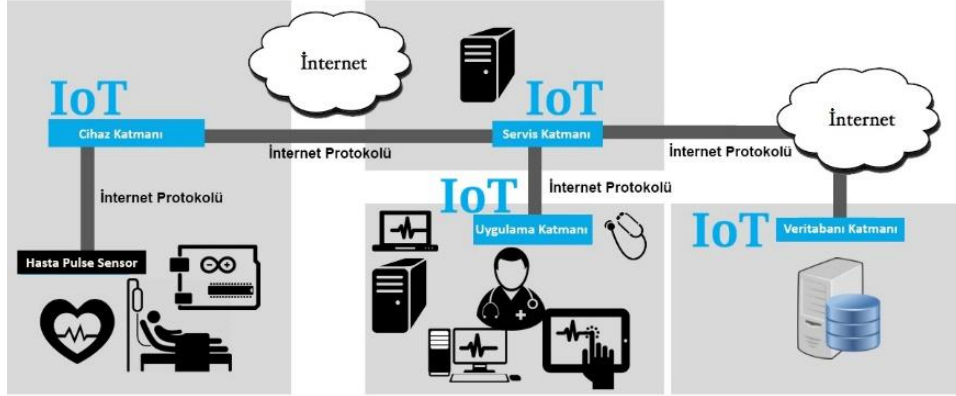
Literatürde uzaktan sağlık izleme alanında birçok farklı hasta profili üzerinde uygulamalar geliştirildiği görülmüştür. Bu uygulama alanlarında IoT teknolojisinin kullanılması ile uygulamaların işlevselliğinin artırıldığı bir gerçektir. Yapılan çalışmalarda her uygulamada farklı bir veritabanı sisteminin kullanıldığı ve mIoT uygulamalarında kullanılan veri tabanlarının etkinliği ve performansları üzerinde az sayıda çalışma yapıldığı görülmüştür.

2. mIoT PLATFORM UYGULAMASI (mIoT PLATFORM APPLICATION)

Bu çalışmada, heterojen mIoT verilerini kullanan ilişkisel ve ilişkisel olmayan veritabanlarının performans testlerini gerçekleştirmek için bir prototip mIoT platform uygulaması geliştirilmiştir. mIoT platform uygulaması, cihaz katmanı, servis katmanı, uygulama katmanı ve veritabanı olmak üzere dört katmanlı bir mimariden oluşmaktadır. Bu katmanların birbirleri ile olan ilişkisi Şekil 1'de gösterilmiştir.

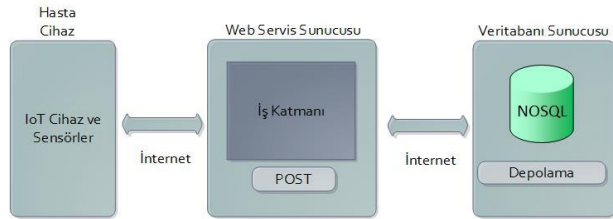
mIoT platform uygulamasının temelinde cihaz katmanı bulunmakta ve bu katman içerisinde algılayıcılar yer almaktadır (Şekil 1). Cihaz katmanı sürekli olarak servis katmanına yazma isteğinde bulunmaktadır. Servis katmanı kendisine gelen okuma veya yazma işlemlerini değerlendirmekte ve yazma işlemi yapacak ise verileri veritabanı katmanına göndererek yazma işlemi gerçekleştirmektedir. Aynı zamanda herhangi bir hastaya ait anlık ve geriye dönük analiz veya rapor istekleri de IoT servis katmanına yapılmaktadır. Uygulama katmanında ise kullanıcı ile etkileşime geçebilecek bir arayüz sayesinde hasta, sağlık personelleri veya hasta yakınları anlık olarak verileri izleyebilmektedir. Uygulama katmanı üzerinden gerçekleştirilen bütün istekler servis katmanında değerlendirildikten sonra işleme alınmaktadır. Kullanıcı tarafından yapılan isteğin niteliğine göre servis katmanı veritabanından veri talep etmekte ve kullanıcıya uygulama katmanı üzerinden istenilen bilgileri sunmaktadır. mIoT platformunda bir veya birden fazla algılayıcı sürekli olarak servis katmanına veri iletebilmektedir. Aynı zamanda bu veriler bir veya birden fazla kullanıcı tarafından farklı analiz ve rapor istekleri için talep edilebilir.

Şekil 2'de geliştirilen mIoT platform uygulamasının blok diyagramı gösterilmiştir. mIoT platform uygulamasında medikal verileri elde etmek için kalp hızı, solunum hızı, cilt sıcaklığı ve ivmeölçer gibi algılayıcılar kullanılmıştır. Bu algılayıcılar üzerinden ölçüm değerleri belirli zaman aralıklarında alınarak mIoT platformunun



Şekil 1. mIoT uygulaması genel mimarisi (mIoT application general architecture)

servis katmanına gönderilmektedir. Örneğin sıcaklık bilgisi her 30 saniyede bir, kalp hızı ölçüm bilgisi ise 60 saniyede bir mIoT platformunun servis katmanına gönderilmektedir. Algılayıcı verilerinin yanında hastaya ait kimlik bilgileri ve hastaya ait özet açıklama bilgisi de belirli zaman aralıklarında IoT platformunun servis katmanına gönderilmektedir. Verileri yazma ve okuma işlemleri web servis sunucusu tarafından yönetilmektedir. mIoT tabanlı cihaz üzerinden alınan verileri web servis sunucusuna göndermek için bir web istemci yazılımı geliştirilmiştir. Web servis sunucusunda ise gelen istekleri karşılamak ve değerlendirmek için RestFUL web servisleri geliştirilmiştir.



Şekil 2. Geliştirilen mIoT uygulaması (Developed mIoT application)

Mesajlaşma yapısı için ise RestFul web servisler ile sorunsuz çalışabilen JSON mesajlaşma yapısı kullanılmıştır. mIoT uygulamalarında hasta üzerinden veri toplayan mIoT destekli cihazlar arasında verilerin sürekli olarak paylaşılabilmesi gerekir. Paylaşılacak istenen veriler her sisteme ve her kullanıcıya standart bir yapıda sunulmalıdır. Bu amaç için Genişletilebilir İşaretleme Dili (XML: Extensible Markup Language) veri yapısı tercih edilebilir. Fakat yapısal, yarı yapısal ve yapısal olmayan verilerin çoğalması ve büyük verilerin devreye girmesi XML'in yerini Javascript Nesne Gösterimi (JSON: Javascript Object Notation)'a bırakmasına neden olmuştur. Popülaritesi gittikçe artan JSON özellikle kendi için özel tasarlanan web servislerle sorunsuz çalışmaktadır.

mIoT cihazlarından elde edilen verilerin veri tabanına JSON veya XML yapısında gönderilmesi gerekmektedir. Fakat SQL veri tabanına depolanmak üzere gelen bir veri, tablo yapısına dönüştürülerek kaydedilmektedir. Bu

durumda veriler, veri tabanına depolanırken veya veri tabanından başka bir platforma gönderilirken tablo yapısı – XML dönüşümü veya tablo yapısı - JSON dönüşümünün sürekli olarak yapılması gerekmektedir. XML dönüşümünün yapıldığı her uygulamada her bir veri için XML başlıklarının oluşturulması gerekmektedir. Bu durum bir veri depolanırken veya veri ağ üzerinde iletilirken boyutunun artmasına neden olmaktadır.

XML yapısında veri tanımlamalarda başlık etiketleri kullanılırken JSON yapısında bu işlem noktalama işaretleri ile basitçe yapılabilmektedir. Etiketleri oluştururken veya veri tabanı işlemleri yaparken XML dönüştürücüler kullanmak gerekir. Fakat JSON yapısı tercih edilirse, JSON yapısına uygun veri tabanı ve uygun web servislerin seçilmesi ile bu dönüşüm gereksinimi ortadan kalkar. Bu sayede veritabanı okuma/yazma ve iletişim hızını arttırmak mümkün olur.

mIoT platform uygulamasının veritabanı katmanında, ikisi açık kaynak kodlu olmak üzere toplam üç farklı ilişkisel veritabanı ve açık kaynak kodlu doküman tabanlı bir NoSQL veritabanı modeli, çeşitli senaryolarda testler gerçekleştirmek amacıyla kullanılmıştır. mIoT platformunda ilişkisel veritabanı olarak MySQL, PostgreSQL, MSSQL ve ilişkisel olmayan veritabanı olarak NoSQL doküman tabanlı MongoDB veri tabanı kullanılmıştır.

mIoT platformunun veritabanı performanslarının test edilmesi amacıyla seçilen veri tabanı modelleri şunlardır;

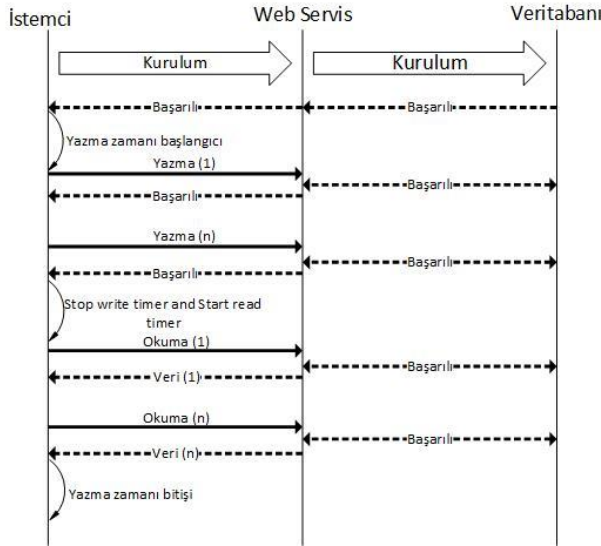
- PostgreSQL açık kaynaklı bir SQL veritabanıdır. 1995 yılından beri yoğun olarak kullanılmaktadır. SQL 2008 standardının çoğunu desteklemek üzere tasarlanmıştır. Ayrıca atomiklik, tutarlılık, izolasyon, dayanıklılık (Atomicity, Consistency, Isolation, Durability - ACID) uyumludur [31].
- MSSQL Microsoft firmasına ait açık kaynaklı olmayan bir veritabanıdır. Birçok kurumsal projede kullanılmaktadır. SQL standartların çoğunu desteklemektedir.
- MySQL açık kaynaklı bir SQL veritabanıdır. Dünya genelindeki en popüler açık kaynak

veritabanı olan MySQL, hızlı ve düşük maliyetli olma özelliğine sahiptir [32].

- NoSQL veri tabanlarının anahtar (anahtar/değer) depolama türü, belge depolama türü vb. çeşitleri bulunmaktadır. MongoDB doküman tabanlı ve açık kaynaklı bir veritabanıdır.

2.1 mIoT Platformu üzerinde veritabanlarının testi (Testing databases on mIoT platform)

Geliştirilen mIoT uygulamasında kullanılan web servis sunucusu, veritabanı sunucusu ve istemci yapısı sabit kalmış sadece veritabanı sistemi değiştirilmiştir. Böylelikle veritabanına ait performans değerlendirmesinde daha gerçekçi sonuçlar elde edilmiştir. Geliştirilen mIoT uygulamasında yapılacak test işlemleri çoklu okuma ve yazma işlemlerini destekleyecek niteliktedir. Şekil 3'te test işlemleri sırasında sistemin genel durumu gösterilmiştir.



Şekil 3. Test işlemleri sırasında sistemin genel durumu (General status of the system during testing transactions)

mIoT uygulaması için gerekli veriler ve veri yapıları Çizelge 1'de gösterilmiştir. Hastaya ait kimlik bilgileri ve hasta üzerinden yer alan çeşitli algılayıcılar sayesinde alınacak veriler ve veri yapıları açıkça belirtilmiştir. Bununla beraber algılayıcıların ölçüm zaman verileri de mIoT uygulamasına gönderilen veriler arasındadır.

Bu çalışma kapsamında gerekli olan bütün yazılımlar için kullanılan donanımlar ve özellikleri Çizelge 2'de gösterilmiştir. Test amaçlı kullanılan veritabanı sistemleri ve versiyonları ise Çizelge 3'te verilmiştir.

Geliştirilen mIoT uygulamasında web servis sunucusu ve kullanıcılar arasında gerçekleşen veri akış yapılandırması Şekil 4'te, web servis sunucusu ile veritabanı sunucusu arasında gerçekleşen veri akış yapılandırması ise Şekil 5'te gösterilmiştir.

Çizelge 1. mIoT uygulamasında kullanılan veri yapıları (Data structures used in mIoT application)

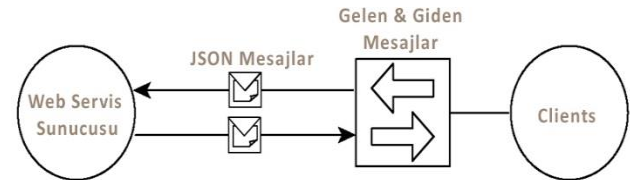
| İsim | Tip |
|----------------|--------|
| Hasta ID | Uuid |
| Hasta Açıklama | String |
| Kalp hızı | Double |
| Solunum hızı | Double |
| Cilt sıcaklığı | Double |
| İvmeölçer | Double |
| Ölçüm zamanı | Date |

Çizelge 2. mIoT uygulamasında kullanılan donanımlar ve özellikleri (Hardware and features used in mIoT application)

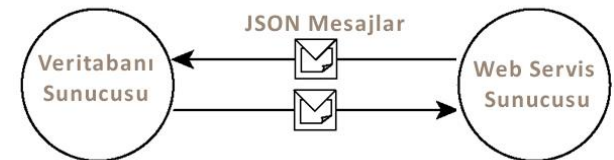
| Donanım Tipi | Web Servis Sunucusu | Veritabanı Sunucusu |
|---------------------|---------------------|---------------------|
| CPU Tipi | Intel i5 | Intel i5 |
| CPU Çekirdek Sayısı | 4 | 4 |
| CPU Hızı | 3.8 GHZ | 3.8 GHZ |
| Bellek Boyutu | 6 GB | 6 GB |
| Disk Boyutu | 1 TB | 1 TB |
| Disk Dönüş Hızı | 7200 RPM | 7200 RPM |
| Ağ Hızı | 1 GB/s | 1 GB/s |

Çizelge 3. Veritabanı sistemleri ve versiyonları (Database systems and versions)

| PostgreSQL | MySQL | MongoDB | MSSQL |
|--------------|-----------------|-----------------|---------------|
| Versiyon 9.3 | Versiyon 5.7.18 | Versiyon 3.2.15 | Versiyon 2014 |



Şekil 4. Web servis sunucusu ve kullanıcılar arasındaki veri akış yapılandırması (Data flow configuration between web service server and users)



Şekil 5. Web servis sunucusu ve veritabanı arasındaki veri akış yapılandırması (Data flow configuration between web service server and database)

Veri akışı sağlandığında veritabanı sunucusundaki veritabanı yönetim sistemi değiştirilerek test işlemleri başarılı bir şekilde tamamlanmıştır. Bunun için sadece bir

bölümden oluşan bir diske sahip veritabanı sunucusu kullanılmıştır. Veritabanı sunucusun yapılandırılması her veritabanı yönetim sisteminin kendi özel durumlarına göre gerçekleştirilmiştir.

3. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

Bu çalışmada geliştirilen mIoT platform uygulamasının üzerinde, ilişkisel ve ilişkisel olmayan veritabanı modellerinin performansı çeşitli işlemlerdeki sorgu zamanı ve yazma zamanı perspektifinde değerlendirilmiştir. Çalışma kapsamında hasta sayısı 40 adet olacak şekilde belirlenmiş ve bir ay içerisinde hastalar üzerinden alınabilecek veriler uygulamaya aktarılmıştır. Kullanıcılar üzerinden veri aktarımı sırasında bağlantı 24 saat aktif durumdadır. Her bir hasta üzerinden 1 milyon satır veri toplanmıştır. Ayrıca veri hazırlama, esneklik ve ölçeklenebilirlik konuları tartışılmıştır.

3.1. Okuma / Yazma Zamanı (Read / Write Time)

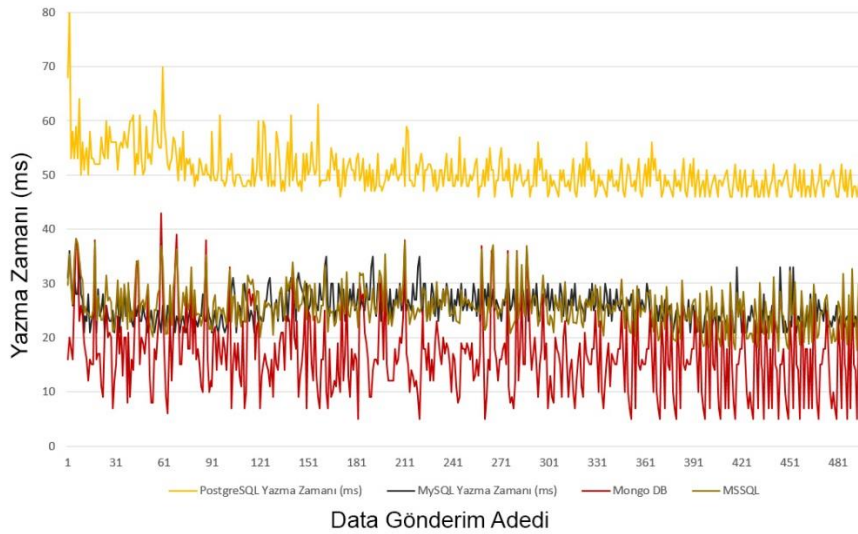
mIoT uygulama verileri için bu çalışmada kullanılan veri tabanlarının performans testleri dört farklı senaryo gerçekleştirilmiştir. Web servis sunucusuna yapılacak olan okuma ve yazma isteklerindeki bu senaryolar şunlardır;

- Tek bir hastaya ait sürekli verilerin yazılması
- Tek bir hastaya ait verilerin okunması
- Birden fazla hastaya ait sürekli verilerin aynı anda yazılması
- Birden fazla hastaya ait verilerin aynı anda okunması

Bu çalışmada kullanılan dört farklı veritabanı (PostgreSQL, MySQL, MongoDB ve MSSQL) için yukarıda belirtilen dört test işlemi senaryosu fiziksel sunucular üzerinde gerçekleştirilmiştir. Kullanılan veri tabanları için belirtilen test senaryolarına ait performans değerlendirilmesi elde edilmiştir.

3.1.1. Tek bir hastaya ait verilerin sürekli yazılması (Continuous writing of data for a single patient)

Algılayıcılar üzerinden gerekli verileri almak için belirli bir süre geçmektedir. Örneğin kalp atış hızını tespit edebilmek için en az 60 sn. geçmesi gerekmektedir. Bu nedenle algılayıcılar üzerinden alınacak veriler 60 sn. ve üzeri bir zamanda web servis sunucusuna gönderilmelidir. Bu senaryoda gerçekleştirilen test işlemlerinde veriler her 75 sn.de bir web servis sunucusuna gönderilmiştir. Test işlemi sonucunda elde edilen verilere ait grafik Şekil 6'da gösterilmiştir.



Şekil 6. Tek bir hastaya ait verilerin sürekli yazılması (Continuous writing of data for a single patient)

Şekil 5'te görüldüğü gibi MongoDB test edilen diğer veritabanı sistemlerine göre üstünlük göstermektedir. MongoDB veritabanına verilerin yazılması sırasında anlık dalgalanmalar olmaktadır. Örneğin bir önceki verinin 38 ms'de yazma sonucu gönderilirken bir sonraki veri de 8 ms 'de yazma işlemi gerçekleştirilmektedir. MongoDB veritabanında ki bu dalgalanmaya karşın diğer veri tabanlarına göre yazma performansı daha yüksektir.

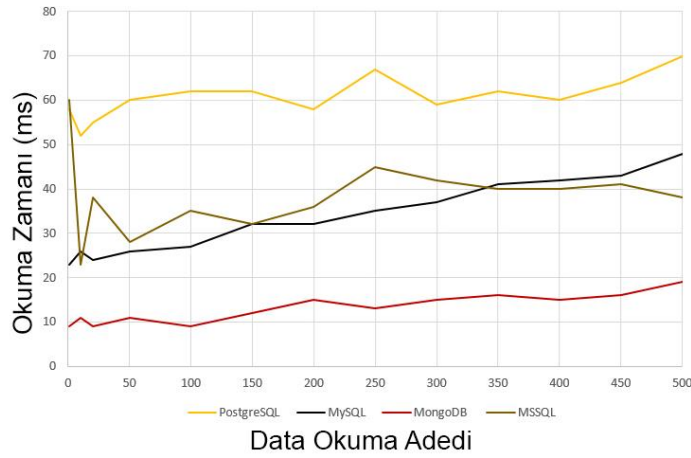
Yazma işlemlerindeki zaman dalgalanması ise en düşük PostgreSQL veritabanında görülmektedir.

3.1.2. Tek bir hastaya ait verilerin okunması (Reading the data of a single patient)

Bu test aşamasında tek bir hastaya ait medikal veriler sadece bir kullanıcı tarafından okunması istenmektedir. Oluşturulan bu senaryoda veri tabanlarının okuma performansı test edilerek elde edilen verilere ait değerler Şekil 7'de gösterilmiştir.

Şekil 7’de görüldüğü gibi MongoDB veritabanından istenilen veri sayısı artmasına karşın okuma zamanındaki performansı düşmemiştir. MongoDB’ye en yakın 23

ms’nin üzerinde okuma işlemlerini gerçekleştirirken MySQL veritabanı yaklaşmaktadır. MongoDB okuma işlemlerini 10 -19 ms arasında gerçekleştirmektedir.

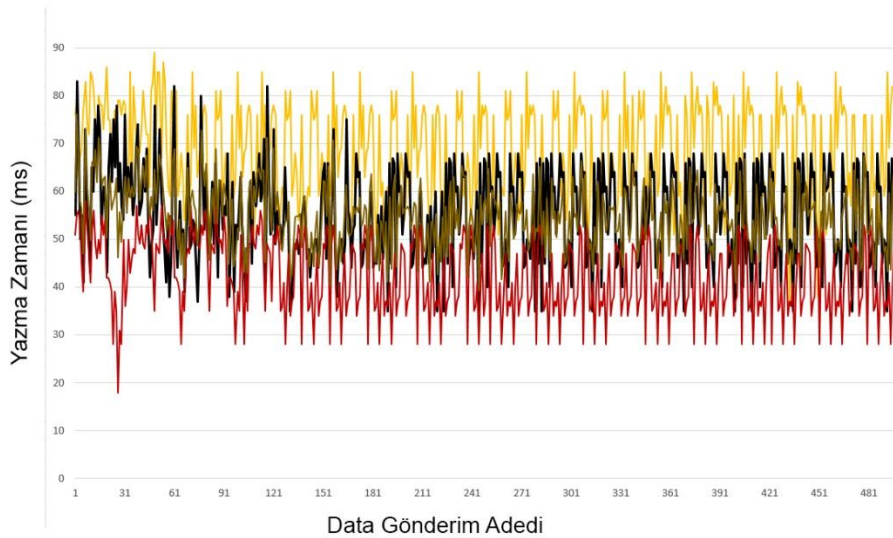


Şekil 7. Tek bir hastaya ait verilerin okunması (Reading the data of a single patient)

3.1.3. Birden fazla hastaya ait sürekli verilerin aynı anda yazılması (Simultaneous writing of data from more than one patient)

Bu test senaryosunda birden fazla hastaya ait sürekli verilerin aynı anda yazılması istenmektedir. Bu nedenle

web servis sunucusuna aynı zaman dilimlerinde farklı hasta verilerinin yazılması isteği yapılmıştır. Bu durumda mIoT uygulamasının cevap verme süresine ait bilgiler Şekil 8’de gösterilmiştir.



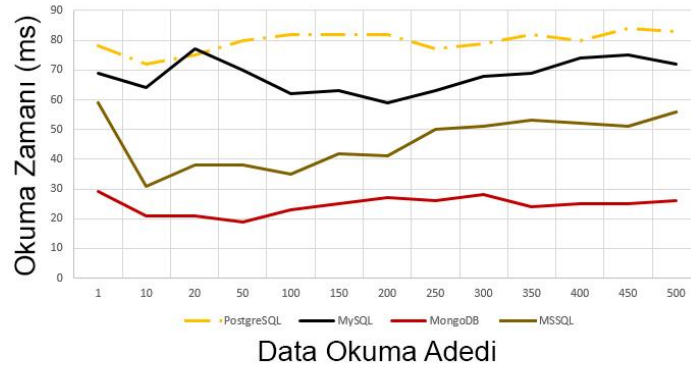
Şekil 8. Birden fazla hastaya ait sürekli verilerin aynı anda yazılması (Simultaneous writing of data from more than one patient)

Sürekli verilerin aynı anda yazılma isteğinde MongoDB test edilen diğer veritabanı sistemlerine göre daha hızlı cevap vermiştir. Bu durum MongoDB veritabanı sisteminin birden fazla kullanıcısı olan ve yazma işlemlerinin sürekli olduğu uygulamalarda daha hızlı sonuçlar verdiğini göstermektedir. MongoDB mIoT uygulaması üzerine aynı anda gelen birden fazla hastaya ait verilerin yazılması işlemlerini 16-55 ms arasında yapmaktadır. Diğer veritabanı sistemlerinin bu senaryo altında yazma süreleri karşılaştırıldığında MongoDB

veritabanının diğerlerine göre üstünlük sağladığı görülmektedir.

3.1.4. Birden fazla hastaya ait verilerin aynı anda okunması (Simultaneous reading of data of more than one patient)

Bu test aşamasında birden fazla hastaya ait medikal verilerin birden fazla kullanıcı tarafından okunması istenmektedir. Oluşturulan bu senaryoda veri tabanlarının okuma performansı test edilmiş ve Şekil 9’daki grafik elde edilmiştir.



Şekil 9. Birden fazla hastaya ait verilerin aynı anda okunması (Simultaneous reading of data of more than one patient)

mIoT uygulamasından birden fazla hastanın verilerini isteyen birden fazla kullanıcı olduğu düşünülmektedir. Bu durumda MongoDB test edilen diğer veri tabanlarına göre üstünlük göstermektedir. MongoDB birden fazla hastaya ait verilerin aynı anda okunması isteğini 18-30 ms arasında cevaplamaktadır.

3.2. Veri Hazırlama (Preparing Data)

Veri hazırlama tabloların tanımı, birincil anahtarlar ve yabancı anahtarlar, tablolar arasındaki ilişkilerin oluşturulması ve veri normalleştirilmesi gibi faaliyetleri içermektedir. Bu süreçler bir ilişkisel veritabanı için gereklidir. Fakat NoSQL veri tabanları şema içermeyen bir veritabanı modelidir ve satır verileri, bunlara uymaya gerek kalmadan doğrudan veri tabanlarına aktarılır. Bu nedenle NoSQL veri tabanları geleneksel veri tabanlarından daha avantajlıdır [4]. NoSQL veri tabanlarında uygulama geliştirme zamanını önemli ölçüde azaltılmıştır.

3.3. Esneklik (Flexibility)

Biçim tanımları, bir medikal sistemi içinde sıklıkla değiştirilir. Sistem önceki bölümlerde açıklandığı gibi verileri yeni formatlarda depolayabilmelidir. Geleneksel veri tabanları statik bir nitelik taşımakta ve başlangıçta başlangıç biçimlerini tanımlamayı gerektirmektedir. Veri modelleme burada izin verilen sayıda sütunla sınırlıdır. Dolayısıyla, formların yeniden tanımlanması ilişkisel veritabanı modelinde bir zorluktur. Buna karşılık, doküman tabanlı veri tabanları, şema gerektirmeden esnek bir yapıya sahiptir. İçerik değiştiğinde, şema yeniden tasarlanmadan veriler eklenebilir.

3.4 Ölçeklenebilirlik (Scalability)

Medikal hizmetlerdeki verilerin sürekli büyümesinden dolayı, medikal uygulamalarda genişletilebilir modeller kullanmak gerekir. İlişkisel veritabanı büyük miktarda veri için planlanmamıştır. Ayrıca, günlük bir ortamda çalışması zordur. Bu nedenle, ölçeklenebilirlik perspektifinden, ilişkisel veri tabanları medikal uygulamalar için doğru model değildir. Buna karşılık,

NoSQL veri tabanları dağıtılmış şekilde büyük miktarda veri için potansiyel olarak hazırlanmıştır. Veriler, birkaç coğrafi bölgedeki farklı makinelere doğrudan yayılabilir. NoSQL veri tabanları bir bulut ortamında ölçeklenebilirliği sağlayan önemli bileşenlerden biridir.

3.5. Güvenlik (Security)

Büyük boyutlara ulaşan medikal verilerin, farklı veritabanlarında parçalanarak saklanması işleminde ağ güvenliği önemlidir. Veritabanı ağ katmanında ilişkisel veritabanı modellerinin çoğu Transfer Katmanı Güvenliği (TLS: Transport Layer Security) protokolünü desteklemektedir. NoSQL kavramı ilk ortaya çıktığında güvenlik gereksinimleri çoğu protokolü desteklememiştir. Fakat günümüzdeki mevcut NoSQL veritabanı sürümleri TLS ve Anahtar Yönetimi Birlikte Çalışabilirlik Protokolü (KMIP: Key Management Interoperability Protocol) protokollerini desteklemektedir. Güvenli kimlik denetimi için Kerberos büyük istemci/sunucu kimlik doğrulama endüstri standardını kullanmaktadır. Ayrıca Sanal Özel Ağlar (VPN: Virtual Private Networks) kullanımı ile iki ağın şifreli veya sınırlı erişimli bir ağ modeli üzerinden veritabanına bağlanmasını izin vermektedir.

NoSQL veritabanı modelleri, ilişkisel veritabanı modellerinin sağladığı birçok güvenlik alt yapısını bünyesinde barındırmaktadır. Fakat her bir güvenlik yapılandırması ayrı bir uzmanlık alanı gerektirmektedir. Genellikle NoSQL kullanıcıları, performans avantajları için IPSEC protokolleri yerine SSL'yi kullanmayı tercih etmektedirler. Verilerin şifreli saklanması veya paylaşılması için ise gelişmiş sürümlerinde yer alan hazır şifreleme modülleri kullanılmaktadır.

Doküman tabanlı model kullanan veritabanı sistemlerinde ise verilerinin şifrelenmesinin birden fazla yolu vardır. Şifreleme uygulama düzeyinde veya harici dosya sistemi düzeyinde disk şifreleme yöntemi ile şifrelenebilir. Fakat bu yaklaşımlar mevcut modele maliyet, performans yükü ve operasyonel karmaşıklık katacaktır.

4. SONUÇ VE TARTIŞMA (RESULT and DISCUSSION)

Bu çalışmada, mIoT verilerinin yönetimiyle ilgili mevcut zorluklar göz önüne alınarak, doküman tabanlı veritabanı yönetimi ve bulut bilişim'e dayanan bir model sunulmuştur. Bu modelin ilişkisel modellere kıyasla performansını değerlendirmek için bir mIoT uygulaması geliştirilmiştir. Sorgu zamanı ele alındığında, doküman tabanlı NoSQL veritabanı modeli, test edilen diğer veritabanı modellerine göre yazma/okuma işleminde hızlı davranmıştır.

Veri hazırlığı göz önüne alındığında, yeni model hazırlık gerektirmeden yarı yapılandırılmış veya yapılandırılmamış verileri saklama kapasitesine sahip olarak işlem gerçekleştirmiştir. Medikal veriler için ilişkisel veri tabanlarında çok fazla alan ve tablo oluşturmak gerekir. Yeni modelin eski modele göre daha fazla esnekliği bulunmaktadır. Yeni modelin ölçeklenebilirliği yükseken, eski modelin bu konuda bazı sınırlamaları vardır.

Mevcut çalışma, mIoT uygulamalarında medikal verilerinin depolanması ve işlenmesi için etkin bir çözüm sağlamıştır. Farklı bağlamlarda yapılan kapsamlı değerlendirmeler, çözümün etkililiğini göstermiştir. Bu modelde, kullanıcıların veri güvenliğine ve gizliliğine güvenmesi önemli bir konudur. Mevcut NoSQL sürümlerinde sağlanan güvenlik alt yapıları yeterli güvenliğin sağlanmasında önemli rol oynamaktadır. Fakat güvenlik seviyesi artırıldığında yaşanan performans kaybı göz ardı edilmeyecek derecededir. Bu nedenle güvenlik/performans konusu ele alınarak bulut ortamında çeşitli güvenlik yaklaşımları araştırılmalı ve mIoT uygulamaları için performansı ve güvenliği yüksek yeni modeller önerilmelidir.

Gelecek çalışmalarda medikal alanda sıklıkla kullanılan veri yapıları ve modellerinin belirlenmesi için çalışmalar yapılacaktır. Bu işlem için saha araştırması gerçekleştirilecektir. Saha araştırması sonucunda belirlenen veri yapıları tez kapsamında belirtilen test senaryolarına uygulanacak ve sonuçlar değerlendirilecektir. Sadece yazı veya numara değerleri alabilen veri yapıları değil aynı zamanda resim, video ve ses gibi farklı veri yapıları ile veritabanı modelleri test edilecektir.

KAYNAKLAR (REFERENCES)

- [1] Evans P. and Annunziata M., "Industrial Internet: Pushing the Boundaries of Minds and Machines", *Research Report*, (2012).
- [2] Khan S. F., "Health care monitoring system in Internet of Things (IoT) by using RFID", *2017 6th Int. Conf. Ind. Technol. Manag*, Cambridge, 198–204, (2017).
- [3] Almotiri S. H., Khan M. A. and Alghamdi M. A., "Mobile health (m-Health) system in the context of IoT", *2016 4th Int. Conf. Futur. Internet Things Cloud Work. W-FiCloud*, Vienne, 39–42, (2016).
- [4] Goli-Malekabadi Z., Sargolzaei-Javan M. and Akbari M. K., "An effective model for store and retrieve big health

- data in cloud computing", *Comput. Methods Programs Biomed*, 132, 75–82, (2016).
- [5] Kang Y., Park I., Rhee J. and Lee Y., "MongoDB-based Repository Design for IoT-generated RFID / Sensor Big Data", *IEEE Sens. J.*, 16(2): 485–497, (2016).
- [6] Archip A., Botezatu N., SSerban E., Herghelegiu P. C. and Zal A., "An IoT based system for remote patient monitoring", *17th Int. Carpathian Control Conf. ICC*, Tatra, 1–6, (2016).
- [7] Cruz J., Brooks D. and Marques A., "Home telemonitoring in COPD: A systematic review of methodologies and patients' adherence", *Int. J. Med. Inform.*, 83(4): 249–263, (2014).
- [8] Pollonini L., Rajan N. O., Xu S., Madala S. and Dacso C. C., "A novel handheld device for use in remote patient monitoring of heart failure patients-design and preliminary validation on healthy subjects", *J. Med. Syst.*, 36(2): 653–659, (2012).
- [9] Msayib Y., Gaydecki P., Callaghan M., Dale N. and Ismail S., "An Intelligent Remote Monitoring System for Total Knee Arthroplasty Patients", *J. Med. Syst.*, 41(6): 1–6, (2017).
- [10] Sotiriou D., "Health Care in Remote Areas", *Journal of Medical Systems*, 1: 69–76, (1995).
- [11] Kumar S., Bansal A., Tiwari V. N., Nayak M. M. and Narayanan R., "Remote health monitoring system for detecting cardiac disorders", *IEEE Int. Conf. Bioinforma. Biomed. IEEE BIBM*, Belfast, 30–34, (2014).
- [12] Harbawi M. A., Ibrahim M. I. and Motakabber S. M. A., "Photoplethysmography based remote health monitoring system", *IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA*, Kuala Lumpur, 26–27, (2013).
- [13] Jameson R., Lorence D. and Lin J., "Data capture of transdermal glucose monitoring through computerized appliance-based virtual remote sensing and alert systems", *J. Med. Syst.*, 36(4): 2193–2201, (2012).
- [14] Babu S., Chandini M., Lavanya P., Ganapathy K. and Vaidehi V., "Cloud-enabled remote health monitoring system", *Recent Trends Inf. Technol. (ICRTIT)*, Chennai, 702–707, 2013.
- [15] Kirtana R. N. and Lokeswari Y. V., "An IoT based remote HRV monitoring system for hypertensive patients", *Comput. Commun. Signal Process. Spec. Focus IoT, ICCSP*, Tamilnadu, (2017).
- [16] Al-Adhab A., Altmimi H., Alhawashi M., Alabduljabbar H., Harrathi F. and AlMubarek H., "IoT for remote elderly patient care based on Fuzzy logic", *Networks, Comput. Commun.*, Hawaii, 1–5, (2016).
- [17] Raghupathi W. and Raghupathi V., "Big data analytics in healthcare: promise and potential", *Heal. Inf. Sci. Syst.*, 2(1): 3, (2014).
- [18] Xiaomin K., Fan B., Nie W. and Yi D., "Design on mobile health service system based on Android platform", *IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. IMCEC*, Xi'an, 1683–1687, (2017).
- [19] Degada A. and Savani V., "Design and implementation of low cost, portable telemedicine system: An embedded technology and ICT approach", *NUiCONE 2015 - 5th Nirma Univ. Int. Conf. Eng.*, Ahmedabad, (2016).
- [20] Sharwardy S. N., Rahman Z., Parveen S., Sarwar H. and Hossain A. M., "A cost-effective web-based teleconsultation system", *8th Int. Conf. Inf. Technol. Asia - Smart Devices Trend Technol. Futur. Lifestyle, Proc. CITA*, Xi'an, 2–5, (2013).

- [21] Hu P., Ning H., Qiu T., Xu Y., Luo X. and Sangaiah A. K., "A unified face identification and resolution scheme using cloud computing in Internet of Things", *Futur. Gener. Comput. Syst.*, 81(1): 582–592, (2018).
- [22] Maurya A. and Bade D. S., "Design of a wireless health monitoring system based on M2M communication", *Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT*, Kanyakumari, 949–953, (2014).
- [23] Pambudi I. T., Hayasaka T., Tsubota K., Wada S. and Yamaguchi T., "Sustainable patient information network (SPaIN) for primary care health center in Indonesia", *25th Annu. Int. Conf. Ieee Eng. Med. Biol. Soc. Vols 1-4 A New Begin. Hum. Heal.*, 25(1): 1421–1424, (2003).
- [24] Sanborn M., "Director's Forum - Developing a Pharmacy Information System Infrastructure", *Hosp. Pharm.*, 42(5), 470–473, (2007).
- [25] Hussain A., Wenbi R., Da Silva A. L., Nadher M. and Mudhish M., "Health and emergency-care platform for the elderly and disabled people in the Smart City", *J. Syst. Softw.*, 110: 253–263, (2015).
- [26] Talamino-Barroso A., Estudillo-Valderrama M. A., Roa L. M., Reina-Tosina J. and Ortega-Ruiz F., "A Machine-to-Machine protocol benchmark for eHealth applications - Use case: Respiratory rehabilitation", *Comput. Methods Programs Biomed.*, 129: 1–11, (2016).
- [27] Paethong P., Sato M. and Namiki M., "Low-power distributed NoSQL database for IoT middleware", *5th ICT Int. Student Proj. Conf. ICT-ISPC*, Nakhon Pathom, 158–161, (2016).
- [28] Poulter A. J., Johnston S. J. and Cox S. J., "Using the MEAN stack to implement a RESTful service for an Internet of Things application", *IEEE World Forum Internet Things*, Milan, 280–285, (2015).
- [29] Farahani B., Firouzi F., Chang V., Badaroglu M., Constant N. and Mankodiya K., "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare", *Futur. Gener. Comput. Syst.*, 78: 659–676, (2018).
- [30] Nkenyereye L. and Jang J. W., "Performance Evaluation of Server-side JavaScript for Healthcare Hub Server in Remote Healthcare Monitoring System", *Procedia Comput. Sci.*, 58(1): 382–387, (2016).
- [31] Van Der Veen J. S., Van Der Waaij B. and Meijer R. J., "Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual", *IEEE Fifth Int. Conf. Cloud Comput.*, Nice, 431–438, (2012).
- [32] Wei-Ping Z., Ming-Xin L. and Huan C., "Using MongoDB to implement textbook management system instead of MySQL", *IEEE 3rd Int. Conf. Commun. Softw. Networks, ICCSN*, Xi'an, 303–305, (2011).
- [33] Gyorodi C., Gyorodi R., Pecherle G. and Olah A., "A comparative study: MongoDB vs. MySQL", *13th Int. Conf. Eng. Mod. Electr. Syst. EMES*, Oradea, 0–5, (2015).