T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

KULLANIMDAKİ VERİLERDE VERİ SIZINTISININ ÖNLENMESİ İÇİN AJAN TABANLI BİR ÇÖZÜM

Veyis ŞEN

DANIŞMAN
Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği, Tezli Yüksek Lisans Programı

Aralık, 2024

TEZ KABUL VE ONAYI

Veyis ŞEN tarafından, **Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN** danışmanlığında hazırlanan **" KULLANIMDAKİ VERİLERDE VERİ SIZINTISININ ÖNLENMESİ İÇİN AJAN TABANLI BİR ÇÖZÜM "** başlıklı bu çalışma, jürimiz tarafından **18/12/2024** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Tez Jürisi

		İmza	**Sonuç**
DANIŞMAN	Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN İstanbul Üniversitesi-Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Prof. Dr. Xxxx XXXX ..... Üniversitesi ..... Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Prof. Dr. Xxxx XXXX ..... Üniversitesi ..... Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Prof. Dr. Xxxx XXXX ..... Üniversitesi ..... Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Prof. Dr. Xxxx XXXX ..... Üniversitesi ..... Anabilim Dalı		☐ Kabul ☐ Ret

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve bilimsel etik kuralları içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını ve her türlü hukuki sorumluluğu aldığımı kabul ederim.

Veyis ŞEN

Sevgili eşime ithaf ediyorum…

BÜTÇE DESTEKLERİ

KULLANIMDAKİ VERİLERDE VERİ SIZINTISININ ÖNLENMESİ İÇİN AJAN TABANLI BİR ÇÖZÜM

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

TEŞEKKÜR

Bu çalışma süresince karşılaştığım zorluklarda desteklerini esirgemeyen kıymetli danışman hocam Sayın Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN'a ve bu tez çalışmasının başlangıcından teslimine kadar desteğini ve yardımını yanımda hissettiğim çok kıymetli hocam Sayın Prof. Dr. Muhammed Ali AYDIN'a tüm kalbimle teşekkür ederim.

Ayrıca lisansüstü eğitimi almam konusunda beni teşvik eden ve takıldığım konularda destek veren değerli dostum Murat KAZANÇ'a ve yoğun tez çalışmamda destekleyici sözleriyle hep yanımda olan eşime teşekkür ederim.

Aralık 2024 Veyis ŞEN

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

σ : Sigmoid aktivasyon fonksiyonu
ft : Unutma kapısının çıktısı
Wf : Unutma kapısının ağırlıkları
ht : Gizli durum
Xt : Mevcut fonksiyon girişi

Kısaltmalar **Açıklama**

BDM : Büyük Dil Modeli
DLP : Data Leakage Prevention (Veri Sızıntısı Önleme)
KVKK : Kişisel Verilerin Korunması Kanunu
GDPR : General Data Protection Regulation (Genel Veri Koruma Düzenlemesi)
RAM : Random Access Memory (Rastgele Erişimli Bellek)
URL : Uniform Resource Locator (Tekdüzen Kaynak Bulucu)
TBMM : Türkiye Büyük Millet Meclisi
NLP : Natural Language Processing (Doğal Dil İşleme)
GPT : Generative Pre-trained Transformer (Üretken Ön İşlemeli Dönüştürücü)
LSTM : Long Short-Term Memory (Uzun Kısa Süreli Bellek)
BERT : Bidirectional Encoder Representations from Transformers (İki Yönlü Dönüştürücü Tabanlı Kodlayıcı Temsilleri)
LLM : Large Language Model (Büyük Dil Modeli)
MLM : Masked Language Models (Maskeli Dil Modeli)
LSB : Least Significant Bit (En Az Önemli Bit)
DLL : Dynmic Link Library (Dinamik Link Kütüphanesi)
JSON : JavaScript Object Notation (JavaScript Nesne Gösterimi)
UTC : Coordinated Universal Time (Eşgüdümlü Evrensel Zaman)
TAI : Uluslararası Atomik Zaman
DBSM : Dictionary Based String Matching (Sözlük Tabanlı Kelime Eşleştirme)

DFA : Deterministic Finite Automata (Deterministik Sonlu Otomat)

API : Application Programming Interface (Uygulama Programlama Arabirimi)

ÖZET

YÜKSEK LİSANS TEZİ

KULLANIMDAKİ VERİLERDE VERİ SIZINTISININ ÖNLENMESİ İÇİN AJAN TABANLI BİR ÇÖZÜM

Veyis ŞEN

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği, Tezli Yüksek Lisans Programı

Danışman : Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN

Kamu kurumları başta olmak üzere her kuruluşun, özellikle iç yazışmalar ve haberleşmelerde kullandıkları verilerin gizliğini sağlamaları, kurumun sağlıklı işleyişi ve kurumlar arası rekabetin sürdürülebilmesi için çok önemlidir. Veri gizliliğini sağlama konusu birçok bilimsel araştırmaya konu olmuştur ve bu çalışmalar sonucunda verinin bulunduğu duruma ve konuma göre çeşitli çözüm önerileri getirilmiştir. Siber güvenlik kavramının önemini arttırmasıyla, bulunan ve önerilen tüm çözümlerin belirli periyotlarla güncellenmesi, esnekleşmesi ve hem yerel hem de küresel olarak uygulanabilirlik ölçeğinden geçirilmesi gerekmektedir. Halihazırda sunulan pek çok çözüm önerisi, yapay zekâ kavramın yaygınlaşması ve büyük dil modellerinin ortaya çıkması sonrası geçerliliğini yitirme noktasına gelmiştir. Literatürdeki çözümlerin Türkçe diline uygun olmaması da bir başka sorun olarak öne çıkmaktadır.

Bu çalışmada öncelikle veri sızıntısı önleme yöntemlerinden olan sözlük tabanlı yapı yerelleştirilerek Türkçe dil desteği ile uygulanmıştır. Kurum içi bilgisayarlara kurulan bir ajan yazılım kullanılarak, çalışanın yazışmaları sözlük sistemiyle anlık olarak kontrol edilmiş,

gerçek zamanlı bulut veri tabanı kullanılarak yine anlık olarak raporlama yapılmış ve sızıntı gerçekleştiği esnada önlem alınmasının sağlanması amaçlanmıştır. İkinci olarak, belirli bloklar halinde biriktirilen yazışmalar, yapay zekâ ile kişisel veri varlığını test etmek amacıyla büyük dil modeline gönderilmiş ve metin içerisinde hassas veri olma durumu tespit edilmeye çalışılmıştır. Büyük dil modelinin rol yapısı kullanılarak çalışmanın farklı sektörlere göre düzenlenebilmesi sağlanmış ve bu şekilde uygulamaya esneklik kazandırılmıştır. Büyük dil modelinin kişisel veri ihlali tespit etme durumu ve yasaklı kelime ihlal tespiti sistem yöneticisine raporlanmıştır.

Aralık 2024 , 68 sayfa.

Anahtar kelimeler: Veri Sızıntısı, Büyük Dil Modeli, Gerçek Zamanlı Veri Tabanı, Ajan Yazılım

ABSTRACT

M.Sc. THESIS

AN AGENT-BASED SOLUTION TO PREVENT DATA LEAKAGE IN-USE STATE

Veyis ŞEN

İstanbul University-Cerrahpaşa

Institute of Graduate Studies

Department of Computer Engineering

Computer Engineering

Supervisor : Assist. Prof. Dr. Gülsüm Zeynep GÜRKAŞ AYDIN

It is very important for every institution, especially public institutions, to ensure the confidentiality of the data they use, especially in internal correspondence and communication, for the healthy functioning of the institution and the sustainability of competition between institutions. The issue of ensuring data confidentiality has been the subject of many scientific studies and as a result of these studies, various solution proposals have been proposed depending on the status and location of the data. With the increasing importance of the concept of cyber security, all solutions found and proposed need to be updated periodically, become flexible and pass through the scale of applicability both locally and globally. Many solution proposals currently offered have come to the point of losing their validity after the widespread use of the concept of artificial intelligence and the emergence of large language models. Another problem is that the solutions in the literature are not suitable for the Turkish language. In this study, first of all, the dictionary-based structure, which is one of the data leakage prevention methods, was localized and implemented with Turkish language support. Using an agent software installed on in-house computers, the employee's correspondence was instantly checked with the dictionary system, and real-time reporting was made using a real-time cloud

database, and it was aimed to ensure that precautions were taken when a leak occurred. Secondly, correspondences collected in certain blocks were sent to the large language model in order to test the existence of personal data with artificial intelligence and an attempt was made to detect the presence of sensitive data in the text. The role structure of the large language model was used to ensure that the study could be organized according to different sectors and thus flexibility was added to the application. The personal data violation detection status of the large language model and the prohibited word violation detection were reported to the system administrator.

December 2024, 68 pages.

Keywords: Data Leakage, Large Language Model, Real Time Database, Agent Software

1. GİRİŞ

Dijitalleşen dünyada veri toplamanın ve toplanılan verilerin anlamlı bilgilere dönüştürülmesinin gelişen teknoloji sayesinde kolaylaşmasına karşın elde edilen verilerin saklanması, yetkisiz kişilerin erişimine kısıtlanması ve korunması giderek zorlaşmakta ve yeni çözümlere açık hale gelmektedir. İnternetin gelişimi ve yaygınlaşmasından sonra hayatımıza giren mobilite kavramıyla beraber bireysel olarak sürekli taşınan ve özellikle paylaşım yapmak amacıyla kullanılan teknolojik cihazlar başta olmak üzere kamu kurumları, bankalar ve ticari kuruluşla gibi yapıların operasyonlarını sağlıklı bir şekilde sürdürebilmek için kurum içi verilerinin gizliliğini sağlamaları esastır.

Veri sızıntıları siber casusluk gibi bilinçli ya da kullanıcı ihmali veya hatası gibi bilinçsiz nedenlerden oluşabilir. Her iki neden kaynaklı sızıntılar için de literatürde çeşitli çalışmalar yapılmış ve muhtemel sızıntıların önüne geçilmeye çalışılmıştır ancak gelişen teknolojiyle beraber yeni siber güvenlik zafiyetleri ortaya çıktıkça, verilerin konumu gibi öznitelikleri değiştikçe örneğin veriler buluta taşındıkça [1], var olan çalışmaların güncellenmesi ve geliştirilmesi ayrıca yeni yöntemlerin tasarlanması gerekli olmuştur.

Yapay zekâ kavramı özellikle büyük dil modellerinin oluşturulması ve bilimsel çalışmaların yanı sıra son kullanıcıya hizmet verir hale getirilmesi sonrasında daha hızlı yaygınlaşmaya başlamıştır. Literatüre giren birçok çalışmada, küçük dil modellerinin geliştirilmesinin yanında büyük dil modeli de kullanılmaya başlanmıştır. Görüntü işlemeden matematiksel hesaplamalara, sağlıktan tarım çözümlerine kadar birçok alanda kullanılan büyük dil modeli, veri sızıntısı özelinde ise kendisine gönderilen veriler üzerinde, belirtilen alana göre kişisel veri olma durumunu tespit edebilmekte, bu verilerin yerel ve uluslararası kişisel veri koruma kanunlarına göre durumunu inceleyebilmekte ve elde ettiği sonuçları istenilen formatta iletebilmektedir [2].

Sohbet robotu kurgusuyla başlayan ve öğrendikleri ile yeni sonuçlar ortaya çıkarabilecek bir yapı haline dönüşen büyük dil modeli 2022 yılından itibaren halka açık kullanıma geçmiş ve büyük teknoloji firmalarından aldığı yatırımlarla gelişimine devam ederek sorulara doğala oldukça yakın cevaplar verir duruma gelmiştir [3].

Bu çalışmada ilk olarak veri sızıntısının önlenmesi konusunda daha önce yapılan akademik çalışmalarla ilgili kapsamlı bir literatür taraması yapılmış ve çalışmaların benzer ve farklı yönleri incelenerek ve bu çalışmayla karşılaştırılarak liste halinde sunulmuştur

Bu çalışmada, yeni bir büyük ve küçük dil modeli geliştirilmemiş, var olan ve birçok alanda kullanılan büyük dil modeli altyapısı veri sızıntılarının önlenmesi amacıyla kullanılmıştır. Veri sızıntıları tüm sektörlerin bir sorunudur ancak bu çalışma özellikle bankacılık sektörü üzerinde oluşması muhtemel veri sızıntılarının önlenmesine odaklanacaktır.

Bu çalışmada programlama dili altyapısı olarak en yaygın kullanılan işletim sistemi [4] olan Microsoft Windows ile uyumu ve sistem kaynaklarına erişmesi için güçlü kütüphaneler barındıran C# dili tercih edilmiştir. C# dilinin kütüphaneleri sayesinde, bu çalışmada kullanılan veri sızıntısı önleme metodu için gerekli bilgiler, donanım seviyesine inilerek toplanmıştır. C# kullanılarak arka planda çalışan ve kullanıcının bilgisayarı kullanması sürecinde herhangi bir şekilde çalışmayı engellemeyen bir yapı hazırlanmıştır [5].

Bu tez çalışmasının organizasyonu şu şekildedir:

Kavramsal çerçeve bölümünde veri ve veri sızıntısı kavramları açıklanmıştır. Kişisel verilerin ve ticari sır niteliği taşıyan bilgilerin korunması için Türkiye'de ve uluslararası hukukta yürürlükte olan yasalardan bahsedilmiştir. Yapay zekâ teknolojisinin gelişimi ve kullanım alanları belirtilmiştir. Yapay zekâ kavramının alt dalları olan makine öğrenmesi ve yapay sinir ağları konuları hakkında bilgi verilmiştir. İlgili çalışmalar bölümünde ise veri sızıntılarını önleme amaçlı yöntemleri inceleyen akademik çalışmalar incelenmiştir.

Yöntem bölümünde, yapılan uygulamanın aşamaları detaylandırılmış ve izlenen yol belirtilmiştir. Yol haritası olarak yönetici yazılımın kodlanması, ajan yazılımının kodlanması, ajan yazılımın donanım kaynaklarına erişerek veri toplaması ve gerçek zamanlı veri tabanındaki sözlük sistemiyle karşılaştırma yapması, büyük dil modeli ile iletişime geçilmesi ve veri sızıntısı kontrolü şeklinde bir planlama yapılmıştır.

Bulgular bölümünde, uygulama örnek verilerle test edilmiş ve sonuçların raporlaması sonucunda elde edilen veriler değerlendirilmiştir.

Tartışma bölümünde, literatür incelemesi sonucunda elde edilen sonuçlar ile bu çalışmanın sonuçları karşılaştırılmıştır.

Sonuç bölümünde çalışmanın sonuçları değerlendirilmiş, veri sızıntısı alanı ile ilgili gelecek araştırmacılara tavsiyelerde bulunulmuş ve yapılabilecek çalışmalardan söz edilmiştir.

2. KAVRAMSAL ÇERÇEVE

Bu bölümde, çalışmada söz konusu edilen veri, yapay zekâ ve büyük dil modeli gibi konulardan bahsedilmiş ve yapılan literatür taraması paylaşılmıştır.

2.1. VERİ SIZINTISI

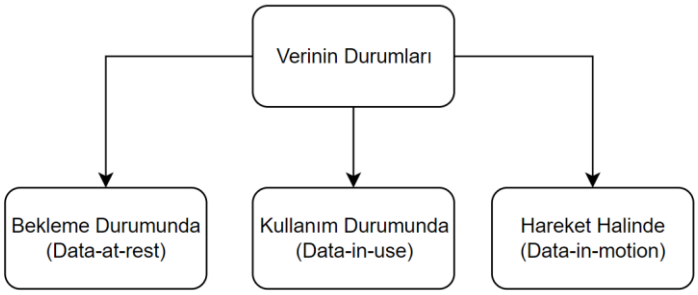
Gerek şahıslar gerek şirketler gerekse de kamu kurum ve kuruluşları açısından farklı seviyelerde olsa da önemli ve çoğu zaman da hayati olan verilerin kasıtlı ya da birçok kez de kasıtsız olarak fark edilmeden yetkisiz veya art niyetli kişilerin eline geçmesine veri sızıntısı denir [6]. Genellikle bir çalışan tarafından gerçekleştirilen işlemler içerisinde, kuruma ait ticari bilgiler veya kurum müşterilerine ait kişisel bilgiler yer alabilmektedir. Bu bilgilerin olağan olarak bulunması gereken ve izin verilmiş güvenli alanlardan çeşitli şekillerde dışarı çıkarılması veri sızıntısı oluşması anlamına gelmektedir. Veri sızıntısı kavramını daha iyi anlayabilmek için öncelikle veri kavramı ve verinin güvenli alanında bulunduğu durumları incelemek gereklidir [7].

2.1.1. Veri

Literatürde "kaydedilebilen, düzenlenebilen ve analiz edilebilen bilgi birimi" [8] olarak ifade edilen veri, ses, görüntü ve yazı dahil pek çok formda bulunabilen ve farklı istatistiksel ve matematiksel işlemler ile anlamlı bilgilere dönüşebilecek yapılar olarak da ifade edilebilir. Kişi ya da kuruma ait veri, bir taraf için anlam ifade etmese bile farklı kişilerin eline geçip uygun yollarla işlenerek başta siber saldırılar olmak üzere birçok amaçla kullanılabilmektedir.

2.1.2. Verinin Durumları

Veri, bulunduğu konuma göre veya işlem görme aşamasındaki durumuna göre Şekil 2.1'de görüldüğü üzere bekleme durumundaki veri(data-at-rest), kullanım durumundaki veri (data-in-use) ve hareket halindeki veri (data-in-motion) olmak üzere 3 farklı durumda incelenir.



Şekil 2.1: Verinin Durumları.

2.1.2.1. Bekleme Durumundaki Veri (data-at-rest)

Verinin herhangi bir transfer işlemi ya da süreci içerisinde olmadığı, depolandığı birimde (sabit disk ya da bulut) işlem göreceği zaman kadar beklediği durumdur. İşlem görmeyen verilerde veri sızıntısı ya da siber güvenlik saldırı, ihtimalinin daha az olduğu düşünülmektedir ancak depolanan alanın güvenliği mutlaka güncel korunma araçlarıyla koruma altında tutulmalıdır.

2.1.2.2. Kullanım Durumundaki Veri (data-in-use)

Verinin, çalışan bir sistem içerisinde, depolandığı yerde erişildiği ve RAM ya da panoda (clipboard) işlem gördüğü durumudur. Bu durumdayken veri üzerinde, kopyalama, silme, düzenleme gibi işlemler yapılmaktadır.

2.1.2.3. Hareket Halindeki Veri (data-in-motion)

Verinin en çok sızıntıya maruz kaldığı durumdur. Veri bu durumda depolandığı konumdan başka bir konuma transfer edilir. Bu transfer işlemi ağ üzerinden (yerel ya da internet) üzerinden gerçekleştirilir. Güvenli alanından çıkan veri, siber saldırılara ve olası veri sızıntılarına da açık hale gelir. Verinin taşınma sürecinde güvenliği sağlamak için çeşitli yöntemler geliştirilmiş olsa da (şifreleme, güvenlik duvarı, saldırı tespit ve önleme sistemleri-IDS/IPS) söz konusu veri sızıntısı olduğunda mutlaka verinin içeriği ile de ilgilenmek ve veri sızıntısı önleme sistemlerini de devreye almak gereklidir.

2.1.3. Veri Gizliliğinin Korunması ve Yasal Düzenlemeler

Veri gizliliği ile ilgili en önemli maddelerden biri kişisel veri kavramıdır. Kişisel veri "kimliği belirli veya belirlenebilir gerçek kişiye ilişkin her türlü bilgi" olarak tanımlanmaktadır [9]. Bu bir kurum için müşterinin adı soyadı, telefon numarası veya kimlik numarası gibi bilgiler olabilir. Kurumlar ve müşterileri arasında imzalanan iş sözleşmelerinde veri

korunmasını da kapsayan birçok madde yer almaktadır. Türkiye'de kişisel verilerin korunması ile ilgili yasal düzenlemeler Dünya'daki benzerlerine paralel olarak yürürlüğe konulmuştur. 6698 sayılı Kişisel Verilenin Korunması Kanunu (K.V.K.K), TBMM'de kabul edilmesinden sonra 7 Nisan 2016 tarihinde 29677 sayılı Resmî Gazete 'de yayımlanarak yürürlüğe girmiştir [10]. Bu kanun ile kişisel verilere erişim, kişinin iznine tabi olmuştur. Ayrıca bu kanun sayesinde kişisel veriyi elde eden kurumun bu veriyi paylaşma, depolama ve işleme hakkındaki izinleri de kanun ile düzenlenmiştir.

Avrupa'da 1990'lı yıllarda kişisel verinin korunması ile ilgili çeşitli kanunlar çıkartılmaya başlamış olsa da toplu olarak yapılan en büyük düzenleme 25 Mayıs 2018 tarihinde çıkartılan Avrupa Genel Veri Koruma Tüzüğü'dür (Genaral Data Protection Regulation-GDPR) [11]. Bu kanun sayesinde Avrupa Birliği'ne (AB) üye ülkeler kendi kişisel veri koruma kanunları çıkarmak yerine tüm AB ülkelerinde geçerli olan kanunla kişisel verilerin korunmasında genel bir yaklaşım benimsenmiştir.

2.1.4. Veri Sızıntısını Önleme Yöntemleri

Verinin elde edildikten sonra depolanma yöntemine, bulunduğu duruma, kullanım şekline veya verinin önem derecesine göre çeşitli veri sızıntısı önleme (Data Leakage Prevention-DLP) yöntemleri geliştirilmiştir. Her bir yöntemin kendi içerisinde avantaj ve dezavantajları bulunmakla beraber insan faktörü her zaman işin bir parçası olduğundan yüzde yüz güvenli bir DLP sisteminden bahsetmek mümkün değildir [12].

Veri depolandığı yere göre yerel makine (Local), sunucu (Server) veya bulut (Cloud) üzerinde olabilir. Her konumun kendine has güvenlik gereksinimleri vardır ve DLP yaklaşımları da bu gereksinimlerdeki zayıf noktalar düşünülerek planlanmalıdır. Şifreleme, erişim yetkisi ve politikalar, antivirüsler ve gelişmiş yapay zekâ sitemleri kullanılarak veri sızıntısı önleme işlemleri gerçekleştirilmeye çalışılabilir.

Verinin hassasiyet durumu incelendiğinde ise kişisel veri veya kurum ticari sırları değerlendirilebilir. Her iki veri türünde de sızıntıya karşı daha çok içerik tabanlı yöntem kullanılmakta ve parmak izi, etiketleme, düzenli ifadeler ve sözlük gibi yapılarla sızıntının önüne geçilmeye çalışılmaktadır. Bu durumdaki verilerde hassas içerik analizi ve bağlam tabanlı taramalar (dosyanın erişim zamanı, oluşturma zamanı, oluşturan kişi vb.) yapılarak oluşabilecek anormal durumlar analiz edilerek muhtemel veri kayıpları önlenmek istenmektedir [13].

Veri sızıntısı önleme konusundaki bir diğer yaklaşım ise verinin anlık durumuna göre geliştirilen önleme yöntemleridir. Bu yaklaşımda verinin kullanım durumunda, hareket halinde veya depolanmış halde bulunduğu varsayılarak her bir durum için farklı yöntemler geliştirilmeye çalışmaktadır. Depolanmış veride şifreleme, erişim kontrolü ve sınıflandırma gibi yöntemler uygulanırken hareket halindeki veride daha çok ağ üzerinden gerçekleşen bir transfer söz konusu oluğundan, ağ trafiğinin çeşitli güvenlik yazılımları ile denetlenmesi, proxy kullanımı ve URL engelleme gibi yöntemler tercih edilmektedir [14].

2.2. YAPAY ZEKÂ

İnsan tarafından belirli amaca yönelik oluşturulan bir model ve bu modelin uygun veri setleri ile eğitilmesi sonucunda gerçek ya da gerçeğe yakın uygulamaların makineler tarafından insan kontrolü olmaksızın gerçekleştirilmesi işlemi temel manada yapa zekâ olarak isimlendirilmektedir [15]. Makine öğrenmesi, derin öğrenme ve doğal dil işleme başlıklarında incelenecek yapay zekâ, büyük dil modellerinin geliştirilmesi ve genel kullanıma sunulmasından sonra daha fazla ilgi çekmiş ve gelişimi hızlanmıştır. Donatıldığı yetenekler ve eğitildiği veri setlerinin zenginliğiyle birçok alanda yaygın kullanıma ulaşan yapay zekâ teknolojileri, kişisel verileri tespit ve ayırt edebilme özelliğiyle de veri sızıntılarını önleme yöntemlerinin önemli bir parçası haline gelmiştir.

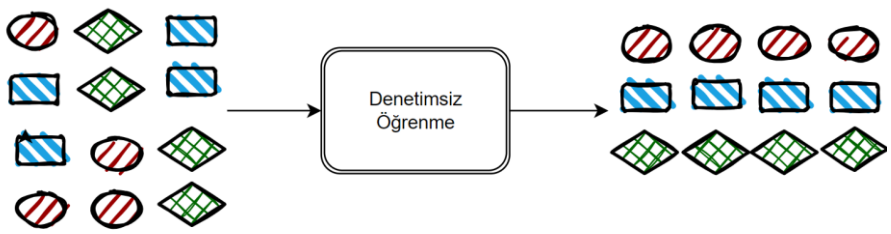
2.2.1. Makine Öğrenmesi

Yapay zekâ kavramının en önemli alt dallarından biri olan makine öğrenmesi ile, denetimli veya denetimsiz şekilde oluşturtulan algoritmalarla eğitilen yazılımların bir çeşit tahmin yaparak insanlara ya da sistemlere yardımcı olması veya yol göstermesi sağlanabilmektedir. Hali hazırda en önemli parçası uygun ve zengin veri setleri olan bu sistemde, veri setlerinin oluşturulabilmesi için gereken verinin veri sızıntıları yoluyla elde edilebilmesi ve veri sızıntısının önlenmesi için makine öğrenmesinin kullanılması arasında bir paradoks olduğu varsayılabilir. Kökeni 1950'li yıllara kadar giden makine öğrenmesinde amaç bir insanın öğrenme tarzını makineler üzerinde de uygulamak ve bu yolla makinelere yeni bilgiler öğretmektir [16]. Öğrendiği bilgiler ile kendini test edecek şekilde programlanan makine, başarısız olsa dahi bunu da bir öğrenme şekli olarak kabul eder ve daha sonraki denemelerinde bu başarısız yolu seçmez veya daha doğru uygulayarak sonuca gitmeye ve en uygun yolu bulmaya çalışır. Günümüzde sıklıkla, eğitildiği veri setleri dahilinde mantığa ve akla uygun en doğru seçimi yapmak, en elverişli yolu seçmek, an kârlı ürünü tespit etmek veya

pazarlama stratejileri geliştirmek gibi birçok alanda aktif olarak kullanılmaktadır. Veri setinin yapısına göre denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme gibi alt dalları bulunmaktadır.

Denetimli Öğrenme, makine eğitiminde kullanılan veri setinin içerisinde yer alan verilerin etiketlenmiş, kategorize edilmiş ve bağımlı değişkenler belirlenmiş halidir.

Denetimsiz öğrenme ise yine veri setindeki etiketlemelerin, değişken türlerinin (bağımlı-bağımsız) olmadığı öğrenme türüdür. Bu türde veriler tam olarak kategorize edilmemiş olabilir ve buna bağlı olarak da sonuçlar net olmayabilir. Burada makine hangi verinin hangi sonucu üreteceği konusunda öğrenmeyi ve karar vermeyi kendisi yapar. Bunun için elindeki verileri ve elde ettiği sonuçları yorumlaması gerekir. Yorumlar sonucunda çeşitli şekillerde veriler arasında ilişkiler kurar ve aynı ilişkide olanları kümeler (clustering). Bu durum Şekil 2.2'de gösterilmiştir.



Şekil 2.2: Denetimsiz Öğrenme [17].

Pekiştirmeli öğrenme yönteminde ise tamamen bir deneme-yanılma sistemi mevcuttur ve makine sürekli deneme yaparak bulduğu sonuçlar üzerinden çıkarımlar yaparak hedefe ulaşmaya çalışır.

2.2.2. Derin Öğrenme

Makine öğrenmesinin bir alt türü olan derin öğrenmede hem denetimli hem de denetimsiz öğrenme kullanılabilir. İnsan beyninin taklit edilmesi" şeklinde genellenen bir tanıma sahip olan derin öğrenmede oluşturulan yapay sinir ağları, mümkün olan her şekilde ve mümkün olan her kaynaktan beslenerek sürekli gelişim ve öğrenme içerisinde olan bir yapıyı temsil eder [18]. Kendisine sağlanan veri setleriyle çıktı arasında belirsiz sayıda ara katman (yapay sinir ağı) bulundurur ve bu ağlar sayesinde en doğru sonucu vermeyi amaçlar. Burada birtakım zorluklar da söz konusudur. Öncelikle doğru sonuç için mümkün olan en fazla sayıda

veri setiyle eğitilmeleri gerekir. Ayrıca yapay sinir ağları çok fazla olasılık ve buna bağlı olarak matematiksel hesap içerdiğinden çok yüksek işlem gücü gerektirir.

2.2.3. Doğal Dil İşleme

LLM mimarileri Doğal Dil İşleme (Natural Language Processing-NLP) dalının gelişmesiyle beraber etkisini arttırmıştır. NLP, bilgisayar ve insan arasında ortak dil kullanmayı veya bilgisayarın insan dilini anlamasını sağlayan bir yapay zekâ alt dalı olarak ortaya çıkmıştır [19]. NLP'nin yaygın olarak kullanılan mimarileri, Transformer ve LSTM (Long Short-Term Memory) olarak sınıflandırılabilir. LSTM ve Transformer mimarilerinin birbirlerine karşı bazı avantaj ve dezavantajları bulunur.

a) Transformer mimarisi: Barındırdığı dikkat mekanizması ve özellikle paralel işlemlere izin vermesiyle diğer mimarilerden öne çıkmıştır. Uzun dizileri işleme noktasındaki başarısına da önemli bir nokta olarak değinilmelidir. 2017 yılında Google tarafından duyurulmuştur ve makale başlığı olarak "Attention is All You Need" seçilmiştir. Kodlayıcı (Enkoder) ve Kod Çözücü (Dekoder) olmak üzere 2 ayrı bölümden oluşmaktadır. Enkoder bölümü, kelimelerin birbiriyle ilişkisini belirlemekten ve bağlam ilişkileri kurarak anlam örgüsünü belirlemekten sorumludur. Dekoder bölümü ise enkoder tarafından üretilen bağlam çıktılarını alıp bir dizi oluşturarak kelime tahmini yapmaktan sorumludur. Transformer mimarisinin diğer mimarilere göre en büyük avantajı paralel işlem yapabilmesidir [20].

b) LSTM (Long Short-Term Memory) mimarisi: 1997 yılında Sepp Hochreiter ve Jürgen Schmidhuber tarafından geliştirilmiştir. Yinelemeli yapay sinir ağlarının en büyük sorunu olarak gösterilen uzun dizilerde zamanla bilgi kaybı yaşama ya da unutma problemini çözmek amacıyla ortaya çıkmıştır. Bu sorunu çözebilmek için zaman bilgisinin taşındığı kapı (gate) yapılarını kullanır ve kelimeler arasındaki bağlamı da yine bu kapılar sayesinde korur. Unutma kapısı (forget gate), girdi kapısı (input gate) ve çıkış kapısı (output gate) olmak üzere 3 kapıdan oluşan hücre (cell) yapısı barındırır. Unutma kapısı, hücre durumundan hangi bilgilerin silinmesi gerektiğini belirler. Bu kapı, bir önceki gizli durum ve mevcut girişle çalışır ve sigmoid aktivasyon fonksiyonu ile 0 ve 1 arasında bir değer üretir. Çıktı 0'a yakınsa bilgi unutulur, 1'e yakınsa bilgi korunur. Kapı formülü şu şekildedir;

$$ft = \sigma(Wf \cdot [ht - 1, xt] + bf) \tag{2.1}$$

Formül değişkenleri:

- ft: Unutma kapısının çıktısı
- Wf: Unutma kapısının ağırlıkları
- $ht-1$: Önceki gizli durum
- Xt: Mevcut giriş
- σ: Sigmoid aktivasyon fonksiyonu

Girdi kapısı ise, mevcut girdiden hangi yeni bilgilerin hücre durumuna eklenmesi gerektiğini kontrol eder. İki aşamadan oluşur: Sigmoid fonksiyonlu kapı, hangi bilginin hücreye eklenmesi gerektiğini belirler; tanh fonksiyonu ise, yeni hücre durumu adaylarını oluşturur. Çıkış kapısı, mevcut hücre durumuna dayalı olarak LSTM hücresinin çıkışını belirler. Bu şekilde bilgileri uzun süreli koruma özelliğini sağlamış olur. LSTM ve Transformer mimarileri ile ilgili karşılaştırma Tablo 2.1'de gösterilmiştir.

Tablo 2.1: Transformer ve LSTM Mimarileri Karşılaştırması.

Özellik	Transformer	LSTM
Diziyi İşleme Şekli	Paralel olarak tüm diziyi işler	Sıralı olarak, birer birer adım adım işler
Dikkat Mekanizması	Self-Attention (Kendine Dikkat) mekanizmasını kullanır	Kapılar (girdi, unutma, çıkış) üzerinden bilgi akışı
Uzun Bağlam Bilgisi	Tüm diziyi aynı anda işleyerek daha uzun bağlamları yakalayabilir	Hücre durumu sayesinde uzun vadeli bağımlılıkları öğrenir
Hesaplama Verimliliği	Paralel işleme yapabildiği için daha hızlıdır	Sıralı olduğu için hesaplama maliyeti daha yüksektir
Paralel İşleme	Evet	Hayır, sıralı işleme gerektirir

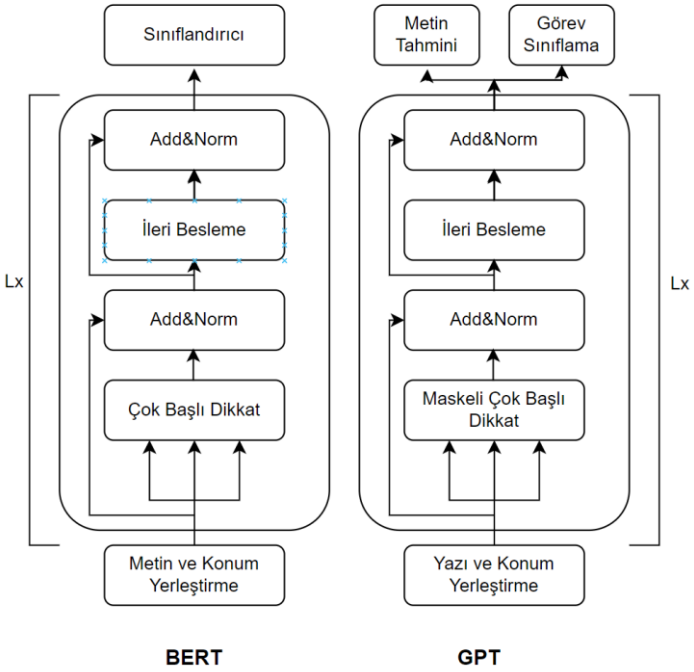
2.2.4. Büyük Dil Modeli

Yapay zekânın bütün alt dallarını birleştiren, etkin bir şekilde kullanan, büyük veri setleriyle beslenen ve sürekli büyüyen yapılara Büyük Dil Modeli (Large Language Model-LLM) denir. Yaygın kullanımı üretken yapay zekâ şeklinde olmasına rağmen kurumsal firmalar

özellikle Ar-Ge çalışmalarında büyük dil modellerini kullanmaktadırlar. İnsan zekasını, konuşmasını, çizim yeteneklerini, hayal gücünü ve insana dair birçok özelliği taklit etme veya simüle etme yeteneğine sahip olan bu modeller, büyük teknoloji firmalarının bu alana yatırım yapmaya devam etmeleriyle çeşitlenmekte ve sürekli eklenen veri setleriyle gelişmeye devam etmektedir.

Veri sızıntıları kasıtlı olabileceği gibi istem dışı da yapılabilir. Bu sızıntıları önlemek için yaygın olarak kullanılan sözlük ve düzenli ifadeler, öngörülemeyen kelime ya da veriler için yetersiz kalabilir. Böyle bir durumda bir çalışanın yazdığı metinlerde KVKK ihlali olan veri olup olmadığını başka bir çalışanın kontrol etmesi teoride bir çözüm gibi düşünülebilir ancak büyük şirketlerin veri trafiği göz önünde bulundurulduğundan bunun pratikte uygun olmayan bir yaklaşım olduğu anlaşılabilir. Üstelik kontrol eden personel de hata yapabilir. Bu durumda kontrol personeli yerine yapay sinir ağlarıyla güçlendirilmiş ve uygun şekilde optimize edilen veri setleriyle eğitilmiş büyük dil modelleri, yazışmalar içerisindeki KVKK ihlallerini tespit edebilir ve oluşturulacak algoritma sayesinde sızıntı gerçekleşmeden önlem alınabilir.

Günümüzde yaygın olarak kullanılan BDM'ler GPT veya BERT algoritmalarını kullanırlar. Şekil 2.3'te bu iki algoritmanın genel yapısı görülebilir. Her iki algoritma da Doğal Dil İşleme (NLP) modelleridir.



Şekil 2.3: BERT ve GPT Modelleri [21].

a) GPT (Generative Pre-trained Transformer) Algoritması: Transformer dil modelleme tekniğini kullanan bu algoritma, çok büyük sayıda veri seti ile eğitilmeyi gerektirir. İşlem yönü olarak soldan sağa(left to right) yapısındadır. Kendisine verilen kelimeleri inceler ve bulduğu sonuçlara göre sıradaki kelimeyi tahmin etmeye çalışır. Sonuca en yakın değeri verebilmesi için ince ayar (fine-tuning) işlemi yapılır. Sahip olduğu dikkat mekanizması sayesinde kelimeyi aldıktan sonra olası en yakın kelimeyi bulmaya çalışır. Her bir kelimeyi bağlamsal ağırlıklarına göre işaretler ve yeni kelimeyi bu ağırlıkların olasılıklarını hesaplayarak tahmin etmeye çalışır. Kullanım olarak en çok metin tamamlama alanında kullanılmasına rağmen, sohbet robotları (chatbots), soru cevaplama, özet çıkarma gibi pek çok alanda işlev yerine getirmektedir.

b) BERT (Bidirectional Encoder Representations from Transformers) Algoritması: Kullanıldığı alanda, girdilere karşılık muhtemel çıktıları daha doğru verebilmek amacıyla tasarlanmıştır. GPT modelindeki soldan sağa yapısı yerine yön belirtmeden, kelimelerin her iki tarafına ve eş veya benzer anlamlarına bakarak sonuç üretmeye odaklanır. Cümle içlerindeki kelimelerin yanında edat ve bağlaçların cümleye kattığı anlamada da odaklanmaktadır. Eğitim şekli olarak GPT modelinden ayrı olarak maskeleme (MLM) tekniğini kullanır. Bu teknikte cümle içerisinden rastgele kelimeler maskelenerek modelin bu kelimeleri tahmin etmesi istenir. GPT'de bulunan metin

tahmini yerine metni anlama ve sınıflandırma üzerine yoğunlaşmıştır. Her iki model ile ilgili karşılaştırma yapılmış ve Tablo 2.2'te gösterilmiştir.

Tablo 2.2: BERT ve GPT Karşılaştırması

Özellik	BERT	GPT
Mimari	Encoder tabanlı Transformer	Decoder tabanlı Transformer
Yönsellik	İki yönlü (bidirectional), metni hem soldan sağa hem sağdan sola isler	Tek yönlü (unidirectional), metni soldan sağa işler
Eğitim Görevi	Maskeli dil modeli (Masked Language Modeling - MLM)	Otoregresif dil modeli (Autoregressive Language Model)
Giriş Verisi	Girdinin bazı kısımları maskelenir ve modelin bu maskeli kelimeleri tahmin etmesi beklenir	Girdinin her kelimesini birer birer işler, sonraki kelimeyi tahmin eder
Çıkış Verisi	Maskelenmiş kelimeleri tahmin eder (tam anlamlı metin olusturmaz)	Tam anlamlı metin üretir, bir kelime dizisini genişleterek devam ettirir

Tablo 2.2 (devam): BERT ve GPT Karşılaştırılması

Bağlam Bilgisi	Tam bağlamlı (kelimenin her iki yanındaki kelimeleri dikkate alır)	Sadece önceki kelimelere dayalı bağlam bilgisini kullanır
Kullanım Amacı	Metin anlamlandırma, metin sınıflandırma, soru yanıtlama, duygu analizi	Metin üretimi, diyalog sistemleri, içerik oluşturma
Eğitim Verisi	İki yönlü dikkat (bidirectional attention) ile dil anlama	Sıralı dikkat (sequential attention) ile dil oluşturma ve dil anlama
Eğitim Stratejisi	Maskelenmiş dil modeli ve cümle çiftleri tahmini	Otoregresif dil modeli, önceki kelimelere dayalı olarak sonraki kelimeleri tahmin eder
Transfer Öğrenme	Özellikle ince ayar (fine-tuning) için güçlü	Metin üretimi odaklı, ince ayar yapılabilir ancak daha çok yaratıcı üretimlerde kullanılır
Avantajları	Daha güçlü dil anlama yetenekleri, bağlamın iki tarafını da görebilmesi	Dil üretiminde daha başarılı, yaratıcı ve doğal metin üretme kapasitesi yüksek
Dezavantajları	Dil üretiminde zayıf, maskelenmiş kelimeler tahminiyle sınırlı	Tek yönlü olduğundan tam bağlam bilgisini öğrenme zorluğu
Kullanım Alanları	Soru cevaplama (Question Answering), doğal dil anlama, metin sınıflandırma	Metin tamamlama, diyalog sistemleri, dil üretimi, yaratıcı yazım

2.3. İLGİLİ ÇALIŞMALAR

Veri sızıntısı kavramı birden çok akademik ve bilimsel alanı ilgilendirdiğinden literatürde, karşılaşılabilecek farklı sorunlara yönelik farklı çözüm önerileri sunulmuştur. Bu çözümlerin büyük çoğunluğunun İngilizce olması ve yerel kullanım için düzenlemeler içermemesi literatür taramasında ilk göze çarpan noktalardandır. Literatür çalışmalarında verileri hassas veya hassas değil şeklinde sınıflamaya çalışan ve hassas olarak sınıflanan dosyaların erişiminden yola çıkarak sızıntı tespiti yapan çalışmalar yer almaktadır. Bu çalışmalar sınıflandırma için makine öğrenmesi yöntemlerini kullanmaktadırlar. Sınıflama sonrasında herhangi bir kullanıcının hassas olarak işaretlenmiş dosyalara erişmesi durumu veri sızıntısı olarak değerlendirilmekte ve belirlenen aksiyonlar(bloklama, erişimi engelleme, bağlantıyı kesme vb.) alınmaktadır. Bir diğer grup çözüm önerilerinde ise ağ üzerinden gönderilen dosyalar veya e-postalar içerik taramasından geçirilmektedir ve bu şekilde sızıntı tespiti yapılmaya çalışılmaktadır. Ayrıca e-posta gönderilebilecek domainlerin veya posta kutularının kısıtlı listeye alınması da bir diğer çözüm önerisidir. Literatürde yer bulmuş çalışmalarda Windows işletim sisteminin minifilter sürücülerini kullanarak bilgisayar takılan harici bellek veya cihazları engelleme veya sızıntı testine sokma ile ilgili konular da yer almaktadır. Veri sızıntısına karşı alınacak önlem önerileri için verinin anlık durumunun bilinmesi önemlidir. Farklı durum ve konumlarda olan veriler için, duruma ya da konuma özel çözümlerin üretilebilmesi pratik bir yaklaşım olarak benimsenmiştir.

Kullanım durumundaki veri üzerinde veri sızıntısı tespit etmek istenen bir çalışmada [22] ana işletim sistemi üzerine bir yazılım olarak kurulan sanal bir işletim sistemi bulunmaktadır. DLP yazılımı ana işletim sisteminde çalışmakta iken kurum çalışanının sanal işletim sistemini kullanması planlanmıştır. Sanal işletim sistemi, ana işletim sistemi için bir program seviyesinde olduğundan işletim sistemindeki yönetici haklarından dolayı (Administrator) erişilemeyen donanım birimlerine veya sistem kaynaklarına erişilmesi planlanmış ve yöneticiyi gözetleyen üst yönetici kurgusuyla veri sızıntıları tespit erişilmeye çalışılmıştır.

Donanım üzerinden veri sızıntısı tespit etmeyi amaçlayan bir çalışmada ise [23] kullanılan hipervizör sistemi sayesinde sanal makine üzerindeki ana bellek (RAM) içeriği analiz edilmeye çalışılmış ve buradaki bilgiler üzerinde hassas veri sızıntısı testleri yapılmıştır. Ana bellek üzerinde veri sızıntısı tespit etmeye çalışılırken desen tabanlı çeşitli tarama

algoritmaları kullanılmıştır. Tarama işlemi periyodik olarak yapılabileceği gibi isteğe bağlı zamanlarda da yapılabilmektedir.

Kurum çalışanlarının hazırlanmış bir sohbet sistemi üzerinden yazışmalar ve dosya transferleri yapmasını amaçlayan bir diğer çalışmada ise [24] çalışanların sohbet yazılım üzerinde yazdıkları mesajlar sözlük tabanlı filtre sistemine sokulmuş ve veri sızıntısı önlenmeye çalışılmıştır. Ek olarak çalışanın sohbet programı üzerinden gönderdiği e-postalar, izinli domainler dışında ise veri sızıntısı uyarısı verilmiştir.

Sınıflandırma ile veri sızıntısı önleme yapmayı amaçlayan bir çalışmada ise [25] LSTM algoritması ile oluşturulan bir model 2000 haber makalesi ile eğitilmiş ve daha sonra belirlenen ölçütlerde sorgulanarak veriler hassas veya hassas değil şeklinde sınıflandırılmaya çalışılmıştır. Hassas veri tespiti durumunda AES256 şifreleme yöntemi kullanarak veriler şifrelenmiş ve bu şekilde veri sızıntısı önlenmek istenmiştir. Çalışma sonunda %93,7 oranında hassas veri tespitine ulaşılmıştır.

Verilerin dosyalar içerisine gizli bir şekilde saklanarak kurum dışarısına çıkarılması yoluyla veri sızıntısı oluşmasına karşı bir önlem geliştirmeyi amaçlayan bir diğer çalışmada [26] bir imaj nesnesi üzerinde çalışılmış ve en az önemli bit (Least Significant Bit-LSB) değiştirme yoluna gidilerek sızıntı önlenmeye çalışılmıştır. Bu çalışmada 2 farklı yöntem kullanılmıştır, ilk olarak LSB, bir OR işlemine tabi tutulmuş ve LSB'nin tamamının 1 yapılması amaçlanmıştır. İkinci yöntemde ise LSB AND işlemine tabi tutularak tüm LSB'ler 0 yapılmıştır. Bu şekilde eğer imaj nesnesinde LSB içerisine gizlenerek kurum dışına çıkarılmak amaçlanan veri varsa, dosya bütünlüğü bozulmadan gizli veri değiştirilerek önlem alma amaçlanmıştır.

İşletim sisteminin sistem dosyalarını ve yönetici haklarını kullanarak sızıntı önlemeye çalışan bir diğer çalışmada [26,27] hassas olarak işaretlenmiş dosya ve klasörlerin üzerinde yapılacak kopyalama, silme ve düzenleme gibi işlemlerin minifilter sürücüsü kullanarak engellenmesi amaçlanmıştır. Aynı şekilde eğer hassas veriler harici bir belleğe transfer edilmek istenirse, bu girişime de yine minifilter sürücüsü yoluyla erişim engel konularak bir veri sızıntısı önlemi almak amaçlanmıştır.

Hassas dosyaların harici bellek yoluyla sızmasına karşı önlem geliştirmeyi amaçlayan başka bir makalede [28] yönetici ve kullanıcı şeklinde 2 rol bulunan bir yazılım geliştirilmiş ve

yönetici tarafından belirlenen kurallara göre kullanıcı ekranında işlem yapılmak amaçlanmıştır. Bu çalışmada ilk olarak minifilter sürücüleri ile cihaza bir harici bellek takılma durumu tespit edilmeye çalışmıştır. Eğer harici bellek takılmışsa, bu belleğe kopyalama işleminin yapılma durumu izlenmiştir. Kopyalama gerçekleştiğinde dosyanın içeriği taranmış, belirlenen kurallara uymayan bir içerik tespit edildiğinde kopyalama işlemi durdurularak harici belleğin bilgisayarla bağlantısı kesilmeye çalışmıştır.

Ağ erişimi üzerinde sızıntı tespiti yapılmaya çalışılan bir çalışmada ise [29] yapay zekâ, sınıflandırma işleminde kullanılmıştır. Ağ geçidi (gateway) üzerinde çalışan bir sistem kurgulanmış ve güvenli ağ olarak adlandırılan kurum içi intranet üzerinden güvensiz olarak adlandırılan kurum dışı internete dosya transferi yapılacağı zaman, gateway üzerindeki makine öğrenmesi yöntemleriyle eğitilmiş sistem dosyayı test etmekte, eğer dosya hassas veri olarak işaretlenmişse şifreleyip göndermekte, eğer hassas değilse herhangi bir işlem yapmamaktadır. İnternet üzerinden gelen trafik de yine kontrol edilmekte şifreli gönderilmiş hassas veriler şifresi çözülerek intranet içerisine iletilmektedir. Bu sistemde dosyanın gönderilmesi veya erişimi ile ilgili bir engelleme yapılmayıp transfer sürecinde şifreleme yöntemi kullanılmıştır.

Dosyaların oluşturma ve son erişim tarihlerini tutar zaman damgası (time stamp) kontrolü ile veri sızıntısı tespiti yapmayı amaçlayan bir diğer çalışmada [30], kuruma ait dosyalar öncelikle gizli ve gizli olmayan fark etmeksizin toplanmıştır. Daha sonra kosinüs benzerliği işlevine sahip Kmeans kullanarak kümeler oluşturulmuş ve her küme için anahtar terimler sıklıklarına göre tanımlanmıştır. İşlem sonucunda her anahtar terim için puanı hesaplanmış ve kurum takviminin tarihlerine göre her belge için zaman damgası atama işlemi gerçekleştirilmiştir.

İncelenen tüm çalışmalar yöntem ve veri durumlarına göre listelenerek Tablo 2.3'te gösterilmiştir.

Tablo 2.3: Mevcut DLP Çalışmalarının Karşılaştırılması.

Çalışma	Yıl	Yöntem	Veri Durumu
Efficient DLP-visor: An efficient hypervisor-based DLP	2021	Kural Tabanlı	Kullanımda

Detection and Prevention of Data Leakage in Transit Using LSTM Recurrent Neural Network with Encryption Algorithm	2023	Makine Öğrenmesi	Hareket Halinde
Hypervisor-based Sensitive Data Leakage Detector	2018	Eşleştirme	3 durumda
Precaution Model for Data Leakage Prevention/Loss (DLP) Systems	2015	Stenografi	Hareket
Corporate chat under DLP–system controlling	2021	Kural Tabanlı	Hareket
File System Minifilter Based Data Leakage Prevention System	2018	Kural Tabanlı	Kullanımda, Beklemede

Tablo 2.3 (devam): Mevcut DLP Çalışmalarının Karşılaştırılması.

Freeware Solution for Preventing Data Leakage by Insider for Windows Framework	2020	Minifilter	3 durumda
Data Leakage Prevention for Data in Transit using Artificial Intelligence and Encryption Techniques	2019	Şifreleme	Hareket

3. YÖNTEM

Bu bölümde bu çalışmada tasarlanan veri sızıntısı önleme sistemi detaylı bir şekilde açıklanmış ve geliştirme süreci ile ilgili süreçler tanımlanmıştır. Süreci meydana getiren adımlar, yönetici (master) programın hazırlanması, ajan (agent) programın kodlanması, gerçek zamanlı veri tabanı sisteminin hazırlanması ve yapay zekâ kullanımı için büyük dil modelinin entegre edilmesi şeklinde sıralanabilir.

Sistem bir ana program üzerinden kontrol edilmektedir. Ana programı kullanan yönetici hem uç noktada çalışacak ajanları kontrol etmekte hem de sistemin kullanılacağı kurumun yapısına göre büyük dil modeli için rol belirleme işlemi yapmaktadır. Ayrıca sözlük tabanlı filtreleme (dictionary based) yapı için gerekli filtre kelimeleri organize etme ve ajanlar tarafından yapılacak veri sızıntısı ihlal raporlarının takibi işlemleri de ana program üzerinden yapılmaktadır. Sistem mimarisi Şekil 3.1'de gösterilmiştir.

Sistem, Microsoft Windows işletim sisteminde çalışmak üzere tasarlanmıştır. .Net Framework yapısının sağladığı ve işletim sisteminin çekirdek yapısının (Kernel) bilgisayar donanımı ile haberleşmesini sağlayan dinamik bağlantı kütüphanelileri ile (Dynmic Link Library-DLL) sistem ile donanım arasına kanca atılarak çalışacak şekilde bir kurgu yapılmıştır.

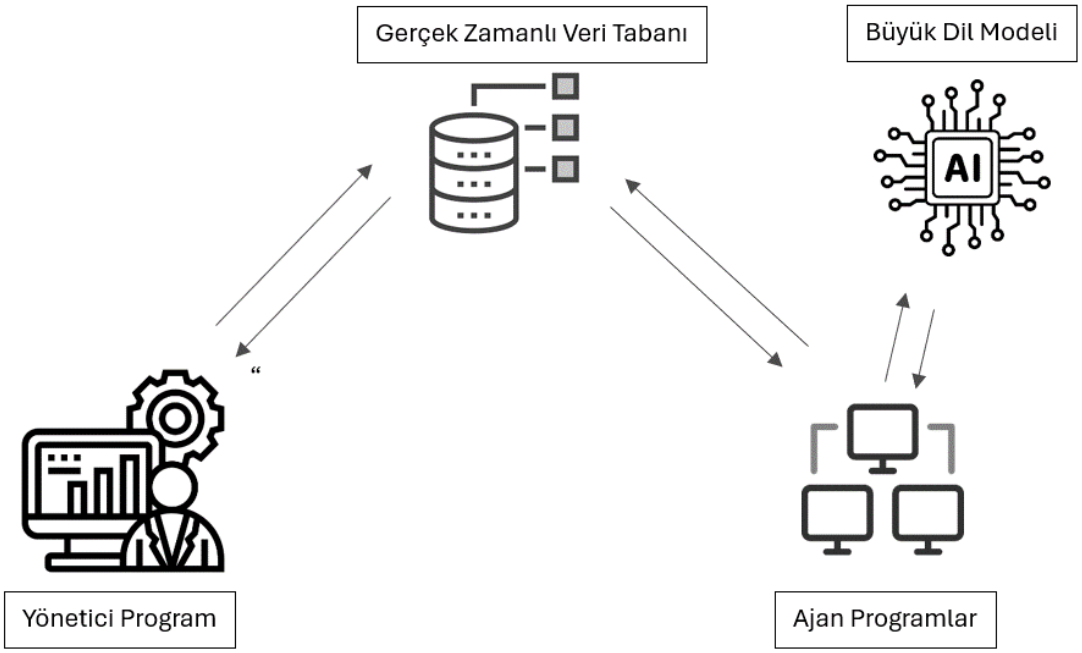
Bu işlem sırasında en büyük amaç kurum çalışanını herhangi bir şekilde etkilemeden ve sistem performansına minimum düzeyde yük bindirerek bir işlem gerçekleştirilmesidir. Tüm işlemler için C# kodlama dili kullanılmıştır.

Veri sızıntısı yazılımın işletim sistemine getirdiği ek yük yaygın kullanılan bir performans analiz programıyla test edilmiş ve sonuçlar bu çalışmanın ajan programı bölümünde paylaşılmıştır.

Sistemin en önemli noktalarından birisi Türkçe dil desteği sunmasıdır. Literatür incelemesinde de gösterildiği üzere, yaygın kullanılan veri sızıntısı önleme sistemlerinde Türkçe karakterler için destek bulunmamaktadır. Türkçe dil desteği sağlamaya çalışan çözüm önerileri oldukça azınlıktadır. Türkçe dil desteği ile ilgili bir diğer sorun ise Türkçe dili ile oluşturulmuş yeterli sayıda veri setinin bulunmamasıdır.

Bir diğer önemli nokta ise gerçek zamanlı raporlama sağlayacak altyapının oluşturulmuş olmasıdır. Ajanlar tarafından elde edilen veriler anlık olarak filtrelemeye sokulmakta ve oluşan sızıntılar yine anlık olarak yöneticiye raporlanmaktadır. Bu sayede sızıntı büyümeden önlem alınma fırsatı oluşturulmaktadır.

Büyük dil modeli KVKK ve GPDR bilgilerini içeren veri setleriyle eğitilmiş olduğundan bu modelin sisteme entegre edilmesiyle kurum çalışanının yazışmaları üzerinde oluşabilecek muhtemel kişisel veri ihlalleri raporlanmaktadır. Büyük dil modeli, çalışılacak kuruma göre rol alabilmekte ve yazışmaları bu role göre inceleyebilmektedir [31]. Ayrıca kullanılan sisteme göre uygun formatta (JSON [32] vb.) dönüş sağlayabilmektedir.



Şekil 3.1: Ajan Tabanlı Veri Sızıntısı Önleme Sistemi.

3.1. AJAN TABANLI VERİ SIZINTISI ÖNLEME SİSTEMİ

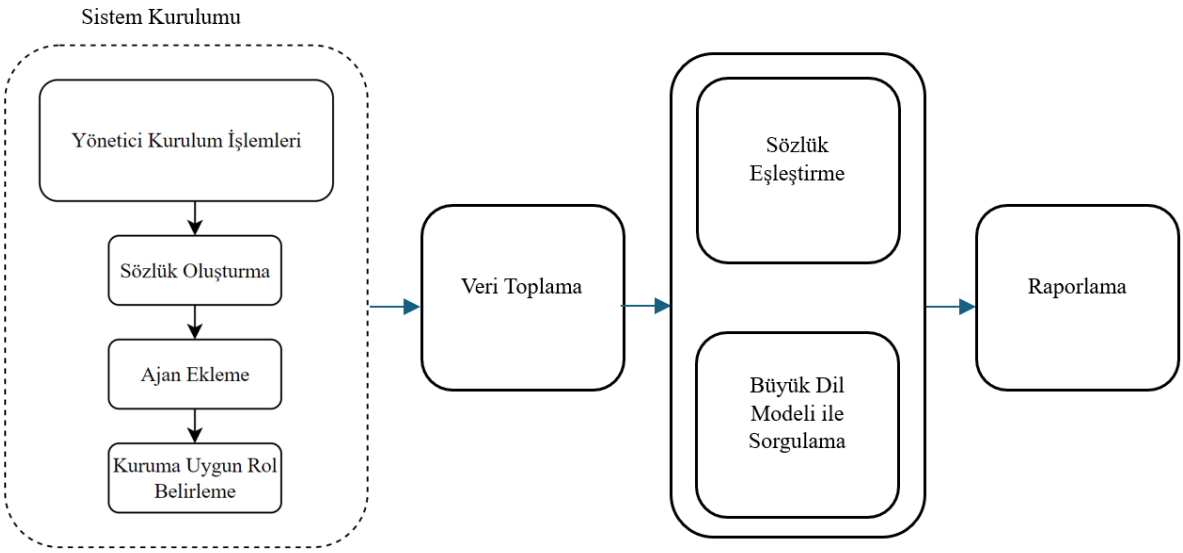
Ajan tabanlı veri sızıntısı önleme sistemi Şekil 3.2'de gösterildiği gibi 5 aşamalı bir izlence ile oluşturulmaktadır. Bu aşamalar: sistemin hazırlanması, verilerin toplanması, sözlük tabanlı kelime eşleme yöntemiyle sızıntı tespiti, büyük dil modeli ile kişisel veri ihlali tespiti ve raporlama şeklindedir.

Sistem kurulumu; kurulacak alt yapının sistemin kullanılacağı çalışma alanı ya da kuruma göre optimize edilmesi, tüm ajanların sisteme dahil edilmesi, yasaklı kelimeler için sözlük hazırlanması ve raporlamanın takip edilmesinin sağlayan altyapının oluşturulması konularını içermektedir.

Verilerin toplanması; bir uç nokta yazılımı olarak (endpoint) çalışanların bilgisayarlarına kurulan ve çalışanın tüm klavye aktivitesini izleyen bir ajan aracılığı ile sızıntı tespitine yönelik veri toplama işlemini kapsamaktadır.

Sözlük tabanlı kelime eşleme yöntemi; ajan tarafından toplanan kelime verilerinin, oluşturulan sözlük altyapısı ile karşılaştırılması ve bu işlem sonucunda veri sızıntısı tespit edilmeye çalışılmasıdır.

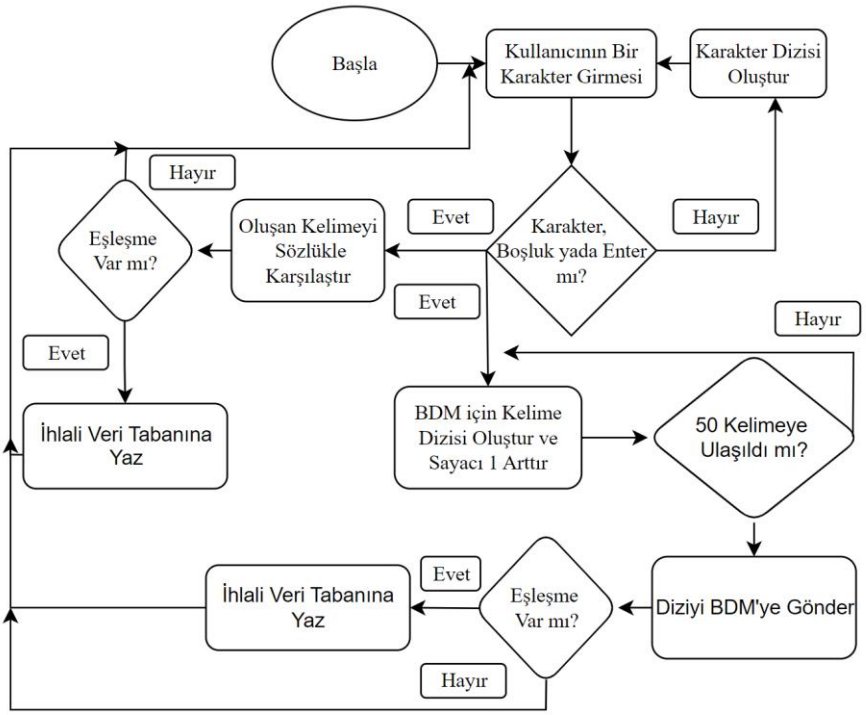
Büyük dil modeli ile kişisel veri ihlali tespiti; ajan tarafından kelime katarı şeklinde toplanan ve belirli uzunluğa ulaşıp bir paragrafa dönüştürülen verilerin, kendisine yönetici tarafından sağlanan rol doğrultusunda büyük dil modeline gönderilmesi ve bu paragraf içerisinde kişisel veri varlığını tespit etmeye çalışılması amacıyla hazırlanmış işlemleri kapsamaktadır.



Şekil 3.2: Ajan Tabanlı Veri Sızıntısı Önleme Sistemi Bölümleri.

Raporlama hem sözlük tabanlı olarak hem de büyük dil modeli aracılığı ile tespit edilen ihlallerin yöneticiye gerçek zamanlı olarak bildirim yapılmasını amaçlayan yöntemleri içerir.

Sistemin çalışma prensibini açıklayan algoritma Şekil 3.3'te algoritmaya ait sözde kod ise Şekil 3.4'te gösterilmiştir.



Şekil 3.3: Sistem Algoritması.

```
1:  array sozluk[]
2:  sozluk[]=getBannedListFromRealtimedatabase()
3:  Input(karakter)
4:  while (karakter!=boşluk or karakter!=enter)
5:          kelime+=karakter
6:  end loop
7:  for i=0 to sozluk.length()
8:          if sozluk[i]== kelime then
9:                  print.to.database "sözlük ihlali, ajanId, kelime"
10:         end if
11: end for
12: kelime.sayisi++
13: paragraf+=kelime()
14: if kelime.count==20 then
15:         bdmyegonder(paragraf)
16:                 if bdmcevap==true then
17:                         print.to.database "kişisel veri ihlali, ajanId, paragraf"
18:                 end if
19: end if
```

Şekil 3.4: Algoritmaya Ait Sözde Kod.

3.1.1. Sistem Kurulumu

Tasarlanan veri sızıntısı önleme sistemi üzerinde sınırsız sayıda ajan yer alabilir ancak tüm ajan yazılımların kontrollerinin tek bir noktadan yapılabilmesi, sistemin güvenliği, kontrol kolaylığı ve oluşabilecek sızıntılara karşı alınacak aksiyonlarda standardizasyon sağlaması açısından faydalı görülmüş ve tasarım bu şekilde gerçekleştirilmiştir. Sistemin

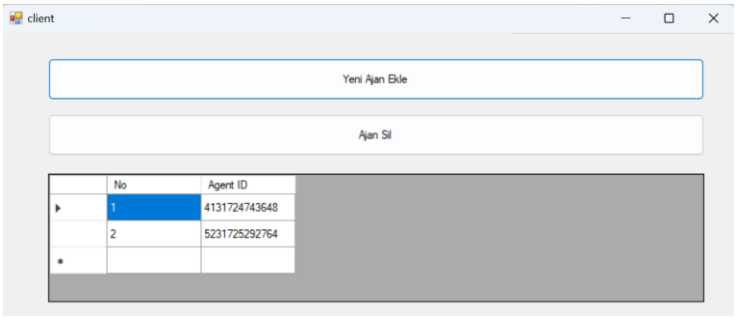
kuruma göre ve kurumun çalışma alanına yönelik olarak optimizasyonu bir yönetici program tarafından sağlanmaktadır. Yönetici program 4 ana bölümden oluşmaktadır;

a) Ajan İşlemleri
b) Sözlük İşlemleri
c) Raporlama İşlemleri
d) Büyük Dil Modeli Ayarları

3.1.1.1. Ajan İşlemleri

Sistemin omurgası ajanlardan oluşmaktadır. Çünkü veri sızıntısına neden olabilecek potansiyeldeki verilerin toplanması, sözlükle karşılaştırılması ve büyük dil modeline gönderilmesi işlemleri ajanlar tarafından yapılmaktadır. Ajan oluşturma işlemi ise yönetici program tarafından gerçekleştirilmektedir. Sistem yöneticisi, kurumdaki her bir uç nokta için yeni bir ajan tanımlaması yapar. Bu işlem için tasarlanan ekran Şekil 3.5'te gösterilmiştir. Ajan ekleme işlemi gerçekleştiğinde eklenen her ajan için sistem, Eşgüdümlü Evrensel Zaman (Coordinated Universal Time -UTC) değerini baz alarak benzersiz bir ID numarası oluşturur. Bu numara anlık olarak gerçek zamanlı veri tabanına da kaydedilir. UTC, Uluslararası Atomik Zaman'a (TAI) dayanan bir zaman standardıdır. Dünya, UTC'ye göre 24 saat dilimine bölünür ve bütün ülkeler, başlangıç meridyenine göre UTC +1, UTC +2 gibi dilimlerde yer alır. Hesaplama yapılırken, Sezyum-133 atomunun 9.192.631.770 titreşimi 1 saniye olarak kabul edilir, bu da mükemmele yakın yüksek bir zaman hassasiyeti sağlar [33].

Kurumdaki herhangi bir uç noktaya veri sızıntısı önleme yazılımı kurulmak istendiğinde, öncelikle yönetici tarafından oluşturulmuş benzersiz ID numarasının girilmesi gerekmektedir. Bu şekilde yetkisiz kurulumların önüne geçmek amaçlanmıştır. Ek olarak raporlama işlemi de yine bu numara üzerinden yapılmakta ve bu ID ajan ve yönetici program arasında ayırt edici bir bağ konumunda bulunmaktadır.



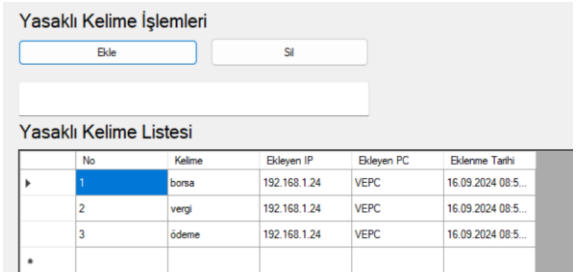
Şekil 3.5: Ajan Ekleme Ekranı.

Herhangi bir nedenle sistemden kaldırılmak istenen bir uç nokta olursa, yönetici tarafından sistemden ID'si silinir. Ajan program, uç noktadan kaldırılmasa dahi yönetici tarafından ID'si silindiğinden veri gönderme, veri tabanı erişimi ve büyük dil modeli bağlantısı kalmayacaktır. Bu şekilde kontrolün tamamen yöneticide olması sağlanmıştır. Ayrıca yönetici herhangi bir zamanda sistemdeki tüm ajanları ve ID'leri yönetici program üzerinden görüntüleyebilir.

3.1.1.2. Sözlük İşlemleri

Yönetici program üzerinden veri sızıntısı olarak kabul edilebilecek, yasaklı veya kullanılması sakıncalı kelimeler Şekil 3.6'da gösterilen ekran aracılığı ile sisteme dahil edilebilir. Sözlük tabanlı sistemler alan yazında yapılan çalışmalarda sıklıkla kullanılmaktadır. Bunun nedeni, kelimelerin kullanıcılar tarafından girilmesi dolayısıyla kullanıcıya veya sistem yöneticisine yasaklamak istediği kelimeyle ilgili seçenek sunması yani esneklik sağlamasıdır. Sözlük yapısı sayesinde gereksiz filtre kelimelere ihtiyaç duymadan ihtiyaca yönelik düzenlemeler yapılabilmektedir. Alan yazında Sözlük Tabanlı Kelime Eşleme (Dictionary Based String Matching-DBSM) olarak geçen bu filtreleme seçeneğinin optimizasyonu ve verimli kullanımı için birçok çalışma yapılmış ve yapılmaya devam etmektedir. Sözlük filtrelemenin iyileştirilmesi konusu bu çalışmanın ana konusu olmadığından, bu sistem en temel haliyle çalışma içerisinde yer almıştır. Sözlük filtrelemesinin çalışma mantığı bu çalışmadaki ajan yazılımının anlatıldığı bölümde detaylı olarak açıklanmıştır.

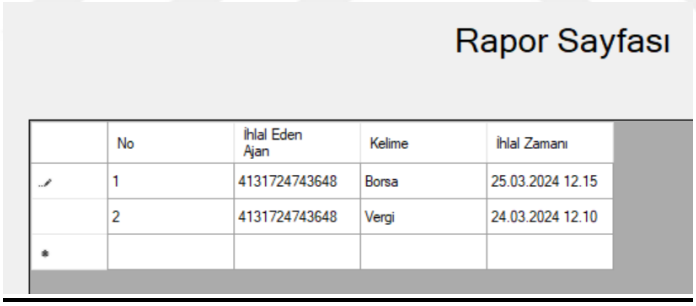
Sözlük içeriği standart değildir. Yönetici program aracılığı ile sistem yöneticisi, çalışılan kuruma özgü olan anahtar kelimelerden sisteme eklemeler yapabilir. Yönetici program üzerinden, sisteme eklenmiş filtre kelimeler listelenebilir ve istenilenler silinebilir. Sisteme eklenmiş filtre kelimeler, ajan tarafından gönderilen kelimeler ile eşleşirse, gerçek zamanlı veri tabanında ilgili ajan altında tarih ve saat verisiyle beraber filtre kelime ile bir düğüm (node) açılır. Oluşan ihlale ait ajan ve tarih verisi bu düğüm atında depolanır ve sistem yöneticisi tarafından raporlanmak istenildiğinde bu düğüme erişilerek listeleme yapılır.



Şekil 3.6: Yasaklı Kelime Ekleme Ekranı.

3.1.1.3. Rapor İşlemleri

Raporlama bölümü ajan tabanlı veri sızıntısı önleme sisteminin en önemli kısımlarındandır. Bütün sistem, oluşabilecek muhtemel sızıntıları tespit amacıyla kurulduğundan, sızıntıları görüp hızlı bir şekilde müdahale edebilmek oldukça önemlidir. Ajan tarafından veri tabanı üzerinden karşılaştırılıp sözlükteki kelimelerle eşleşme sağlayan kelimeler, gerçek zamanlı veri tabanı üzerinde raporlar (reports) düğümü altında toplanmaktadır. Sistem yöneticisi, oluşmuş olan ihlalleri görüntüleyebilir ve ilgili ID numarasına sahip uç nokta için kurum içi düzenlenmiş eylem planına göre hareket edebilir. Bu ekran Şekil 3.7'de görülebilir.



Şekil 3.7: Raporlama Ekranı.

3.1.1.4. Büyük Dil Modeli İşlemleri

Büyük Dil Modeli (BDM), sistem kaynaklarını kullanmak isteyen yazılımlarla bir Uygulama Programlama Arabirimi (Application Programming Interface-API) aracılığı ile haberleşir. Veriler büyük dil modeline API kullanılarak gönderilir. API; 2 veya daha fazla uygulamanın belirli protokoller ve iletişim modelleri üzerinden veri alışverişi yapması ve iletişime geçmesini sağlayan bir ara yazılımdır [34]. Farklı büyük dil modellerinin kendilerine has API sistemlerine bulunmaktadır. Bu çalışmada kullanılan büyük dil modeli, sağladığı API kodlarının çalışabilmesi için uygulama yazılımında rol ve içerik bilgisinin gönderilmesini istemektedir [35]. Tanımlanacak rol, sistem (system) veya kullanıcı (user) rolü şeklinde olabilir.

a) Sistem rolü: Doğrudan BDM'nin vereceği cevapları özelleştirmek için kullanılacak roldür. Bu role giren BDM, kendisine yöneltilen sorulara dahil olduğu rolde belirtilen özelliğe göre yanıt verecektir. Örneğin, geliştirilen sistem bankacılık sektöründe veri sızıntısı önleme amaçlı kullanılacaksa; şeklinde bir rol verilebilir. Şekil 3.8'de örnek bir rol eklemesi gösterilmiştir

```
"messages": [
{ "role": "system",
 "content": "Sen bir coğrafyacısın ve veri sızıntısı önleme, siber güvenlik ve kişisel veriler konusunda uzmansın."}
] }
```

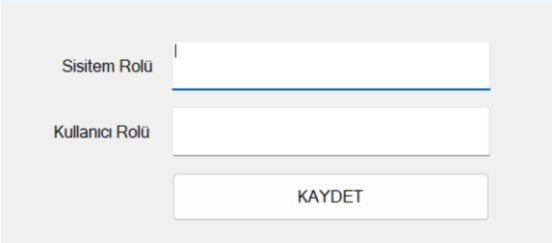
Şekil 3.8:BDM Sistem Rolü.

b) Kullanıcı Rolü: Bu rolde BDM, kendisine yöneltilen soruları karşısında standart bir kullanıcı olduğunu varsayarak yanıtlayacak ve iletişim birebir konuşma şeklinde devam edecektir. Kullanıcı için örnek bir rol Şekil 3.9'da gösterilmiştir.

```
"messages": [
{
  "role": "user",
  "content": "Türkiye'nin başkenti neresidir?"
```

Şekil 3.9:BDM Kullanıcı Rolü.

c) Ajan tabanlı veri sızıntısı önleme sisteminde, girilen kelimelerin sözlükle karşılaştırılmasına paralel olarak, oluşması muhtemel kişisel veri sızıntılarına karşı büyük dil modeli tarafından da kontrol edilmesi planlanmıştır. BDM'nin çalışabilmesi için gerekli olan rol, kurumun çalışma alanı, istek ve ihtiyaçları için özelleştirilebilir nitelikte planlanmıştır. Sistemin kullanılacağı kuruma göre belirlenen rol, yönetici programdaki Ayarlar menüsünden yapılmaktadır. Şekil 3.10'da görülen ekranda hem sistem hem de kullanıcı için rol ataması yapılabilmektedir.



Şekil 3.10: Rol Ekleme Ekranı.

3.1.2. Verilerin Toplanması

Sistemin veri sızıntılarını tespit altyapısının kurulmasından sonra veri toplama işlemi başlatılır. Bu işlem için her bir uç noktaya çalışanın klavye hareketlerini takip eden bir ajan kurulumu yapılır. Ajan yazılımı kurum içerisinde veri sızıntısı kontrolü yapılmak istenen her bir uç noktaya kurulan ve çalışanın ve sistemin performansını etkilemeden arka plandan çalışarak veri toplayan, topladığı verileri hem sözlük filtrelemesinde hem de BDM üzerinden kontrol yaptırarak sızıntı tespit eden bir araçtır.

Sistem bir uç noktaya kurulmak istenildiğinde, ilk olarak uç nokta üzerinde daha önceden kurulu olup olmadığını denetleyecek bir şekilde kodlanmıştır. Bu işlem için, kurulum aşamasında girilen benzersiz ajan ID'si, tarih ve saat bilgilerinin işletim sistemindeki program klasörüne bir dosya içerisine kaydetme yöntemi kullanılmıştır. Program açılışta bu bilgileri sorgulayacak ve eğer erişim sağlayıp daha önce kurulduğu tespit edilirse otomatik olarak çalışmaya ve veri toplamaya başlayacaktır. Eğer dosyaya erişemezse veya dosya içeriği boşsa bunu ilk kurulum olarak değerlendirecek ve bu aşamada kuruluma devam edebilmek için yönetici program tarafından oluşturulan ID bilgisinin sisteme girilmesi gerekecektir.

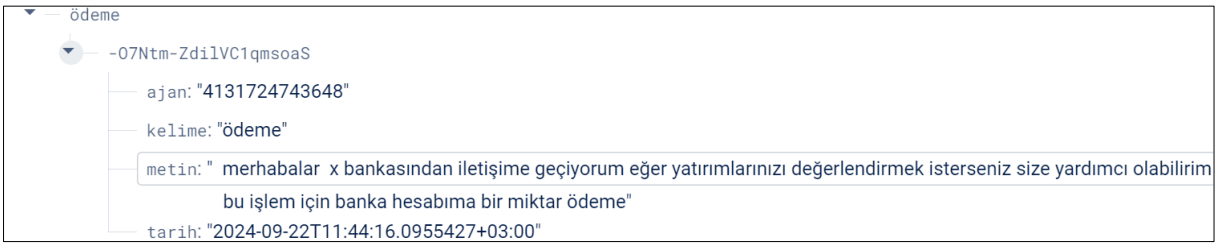
Sistem ilk kurulumda ve sonrasında günlük olarak gerçek zamanlı veri tabanından sözlük tablosunu bir listeye çekmekte ve sorgulamaları çevrimdışı (offline) olarak sürdürmektedir. Bu da sürekli yazılan her kelimede veri tabanı erişimi gerekliliğini ortadan kaldırmakta ve performans artışı sağlayarak sistem kaynaklarının gereksiz kullanımının önüne geçmektedir. Kurumun ihtiyaçlarına göre bu süre kısaltılabilir veya elle (manuel) olarak filtre veri tabanı güncellemesi yapılabilir.

Çalışmaya başlayan yazılım, işletim sisteminin DLL dosyalarını kullanarak sistem kaynaklarına kanca (Hook) atmakta ve klavyeden basılan her karakteri yakalamaktadır. Yakalanan karakterler, kullanıcı bir boşluk veya enter tuşuna basıncaya kadar birleştirilerek bir

dizi elde edilmektedir. Eğer boşluk ya da enter tuşuna basılmışsa dizi sonlandırılmakta ve gerçek zamanlı veri tabanı üzerinde yönetici tarafından eklenerek oluşturulan sözlükle kıyaslayarak sızıntı tespit çalışması yapmaktadır. Bu aşamada klavyeden girilen Unicode karakterler dönüşüme sokulmakta ve alfabe harici karakterler ayıklanmaktadır. Standart olarak yer almayan Türkçe karakterler (ç,ö,ş,ü,ğ,ı) bu noktada kod karşılaştırması yapılarak sisteme dahil edilmektedir. Bu sayede Türkçe dil desteği de sisteme eklenmiş olmaktadır. Kelime oluşturma işlemi, kullanıcı klavyeden giriş (Enter) ya da boşluk (Space) tuşuna basıncaya kadar devam etmektedir. Karakter dizileri birbirlerinden bu 2 tuşla ayrılmakta ve sorgulanmak üzere kelimeler oluşturulmaktadır.

3.1.3. Sözlük Tabanlı Kelime Eşleme Yöntemiyle Sızıntı Tespiti

Sistem uç noktalara kurulduktan sonra otomatik olarak veri toplamaya başlar. Kullanıcının yazdığı her kelime veri toplama maddesinde anlatıldığı şekliyle tespit edilir. Sistem herhangi bir zamanda sözlükteki kelime ile eşleşen bir karakter dizisiyle karşılaşırsa veri tabanına tarih, saat, veri ihlali yapan uç nokta ID'si ve ihlal edilen kelimeyi raporlamaktadır. Bu noktada sadece ihlal edilen kelime tek başına bir anlam içermeyebileceğinden veya farklı anlamlar içerebileceğinden dolayı ihlal gerçekleştiği kelime dahil olmak üzere son 20 kelimeden oluşan metin bloğu yöneticiye raporlanmaktadır. Bunun için, yazılanlar 20 kelimelik diziler halinde ayrı bir şekilde depolanmaktadır. Şekildeki örnekte yönetici tarafından yasaklı kelime olarak belirlenmiş olan "ödeme" kelimesi için bir ihlal gerçekleşmiş ve yasaklı kelime dahil ihlal gerçekleşinceye kadar yazılan son 20 kelime raporlanmıştır. Bu durum Şekil 3.11'de gösterilmiştir.



Şekil 3.11: Gerçek Zamanlı Veri Tabanına İhlal Kaydı Girilmesi.

Kelime oluşturma ve sözlükten yasaklı kelime kontrol işlemi Deterministik Sonlu Otomat (Deterministic Finite Automata-DFA) kullanılarak şöyle gösterilebilir [36];

- Kullanıcı giriş sonlandırma karakterlerine (enter veya boşluk) tuşuna basıncaya kadar her seferinde bir harf girmektedir ve bu harfler birbirine eklenmektedir.
- Giriş (enter) veya boşluk (space) tuşuna basıldığında harf alma işlemi sonlandırılmakta ve birleştirilen harflerden bir dizi(kelime) oluşturulmaktadır.
- Her adımda bu kelime adayı yasaklı kelime sözlüğüyle karşılaştırılmaktadır.
- Eğer dizi, sözlük ile eşleşirse bu bir $qkabul$ durumu anlamına gelmekte, kullanıcıya mesaj gösterilmekte ve işlem $q0$'a dönmektedir.
- Eğer dizi, sözlük ile eşleşmezse bu bir $qret$ durumu anlamına gelmekte ve raporlama yapılmadan işlem yine $q0$'a dönmektedir.

DFA Modeli:

1. Durumlar (States): Her durum, kelime oluşturma ve sözlükten kontrol etme işleminin bir adımını temsil eder. Başlangıç durumu, henüz hiçbir harfin girilmediği durumdur.

   a) $q0$ (Başlangıç durumu): Kullanıcı henüz harf girmemiş veya yeni bir kelime oluşturmaya başlama aşamasındadır.
   b) $qkelime$ (Kelime oluşturma durumu): Kullanıcı herhangi bir harfe bastığında bu duruma geçilir ve harfler birleştirilerek geçici bir kelime oluşturulur. Bu durum, kullanıcı ENTER veya SPACE tuşuna basana kadar devam eder.
   c) $qkontrol$ (Kontrol durumu): Kullanıcı ENTER veya SPACE tuşuna bastığında, birleştirilen harfler bir kelime olarak kabul edilir ve yasaklı kelimeler sözlüğünde aranır.
      - Eğer kelime yasaklıysa, TRUE mesajı gösterilir ve $qkabul$ durumuna geçilir.
      - Eğer yasaklı değilse, FALSE mesajı gösterilir ve $qret$ durumuna geçilir.

   d) $qkabul$ (Kabul durumu): Kelime yasaklı olduğunda bu duruma geçilir ve raporlama yapılır.
   e) $qret$ (Reddedilme durumu): Kelime yasaklı olmadığında bu duruma geçilir ve işlem burada sonlanır, ardından başlangıç durumuna dönülerek yeni harf almaya başlanır.

2. Giriş Alfabesi (Input Alphabet): Türkçe ve İngilizce alfabedeki harflerin birleşimine ek olarak, ENTER veya SPACE tuşlarıdır. (a, b, c, ç, ... ,x, y, z, ENTER, SPACE).

3. Durum Geçişleri (Transitions): Her harf alındığında sonlandırma karakterlerinden birisi basılmış mı diye kontrol edilir. Eğer sonlandırma karakterlerine basılmamışsa, yeni bir harf daha alınır ve sürece devam edilir. Eğer basılmışsa kontrol durumuna geçilir.

   a) $q0 \rightarrow qkelime$: İlk harf girildiğinde $qkelime$ durumuna geçilir ve harf birleştirilmeye başlanır.
   b) $qkelime \rightarrow qkelime$: Her harf girildiğinde, mevcut harfler birleştirilir ve yeni bir geçici kelime oluşturulur.
   c) $qkelime \rightarrow qkontrol$: Kullanıcı ENTER veya SPACE tuşuna bastığında, birleştirilen harflerle bir kelime oluşturulur ve yasaklı kelimeler sözlüğünde arama yapılır.
   d) $qkontrol \rightarrow qkabul$: Eğer oluşturulan kelime yasaklıysa, $qkabul$ durumuna geçilir.
   e) $qkontrol \rightarrow qret$: Eğer kelime yasaklı değilse, $qret$ durumuna geçilir.
   f) $qkabul \rightarrow q0$: Kullanıcıya uyarı mesajı gösterildikten sonra tekrar yeni bir kelime oluşturmaya başlanır.
   g) $qret \rightarrow q0$: Yeni harfler alınıp yeni bir kelime oluşturmaya başlanır.

4. Kabul Durumları (Accepting States): Oluşan kelimenin sözlükte bulunan bir kelime ile eşleşme durumudur.

Kümelerin Oluşumu:

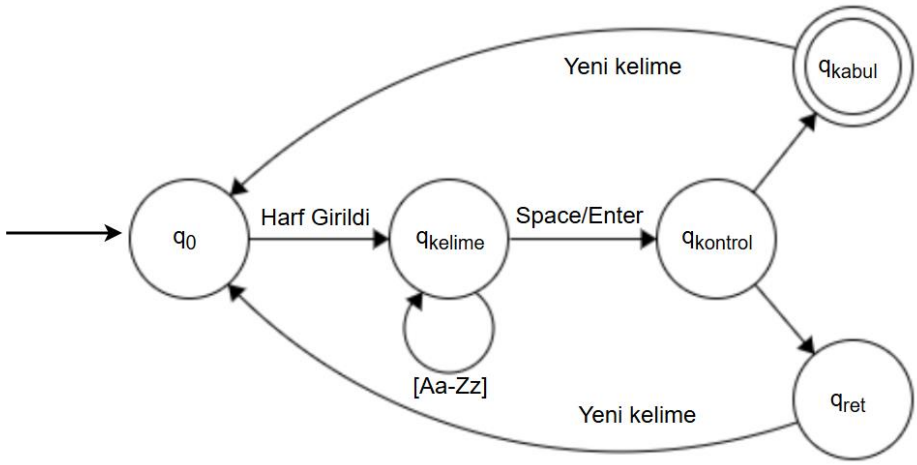
Her harf alındığında, yeni bir alt küme oluşturulur. Örneğin, ‘c’ girildiğinde bu bir küme oluşturur. Sonra ‘e’ girilmişse, bu da kümenin genişlemesine neden olur (yeni dizi ‘ce’ olur). Bu şekilde her yeni harf bir alt küme gibi düşünülebilir ve bu alt kümeler sonlandırma karakterleri basılıncaya kadar oluşturulmaya devam eder.

Örnek:

$$\boldsymbol{\Sigma}: \{Türkçe\ ve\ İngilizce\ Alfabe\ +\ boşluk\ +\ enter\} \qquad (3.1)$$

- Başlangıç durumu q_0: kullanıcıdan harf bekliyor.
- Geçiş $q0 \rightarrow qkelime$: "c" harfi girildi, birleştirme devam ediyor.

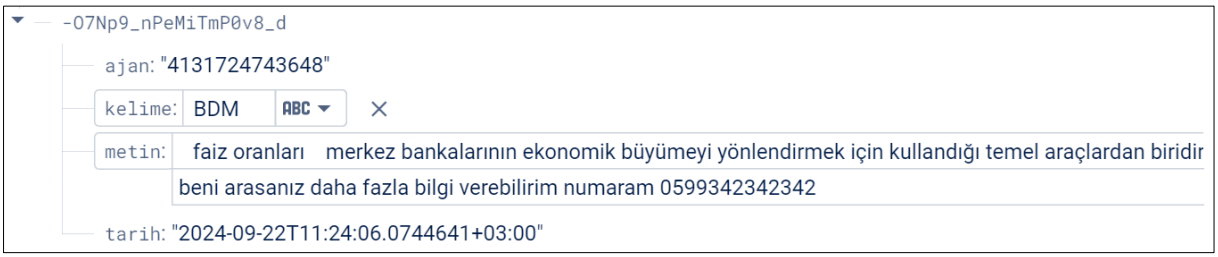
- Devam eden geçişler: "ce" harfi oluştu, birleştirme devam ediyor.
- Boşluk veya enter tuşuna basıldı: Birleştirme sonlandırıldı ve $qkontrol$ durumuna geçildi .
- Son durum $qkabul$: Kelime sözlükte bulundu. Durum $qkabul \rightarrow q0$'a geçiş yapar ve yeni bir kelime için otomata sıfırlanır. Tüm adımlar Şekil 3.12'de gösterilmiştir.



Şekil 3.12: Yasaklı Kelime Tespit Otomatı.

3.1.4. Büyük Dil Modeli ile Kişisel Veri İhlali Tespiti

Büyük dil modelinin kullanımı için izlenen yol şu şekildedir. Yönetici tarafından kurumun türü ve çalışma alanına göre sisteme girilmiş olan rol üzerinden işlem yapılmaktadır. Büyük dil modeline karakterler halinde ya da kelimeler halinde sızıntı tespiti yaptırmak, BDM'nin ortalama yanıt süresine göre [37] teoride yapılabilir gibi görünse de pratikte uygun değildir. Bu yüzden uç nokta kullanıcısının yazdığı kelimeler 20 kelimelik bloklar halinde uç uca eklenerek bir paragraf haline getirilip BDM'ye gönderilmektedir. BDM'den dönen değere göre de raporlama yapılmaktadır. Örnek bir veri sızıntısı şekil 3.13'de gösterilmiştir. Burada çalışanın kendi telefon numarası, ihlal olarak değerlendirilerek son 20 kelime raporlanmıştır.



Şekil 3.13: Gerçek Zamanlı Veri Tabanı Kişisel Veri İhlal Kaydı Girilmesi.

Test aşamasında, yazılımın bankacılık sektöründe kullanılacağı bir ortam kurgulanmış ve BDM için aşağıdaki roller yazılmıştır;

Sistem Rolü: “Sen finans sektöründe çalışıyorsun, siber güvenlik, veri sızıntısı ve kişisel verilerin korunması konusunda uzmansın.”

Kullanıcı Rolü: “Sana vereceğim paragraf içerisinde kişisel veri olup olmadığını denetleyip sonucu “evet” ya da “hayır” diyerek bana iletebilir misin?”

Eğer BDM’den dönen değer “Evet” ise veri tabanında raporlama bölümüne ihlal kaydı girilmekte ve hazırlanmış olan paragraf veri tabanına yazılmaktadır.

3.1.5. Raporlama

Sistemin raporlanması, sistem kurulumunda hazırlanan gerçek zamanlı veri tabanı altyapısı doğrultusunda sızıntının gerçekleştiği an bildirimde bulunma şeklinde gerçekleşmektedir. Bu işlem, veri tabanında ilgili düğümlerin (node) dinlenmesiyle gerçekleştirilir. Raporlama düğümü üzerinde herhangi bir yeni kayıt eklenmesi durumunda dinleyiciye bildirimde bulunulur. Dinleyici de bu bildirimi yöneticiye ileterek belirlenmiş önlemlerin alınması sağlanır. Önlem alma işlemleri bildirime göre otomatik olarak veya bildirimin içeriğine ve önem derecesine göre isteğe bağlı olarak yapılabilir.

4. BULGULAR

Ajan tabanlı veri sızıntısı önleme sistemi, içerisinde yasaklı kelimeler ve kişisel veriler bulunduran bir metin ile test edilmiş ve test sonuçları değerlendirilmiştir. Aynı zamanda test işlemi süresince test için kullanılan bilgisayarın işlemci ve ana bellek kayıtları tutularak uygulamanın sistem üzerinde oluşturacağı yük belirlenmiştir.

4.1. TEST VERİSİ SONUÇLARININ DEĞERLENDİRİLMESİ

Veri sızıntısı önleme sitemi için Tablo 4.1'de gösterilen şekliyle bir senaryo hazırlanmış ve bu senaryoya göre yönetici program ve uç nokta ajanlarının kurulumları gerçekleştirilmiştir.

Tablo 4.1: Sistem Test Senaryosu.

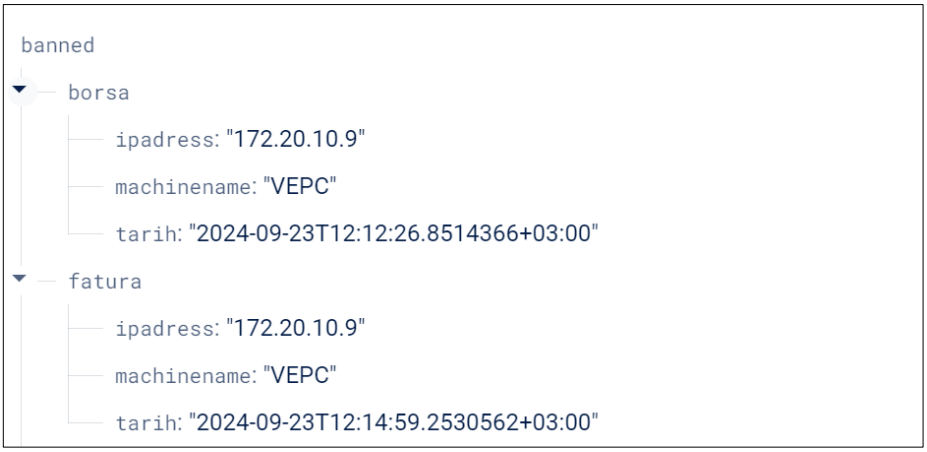
Sektör:	Bankacılık
Uç Nokta Sayısı:	6

Sistem yöneticisi Tablo 5'te listelenen yasaklı kelimeleri belirlemiş ve yönetici program üzerinden sisteme giriş yapmıştır. Kelimeler belirlenirken sızıntı olarak değerlendirilecek olanların farklı yazılışları da sisteme dahil edilmiştir. Ayrıca Türkçe dil desteği testi yapılabilmesi için hem İngilizce hem de Türkçe harfler içeren kelimeler seçilmiştir.

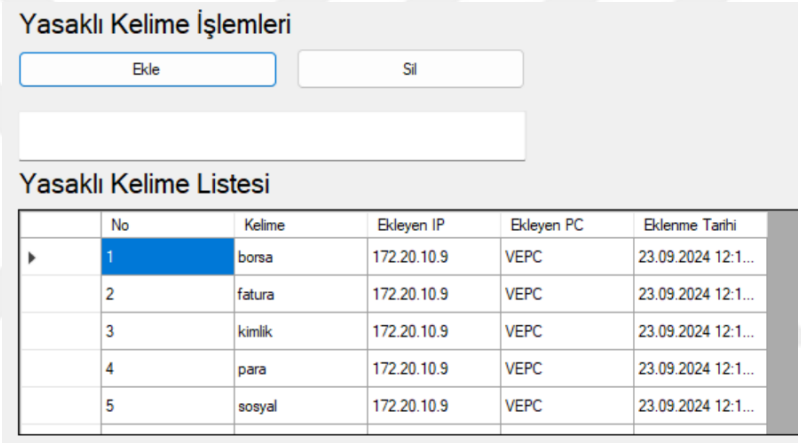
Tablo 4.2: Örnek Yasaklı Kelime Listesi.

Ödeme	Tax	Borsa
Ödemeler	Telefon	Para
Yatırım	Instagram	Virman
Kimlik	Sosyal	Fatura

Belirlenen yasaklı kelimeler Şekil 4.1'de görüldüğü şekliyle yönetici program aracılığı ile gerçek zamanlı veri tabanına eklenmiştir. Sisteme eklenmiş olan yasaklı kelimeler yönetici ekranında Şekil 4.2'te olduğu gibi görüntülenmektedir.



Şekil 4.1: Yasaklı Kelimelerin Veri Tabanına Kayıt Edilmesi.



Şekil 4.2: Yasaklı Kelimelerin Sisteme Eklenmesi.

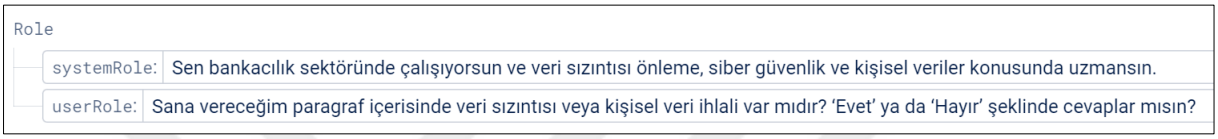
Büyük dil modeli aracılığı ile kişisel veri sızıntılarının tespiti için ise sistem yöneticisi Tablo 4.3'teki rolleri eklemiştir.

Her bir uç nokta için içerisinde hem yasaklı kelimeler hem de kişisel verilerin olduğu paragraflar hazırlanmış ve uç nokta kullanıcılarının bu paragrafları sisteme girmeleri istenmiştir.

Tablo 4.3: Test Amaçlı Sistem Rolleri.

Sistem Rolü:	Sen bankacılık sektöründe çalışıyorsun ve veri sızıntısı önleme, siber güvenlik ve kişisel veriler konusunda uzmansın.
Kullanıcı Rolü:	Sana vereceğim paragraf içerisinde veri sızıntısı veya kişisel veri ihlali var mıdır? 'Evet' ya da 'Hayır' şeklinde cevaplar mısın?

Belirlenen roller yönetici program aracılığı ile gerçek zamanlı veri tabanına eklenmiştir. Bu işlem Şekil.4.3'te gösterilmiştir.



Şekil 4.3: Rollerin Veri Tabanındaki Görünümü.

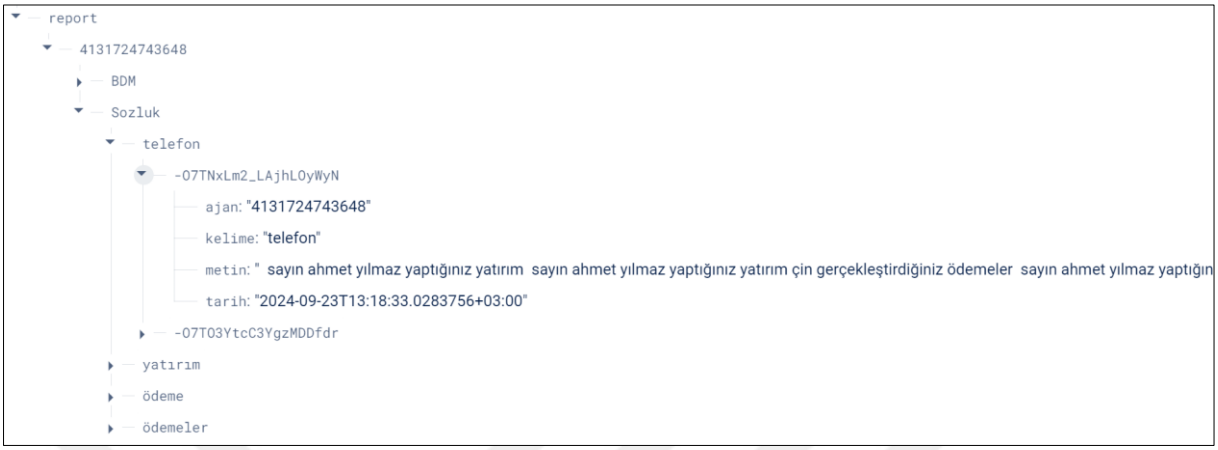
Uç nokta kullanıcısının sistemi test etmesi için belirlenen paragraf Tablo 4.4'te belirtildiği şekildedir.

Tablo 4.4 : Sistem Test Metni.

Sayın Ahmet Yılmaz, Yaptığınız yatırım için gerçekleştirdiğiniz ödemeler tarafımıza başarılı bir şekilde ulaşmıştır. Ödeme sırasında kullandığınız telefon numaranızın 0532 123 4567 olduğunu görüntülemekteyim. Ödemelerin doğrulanması için birazdan telefon numaranıza bir SMS göndereceğim. Lütfen SMS ile gelen şifreyi bana iletir misiniz? Saygılarımızla, XYZ Bankası Müşteri Hizmetleri

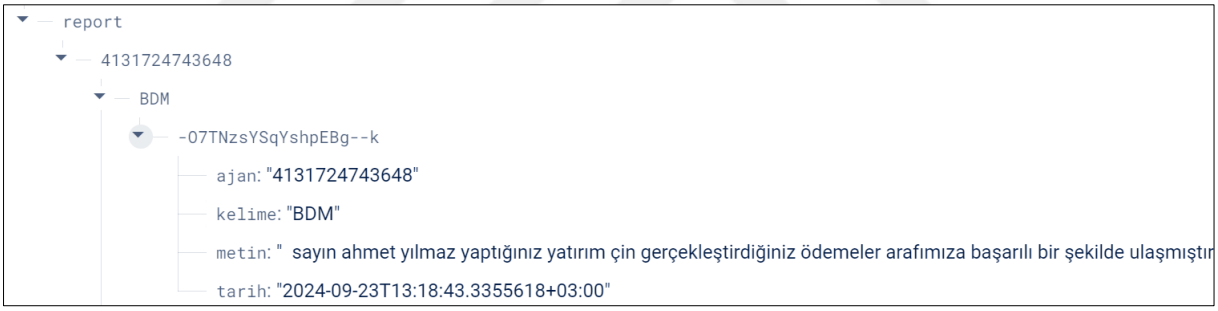
Bu paragrafta, sistem yöneticisi tarafından belirlenen yasaklı kelimelerden "ödemeler", "ödeme", "telefon" ve "yatırım" kelimeleri geçmekte ve kişisel veri ihlali olarak ise müşterinin ismi ve telefon numarası yer almaktadır. Belirlenen paragrafın uç nokta kullanıcısı tarafından sisteme girilmesi sonrasında sistem belirlenen kelimeleri başarıyla yakalamış ancak yasaklı kelimelerle aynı kökten olan "ödemelerin" kelimesini yasaklı olarak kabul etmemiştir. Paragraf

girişi sonrası veri tabanı sözlük alanında raporlama ekranı görünümü Şekil 4.4'te gösterildiği gibidir.



Şekil 4.4: Yasaklı Kelime İhlali Veri Tabanı Görünümü.

Büyük dil modelinin tespit etmesi beklenilen isim ve telefon numarası gibi kişisel verilerin durumu ise rapor ekranında Şekil 4.5'te gösterilmiştir.



Şekil 4.5: Kişisel Veri İhlali Veri Tabanı Görünümü.

Her 2 ekran da incelendiğinde BDM tarafından kişisel veri ihlali olan paragraf , saat, tarih ve ihlali yapan ajan alanlarıyla listelenerek raporlanmıştır.

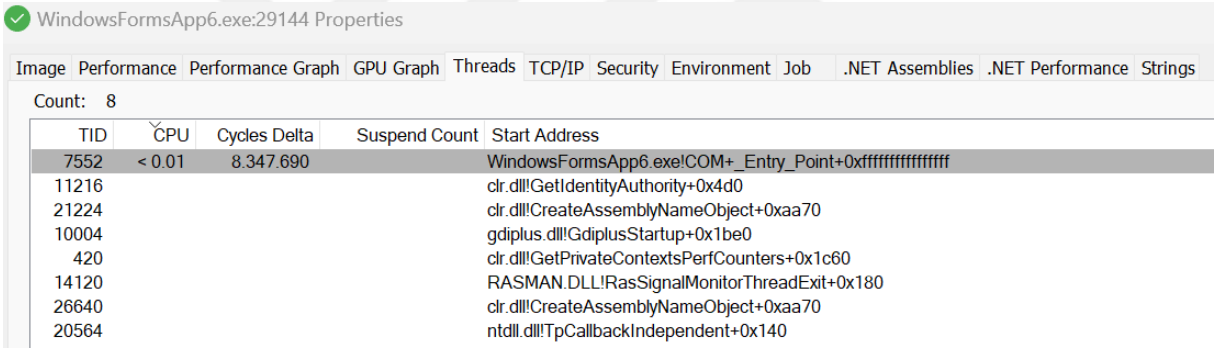
Yasaklı kelimelerin eklendiği sözlükte geçen kelimeler ise %100 doğrulukta tespit edilerek yine ihlal yapan ajan, ihlal öncesi yazılan son 20 kelime ve zaman bilgisiyle beraber raporlanmıştır. Burada "ödemelerin" kelimesinin raporlanmaması geliştirilmesi gereken alanlar arasına not edilmiştir. Ayrıca BDM modeline gönderilen verinin incelenmesi sonucunda dönen değer, farklı altyapılara sahip sistemlerinde kullanılmaya uygun olacak şekilde standart bir ara form olan JSON formatında da görüntülenebilir. Test paragrafının kişisel veri ihlali tespiti sonucu JSON formatlı Şekil 4.6'da gösterildiği gibidir.

```
{
    “kişisel_veri_ihlalleri”: {
        “isim”: “Ahmet Yılmaz”
    },
    “Telefon_Numarası”: {
        “1.Telefon”: “0532 123 4567”
}
}
```

Şekil 4.6: Kişisel Veri İhlali JSON Çıktısı.

4.2. PERFORMANS ANALİZİ

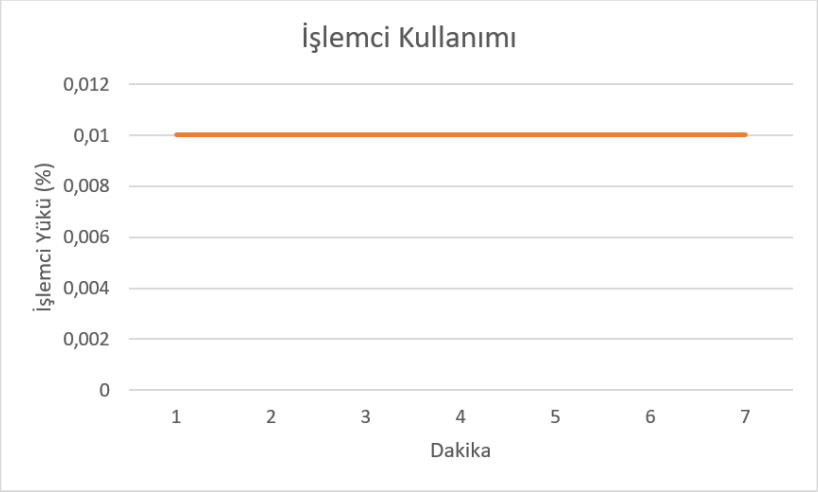
Ajan yazılımın çalışma zamanındaki sisteme yükü Microsoft Windows Process Explorer [38] yazılımıyla test edilmiş, işlemci ve bellek kullanımları uç nokta kullanıcısının klavye girdi hızlarına göre kaydedilmiş (Şekil 4.7) ve performans-zaman grafiğinde (Şekil 4.8) gösterilmiştir. Veri giriş hızından ve miktarından bağımsız olarak uygulamanın işlemci yükünün tüm işlemci gücünün 0,01%'inden daha az olduğu tespit edilmiştir.



Şekil 4.7: Uygulama Sistem Yük Ekranı.

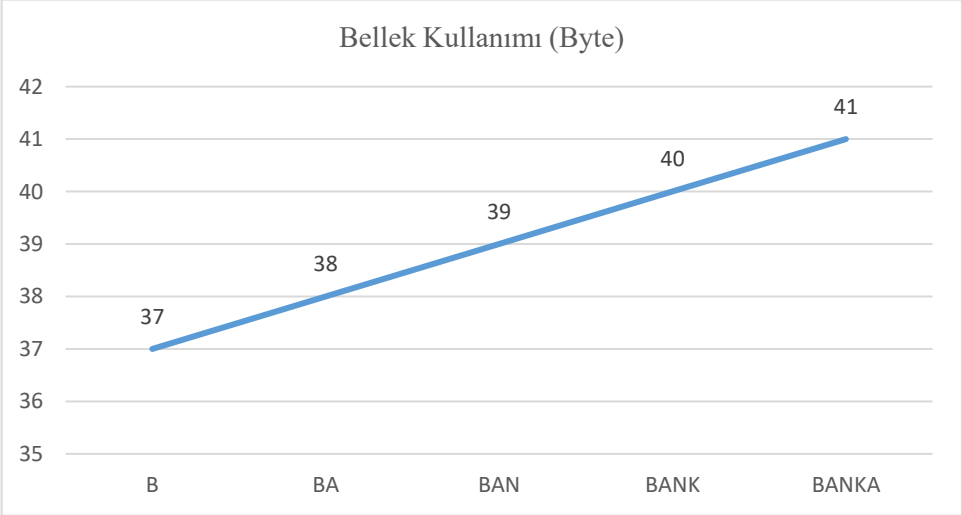
Uygulamanın sistem üzerinde oluşturduğu yük miktarını belirlemek için test amaçlı kullanıcı tarafından 674 kelime ve 11 paragraftan oluşan bir metnin aralıksız sisteme girilmesi sağlanmıştır. Metin içerisinde hem yönetici tarafından oluşturulmuş yasaklı kelimeler hem de yasaklı olmamasına rağmen kişisel veri ihlaline sebep olabilecek veriler yer almaktadır. Kullanıcı bu metni 7 dakika 23 saniyede sisteme girmiş ve bu süre içerisinde uygulamanın işlemci ve bellek kullanımı kayıt altına alınmıştır.

Test sonuçlarına göre bellek kullanımı, veri giriş hızına bağlı olarak değişkenlik göstermektedir ancak işlemci kullanımı sabit olarak 0,01% değerinin altında kalmıştır. Bu da çalışmanın temel hedeflerinden birisi olan son kullanıcıya performans yükü oluşturmama maddesinin gerçeklenmiş olduğunu göstermektedir.



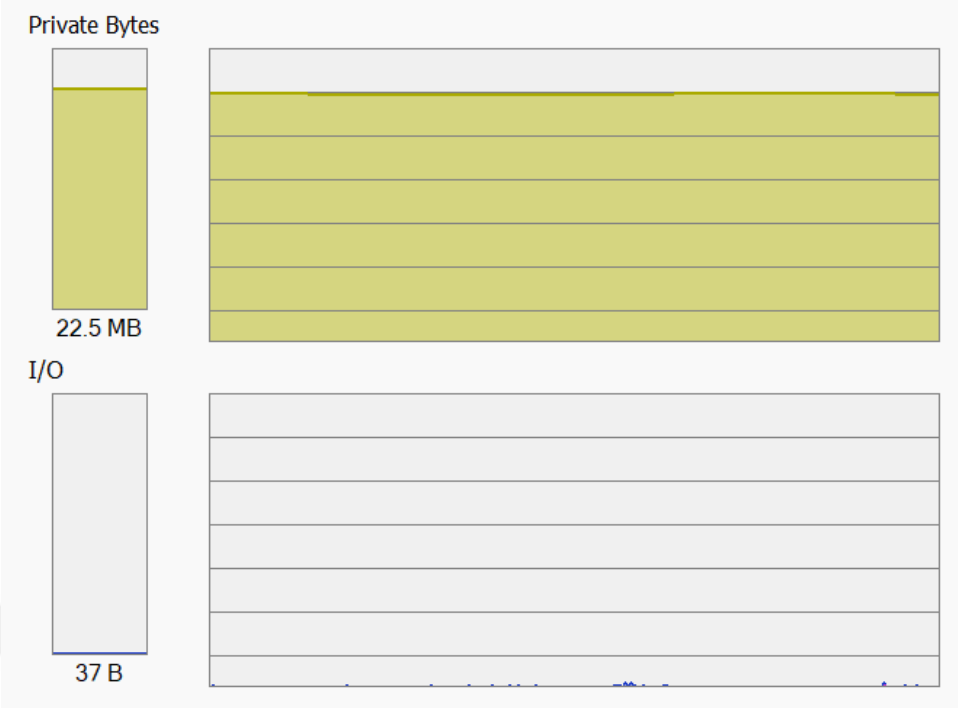
Şekil 4.8: Zamana Göre İşlemci Yükü.

Bellek kullanımında ise G/Ç (I/O) işlemi olarak klavyeden basılan ilk harf 37B yer kaplarken eklenen her harf ilave 1B olarak kaydedilmiştir. Örnek olarak "B" harfine basılıp kelimeyi bitirmek için ön tanımlı olan boşluk veya enter tuşuna basıldığında 37B değeri izlenmiştir. İkinci kelime "BA" olarak seçildiğinde bellek kullanımının 38B'a yükseldiği tespit edilmiştir. Örnek olarak yazılan "BANKA" kelimesinin harf sayısına bağlı bellek kullanım grafiği Şekil 4.9'de gösterilmiştir.



Şekil 4.9: Harf Girdisi Başına Bellek Kullanımı

Şekil 4.10'da görüldüğü üzere uygulamanın çalışma zamanında ana bellekte (RAM) kapladığı alan ise 22.5 MB olarak kaydedilmiştir.



Şekil 4.10: Harf Girdisi G/Ç Bellek Kullanımı.

Yasaklı kelime için sözlük kontrolü her kelimede yapılmaktadır ancak tek bir kelime kendi başına kişisel veri ihlali anlamına gelmeyebilir. Kullanıldığı cümleye göre bir kişisel veri durumuna gelebilir. Bu yüzden kelimeler tek tek BDM ile sorgulanmak yerine 20 kelimelik paragraflar haline getirilip o şekilde BDM'ye gönderilmiştir. BDM 20 kelimelik blokları, kendisine daha önceden verilmiş role göre incelemiş ve kişisel veri analizi yapmıştır. Sistem yük analizinde 20 kelimelik blokların gönderilmesi esnasında anlık G/Ç kullanım 860B olarak ölçülmüştür. Bu sonuca göre de uygulamanın bellek yükünün standart uç nokta için performansı etkilemeyecek düzeyde olduğu görülmüştür.

4.3. BULGULARIN DEĞERLENDİRİLMESİ

Ajan tabanlı veri sızıntısı önleme sistemi, veri sızıntılarına karşı literatürdeki çalışmalar incelenerek hazırlanmış ve benzer çalışmalarda eksik görülen alanların tamamlanması ile literatüre katkı yapmak amaçlanmıştır. Araştırma sonucunda, önerilen veri sızıntısı önleme yöntemlerinin, bilgisayardan dışarıya fiziksel aygıtlar veya internet aracılığı ile veri çıkışının engellenmesi, bilgisayarda oluşturulan dosyaların hassaslık durumlarına göre sınıflandırılması ve bu dosyalara erişim durumunda önleme sistemlerinin çalıştırılması şeklinde genellenebileceği görülmüştür. Bu çalışmada hedeflenen ise veri sızıntısına konu edilebilecek bilgilerin henüz dosyaya dönüşmeden, kullanıcının yazdığı an tespit edilmeye çalışılması ve bu

amaçla bir yasaklı kelime sözlüğü oluşturulup kelimelerin yazıldığı an yakalanabilmesi için sistemi sürekli dinleyen bir ajan sisteminin hazırlanmasıdır. Ayrıca hazırlanan bu ajan, yazılanları belirli bloklar haline getirerek BDM üzerinde kişisel veri ihlali sorgusuna da tabi tutacaktır. Tüm bu işlemlerin uç noktanın performansını etkilemeyecek düzeyde olması ve Türkçe dil desteği içermesi de hedefler arasındadır.

Hazırlanan sistem, çeşitli uzunlukta metin girdileri ile test edilmiştir. Bu metimler içerisinde yasaklı kelimeler, yasaklı kelimelerin ek almış halleri, Türkçe ve İngilizce içerikli kelimeler ve kişisel veri ihlaline konu edilebilecek veriler bulunmaktadır.

Yapılan test sonuçlarına göre ajan yazılımın sistem performansına etkisinin çok düşük olduğu gösterilmiştir. Raporlama bölümü incelendiğinde ise yasaklı kelimeler arasında Türkçe karakterler içeren kelimelerin de sistem tarafından 100% oranda yakalandığı görülmüştür.

Metin içerisine eklenen kişisel veriler, BDM sorgusu sonucunda başarıyla tespit edilmiş ve bu ihlal de raporlama şeklinde sistem yöneticisine bildirilmiştir.

Ajan'ın yasaklı kelimelerin ek almış hallerini yasaklı kelime olarak yakalayamadığı görülmüştür. Bununla ilgili muhtemel çözüm önerileri tartışma bölümüne ilave edilmiştir.

Sistemin yasaklı kelime yakalandıktan sonra ve kişisel veri ihlali tespit edildikten sonra nasıl bir aksiyon alacağı bu çalışmanın dışında tutulmuştur. Bu durumlarla ilgili de yapılabilecekler tartışma bölümünde değerlendirilmiştir.

Uç nokta kullanıcısının klavye aracılığı ile yazdığı bilgileri herhangi bir ihlal olmadığında sistem yöneticisi dahil olmak üzere ikinci bir kimsenin görme yetkisi bulunmamaktadır. Sadece bir ihlal gerçekleştiğinde raporlama amacıyla sınırlı sayıda kişi tarafından görülebilecektir. Bunun için de uygulama başlangıcında uç nokta kullanıcısına bu durum bildirilmekte ve kullanıcının kabulü sonrasında ajan programı çalıştırılmaktadır. Bu şekilde de kişisel verilerin gizliliği ilkesine uyulması amaçlanmıştır.

5. TARTIŞMA

Veri sızıntısının önlenmesi, muhtemel sızıntı yollarının kapatılması veya önlem alınması şeklinde yaklaşımlarla literatürde yer bulmuştur. Çalışmalarda hem kasıtlı veri sızmaları hem de istem dışı sızıntılara karşı ne gibi önemler alınabileceği ile ilgili tavsiyelerde bulunulmuştur. Literatürde geçen çalışmalar dosyanın sınıflandırılması veya erişim anında dosya ismi, oluşturma tarihi, dosyanın gönderileceği e-posta adresi gibi verilerin kontrol edilmesiyle muhtemel sızıntıların önüne geçmeyi amaçlamaktadır. Ajan tabanlı veri sızıntısı önleme sistemi ise dosya daha oluşmadan, henüz yazım aşamasında sızıntı tespiti yapmakta ve bu işleme paralel olarak büyük dil modelini kullanarak kişisel veri ihlal tespiti yapmaktadır. Bu şekilde sorunu kaynağında çözmeyi amaçlamaktadır. Bu yönüyle de literatürdeki diğer çalışmalardan ayrılmaktadır.

Literatürdeki diğer veri sızıntısı önleme sistemlerinin başvurduğu yöntem olan sınıflandırma ve erişim anında kontrol, çok fazla sistem yükü oluşturmakta, kullanıcının çalışma performansını etkilemekte ve dolayısıyla iş verimini düşürmektedir. Ayrıca performans sorununu çözmeye çalışmak için ihtiyaç duyulan güçlü bilgisayarlar ek maliyet oluşturmaktadır. Ajan tabanlı veri sızıntısı önleme sisteminde ise arka planda çalışan ajanın sisteme yükü sistem performansını ve kullanıcı iş verimini etkilemeyecek düzeydedir. Büyük Dil Modeli'nin yaptığı tespit işlemi bulut üzerinde gerçekleştiğinden yine son kullanıcının sistem kaynaklarını etkileyen bir durum söz konusu değildir. Ayrıca ihlal verileri ve ajan bilgileri yine bulut üzerinde tutulduğundan veri tabanı ile ilgili uç birimde kaynaklar sadece erişim amaçlı kullanılmaktadır. Bu da performans kaybına neden olmamaktadır.

Bu çalışmaya benzer şekilde ajan tabanlı DLP sistemi kurgulayan bir başka çalışmada [39] çoklu ajan sistemi kullanılmış, bir ajanla kullanıcının tüm aktivitesi kayıt altına alınmış ve elde edilen tüm veriler bir log dosyasında saklanmış, diğer bir ajanla ise klasik DLP teknikleriyle korunan içeriklere erişmeye çalışan bir kullanıcı olma durumu tespit edilmiş ve erişim durumunda log dosyası analizi yapılarak bir güven endeksi kurgulanmış, bu endekse göre de erişim yetkisi verilmiş veya kısıtlama yapılmıştır. Teoride mümkün gibi görünen bu sistem ile bu çalışma ile kıyaslandığında sistem üzerine binen yük ve sızıntı tespit işleminin hızı konusunda açık fark görülecektir. İlgili çalışmada hassas veri içeren bir dosyaya erişilmeye çalışıldığında başlatılan prosedür, ek ajanlar yardımıyla ilave bir güvenlik katmanı sağlamış gibi görünse de erişim sırasında log dosyasının taranması, güven endeksinin oluşması ve bu

işlem sonucunda “güvenilir” ya “güvenilmez” kararının verilmesi hem sisteme ek yük getirmekte hem de gerektirdiği işlem zamanı ile pratikte kullanışsız görünmektedir. Bu çalışmada kurgulanan ajan tabanlı veri sızıntısı önleme sisteminde ise ajan sadece tespit işlemi yapmakta, herhangi bir log dosyası tutmamaktadır. Bu da sisteme ek bir yük oluşturmamıştır.

Tablo 5.1: Ajan Tabanlı DLP Sistemlerinin Karşılaştırılması.

	Ajan Tabanlı Veri Sızıntısı Önleme Sistemi	Güvene Dayalı Veri Sızıntısı Önleme Sistemi
Ajan Sayısı	1	2
Log Tutma	Yok	Var
Klavye Hareketlerini İzleme	Var	Yok
Sistem Yükü	Düşük	Yüksek
Gerçek Zamanlı Tespit	Var	Yok
Yönetici Paneli	Var	Yok

Veri sızıntısı önleme sistemleri için yapılan araştırmada ajan tabanlı sistemlerde, işletim sisteminin İngilizce altyapısından kaynaklı Türkçe dil desteğinin bulunmadığı gözlenmiştir. Özellikle Windows işletim sistemin sistem kaynaklarına erişmek için kullandığı dinamik link kütüphane dosyaları (DLL), kullanıcının işlemlerini etkilemeden klavye tuşlarını okumak için kullandığı LowLevelKeyboardProc fonksiyonunda Türkçe karakterleri içermeyen ANSI kod sistemini kullanmaktadırlar. Türkçe karakterler için UNICODE desteği gereklidir ve bu dönüşüm ajan tabanlı veri sızıntısı önleme sitemi içerisinde yapılmıştır. Bu sayede Türkçe karakterleri içeren kelimeler de sızıntılara konu edilebilmiştir.

Bu çalışmada hedeflenen, kullanıcının klavye girdilerini DLP teknikleri ile inceleme yöntemine benzer bir çalışmada [40], kurum çalışanlarının kendi aralarında mesajlaşabilmesi için bir sohbet uygulaması geliştirilmiş ve uygulamada yazılan kelimeler DLP taramasından geçirilmiştir. Bu şekilde veri sızıntısının önlenmesi amaçlanmıştır. Anacak eğer kullanıcı, veri sızıntısına konu olacak yazışmaları sohbet (chat) uygulaması hariç başka bir alanda örneğin e-posta gönderiminde ya da dosya oluşturmada yaparsa kullanıcının yazdıkları DLP taramasından geçirilemeyeceği için sistem işlevsiz kalmaktadır. Ajan tabanlı veri sızıntısı önleme sisteminde ise çalışanın uç nokta içerisinde yazdığı tüm yazışmalar kontrol edildiğinden sistem sürekli bir tarama modundadır ve kullanıcının tüm klavye aktivitesi kontrol edildiğinden herhangi bir alanda yazdığı tüm yazılar veri sızıntısı testine tabi tutulur.

6. SONUÇ VE ÖNERİLER

Bu çalışma güncel veri sızıntısı önleme yöntemlerinin benimsediği sınıflandırma veya dosyaya erişim sırasında tarama yaparak içerikte sızıntı tespiti yapma yöntemlerinden farklı olarak sızıntıya konu olan kelime veya içeriğin henüz yazılma aşamasında tespit edilebileceği fikrinden hareketle ajan tabanlı bir veri sızıntısı önleme tespiti önermiştir. Bu amaçla kurum içerisindeki uç noktalara kurulan ajanlar aracılığıyla toplanan veriler sözlük filtresinden geçirilmiş ve tespit edilen kelimeler için sızıntı bildiriminde bulunulmuştur. Ayrıca periyodik olarak yazılan kelimelerden oluşan bloklar büyük dil modeline gönderilerek kişisel veri analizi yapılmıştır. Yapılan çalışmalar sonucunda sızıntıya konu olan ve yönetici tarafından sözlük sistemine girilen verilerin %100 oranında tespit edilebildiği doğrulanmıştır.

Telefon numarası, adres veya isim gibi kişisel verilerin büyük dil modeli tarafından hassas veri olarak algılanması sonucunda kullanıcının yazıları içerisinde var olan kişisel veriler de yöneticiye sızıntı ihtimali göz önünde bulundurularak raporlanmıştır. Bu amaçla yapılan testlerde kişisel veri ihlallerinin tespit edilebildiği doğrulanmıştır.

Çalışmada tespit edilen sızıntılar gerçek zamanlı veri tabanı üzerinde tutulmuş, bu şekilde de oluşması muhtemel sızıntıların yöneticiye anlık olarak raporlanması ve isteğe bağlı olarak aksiyon alınabilmesi amaçlanmıştır.

Bu çalışma sızıntıyı tespit etmeye odaklanmış olup tespit sonrası alınacak önlemler kapsam dışında tutulmuştur. Sistemin geliştirilmesine yönelik yapılacak çalışmalarda sızıntı anında internet bağlantısını kesme, ekranı kapatma veya klavyeyi kilitleme benzeri önlemler alınabilir. Ayrıca sözlükte yazılan yasaklı kelimelerin dil bilgisi kurallarına göre ek almış halleri sisteme otomatik dahil edilerek yasaklı kelime havuzu genişletilebilir. Büyük dil modelinde ise çalışmanın kullanılacağı bölgeye ya da kuruma göre istenen kişisel veri türleriyle modelin eğitilmesi başarıyı arttıracaktır.

KAYNAKLAR

[1]. Salih, B.M., Jasim Mohammad , O.K., 2024, Cloud data leakage, security, privacy issues and challenges: review, *Procedia comput sci* 242 592–601. https://doi.org/10.1016/J.PROCS.2024.08.113.

[2]. Anon., *What are large language models (LLMs)*,.2024, https://www.ibm.com/topics/large-language-models [Ziyaret tarihi: 10 Ağustos 2024].

[3]. Casheekar, A., Lahiri, A., Rath, K., Prabhakar, K.S., Srinivasan, K., 2024, A contemporary review on chatbots, AI-powered virtual conversational agents, ChatGPT: Applications, Open challenges and future research directions *comput sci rev* 52 100632. https://doi.org/10.1016/J.COSREV.2024.100632.

[4]. Anon., *Desktop operating system market share 2013-2024*,. 2024, https://www.statista.com/statistics/218089/global-market-share-of-windows-7/ , [Ziyaret tarihi: 10 Ağustos 2024].

[5]. Dadkhah ,M., Jazi, M., Ciobotaru, A.M., Barati, E.,2014, An introduction to undetectable keyloggers with experimental testing,. *International journal of computer networks and communications security* 4 3

[6]. Shaj, V., Kaliyamurthie, K.P., 2013, A review on data leakage detection, *International journal of computer science and mobile computing*,2 4

[7]. Alneyadi, S., Sithirasenan, E., Muthukkumarasamy, V., 2016, A survey on data leakage prevention systems, *Journal of network and computer applications* 62 137–152 https://doi.org/10.1016/J.JNCA.2016.01.008.

[8]. Ş. Işıklı, 2014, Büyük veri, epistemoloji ve etik tartışmalar, *Online academic journal of information technology* 5 17. https://doi.org/10.5824/1309--1581.2014.4.006.x.

[9]. Anon., *Kişisel veri nedir? Ne demektir? Neleri içerir?*,. 2024, https://www.kisiselverilerinkorunmasi.org/kisisel-veri-nedir-ne-demektir/, [Ziyaret tarihi: 13 Eylül 2024].

[10]. Anon., *Mevzuat bilgi sistemi*,. 2024, https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5, [Ziyaret tarihi: 13 Eylül 2024].

[11]. Anon. *General data protection regulation (GDPR) – Legal text*,. ,2024, https://gdpr-info.eu/ [Ziyaret tarihi: 13 Eylül 2024].

[12]. Ogundare, E., 2024, The human factor in cyber security,.*Researchgate*

[13]. Zaini, N., Zolkipli , M., 2024, A survey on balancing data loss prevention (dlp) with user privacy in a data-driven world, *Journal of innovation and technology 2024*. https://doi.org/10.61453/joit.v2024no10.

[14]. Yadav, I., Gupta, H., 2023, Designing data loss prevention system for the enhancement of data ıntegrity in cyberspace, Proceedings - *IEEE 2023 5th international conference*

on advances in computing, communication control and networking, 2023 1361–1365. https://doi.org/10.1109/ICAC3N60023.2023.10541823.

[15]. Anon., *What is artificial intelligence (AI)?, 2024*, https://www.ibm.com/topics/artificial-intelligence [Ziyaret tarihi: 13 Eylül 2024].

[16]. França, R.P., Borges Monteiro, A.C., Arthur, R., Iano, Y., 2020, An overview of deep learning in big data, image, and signal processing in the modern digital age, *Trends in deep learning methodologies: algorithms, applications, and systems* 63–87. https://doi.org/10.1016/B978-0-12-822226-3.00003-9.

[17]. Efe, M., Kayı Cangır, A., 2022, Artificial intelligence, machine learning and medical applications, *Journal of ankara university faculty of medicine* 75 1–6. https://doi.org/10.4274/ATFM.GALENOS.2022.78557.

[18]. Hao, Z., 2019, Deep learning review and discussion of its future development, *Maricopa advanced technology education center web of conferences* 277 02035. https://doi.org/10.1051/matecconf/201927702035.

[19]. Scarcello, F., 2018, Artificial intelligence, *Ancyclopedia of bioinformatics and computational biology: abc of bioinformatics* 1–3 287–293. https://doi.org/10.1016/B978-0-12-809633-8.20326-9.

[20]. Lin, T., Wang, Y., Liu, X., Qiu ,X., 2022, A survey of transformers, *Artificial intelligence open* 3 111–132. https://doi.org/10.1016/J.AIOPEN.2022.10.001.

[21]. Qiu, Y., Jin, Y., 2024, ChatGPT and finetuned BERT: A comparative study for developing intelligent design support systems, *Intelligent systems with applications* 21 200308. https://doi.org/10.1016/J.ISWA.2023.200308.

[22]. Kiperberg, M., Amit, G., Yeshooroon, A., Zaidenberg, N.J., 2021, Efficient dlp-visor: an efficient hypervisor-based dlp, *Proceedings - 21st ieee/ association for computing machinery international symposium on cluster, cloud and internet computing*, 2021 344–355. https://doi.org/10.1109/CCGRID51090.2021.00044.

[23]. Chang, S.H., Mallissery, S., Hsieh, C.H., Wu, Y.S., 2018, Hypervisor-based sensitive data leakage detector, *Proceedings - 2018 ieee 18th international conference on software quality, reliability, and security*, 2018 155–162. https://doi.org/10.1109/QRS.2018.00029.

[24]. Cherckesova, L., Safaryan, O., Reshetnikova, I., Nikishina, T.,Korochentsev, D.,2021, Corporate chat under DLP-system controlling, *Environment, energy and earth sciences web of conferences, edp sciences*, https://doi.org/10.1051/e3sconf/202127308048.

[25]. Abiodun, M.K., Adeniyi, A.E., Victor, A.O., Awotunde, J.B., Atanda, O.G., Adeniyi, J.K., 2023, Detection and prevention of data leakage in transit using lstm recurrent neural network with encryption algorithm, *2023 International conference on science, engineering and business for sustainable development goals*, . https://doi.org/10.1109/SEB-SDG57117.2023.10124503.

[26]. Tahboub, R., Saleh, Y.,2015, Precaution model for data leakage prevention/loss (dlp) systems, *The 4th palestinian ınternational conference on computer and ınformation technology*

[27]. Buda, A., Colesa ,A., 2018, File system minifilter based data leakage prevention system, *Proceedings - 17th ieee international conference: networking in education and research,* 2018. https://doi.org/10.1109/ROEDUNET.2018.8514147.

[28]. Thombre, S., 2020, Freeware solution for preventing data leakage by insider for windows framework, *2020 International conference on computational performance evaluation*, 2020 44–47. https://doi.org/10.1109/COMPE49325.2020.9200160.

[29]. Ghouse, M., Nene, M.J., Vembuselvi, C., 2019, Data leakage prevention for data in transit using artificial ıntelligence and encryption techniques, *2019 6th International conference on advances in computing, communication and control*, 2019. https://doi.org/10.1109/ICAC347590.2019.9036839.

[30]. Peneti, S., 2016, Data leakage prevention system with time stamp, *2016 International conference on ınformation communication and embedded systems*. https://doi.org/10.1109/ICICES.2016.7518934.

[31]. Anon., *ChatGPT-4 roles*, 2024, https://www.w3schools.com/gen_ai/chatgpt-4/chatgpt-4_roles.php [Ziyaret tarihi: 26Ekim 2024].

[32]. Anon., *JSON*,. https://www.json.org/json-en.html, 2024, [Ziyaret tarihi: 26 Ekim 2024].

[33]. Panfilo, G., 2016, The coordinated universal time, *Instrum meas mag* 19 28–33. https://doi.org/10.1109/MIM.2016.7477951.

[34]. Anon., *API nedir? - Uygulama programlama arabirimlerine ayrıntılı bakış - AWS*,. 2024, https://aws.amazon.com/tr/what-is/api/ [Ziyaret tarihi: 25 Ağustos 2024].

[35]. Anon. *API reference - OpenAI API*,. 2024, https://platform.openai.com/docs/api-reference/messages [Ziyaret tarihi: 20 Ekim 2024].

[36]. Sundaram, O.V.S., Murugesan , N., 2015, A general approach to dfa construction, *International journal of research in computer science*. https://www.researchgate.net/publication/280561964.

[37]. Samala, A.D., Rawas ,S., 2024, Generative ai as virtual healthcare assistant for enhancing patient care quality, *International journal of online and biomedical engineering* 20 174–187. https://doi.org/10.3991/ijoe.v20i05.45937.

[38]. Anon., *Process explorer -s*,. 2024, https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer [Ziyaret tarihi: 23 Ekim 2024].

[39]. Moudni, M., El Ziyati ,E., 2023, *Data leakage prevention approach based on ınsider trust calculation, 2023 10th International conference on wireless networks and mobile communications*, pp. 1–6. https://doi.org/10.1109/WINCOM59760.2023.10322935.

[40]. Cherckesova, L., Safaryan, O., Reshetnikova, I.,Nikishina ,T., Korochentsev, D., 2021, Corporate chat under DLP-system controlling, *Environment, energy and earth sciences web of conferences, edp sciences*, 2021. https://doi.org/10.1051/e3sconf/202127308048.

İNTİHAL RAPORU İLK SAYFASI

Veyis ŞEN

ORİJİNALLİK RAPORU

%8	%6	%5	%4
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	%3
2	acikbilim.yok.gov.tr İnternet Kaynağı	%1
3	www.researchgate.net İnternet Kaynağı	<%1
4	cames.ippt.pan.pl İnternet Kaynağı	<%1
5	hdl.handle.net İnternet Kaynağı	<%1
6	Submitted to Anadolu University Öğrenci Ödevi	<%1
7	Al-Sanabani, Hussein. "Eklentiler Kullanarak Veri Kaybını Engelleme", Sakarya Universitesi (Turkey), 2022 Yayın	<%1
8	dspace.ankara.edu.tr İnternet Kaynağı	<%1

ETİK KURUL İZİN YAZISI

Uyarı: Canlı denekler üzerinde yapılan tüm araştırmalar için Etik Kurul Belgesi alınması zorunludur.

☐ **Etik Kurul izni gerekmektedir.**

☒ **Etik Kurul izni gerekmemektedir.**

Veyis ŞEN

KURUM İZNİ YAZILARI

Uyarı: Canlı ve cansız deneklerle yapılan tüm çalışmalar için kurum izin belgelerinin eklenmesi zorunludur. Gizlilik ve mahremiyet içeren durumlarda kurum adı kapatılmalıdır.

☐ **Kurum izni gerekmektedir.**

☒ **Kurum izni gerekmemektedir.**

Veyis ŞEN