KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TELİF HAKLARI KORUMASI İÇİN İLİŞKİSEL VERİ TABANLARINDA VERİ TÜRÜNDEN BAĞIMSIZ SAYISAL DAMGALAMA

DOKTORA TEZİ

Bilg. Yük. Müh. Mustafa Bilgehan İMAMOĞLU

HAZİRAN 2018
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TELİF HAKLARI KORUMASI İÇİN İLİŞKİSEL VERİ TABANLARINDA VERİ TÜRÜNDEN BAĞIMSIZ SAYISAL DAMGALAMA

Bilgisayar Yük. Müh. Mustafa Bilgehan İMAMOĞLU

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"DOKTOR (BİLGİSAYAR MÜHENDİSLİĞİ)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 29.05.2018
Tezin Savunma Tarihi : 21.06.2018

Tez Danışmanı : Prof. Dr. Mustafa ULUTAŞ

Trabzon 2018

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Ana Bilim Dalında
Mustafa Bilgehan İMAMOĞLU Tarafından Hazırlanan

TELİF HAKLARI KORUMASI İÇİN İLİŞKİSEL VERİ TABANLARINDA VERİ TÜRÜNDEN BAĞIMSIZ SAYISAL DAMGALAMA

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 05 /06 /2018 gün ve 1756 sayılı kararıyla oluşturulan jüri tarafından yapılan sınavda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof. Dr. Rıfat YAZICI .....

Üye : Prof. Dr. Fatih Vehbi ÇELEBİ .....

Üye : Prof. Dr. Mustafa ULUTAŞ .....

Üye : Prof. Dr. Ali GANGAL .....

Üye : Dr. Öğr. Üyesi Hüseyin PEHLİVAN .....

Prof. Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

İlişkisel veri tabanlarında sayısal hakların korunması amacıyla damgalama tekniklerinin tasarımı ve geliştirilmesi etkin bir araştırma alanıdır. Mevcut damgalama tekniklerinin halen iyileştirilmesi gereken problemleri bulunmaktadır. Bu tez çalışmasında var olan damgalama teknikleri iyileştirilmeye çalışılmış ve farklı veri türlerine damga saklayabilen yeni teknikler önerilmiştir.

Çalışmalarımda danışmanlığımı üstlenip ilgisini, desteğini ve tecrübelerini esirgemeyen değerli hocam, danışmanım Sayın Prof. Dr. Mustafa ULUTAŞ'a sonsuz teşekkürlerimi bir borç bilirim. Doktora süresince fikirlerine başvurduğum jüri üyeleri Prof. Dr. Ali GANGAL ve Dr. Öğretim Üyesi Hüseyin PEHLİVAN'a ayrıca teşekkür ederim.

Tez konumun seçiminde bana fikir veren, görüş ve önerileri ile çalışmalarımda yol gösteren değerli arkadaşım Dr. Öğretim Üyesi Güzin ULUTAŞ'a teşekkürü bir borç bilirim.

Fikirlerine başvurduğum değerli hocam Prof. Dr. Bedri SERDAR ve Murat ÖZTÜRK'e teşekkür ederim.

Bu günlere gelmemi sağlayan sevgili aileme, tez çalışmam süresince gösterdikleri sabırdan dolayı sevgili eşime ve kızlarıma teşekkür ederim.

Mustafa Bilgehan İMAMOĞLU

Trabzon 2018

TEZ ETİK BEYANNAMESİ

Doktora Tezi olarak sunduğum "Telif Hakları Koruması İçin İlişkisel Veri Tabanlarında Veri Türünden Bağımsız Sayısal Damgalama" başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. Mustafa ULUTAŞ'ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 28/05/2018

Mustafa Bilgehan İMAMOĞLU

İÇİNDEKİLER

Sayfa No

Doktora Tezi

ÖZET

TELİF HAKLARI KORUMASI İÇİN İLİŞKİSEL VERİ TABANLARINDA VERİ TÜRÜNDEN BAĞIMSIZ SAYISAL DAMGALAMA

Mustafa Bilgehan İMAMOĞLU

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Mustafa ULUTAŞ
2018, 114 Sayfa

Son yıllarda bilgi teknolojilerindeki hızlı gelişim internet üzerinden veri paylaşımını kolaylaştırmıştır. Merkezi veya dağınık veri tabanlarında veri toplanarak uygulamalar aracılığıyla yetkilendirilmiş kullanıcılara erişim ve kullanım hakkı sunulmaktadır. Dijital verilerin izinsiz kopyalanması, değiştirilmesi ve yetkisiz kullanıcılar tarafından paylaşılması sayısal hakların korunması problemini beraberinde getirmiştir. Kriptografi gizli bilgilerin güvenli bir şekilde paylaşılmasını sağlamasına rağmen, şifrelenmiş bilgiyi orijinal içerik ile ilişkilendiremeyeceği için telif hakkı korumasında kullanılamaz. Son yıllarda sayısal hak yönetiminde sayısal damgalama olarak isimlendirilen yöntemler kullanılmaktadır. Özel olarak oluşturulmuş gizli bir bilginin sayısal ortamın içerisine saklanmasını sağlayan yöntemler, verinin korunmasında ve sahiplik bilgisinin belirlenmesinde ek güvenlik katmanı sağlamaktadır. Çoklu ortam dosyalarındaki hak yönetimi için önerilen sayısal damgalama algoritmaları, veri tabanları üzerinde doğrudan uygulanamamaktadır. Araştırmacılar dijital hakların korunması ile ilgili sorunları önlemek için veri tabanlarının damgalanmasını sağlayan özel yöntemler önermişlerdir. Literatürdeki çalışmaların çoğu veri tabanının barındırdığı belirli bir veri türüne uygulanabilmektedir. Bu tez kapsamında yapılan çalışmada farklı veri türleri için saldırılara karşı dayanıklı, damgalama kapasitesi yüksek veri tabanı damgalama yöntemleri önerilmiştir. Önerilen yöntemler, sonuçlardan da gözlemlenebileceği gibi yüksek damgalama kapasitesinde farklı ataklara karşı literatürdeki benzer yöntemlerden daha üstün sonuçlar vermektedir.

Anahtar Kelimeler: Sayısal damgalama, veri tabanı damgalama, veri tabanı, ateş böceği algoritması, homoglif dönüşüm

PhD. Thesis

SUMMARY

DATA TYPE INDEPENDENT RELATIONAL DATABASE WATERMARKING FOR COPYRIGHTS PROTECTION

Mustafa Bilgehan İMAMOĞLU

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Prof. Dr. Mustafa ULUTAŞ
2018, 114 Pages

Rapid developments in Information Technologies have eased data sharing through the Internet recently. Right to access and use data stored in centralized or distributed databases are served to authenticated users through applications. Copying or modification of digital data without permission and sharing by unauthorized users have brought copyright protection problem. Even though cryptography can be used to share sensitive information securely, it can not be used to protect copyright since encrypted information can not be associated with the original. Digital watermarking methods are used lately in digital rights management. Methods to hide specially crafted secret information or watermark into digital media provide additional security mechanism in protecting the data and determining the ownership information. The digital watermarking algorithms proposed for rights management in multimedia files can not be applied directly on databases. Researchers have suggested special methods for database watermarking to prevent problems with the protection of digital rights. Most of the studies in the literature can be applied to a specific data type that the database contains. In the works has been done in thesis, it is proposed watermarking methods which are resistant to attack and have high watermark capacity for different data types. Proposed methods have high watermark capacity and superior results than similar methods in the literature against different attacks, as can be observed from the results.

Key Words: Digital watermarking, database watermarking, firefly algorithm, homoglyph transformation

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

SEMBOLLER DİZİNİ

FFADEW	: Ateş Böceği Algoritması Destekli Fark Genişleme Tabanlı Damgalama Yöntemi
GADEW	: Genetik Algoritma Destekli Fark Genişleme Tabanlı Damgalama Yöntemi
LSB	: En Anlamsız Bit
MD5	: Mesaj Özeti Algoritması 5
NP-Hard	: Deterministik Olmayan Polinom Zorluğu
SHA-256	: 256 bitlik Güvenli Hash Algoritması
SHA-512	: 512 bitlik Güvenli Hash Algoritması
ULC	: Üst Alt Kodlama
XML	: Genişletilebilir İşaretleme Dili
XSLT	: Genişletilebilir Stil Sayfası Dili
Web	: Dünya Çapında Ağ

1. GENEL BİLGİLER

1.1. Giriş

Bilgi Teknolojileri alanındaki hızlı gelişim ile beraber farklı formatlardaki dijital verinin web ortamında paylaşımı artmıştır. Dijital veriler yapılarından dolayı kolaylıkla kopyalanabilir, değiştirilebilir ve farklı amaçlar için yeniden dağıtılabilirler. Bu sebeple paylaşılan verinin gizliliği sağlanmalıdır. Bu gizlilik sağlanırken aynı zamanda verinin bütünlüğü korunmalı ve kullanılabilirliği devam etmelidir. Dijital Hak Koruma mekanizmaları dijital verinin sahipliğini kanıtlamak ve yasal olmayan hareketini önlemede büyük rol oynamaktadır.

Kriptografi, şifreleme ve şifre çözme adı verilen iki fazı bulunan bilgi saklama tekniğidir [1]. Şifreleme fazında, mesaj iletişim kanalına verilmeden önce kodlanmaktadır. Kodlanmış mesaj iletişim kanalına erişen kişiler tarafından ele geçirilse bile kullanılamaz halde olacaktır. Şifre çözme fazında ise mesajın alındığı noktada kodlanmış mesaj çözülerek orijinal mesaj içeriği elde edilmiş olur. Mesajın kodlanması ve kod çözümü aşamasında mesaj gönderici ve alıcı taraf tarafından belirlenmiş ortak gizli anahtar kullanılmaktadır. Bu sayede kodlanmış mesaj iletişim kanalına yapılan saldırıda ele geçirilse bile, gizli anahtar bilinmeden kodun çözülmesi söz konusu değildir. Tüm bu özelliklerine rağmen kriptografi şifrelenmiş bilgiyi orijinal içerik ile ilişkilendiremeyeceği için telif hakkı korumasında kullanılamaz. Kriptografinin bu eksikliği sayısal damgalama yaklaşımı ile çözülmüştür.

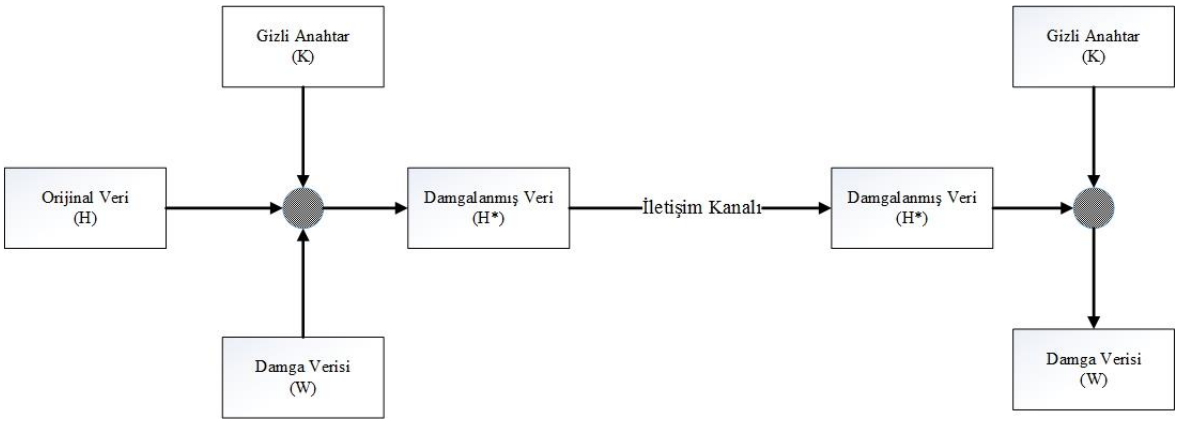
1.2. Sayısal Damgalama

Damga terimi geleneksel kâğıt imalatının geçmişinden günümüze ulaşmıştır. Islak elyafın suyunun atılması için preslendiği aşamada, kâğıdın damgalanmış ve damgalanmamış alanları arasındaki kontrast farkı gözle görülür olarak damganın oluşmasını sağlamaktaydı. Damgalama 13. yüzyılda ilk olarak kâğıt markasını ve üretildiği değirmeni belirlemek için kullanılıyordu. Sonrasında kâğıt üretiminin onaylanması için kullanılsa da günümüzde halen birçok ülke tarafından pulların, kâğıtların ve paraların üzerlerinde damga yerleştirme yaklaşımı devam etmektedir. Gelişen teknoloji ile birlikte bilginin dijitalleşmesi ve internet

yardımıyla hızı bir şekilde yer değiştirmesi sonucunda bilginin sahipliğinin korunması ve orijinalliğinin kanıtlanması önem kazanmaya başlamıştır.

Sayısal Damgalama, gizli bir telif hakkı bilgisinin (işaretlerin) yetkisiz kişiler tarafından değiştirilmesi veya yok edilmesi durumunda verinin kullanılabilirliğini etkileyecek şekilde metin, resim [2], veri tabanı, video [3], ses [4, 5] veya yazılım [6-9] gibi içeriklere yerleştirilmesi sürecidir [10-13]. Yerleştirilen damga verisi, veri sahibini veya veriye erişim yetkisi olan kullanıcıları belirleyen bilginin ikilik tabandaki karşılığıdır. Klasik damgalama süreçlerinde olduğu gibi basılı veya gözle görülebilir bir bilgi değildir. Saldırganların damga verisini kolaylıkla belirleyememesi için yerleştirilen damga bilgisinin veriye homojen şekilde dağıtılması gerekir.

Şekil 1.1.'de klasik sayısal damgalama sisteminin akış diyagramı görülmektedir. Damgalama süreci damga yerleştirme ve damga çıkarım adı verilen iki fazdan oluşmaktadır. Damga yerleştirme fazında, verinin sahibi tarafından belirlenen ve sadece kendisinin bildiği gizli anahtar κ kullanılarak damga verisi W orijinal verinin H içerisine yerleştirilir. Damgalanmış veri tabanı H^* herkese açık ağlarda veya internette paylaşıldıktan sonra, verinin sahipliği konusunda ihtilaf olması durumunda veri tabanı sahibi tarafından bilinen gizli anahtar κ kullanılarak damga çıkarım fazı gerçekleştirilir.



Şekil 1.1. Klasik sayısal damgalama sistem akış şeması

Yerleştirilen damga verisinin orijinal veri üzerinde kullanımı etkileyecek düzeyde bozulmalara sebep olmaması çok önemlidir. Aksi takdirde güvenliği sağlanmış fakat kullanılamaz bir veri elde edilmiş olacaktır. Güvenliğin yanında damgalama kapasitesinin yüksek olması da sistemin kullanılabilirlik alanlarını artırmaktadır. İdeal damgalama

sistemlerinin büyük miktarda damga verisini, orijinal verinin kullanılabilirliğini bozmadan yerleştiriyor olması beklenir.

1.3. Veri Tabanı Damgalama

Veri tabanı uygulamalarının kullanımı arttıkça verinin sahiplik ve bütünlük kontrolü problemleri yaşanmaya başlanmıştır. Veri tabanı sistemlerinin doğasından dolayı sıklıkla yeni kayıt ekleme, kayıt güncelleme ve kayıt silme gibi işlemlere tabi tutulurlar. Bu özelliklerinden dolayı resim, video ve ses üzerinde daha önce yapılmış damgalama çalışmaları, veri tabanları üzerinde gerçekleştirilecek damgalama sürecine uygun değillerdir [1-5]. Multimedya ve veri tabanı damgalama arasındaki farklar Tablo 1.1'de verilmiştir.

Tablo 1.1.Multimedya ve veri tabanı damgalama arasındaki farklar

Özellik	*Multimedya Damgalama*	*Veri Tabanı Damgalama*
Gereksiz Veri	Fazla	Az
Damga Konumu	Değişmez	Değişebilir
Güncelleme	Olmaz	Sıklıkla ekleme, silme, güncelleme olur
Boyut	Sabit	Değişken
Veri Türü	Sadece Sayısal	Sayısal, Metin, Tarih, Kategorik, vs.

Damganın yerleştirilebileceği fazlalık (gereksiz) veri miktarı multimedya dosyalarında gereğinden fazla iken veri tabanında bu özellikteki veriler daha azdır. Multimedya dosyalarda damga verisinin konumu değişmezsen, veri tabanlarına yerleştirilen damga verisinin konumu veri tabanı üzerinde gerçekleştirilecek satır ekleme ve satır silme işlemlerinden dolayı değişebilmektedir. Multimedya dosyalar üzerinde damgalamadan sonra değişim olmazken, veri tabanı sıklıkla yeni satır ekleme, değer güncelleme veya satır silme gibi işlemler sebebiyle güncellenmektedir. Multimedya dosyasının sınırları belirli iken, veri tabanının boyutu farklılık göstermektedir. Multimedya dosyasında damga verisinin yerleştirileceği veri nümerik ve sınırları belirli iken, veri tabanı sistemlerinde nümerik, kategorik ve nümerik olmayan veri türleri bulunabilmektedir.

Veri tabanları üzerinde damga yerleştirme konusundaki literatürdeki ilk çalışma 2002 yılında Rakesh Agrawal vd. tarafından yapılmıştır [14]. Yöntem içerisinde kullanılmış olan

(1.1)'de ifadesi verilmiş fonksiyonda, özüt fonksiyonu $\mathcal{H}$ ile, gizli anahtar değeri κ ile ve birleştirme işlemi ise "||" sembolü ile gösterilmiştir. Bu fonksiyonun üretecek olduğu değer r ilişkisinde yer alan kaydın o anki birincil anahtar değeri P ve özel anahtar değeri κ ile doğrudan ilişkilidir.

$$\mathrm{F(r.P)} = \mathcal{H}\left(\kappa \parallel \mathcal{H}\left(\kappa \parallel (\mathrm{r.P})\right)\right) \tag{1}$$

Yöntem her bir veri tabanı kaydı için (1) ifadesi ile hesaplanan değerin γ değeri ile modüle sonucu 0 olan satırları damgalanmak üzere seçmektedir. γ damgalanacak kayıt oranını belirleyen değerdir. Bir sonraki aşamada, seçilen kaydın damgalanacak alanın belirlenmesi gereklidir. Bu işlem (1) ifadesinden hesaplanan değerin ν değeri ile modüle sonucu ile gerçekleştirilir. ν değeri kullanıcı tarafından belirlenen ve damgalama işleminde kullanılabilecek alan sayısıdır. Damganın yerleştirileceği alan belirlendikten sonra, damga bitinin saklanacağı LSB'nin konumu hesaplanır. Bu işlemde yine kullanıcı tarafından değeri belirlenen ξ değeri (1) ifadesinden hesaplanan değer ile modüle edilmektedir. Damga olarak yerleştirilecek veri daha öncesinde belirlenmek yerine, birincil anahtar değeri ile kullanıcı tarafından belirlenen özel anahtar değerinin özüt fonksiyonu sonucuna göre belirlenmiştir. Özüt fonksiyonun elde ettiği değer çift ise belirlenen konumdaki bit değeri 0 ile, aksi durumda 1 ile değiştirilmektedir. Damgalama esnasında kullanılan yöntem yalancı kod ifadesi Şekil 1.2'de verilmiştir.

Damga çıkarım sürecinde, damga yerleştirilirken kullanılan özel anahtar değerinin ve LSB oran değerinin bilindiği varsayılmaktadır. Aksi durumda çıkarılan damga değeri ile yerleştirilen damga değeri eşleşmeyecektir. Damga yerleştirme aşamasında olduğu gibi özüt fonksiyonu yardımıyla seçilecek kaydın belirlenen özellik değerinden damga biti çıkarılacaktır. Çıkarılan bit değeri, yeniden hesaplanan $\mathcal{H}\left(\kappa \parallel (r.P)\right) mod\ 2$ değeri ile kıyaslanacak ve doğrulama gerçekleştirilecektir. Doğru olarak elde edilen damga bitleri sayısı belirli bir α eşik değerini geçerse, kontrolü gerçekleştirilen veri tabanının damgası doğrulanacaktır. Damga çıkarma algoritmasına ilişkin yalancı kod ifadesi Şekil 1.3.'de verilmiştir.

Giriş: R ilişkisi, gizli anahtar κ, oran $1/\gamma$, kullanılacak LSB oranı ξ, damgalanabilir özellikler $\{A_1, A_2, \dots, A_v\}$

Çıkış: Damgalanmış ilişki R_w

forall kayit $r \in R$ **do**

  if $F(r.P)\ mod\ \gamma = 0$ then

    $i = F(r.P)\ mod\ v$;

    $j = F(r.P)\ mod\ \xi$;

    $r.A_i^j = \mathcal{H}\big(\kappa \parallel (r.P)\big)\ mod\ 2$;

  end

end

Şekil 1.2. Damga yerleştirme algoritması

Giriş: R ilişkisi, gizli anahtar κ, oran $1/\gamma$, kullanılacak LSB oranı ξ, eşik değeri α

Çıkış: Damga durumu $\in \{true, false\}$, $essayi = 0, toplamsayi = 0$

forall kayit $r \in R$ **do**

  if $F(r.P)\ mod\ \gamma = 0$ then

    $i = F(r.P)\ mod\ v$;

    $j = F(r.P)\ mod\ \xi$;

    if $\mathcal{H}\big(\kappa \parallel (r.P)\big)\ mod\ 2 == r.A_i^j$ **then**

      $essayi = essayi + 1$;

    end

    $toplamsayi = toplamsayi + 1$;

  end

end

$\varphi = min(\theta): B(\theta, toplamsayi, 1/2) < \alpha$

if $essayi \geq \varphi$ **then**

  $return\ true$;

else

  $return\ false$;

end

Şekil 1.3. Damga çıkarma algoritması

1.4. Veri Tabanı Damgalama Yöntemlerinin Ortak Özellikleri

İlişkisel veri tabanlarının sahip olduğu farklılıklardan damgala gerçekleştirecek sistemlerin sahip olması gereken özellikler bulunmaktadır. İlk olarak veri tabanı içerisine yerleştirilen damga verisinin veri tabanının kullanılabilirliğini ortadan kaldırmaması gerekir. Aksi takdirde güvenliği sağlanmış fakat kullanılamayan bir veri tabanı elde edilmiş olacaktır. Ayrıca orijinal veri tabanında damgalama işleminden sonra anlamsız veri oluşmamalıdır. Anlamsız veriler saldırganlar tarafından damga konumunun belirlenebilmesine yardımcı olacaktır.

İlişkisel veri tabanlarında tablolar arasındaki ilişki sütun değerleri üzerinden belirlenmektedir. Bu tür alanlar kategorik veri taşımaktadırlar. Örneğin, personel bilgilerinin tutulduğu bir veri tabanı sisteminde cinsiyet türlerinin tanımlandığı REF_CINSIYET isimli tablonun "Bay" ve "Bayan" verilerini tutan AD metin alanı ile tablodaki kayıtların tekilliğini sağlayan ID nümerik alanı olduğu varsayılsın. Personel bilgilerinin tutulduğu PERSONEL tablosunda ilgili personel kaydının cinsiyet bilgisini tutacak olan alan değeri REF_CINSIYET tablosundaki kayıtların ID alan değeri ile eşleşmesi gereklidir. Damga yerleştirme sürecinde bu ilişkinin yok edilmemesi, devamlılığının sağlanması gereklidir.

Damgalama sisteminin güvenliği kullanıcı tarafından belirlenen ve kendisinde saklı olan gizli anahtar vasıtasıyla sağlanmalıdır. Damgalama sisteminde damga yerleştirme damga çıkarım metotları herkese açık olmasına rağmen, gizli anahtar olmadan damga çıkarılamamalıdır. Bu sayede damgalama ve damga çıkarım algoritmaları herkese açık olmasına rağmen, gizli anahtara sahip olmayan hiç kimse damga verisini elde edemeyecektir.

Saldırganlar damga yerleştirilmiş veri tabanları üzerinde saldırılar gerçekleştirerek damganın tamamını veya bir kısmını ortadan kaldırmaya çalışacaklardır. Damga yerleştirme sisteminin saldırıya uğramış veri tabanından sahiplik kontrolü yapılabilecek düzeyde damga verisini belirleyebilmesi beklenir. Damgalama sisteminin damga verisini elde edebilmesi için orijinal veriye veya yerleştirilen damga verisine ihtiyacı olmaması tercih edilen bir özelliktir. Aksi durumda orijinal verinin veya yerleştirilen damga verisinin harici bir ortamda saklanması ve değiştirilmediğinden emin olunması gerekecektir.

Damgalanmış veri tabanına yeni satırlar eklendiğinde, damgalama sisteminin sadece yeni eklenen satırlar üzerinde damga yerleştirme sürecini koşabilmesi gerekir. Bu sayede yeni eklenen kayıtların damgalanma işlemi gerçek zamanlı olarak uygulanabilir olacaktır.

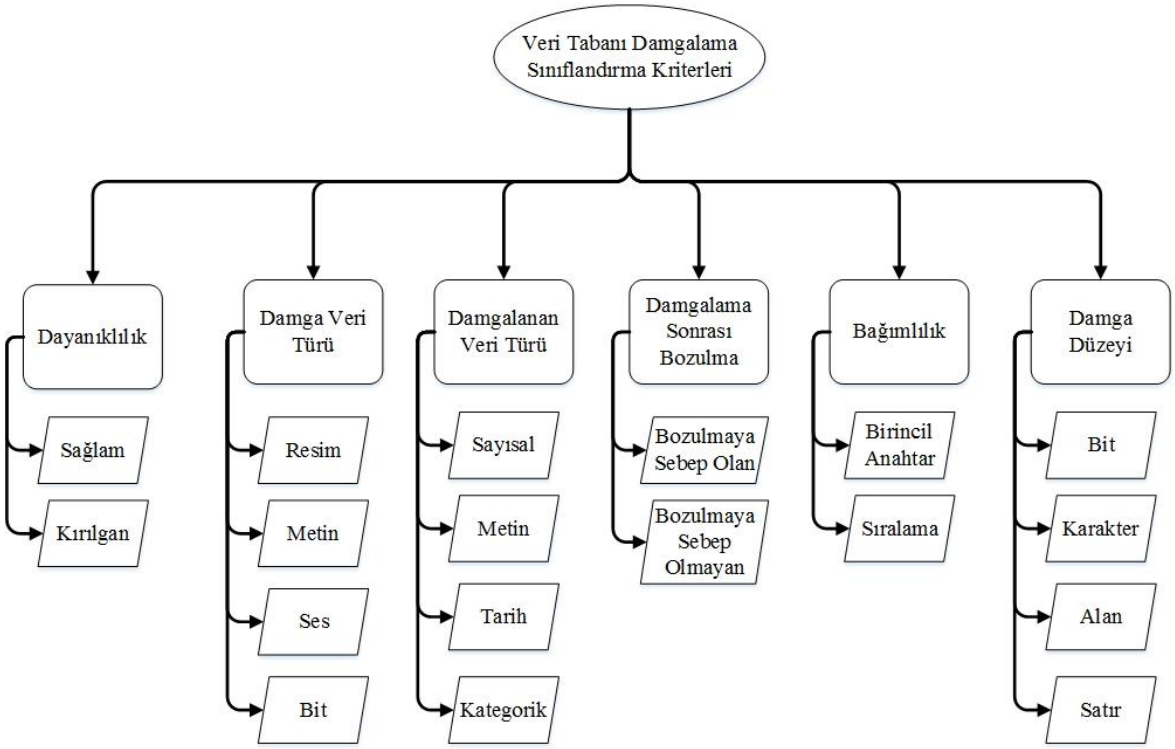
Veri tabanına yerleştirilen damga verisinin boyutunun büyük olması, damgalama sisteminin kullanım alanının geniş olmasını sağlayacaktır. Bu sebeple aynı veri tabanına veri kullanılabilirliğini bozmadan daha yüksek miktarda veri saklayabilen sistemler tercih edilmektedir.

Veri tabanını izinsiz kullanmak veya bütünlüğünü bozmak isteyen saldırganlar tarafından damgalama sistemini etkisiz bırakacak saldırılar yapılmaktadır. Damgalama sisteminin bu saldırılara dayanıklı olması gereklidir. Saldırganların verinin kullanılabilirliğini ortadan kaldırmadan damga verisini yok edememeleri gerekir. Damga verisinin saldırılara karşı dayanıklılığı damga yerleştirme sisteminin sağlamlığını ortaya koymaktadır.

Damgalama sisteminin damga yerleştirmediği kayıttan damga çıkarımı yapmaması gerekir. Böyle bir durumda çıkarılan damga bilgisi hatalı olacak ve güvenlik zafiyetine sebep olacaktır. Damgalanmış veri tabanlarına gerçekleştirilen saldırıların bir kısmı damgalama sistemini şaşırtarak yerleştirmediği damgayı çıkarmasına sebep olmaya çalışmaktadırlar. Damgalama sisteminin damga yerleştirdiği veri tabanından damgayı sorunsuz olarak çıkarabilmesi beklenir. Damgalanmış bir veri tabanından damga çıkarım süreci sonucunda damga verisinin çıkarılamaması, veri tabanının damgalanmamış olduğuna karar verilmesine sebep olacaktır.

1.5. Veri Tabanı Damgalama Yöntemlerinin Sınıflandırılması

İlk veri tabanı damgalama çalışmasından bugüne kadar yapılan çalışmaların genel özellikleri dikkate alındığında veri tabanı damgalama yöntemlerinin sınıflandırılmasında sıklıkla kullanılan kriterler Şekil 1.4.'de verilmiştir. Veri tabanı damgalama yöntemleri saldırılara karşı dayanıklılık durumlarına göre Sağlam ve Kırılgan olarak sınıflandırılmaktadırlar. Sağlam yöntemler saldırılara karşı dayanıklıdırlar. Bu yöntemlerden saldırı yapılmış veri tabanından damga verisini başarıyla çıkarabilmeleri beklenir. Kırılgan yöntemlerin özelliği damgalanmış veri tabanında saldırganlar tarafından gerçekleştirilecek en ufak bir değişikliği bile algılayabilmeleri ve yapabiliyorlarsa bozulmanın konumunu belirleyebilmeleridir. Bu sebeple kırılgan yöntemler saldırılara karşı dayanıklı değildirler.



Şekil 1.4.Veri tabanı damgalama yöntemlerinin sınıflandırma kriterleri

Damga olarak yerleştirilen verinin türü de damgalama yöntemlerinin sınıflandırılmasında kullanılan kriterlerden biridir. Damga verisi olarak metin kullanan yöntemler, kullanıcı tarafından belirlenen bir metin parçasını veri tabanı içerisine yerleştirmektedirler. Kullanıcının sesini veya kurumsal bir ses dosyasını damga verisi olarak kullanan çalışmalar olduğu gibi, resim dosyasını (örneğin firma logosu) damga verisi olarak veri tabanına yerleştiren çalışmalar da bulunmaktadır. Bazı çalışmalar ise damga verisini anlamlı veya anlamsız bit dizilerinden elde etmektedirler.

Damga verisinin türünün yanında damganın yerleştirildiği verinin türü de sınıflandırma kriterleri arasındadır. Literatürdeki çalışmaların büyük çoğunluğu sayısal veriler üzerine damga yerleştirme yapmaktadır. Metin, tarih, XML, kategorik veri alanları üzerine yapılan çalışmalar da gün geçtikçe artmaktadır.

Damgalama yöntemlerinin damgalama sonrası verinin kullanılabilirliğini ortadan kaldırmaması gerekir. Bu durum yöntemin orijinal veri üzerinde bozulmaya sebep olmayacağı anlamına gelmemektedir. Literatürde yapılan çalışmaların büyük kısmı damga yerleştirirken orijinal veri üzerinde kullanımını etkilemeyecek oranda bozulmaya sebep olmaktadır. Bazı çalışmalar damga çıkarım sonrası bu bozulmayı ortadan kaldırarak damgalama öncesi orijinal veriyi elde edebilirken, bazıları sadece yerleştirilen damgayı elde

edebilmektedir. Bu durumda damga yerleştirme sonrası oluşan bozulma kalıcı olmaktadır. Bozulmaya sebep olmayan çalışmaların büyük kısmı aynı zamanda saldırıya karşı dayanıksız olan kırılgan yöntemler olmakla birlikte, bozulmaya sebep olmayan sağlam yöntemler de bulunmaktadır.

Damga yerleştirme ve çıkarım aşamalarında işlem görecek kaydın belirlenmesinde çalışmaların büyük çoğunluğu veri setinin birincil anahtarına bağımlıdırlar. Birincil anahtarın olmadığı durumlarda bu çalışmalar bir kısmı kullanıcı veya yöntem tarafından otomatik belirlenen alanlar yardımıyla birincil anahtar oluşturulup damgalama işlemi gerçekleştirilmektedir. Bir kısım çalışmalar ile damga yerleştirme sürecinde kayıtların belirlenen kritere göre sıralı olmasına bağımlıdırlar.

Damga verisinin yerleştirildiği düzey veri tabanı damgalama yöntemlerinin sınıflandırılmasında kullanılan bir başka kriterdir. Çalışmaların bir kısmı damga verisini bit düzeyinde yerleştirirken, diğer bir kısım alanın içerdiği karakter veya karakterleri damgalamada kullanabilmektedir. Alanın değerini damga değeri ile değiştiren çalışmalar olduğu gibi satırın tamamını damgalayan çalışmalar da bulunmaktadır.

1.5.1. Sağlam Damgalama Yöntemleri

Literatürde damgalama sonrası orijinal veri üzerinde bozulmaya (değişime) sebep olan birçok veri tabanı damgalama algoritması bulunmaktadır. Bu algoritmaların ortak özelliği, damgalama gerçekleştirdikleri veri tabanlarının veriler üzerinde oluşacak küçük değişiklikleri tolere edebilir oluşudur. Bu algoritmaların birçoğu telif hakları korunması ve sahiplik kanıtı için kullanılmaktadırlar. Yerleştirilen damga verisi bit veya karakter düzeyinde olabileceği gibi, satır veya sütun bazında da olabilmektedir.

Telif hakkı koruması sağlayan sağlam şemaların birçoğu veri bütünlüğünü ve kullanılabilirliğini etkileyen orijinal içeriğin bozulmasına sebep olmaktadır. Bu şemalar damgalarını yerleştirmek için ilişkisel veri tabanlarının numerik veya kategorik alanlarını kullanmaktadırlar. Kullanılan damga bilgisi anlamsız bir şekilde oluşmuş bit dizisi olabileceği gibi resim, konuşma veya sahip bilgisi gibi anlamlı bit dizisi de olabilir.

Sağlam damgalama şemalarının, damganın tamamen yok edilmesi veya algılanamaz hale getirilmesi amaçlı yapılan saldırılara dayanıklı olması gerekir. Bu sebeple algoritma tasarımının amacı damgalanmış kayıtlara yapılan silme, ekleme, değiştirme gibi saldırılara karşı şemanın dayanıklı olmasıdır. Sağlam damgalama şemalarının temel varsayımı, ilişkisel

veri tabanının orijinal verileri üzerinde gerçekleştirilecek küçük değişikliklerin müsamaha edilebilir olduğudur. Bununla birlikte, tıbbi veriler gibi hassas veri tabanlarındaki değişiklikler, her ne kadar küçük de olsa, veri tabanının kullanılabilirliğini etkilemekte, hatta veri tabanının bütünlüğünün ihlaline sebep olabilmektedir. Bu sebeple, sağlam damgalama şemaları, hassas veri içeren ilişkisel veri tabanları için uygun değildir.

Sağlam damgalama yöntemlerinin damga yerleştirirken kullandıkları alan veri türü, damgalama sonrası bozulmaya sebep olma durumları ve damgalama öncesi orijinal veriye geri dönebilme özelliklerine ait özet bilgi Tablo 1.2.'de verilmiştir.

Tablo 1.2. Sağlam damgalama yöntemleri

Çalışma	*Yıl*	*Veri Türü*	*Bozulma*	*Orijinal Veriye Dönebilme*
Agrawal ve Kiernan [14]	2002	Sayısal	Olur	Dönülemez
Hu vd. [15]	2009	Sayısal	Olur	Dönülemez
Peng vd. [16]	2007	Sayısal	Olur	Dönülemez
Odeh vd. [17]	2008	Metin	Olur	Dönülebilir
Sion vd. [18]	2004	Kategorik	Olur	Dönülemez
Pournaghshband [19]	2008	-	Olmaz	Dönülebilir
Gross-Amblard [20]	2003	Sayısal	Olur	Dönülemez
Kamran vd. [21]	2013	Sayısal	Olur	Dönülemez
Wilfred vd. [22]	2005	Sayısal, Metin	Olur	Dönülemez
Gu vd. [23]	2006	Sayısal, Metin	Olur	Dönülemez
Yao vd. [24]	2006	-	Olmaz	Dönülebilir
Saad [25]	2010	Metin	Olur	Dönülemez
Saad [26]	2012	-	Olmaz	Dönülebilir
Tchokpon vd. [27]	2012	-	Olmaz	Dönülebilir
Wen vd. [28]	2018	-	Olmaz	Dönülebilir
Gupta vd. [29]	2008	Sayısal	Olur	Dönülebilir
Jawad vd. [30]	2013	Sayısal	Olur	Dönülebilir
Bhattacharya vd. [31]	2009	-	Olmaz	Dönülebilir
Hanyurwimfura vd. [32]	2010	Metin	Olur	Dönülebilir
Farfoura vd. [33]	2012	Sayısal	Olur	Dönülebilir

Agrawal ve Kiernan tarafından veri tabanı damgalama alanında yapılan ilk çalışma damga verisini sayısal alan değerinde bit düzeyinde değişiklik yaparak yerleştirmektedir [14]. Damgalanacak olan satırlar, alanlar ve alanın bit seviyesi özüt fonksiyon kullanılarak

sadece kullanıcı tarafından bilinen özel anahtar yardımıyla belirlenmektedir. Damga verisi kullanıcı tarafından belirlenememekte, damgalanacak alan değerine göre farklılık göstermektedir. Bu özelliğiyle veri tabanına özgü damga verisi oluşturan çalışma, damga çıkarım sonrası orijinal verileri elde edememekte, veri tabanında bozulmaya sebep olmaktadır.

Hu vd. tarafından önerilen yöntem damga olarak ikilik düzene çevrilmiş resim bilgisi kullanmaktadır. Resmin özellikle telif hakkını ispatlama aşamasında yararlı olacağı düşünülmektedir. Yöntemin damgalanacak satırları düzgün olarak seçmemesi, güvenli olarak görülmeyen [34] MD5 algoritmasını kullanması, kategorik veriler üzerinde bozulmalara sebep olan değişiklikler yapması eksik yönleri olarak sıralanabilir [15].

Peng vd. tarafından önerilen yöntem, damga olarak seçilen resim verisinin kesirli sayı içeren alanlar üzerine saklanmasını sağlamaktadır. Yöntem damga saklanan alandaki kesirli sayının ondalık değerin kaybolmasına sebep olmaktadır [16].

Odeh vd. tarafından önerilen çalışmada damga verisi olarak kullanılan resim bilgisi birden çok kelime içeren alanlar içerisine saklanmıştır. Metin alanlar üzerinde saklama yapıldığı için damgalama kapasitesi yüksek olurken, resim verisi metin veriler üzerinde bozulmaya sebep olmuştur [17].

Sion vd. tarafından önerilen yöntem satırları sıralayıp gruplara ayırdıktan sonra standart sapma değerine yakın olan satıra damga verisini saklamaktadır. Çalışma tek bir alan bağımlı olmasının yanında ekstra yük bilgisine sahiptir [18].

Saldırılara karşı dayanıklı olmanın bir diğer yöntemi de orijinal veri tabanına ait olmayan sahte yeni kayıtlar oluşturarak damgalama yapma yöntemidir. Bu durum orijinal verilerin değiştirmese de bazı sorgulamaların sonucunun beklenenden farklı olmasına sebep olabilmektedir. Pournaghshband'ın gerçekleştirdiği çalışma bu yöntemi kullanırken, veri tabanının tutarlılığını etkilememek için birincil anahtarın kısıtlamasına uygun olarak eklenen her bir satıra yeni bir anahtar değeri oluşturmaktadır. Damga çıkarım aşamasında ise sadece veri tabanı sahibi tarafından belirlenen kriterlere uyan sahte satırların varlığı aranmaktadır [19].

Damgalama sonrası oluşacak bozulmanın kontrollü bir şekilde, belirlenen sınırlar içerisinde kalmasını sağlayan ve bu yöntemle damga verisini saklayan yöntemlerden biri Gross-Amblard tarafından gerçekleştirilmiştir. Yöntem daha önceden bilinen toplam sorgularının sonucunu değiştirmeyecek şekilde veriler üzerinde değişiklikler yaparak damga yerleştirmektedir. Orijinal veri üzerinde yapılan sorgu ile damgalanmış veri üzerinde yapılan

sorgunun sonuçları aynı ise minimum ve kontrollü bozulma ile damgalamanın gerçekleştirildiği kabul edilmiştir [20].

Kamran vd. tarafından sonuçları önceden bilinen veri madenciliği sürecinin sonuçlarını korumak için önerilen yöntemde alanlar önemlerine göre gruplanmakta ve ardından madencilik ile ilgili veri kümesinin istatistiki özelliklerinden yerel ve global bazı sınırlamalar üretilmektedir [21].

Wilfred vd. XML veri alanlarına veri türünü dikkate alarak damga yerleştirme çalışması yapmışlardır [22]. Damgalama yapılacak veri nümerik olması durumunda Agrawal'ın yöntemi ile LSB üzerine veri saklama gerçekleştirilirken, metin alana veri saklarken WordNet [35] sinonim havuzunu kullanmışlardır. Çalışmanın sakladığı veri miktarının düşüklüğünün yanı sıra, gömülen damganın çıkarılabilmesi için damgalama aşamasında kullanılan parametrelere ihtiyaç duyulması sebebiyle bu parametrelerin harici bir kaynakta saklanması zorunluluğu bulunmaktadır. Metinlere damga yerleştirme aşamasında seçilen sinonim karşılığın farklı uzunlukta olmasından dolayı da veri boyutunun değişimi söz konusudur.

Gu vd. XML verilere parmak izi eklemeyi sağlayan çalışmaları ile metin ve nümerik verilerin farklı türdeki parmak izi verisi ile damgalanmasını sağlamışlardır [23]. Çalışmalarında damga verisi ikilik düzene dönüştürülerek, birincil anahtar ve özüt fonksiyonu yardımı ile damgalanacak satırlar seçilmektedir. Verinin türünün nümerik olması durumunda LSB değişimi, metin olması durumunda ise sinonim kullanarak damgalama gerçekleştirilmiştir. Damganın geri elde edilmesi için damga verisinin uzunluğunun saklanma gereksinimi, damgalama öncesi orijinal verinin elde edilememesi, orijinal veri üzerinde bozulmaya sebep olması algoritmanın zayıf yönleridir.

XML verinin bütünlüğünün sağlanması amacıyla Yao vd. yapmış olduğu çalışmada nümerik alanlar dikkate alınmadan, XML etiketleri ULC (Upper-Lower Coding) yöntemi ile damgalanmıştır [24]. Yöntem veri boyutunu değiştirmeden XML verisinin bütünlüğünü sağlamayı amaçlamaktadır. Etiketler üzerinde yapmış olduğu değişiklikten dolayı damga verisi saldırgan tarafından kolaylıkla belirlenebilmektedir.

Saad tarafından yapılan çalışmada parmak izi verisi ile damgalama gerçekleştirilmiştir [25]. Saad'ın çalışmasında gruplama kullanımı ve parmak izinin oluşturulma aşamasında geliştirmeler bulunmaktadır. Verinin bozulmasını engellemek için nümerik alanlar üzerinde değil, metin alanlar üzerinde damgalama yapılmıştır. Metinlerin büyültülüp/küçültülmesi yöntemi ile 1/0 verisinin yerleştirilmesi sağlanmıştır.

Saad, daha önce yapmış olduğu çalışmayı [25] geliştirerek damgalama aşamasında veri metninin karakterlerinin büyültülmesi yerine XML verisinin kapatma etiketinin sonuna "boşluk" eklemeyi tercih etmiştir [26]. Damgalama sonrası veri boyutunun değişimi ve damganın algılanabilir olma problemi bu çalışmada da söz konusudur.

Tchokpon vd. yapmış oldukları çalışmada rastgele üretilen damga bitlerini veri içerisinde gömmek için "boşluk ekleme" yöntemini kullanmışlardır [27]. Veri üzerinde gerçekleştirilen sorgular dikkate alınarak en önemli etiket belirlenmeye çalışılmaktadır. Belirlenen bu etiketin içerdiği veriye damga bit değeri "0" olduğu durumda boşluk eklenmiş, "1" olduğu durumda veri değiştirilmemiştir.

Wen vd. XML dokümanların damgalanmasını sağlayan iki farklı yaklaşım sunmuşlardır [28]. İlk yaklaşımda XML verinin görünüm şeklini veya formatını belirleyen XSLT verisi kullanılarak damgalama gerçekleştirilmiştir. İkinci yaklaşımda ise XML verinin işlevsel bağımlılığı dikkate alınmıştır. XSLT verisi olmayan XML veri tabanlarında yaklaşımın uygulanabilirliğinin olmaması, damgalama sonrası oluşturulan dosyanın harici bir ortamda saklanma gereksinimi damgalama algoritmasının zayıf yönlerini oluşturmaktadır.

Gupta vd. Fark Genişleme Tabanlı Damgalama yöntemini kullanarak sayısal damgalama gerçekleştirmişlerdir [29]. Ortalama değeri genişleterek damga verisinin saklanmasını sağlayan Fark Genişleme Tabanlı Damgalama yöntemi ilk olarak 2004 yılında Alattar tarafından önerilmiştir [36]. Resimler üzerinde uygulanabilen ve orijinal veriye geri dönüşümün mümkün olduğu bu yöntem komşu iki pikselin arasındaki farkı değiştirme yolu ile damgalama işlemini gerçekleştirmektedir. Damga çıkarım süreci sonrasında pikselleri orijinal değerlerine döndürebilmektedir. Damganın yerleştirileceği satırdaki iki alan değerini, piksel değeri yerine kullanmaktadır. Yöntem içerisinde kullanılacak ifade (2)'de verilmiştir. Damga verisinin saklanacağı alanlar sırasıyla A_1 ve A_2 ile alan değerlerinin aritmetik ortalaması avg ile alan değerleri arasındaki fark ise d ile gösterilmiştir.

$$avg = \left\lfloor \frac{A_1 + A_2}{2} \right\rfloor, d = A_1 - A_2 \quad (2)$$

⌊x⌋ sembolü taban fonksiyonu olarak isimlendirilir ve x değerine eşit veya küçük olan en büyük tamsayıyı belirtmektedir. Fark değerini tutan d ifadesi damga biti b değerini

tutacak şekilde (3)'de değiştirilmektedir. Damgalama sonrasında alanların yeni değerlerini tutan $\widetilde{A_1}$ve $\widetilde{A_2}$ değerleri de (3) ifadesinde hesaplanmaktadır.

$$\tilde{d} = 2d + b, \widetilde{A_1} = avg + \left\lfloor \frac{\tilde{d}+1}{2} \right\rfloor, \widetilde{A_2} = avg - \left\lfloor \frac{\tilde{d}}{2} \right\rfloor \quad (3)$$

Metot aşağıda verilmiş adımlar izlenerek damgalanmış veriden orijinal veriyi ve damga bitini elde edebilmektedir.

Adım 1. Alanların fark ve aritmetik ortalama değerleri hesaplanır, $\acute{\text{a}}\text{vg} = \left\lfloor \frac{\widetilde{A_1}+\widetilde{A_2}}{2} \right\rfloor, \tilde{\text{d}} = \widetilde{A_1} - \widetilde{A_2}$

Adım 2. Damga biti fark değerinden elde edilir, $\text{b} = \tilde{\text{d}} - 2\lfloor \tilde{\text{d}}/2 \rfloor$

Adım 3. Alanların damgalama öncesi sahip oldukları orijinal değerleri hesaplanır, $\text{A}_1 = \acute{\text{a}}\text{vg} + \lfloor (\lfloor \tilde{\text{d}}/2 \rfloor + 1)/2 \rfloor, \text{A}_2 = \acute{\text{a}}\text{vg} - \lfloor (\lfloor \tilde{\text{d}}/2 \rfloor)/2 \rfloor$

Örneğin; damga yerleştirilecek alan değerleri sırasıyla $\text{A}_1 = 16$ ve $\text{A}_2 = 24$ olduğunda, fark ve aritmetik ortalama değerleri $\text{d} = 16 - 24 = -8$ ve $\text{avg} = \lfloor (16 + 24)/2 \rfloor = 20$ olarak hesaplanır. Damga bitinin $\text{b} = 1$ olduğu varsayılarak yeni fark değeri $\tilde{\text{d}} = 2(-8) + 1 = -15$ olarak hesaplanır. Damgalama sonrası yeni alan değerleri $\widetilde{\text{A}_1} = 20 + \lfloor (-15 + 1)/2 \rfloor = 13$ ve $\widetilde{\text{A}_2} = 20 - \lfloor -15/2 \rfloor = 28$ olarak hesaplanır. Damgalanan verinin geri elde edilmesi aşamasında fark ve aritmetik ortalama değerleri sırasıyla -15 ve 20 olarak hesaplanacaktır. Çıkarılan damga verisi $\text{b} = -15 - \big(2(-8)\big) = 1$ olarak hesaplanır. Damga verisi çıkarıldıktan sonra alanların orijinal değerlerin çıkarımı $\text{A}_1 = 20 + \lfloor (\lfloor -15/2 \rfloor + 1)/2 \rfloor = 16, \text{A}_2 = 20 - \lfloor (\lfloor -15/2 \rfloor)/2 \rfloor = 24$ şeklinde olacaktır.

Jawad vd. Genetik Algoritma ve Fark Genişleme Tabanlı Damgalama yöntemlerini beraber kullanarak saldırılara karşı dayanıklı ve geri dönüşüm sağlayan GADEW ismini verdikleri damgalama algoritması önermişlerdir [30]. GADEW, Fark Genişleme Tabanlı damgalama yöntemini geri dönüşümlü bir şekilde kullanarak bozulma oranını minimize ederken, damgalama kapasitesini de artırmıştır.

Bhattacharya vd. tarafından önerilen bozulmaya sebep olmayan damgalama yöntemi özüt fonksiyonu yardımıyla satırları gruplara ayırmaktadır. Kategorik veri tutan alanları kullanan algoritma satırların yeniden sıralanması ile damga verisinin saklanmasını sağlamaktadır [31].

Hanyurwimfura vd. tarafından önerilen yöntem birden çok kelime içeren metin alanlarına damga gizleme özelliğine sahiptir. Damgalama sürecinde Lavenhtein uzaklık metodunu kullanmaktadır. Damga bit değerine bağlı olarak kelimenin konumu yatay olarak kaydırılmaktadır. Damgalanmak üzere kullanılacak satır ve alanlar rastgele olarak seçilmektedir [32].

Farfoura vd. tarafından önerile geri dönüşümlü damgalama yöntemi Thodi ve Rodriguez tarafından önerilen [37] Tahmini Hata Genişleme yöntemini kullanmaktadır. Zaman Damgası hizmetini kullanan yöntem birincil anahtara bağımlı ve veri tabanı değerlerinin küçük değişiklikleri tolere edebildiğini varsaymaktadır [33].

1.5.2. Kırılgan Damgalama Yöntemleri

Kırılgan damgalama alanında yapılan çalışmaların büyük bir kısmı görüntü, ses ve videolar üzerinde yapılmıştır. Yakın geçmişte diğer veri alanlarının önem kazanması ile birlikte metinler ve ilişkisel veri tabanları üzerinde de kırılgan damgalama şemaları kullanılmaya başlanmıştır.

Prabhakar vd. tarafından önerilen kırılgan damgalama yönteminde, veri içerisinden seçilen alt kümelerin değer dağılımı ayarlanarak sanal damganın eklenmesi sağlanmıştır. Damlamada kullanılan alan üzerindeki değişiklik veya alanın kaldırılması damganın yok edilmesine sebep olacaktır. Ayrıca verilerin dağılımını ayarlamanın zaman alması ve oluşturulan verinin saklamak için gerekli alanın boyutunun büyük olması da yöntemin zayıflıkları arasındadır [38].

Kategorik verileri kullanarak veri tabanı üzerindeki izinsiz değişiklikleri algılamak amacıyla Guo vd. tarafından geliştirilen yöntem, veri tabanındaki satırları gruplayarak, her bir grubun kendi içerisinde damgalanmasını sağlamaktadır. Öncelikle kullanıcının belirlediği gizli anahtar, veri tabanının birincil anahtarı ile beraber ilgili satırın tüm alan değerlerini kullanarak alan bazında bir özüt değer elde edilmektedir. İlgili satırın birincil anahtar değeri üzerinden elde edilen ikinci özüt değeri ile satırlar gruplara ayrılmaktadır. Bu gruplama sanal bir gruplama olduğu için satırların fiziksel konumları üzerinde değişiklik yapılmamaktadır. Gruplar oluşturulduktan sonra grup içerisinde belirlenen satırların yer değiştirmesi ile damgalama işlemi gerçekleştirilmektedir [39].

Guo vd. tarafından yapılan bir diğer çalışmada veri tabanı gruplara ayrılarak alanların en anlamsız 2 bit değerine veri saklanmıştır. Yöntem tüm alanların sayısal olduğunu kabul

etmiştir. Her bir alanın en anlamsız 2 bit değeri üzerinde gerçekleştirilecek bozulmayı tolere edebileceği varsayılmıştır. Satırların birincil anahtar ve kullanıcı tarafından belirlenen gizli anahtar değeri kullanılarak damga yerleştirilecek satırlar seçilmiş ve gruplara ayrılmıştır [40].

Kamel'in önermiş olduğu kırılgan damgalama şeması veri tabanı ilişkilerini gruplara ayırarak her bir gurubun bağımsız olarak damgalanmasını gerçekleştirmektedir. Damga verisinin sayısal olduğunu kabul eden çalışma verinin yeniden indekslenmesi yöntemi ile veri saklama sürecini gerçekleştirmektedir [41].

Khataeimaragheh ve Rashidi tarafından önerilen çalışma nümerik alanların en anlamsız 2 bitinde değişikliğe giderek kırılgan damgalama yöntemi ile veri tabanı bütünlüğünü sağlamaya çalışmaktadır. Yöntem damga yerleştirilecek alanların sayısal olmasını zorunlu kılarken, damga çıkarım aşamasında bozulmanın olduğu alanı belirleyerek orijinal veriyi geri döndürebilme özelliğine de sahiptir [42].

Hamadou vd.'nin önermiş olduğu kırılgan damgalama şeması, veri tabanı alanlarının özüt değerleri dikkate alınarak yeniden sıralanması üzerine kuruludur [43]. Bu sayede orijinal veri tabanı herhangi bir tahribata uğramamaktadır. Veri tabanı kaydının her bir alanının en anlamsız bitleri kullanılarak damga verisi oluşturulur. Oluşturulan damga verisi sertifika yetkilisine kayıt ettirilir. Şema alan adlarına göre sanal bir sıralamaya dayandığından, saldırgan tarafından alan adlarının değiştirilmesi sonrasında saldırının belirlenmesi mümkün olmayacaktır.

Khan vd. veri tabanının ilişki özelliklerini kullanan, sıfır damgalama özelliğine sahip veri tabanı üzerinde yapılacak değişiklikleri algılayabilen kırılgan bir damgalama yöntemi önermişlerdir. Sayısal alan değerlerinin kullanıldığı yöntem bu alanlar üzerinde yapılacak saldırılara karşı dayanıklı değildir [44].

Iqbal vd. tarafından önerilen yöntem veri tabanı ilişkilerini 3 gruba ayırarak her bir gruptan kırılgan bir özüt değeri hesaplamaktadır. Oluşturulan özüt değeri gruptaki satırların sayısal alan değerlerinin en anlamsız bitlerine işlenerek saklanmaktadır. Yöntem veri tabanındaki sayısal alanlarında oluşacak küçük değişikliklerin tolere edilebilir olduğunu varsaymaktadır [45].

Prasannakumari tarafından önerilen kırılgan yöntem ise veri tabanına sahte alan ekleyerek damgalama işlemini gerçekleştirmektedir. Yeni eklenen ve damga verisini barındıran alanın değerleri, orijinal veri tabanının alan değerleri üzerinde gerçekleştirilen toplama işlemi ile belirlenmektedir [46].

Camara vd. tarafından önerilen yöntem orijinal veri tabanı üzerinde değişikliğe sebep olmadan damgalama gerçekleştirmektedir. Yöntem veri tabanını gruplara ayırarak, her bir gruptan matematiksel işlemler sonucu damga verisi oluşturmaktadır [47].

Kırılgan damgalama yöntemlerinin damga yerleştirirken kullandıkları alan veri türü ve damgalama sonrası bozulmaya sebep olma özelliklerine ait özet bilgi Tablo 1.3.'de verilmiştir.

Tablo 1.3. Kırılgan damgalama yöntemleri

Çalışma	*Yıl*	*Veri Türü*	*Bozulma*
Prabhakar vd. [38]	2002	Sayısal	Olmaz
Guo vd. [39]	2004	Sayısal	Olmaz
Guo vd. [40]	2006	Sayısal	Olur
Kamel [41]	2009	Sayısal	Olmaz
Khataeimaragheh ve Rashidi [42]	2010	Sayısal	Olur
Hamadou vd. [43]	2011	-	Olmaz
Khan vd. [44]	2013	Sayısal	Olmaz
Iqbal vd. [45]	2012	Sayısal	Olur
Prasannakumari [46]	2009	-	Olmaz
Camara vd. [47]	2014	Sayısal	Olmaz

1.6. Saldırı Türleri

Damgalanmış veri tabanının izinsiz kullanımını amaçlayan saldırganlar damga verisinin tamamını veya damga çıkarım sürecini olumsuz etkileyecek oranını bozmak için saldırılar düzenlemektedirler. Bu saldırıların bazıları damga verisini yok etmek için yapılırken, bazı saldırılar saldırganların kendi damga verilerini eklemek amacını gütmektedir. Veri tabanının tuttuğu veri türüne bağlı olarak saldırı türleri de değişiklik göstermektedir.

1.6.1. Alt Küme Ekleme Saldırısı

Saldırganın damga verisini yok etmek amacıyla verisini rastgele oluşturduğu satır veya sütunları veri tabanına eklemesiyle gerçekleştirdiği saldırıdır. Damga çıkarım aşamasında yerleştirilmemiş damganın çıkarılarak damga çıkarım başarısının düşürülmesi

amaçlanmaktadır. Veri tabanında tutulan verinin türünden bağımsız olarak yapılabilecek saldırı türlerindendir.

1.6.2. Alt Küme Değiştirme Saldırısı

Saldırganın belirlediği satırlardaki sütun değerleri üzerinde gerçekleştirmiş olduğu değişim saldırısıdır. Damga yerleştirilen değerler üzerinde gerçekleştireceği değişimin damgalama sisteminin damga çıkarım başarını düşürmesini amaçlamaktadır. Veri tabanında tutulan verinin türünden bağımsız olarak yapılabilecek saldırı türlerindendir.

1.6.3. Alt Küme Silme Saldırısı

Saldırganın veri tabanından belirlediği satırları silme yöntemiyle damga verisini yok etme amacıyla gerçekleştirdiği saldırıdır. Damga çıkarım aşamasında başarının düşürülmesi amaçlanmaktadır. Damga yerleştirme yöntemlerini en yüksek düzeyde etkileyen saldırı türüdür. Damga verisinin yerleştirildiği satır veya alanın yok edilmesi damga verisinin tamamen kaybedilmesine sebep olabilmektedir. Veri tabanında tutulan verinin türünden bağımsız olarak yapılabilecek saldırı türlerindendir.

1.6.4. Bit Değiştirme Saldırısı

Saldırganın belirlediği değerler üzerinde bit düzeyinde değişiklik gerçekleştirdiği saldırı türüdür. Değiştirilecek bit sayısı ve değişim türü saldırgan tarafından belirlenmektedir. Bit değer veya değerleri rastgele değiştirildiği gibi, bazı durumlarda bit değerlerinin tersi alınarak saldırı düzenlenebilmektedir. Sayısal değer tutan alanlar üzerinde gerçekleştirilen saldırı türüdür.

1.6.5. Sahte Sahiplik Saldırısı

Saldırganın sahiplik iddia edebilmek amacıyla kendi damgasının yer aldığı satırları veri tabanına eklediği saldırı türüdür. Saldırganın eklediği satırların veri tabanı sahibi tarafından eklenen damganın çıkarım başarısını düşürmesi ve kendi damgasının sahiplik

iddiasında kullanılabilmesi amaçlanmaktadır. Veri tabanında tutulan verinin türünden bağımsız olarak yapılabilecek saldırı türlerindendir.

1.6.6. Sıralama Saldırısı

Veri tabanına yerleştirilmiş damga verisinin bütünlüğünü bozmak için satırların veya sütunlarının yerlerinin değiştirildiği saldırı türüdür. Damgalama yöntemlerinin bazıları damga çıkarımı için satır veya sütunların damga yerleştirme sürecindeki sıralarında olmalarına bağımlıdırlar. Satır veya sütunların yerlerinin değiştirilmesi damga çıkarımını olumsuz etkileyecektir. Veri tabanında tutulan verinin türünden bağımsız olarak yapılabilecek saldırı türlerindendir.

1.6.7. Satır Bazlı Çoklu Saldırı

Birden çok saldırının art arda yapılmasından oluşan saldırı türüdür. Yanlış damganın çıkarılması amacıyla ilk olarak Alt Küme Ekleme saldırısı gerçekleştirilir. Oluşan yeni veri tabanına damga yerleştirilmiş satırların yok edilmesi için Alt Küme Silme saldırısı uygulanır. Elde edilen veri tabanına son olarak Bit Değiştirme saldırısı uygulanarak damga çıkarım oranının düşürülmesi amaçlanmıştır. Tek tür saldırıların başarılı olamadığı durumlarda daha komplike saldırı türü olan Satır Bazlı Çoklu saldırının damganın yok edilmesi konusunda başarılı olduğu görülmüştür.

1.6.8. Alan Bazlı Çoklu Saldırı

Bit değiştirme ve Alt Küme Değiştirme saldırılarının art arda uygulanmasıyla gerçekleştirilen saldırı türüdür. Satır Bazlı Çoklu saldırı türünde olduğu gibi, birden çok seviyeli saldırı ile dayanıklı damga yerleştirme yöntemlerine karşı damganın yok edilmesi amaçlanmaktadır.

1.6.9. Element Sıra Karıştırma Saldırısı

XML veri tutan veri tabanlarında uygulanan saldırı türlerinden biridir. XML'in yapısal özelliğinden dolayı aynı seviyedeki düğümlerin sıralarının değiştirilmesi XML verinin bütünlüğü bozmamaktadır. Bu özellikten yararlanan saldırı türü XML verideki düğümlerin sıralarını değiştirerek, sıralamaya bağımlı olan damgalama yöntemlerine karşı başarı elde etmeyi amaçlamaktadır.

1.6.10. Etiket Ad Karıştırma Saldırısı

XML etiketleri büyük/küçük harf duyarlıdırlar. "<bilgi>" etiketi ile "<BilGi>" etiketi farklı etiketlerdir. XML verinin bütünlüğünün bozulmaması için veriyi tutan düğümün başlangıç ve bitiş etiketinin karakterlerinin aynı olması gerekir. Yani, "<bilgi>Ad</bilgi>" düğüm verisini "<BilGi>Ad</BilGi>" haline getirmek veriyi bozmayacaktır. Fakat bu değişim damga verisini etiket içerisine saklayan damgalama yöntemleri üzerinde etkili olup, damga çıkarım sürecini olumsuz etkileyecektir.

1.6.11. Alan Silme veya Karıştırma Saldırısı

XML düğümünün sahip olduğu alan değerlerinin silinmesi veya düğüm içerisindeki sırasının değiştirilmesi üzerine kurulu olan saldırı, XML veriler tutan veri tabanlarında damgalama gerçekleştirirken düğüm alanlarını kullanan yöntemleri etkilemeyi amaçlamaktadır.

2. YAPILAN ÇALIŞMALAR

Günümüzde internete erişim hızı ve erişen cihaz sayısındaki artış veri hareketliliğini ortaya çıkarmıştır. Daha büyük boyuttaki veriler daha hızlı bir şekilde yer değiştirmektedir. Küçük boyuttaki doküman, resim ve videoların yanı sıra büyük boyuttaki veri tabanlarının paylaşımı da gün geçtikçe artmaktadır. Veri tabanları yapılarından dolayı kolaylıkla çoğaltılabilir, içeriği değiştirilebilir ve yeniden yayınlanabilirler. Ayrıca kullanım alanlarından dolayı içeriğinin değişmesi veya silinmesi durumları ile de karşı karşıyadırlar. Tüm bu durumlar veri tabanları üzerinde sahiplik ve bütünlük kontrolünün sağlanması problemini ortaya çıkarmıştır. Literatürde 2002 yılından bugüne kadar veri tabanlarının güvenliğini sağlamak amacıyla sayısal damgalamanın kullanıldığı çalışmalar yapılmıştır.

Tez kapsamında ilişkisel veri tabanlarında sahiplik kontrolünün yapılması amacıyla uygulanabilecek çalışmalar aşağıda maddeler halinde verilmektedir.

1. Literatürdeki benzer çalışmalar dikkate alındığında sık kullanılmayan tarih alanına damga yerleştirme yöntemiyle veri tabanının güvenliği sağlanmaya çalışılmıştır.
2. Sayısal alanlar üzerinde sıklıkla kullanılan Fark Genişleme Tabanlı Damgalama yöntemi yerine önerilen nümerik dönüşüm yöntemi ile damgalama kapasitesi artırılırken, çalışma süresinde iyileşmeler görülmüştür. Fark Genişleme Tabanlı Damgalama yöntemi için en az 2 sayısal alan gerekli iken, önerilen yöntem tek sayısal alan üzerinde çalışabilmektedir.
3. Damgalama sonrası veri tabanındaki bozulmayı minimize etmek amacıyla damgalanacak satırların alan seçiminde Ateş Böceği optimizasyon algoritmasının kullanıldığı yeni bir damgalama yöntemi önerilmiştir. Literatürdeki benzer çalışmalar ile karşılaştırıldığında damgalama kapasitesinde artış, saldırganlar tarafından damganın algılanabilirliğinde iyileştirmeler sağlayan yöntem düşük çalışma süresine sahiptir.
4. XML veri içeren veri tabanları için veri boyutunu değiştirmeden, anlamsal bütünlüğü veya içeriği bozmadan damgalama gerçekleştiren yeni bir damgalama algoritması geliştirilmiştir.
5. Sayısal ve metin alan içeren veri tabanlarında her iki veri türüne de damga yerleştirebilen yeni bir damgalama yöntemi gerçekleştirilmiştir. Önerilen

yöntem sayısal alanlar yanında metinler üzerinde de damgalama gerçekleştirdiği için damgalama kapasitesi literatürdeki benzer çalışmalara göre yüksektir.

Bu tez çalışmasında kapsamında gerçekleştirilen çalışmalar MicroSoft SQL Server veri tabanı yönetim sistemi üzerinde MicroSoft Visual Studio IDE ortamında C# programlama dili ile hazır paketler olmaksızın kodlanarak gerçekleştirilmiştir. Maddeler halinde verilmiş olan çalışmaların detayları sırasıyla ilerleyen bölümlerde verilmektedir.

2.1. İlişkisel Veri Tabanları İçin Bölümleme Tabanlı Damgalama Yöntemi

Literatürde gerçekleştirilen çalışmalar incelendiğinde büyük bir çoğunluğunun sayısal alan veya alanlar kullanılarak damgalama yapıldığı görülmektedir. Metin, XML, kategorik veya tarih verisi barındıran sayısal olmayan alanlar üzerine yapılan çalışmalar nispeten daha az sayıdadır. Bu sebeple damganın yok edilmesine veya değiştirilmesini amaçlayan saldırıların birçoğu sayısal veriler üzerine odaklanmıştır.

Sayısal alanların yanı sıra tarih verisi de veri tabanlarında sıklıkla karşılaşılan alanlardan biridir. Fakat veri tabanı damgalama yöntemlerinde pek de sık kullanılmamıştır. Odeh vd. tarafından gerçekleştirilen çalışma tarih bilgisi barındıran alan değerlerini kullanarak damga verisini saklamayı amaçlayan az sayıdaki çalışmalardan biridir [48]. Çalışma damga verisi olarak seçilen resmi ikilik düzene döndürerek, her bir bit değerini veri tabanının tarih alanında saklamaya çalışmıştır. Veri tabanlarında tarih verisini tutan alanlar gün ve saat bilgisi olarak 2 farklı bilgiyi aynı alanda saklamaktadırlar. Araştırmacılar tarih alanının saniye bilgisindeki küçük değişimlerin kabul edilebileceğini varsayarak damga bitini saniye verisine saklamışlardır.

Tez çalışması kapsamında gerçekleştirilen yöntem de [48] çalışmasında benzer şekilde tarih tipindeki verileri tutan alanlar üzerinde damgalama işlemini gerçekleştirmektedir. Damgalanacak satırların seçiminde SHA-256 özüt fonksiyonu kullanılmıştır. Saldırılara karşı dayanıklı olması amacıyla damgalanmak üzere seçilen aday satırlar arasında alt bölümleme uygulaması yapılmıştır. Damga verisi (resim, ses, metin, vb.) ikilik düzene çevrilerek 5'er bitlik gruplara ayrılmıştır. Her 5 bitlik değerin 10'luk tabandaki karşılığı ilgili grubun içerisindeki tüm aday satırların tarih verisi barındıran alan (birden çok alan olması durumunda seçim yapılacaktır) değerinin saniye bilgisine saklanmaktadır. Damganın ortadan kaldırılması veya bozulması amacıyla gerçekleştirilecek saldırılar sonucu aynı

gruptaki damgalanmış satırlardan en az 1 tanesi bile kurtarılması durumunda, ilgili grubun damga verisi kayıpsız olarak kurtarılmış olacaktır. Çıkarılan damga verisinin doğruluğu için de çoğunluk oylaması yapılarak saldırılar ile değiştirilmesi mümkün olan damga verilerinin kayıpsız kurtarılması amaçlanmıştır.

2.1.1. Damga Yerleştirme Algoritması

Damga yerleştirme algoritması veri tabanından damgalanacak satırların seçilmesi ve ilgili satırların tarih alanının saniye verisi içerisine damga değerini saklamaktadır. Damga verisinin türü (metin, resim, ses, vb.) algoritma için önem arz etmemektedir. Damga olarak seçilen veri ikilik düzene çevrilmektedir. Saniye değeri 0-59 sayısal değer aralığında olduğundan ikilik düzendeki damga verisi 5 bitlik parçalara ayrılır. Bu sayede damga verisi 0-31 sayısal aralığında değerlerden oluşmuş olacaktır ki, bu da damganın saklanacağı saniye alanının sınırları içerisindedir.

Damgalamada kullanılan satırların seçiminde (3) eşitliğinden yararlanılmıştır.

$$\mathcal{H}\big((r.P)||\kappa\big)\ mod\ \kappa = 0 \qquad (3)$$

$\mathcal{H}$ ifadesi SHA-256 özüt fonksiyonunu, $r.P$ ilgili satırın birincil anahtar değerini, κ kullanıcı tarafından belirlenen gizli anahtar değerini, $||$ ifadesi birleştirme operatörünü, mod ifadesi ise matematiksel modüle işlemini yerine getirmektedir. Veri tabanındaki her bir satır için (3) ifadesini sağlayanlar damgalanmak üzere seçilecektir.

Seçilen kaydın birden çok tarih alanı olması durumunda, bu alanlar arasından ikinci bir seçim işlemi uygulanarak damga yerleştirilecek alan belirlenecektir. Satırın sahip olduğu her bir tarih alanının saat ve dakika değeri ile kullanıcı tarafından belirlenen gizli anahtar değeri birleştirilerek özüt fonksiyonuna girdi olarak verilecektir. Özüt fonksiyonunun üreteceği değerin κ tabanındaki karşılığı 0'a eşit olan tarih alanını damga yerleştirmede kullanılmak üzere seçilecektir.

Damga yerleştirilecek satır ve alan belirlendikten sonra saklanacak damga verisinin oluşturulmasına sıra gelmiştir. İkilik tabana döndürülerek 5'er bitlik parçalara ayrılan damga verisinin her bir parçası bir grup olarak düşünüldüğünde, damgalanmak üzere seçilen satırın tarih alanının saniye değerine hangi grup damga değerinin saklanacağı (4) ifadesi yardımıyla belirlenir.

$$\mathcal{H}\big((r.P)||\kappa||(r.P)\big)\ mod\ N_p \quad (4)$$

(4) ifadesinde yer alan N_p ifadesi grup sayısını göstermektedir. Damgalanmak üzere seçilen satırın birincil anahtar değeri ve kullanıcı tarafından belirlenen gizli anahtar değeri yardımıyla elde edilen özüt değerinin N_p tabanındaki karşılığı saklanacak olan damga grup indisini belirlemektedir. Grup bilgisi kullanılarak damga verisinin yerleştirilmesi, yöntemi literatürde kullanılan saldırılara karşı dayanıklı hale getirmektedir. Damgalama algoritmasına ilişkin yalancı kod ifadesi Şekil 2.1.'de verilmiştir.

Giriş: R veri kümesi, gizli anahtar κ, veri alanları $\{A_1, A_2, \dots, A_v\}$, grup sayısı N_p, damga verisi W
Çıkış: Damgalanmış veri R_w
forall kayit $r \in R$ **do**
  if $\mathcal{H}\big((r.P)||\kappa\big)\ mod\ \kappa = 0$ then
    forall alan $a \in \{A_1, A_2, \dots, A_v\}$
      if $\mathcal{H}\big((a.SS)||(a.DK)||\kappa\big)\ mod\ \kappa = 0$ **then**
        $a.SN = W\left(\left(H\big((r.P)||\kappa||(r.P)\big)\ mod\ N_p\right) + 1\right)$
      end
    end
  end
end

Şekil 2.1. Damga yerleştirme algoritması

2.1.2. Damga Çıkarım Algoritması

Önerilen yöntem damgalanmış veri tabanından damgayı çıkarabilmek için orijinal veri tabanına ihtiyaç duymamaktadır. Damga çıkarım işlemi için kullanıcının damgalama sürecinde belirlediği gizli anahtar değeri ve grup sayısı yeterlidir. Damga çıkarım algoritmasına ait yalancı kod Şekil 2.2'de verilmiştir.

Giriş: Damgalanmış veri R_w, gizli anahtar κ, veri alanları $\{A_1, A_2, \dots, A_v\}$, grup sayısı N_p, damga aday listesi $\{D_1, D_2, \dots, D_{N_p}\}$

Çıkış: Damga verisi W

forall kayit $r \in R$ **do**

if $\mathcal{H}((r.P)||\kappa)\ mod\ \kappa = 0$ then

forall alan $a \in \{A_1, A_2, \dots, A_v\}$

if $\mathcal{H}((a.SS)||(a.DK)||\kappa)\ mod\ \kappa = 0$ **then**

$x = (H((r.P)||\kappa||(r.P))\ mod\ N_p) + 1$

$D_x.Add(a.SN)$

end

end

end

end

forall $x \in \{1,2, \dots, N_p\}$ **do**

$W(x) = majority(D_x)$

End

Şekil 2.2. Damga çıkarma algoritması

İlk olarak damga yerleştirme aşamasında satır seçme kriteri olarak kullanılan (3) ifadesi yardımıyla damga çıkarılacak satırlar belirlenir. Damga çıkarılacak satırlar belirlendikten sonra satırın damgalanmış alanının bulunması gerekmektedir. Damga çıkarılacak aday satırın birden çok tarih verisi bulunduran alanı olması durumunda damga yerleştirme aşamasında kullanılan kriter kullanılarak satırın her bir tarih alanının saat ve dakika değeri ile kullanıcının belirlediği gizli anahtar değeri birleştirilerek özüt fonksiyonuna girdi olarak verilmektedir. Özüt fonksiyonunun üreteceği değerin κ tabanındaki karşılığı 0'a eşit olan tarih alanının saniye değeri damganın çıkarılacağı alan olarak belirlenir.

Yerleştirilen damga değerinin saldırılar sebebiyle bozulma ihtimaline karşı çıkarılan grup damga verileri üzerinde çoğunluk oylaması yapılacaktır. Bu amaçla her bir damga grubu için boş bir dizi oluşturulacaktır. Çıkarılan damga değeri karşılık düşen grup dizisine yerleştirilecektir. Damga çıkarım işleminin ardından grupları temsil eden diziler üzerinde

değerlendirme yapılmaktadır. Dizideki değerlerin görülme sıklığına göre gruba saklanmış olan damga değeri belirlenmektedir. En fazla bulunan değer, ilgili gruptan çıkarılacak olan damga değerini belirlemektedir. Şekil 2.2.'de verilen damga çıkarım algoritmasına ait yalancı kodda bu işlemi $majority$ metodu gerçekleştirmektedir. Örneğin 5 kayıttan oluşan bir grup için üretilen dizi $\{12,43,12,13,12\}$ şeklinde ise, grubun içerisinden çıkarılan damga değeri 12 olarak karar verilir.

2.2. İlişkisel Veri Tabanları İçin Nümerik Dönüşüm Tabanlı Damgalama Yöntemi

Sayısal alanlar üzerinde damgalama gerçekleştiren çalışmalar incelendiğinde genellikle 2 sayısal alana ihtiyaç duymaları veya sayısal değerler üzerinde sebep oldukları değişimin fazla olması dikkati çekmektedir. Çalışma kapsamında önerilen yöntem ise damgalama kapasitesini artırmak amacıyla tek bir sayısal alan üzerinde çalışabilmekte, damgalama sonrası oluşacak değer değişimini minimize edebilmekte ve basit matematiksel işlemleri kullanması sayesinde çalışma süresini kısaltabilmektedir. Saldırılara karşı dayanıklı olması amacıyla damgalanmak üzere seçilen aday satırlar arasında alt bölümleme uygulaması yapılmıştır. Damga verisinin türünden (resim, ses, metin vb.) bağımsız olarak damgalama yapabilen yöntem damga verisini ikilik düzene çevirerek her bir bit değerini ilgili grubun içerisindeki tüm aday satırlara saklamaktadır. Saldırılar sonucu aynı gruptaki damgalanmış satırlardan en az 1 tanesi bile kurtarılması durumunda, damga verisi kayıpsız olarak kurtarılacaktır. Çıkarılan damga verisinin doğruluğu için de çoğunluk oylaması yapılarak saldırılar ile değiştirilmesi mümkün olan damga verilerinin kayıpsız kurtarılması amaçlanmıştır.

Önerilen yöntem sayısal alanları kullanarak damga verisini saklamaktır. Damganın yerleştirilmesi sürecinde sayısal alanlar üzerinde kullanılan nümerik dönüşüm (5) ifadesinde verilmiştir [49, 50]. Veri kaybına sebep olmadan orijinal değerlere geri dönebilmeyi desteklenen dönüşüm ifadesinde veri gömülecek sayısal değerler $\mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{n})$ vektörü, damga verisinin saklanması (damgalama) sonucu elde edilecek sayısal değerler $\acute{\mathrm{x}} = (\acute{\mathrm{x}}_1, \acute{\mathrm{x}}_2, \dots, \acute{\mathrm{x}}_\mathrm{n})$ vektörü ile ifade edilmiştir.

$$\begin{cases} \acute{x_1} = 2 * x_1 - 2f\big(a(x)\big) + b_1, \\ \qquad\qquad . \\ \qquad\qquad . \\ \acute{x}_{n-1} = 2x_{n-1} - 2f\big(a(x)\big) + b_{n-1}, \\ \qquad \acute{x_n} = 2x_n - a(x) \end{cases} \tag{5}$$

(5) ifadesindeki $b = (b_1, b_2, \dots b_{\mathrm{n}-1})$ vektörü sayısal değerler içerisine saklanacak olan damga verisidir. $\mathrm{a}(x)$ ifadesi x vektörünü oluşturan değerlerin aritmetik ortalamasının en yakın tam sayıya yuvarlatılmış değeri, f(z) ifadesi ise z değerinin yarısının yukarı yuvarlanmış tam sayı değerini göstermektedir. Genel form dikkate alındığında, n adet tamsayı değerinden oluşan veri kümesine $n-1$ adet damga biti gömülebildiği anlaşılacaktır.

Damga çıkarım sürecinde damgalama öncesi veri tabanının orijinal değerlerinin elde edilmesini sağlayan dönüşüm (6) ifadesinde verilmiştir.

$$x_i = h(\acute{x_\iota}) + a\big(h(\acute{x})\big) + \acute{x_n}\ mod\ 2, \quad i \in \{1, \dots, n\} \tag{6}$$

$h(\acute{x})$ ifadesi $\acute{x}$ değerinin yarısının taban değerini hesaplamaktadır. $\mathrm{z} = (\mathrm{z}_1, \mathrm{z}_2, \dots, \mathrm{z}_\mathrm{n})$ vektörü için $h(\mathrm{z}) = \big(h(\mathrm{z}_1), h(\mathrm{z}_2), \dots, h(\mathrm{z}_\mathrm{n})\big)$ şeklinde olacaktır. mod matematiksel modüle işlemini yerine getirmektedir. Damgalama sürecinde saklanan damga verisini barındıran b vektör değerleri damga çıkarım sürecinde (7) ifadesi ile hesaplanmaktadır.

$$b_j = \acute{x_j}\ mod\ 2, \quad j \in \{1, \dots, n-1\} \tag{7}$$

Önerilen yöntemde damga verisi ikili düzene çevrilerek kullanılmaktadır. Bu sayede damga verisinin türü (resim, metin, ses, vb.) önemli değildir. Veri tabanında damgalama için kullanılacak kayıtların seçiminde (8) ifadesinden faydalanılmaktadır. İlgili satırın birinci anahtar değeri $r.P$ ile, özüt fonksiyonu $\mathcal{H}$ ile, gizli anahtar değeri κ ile ve birleştirme işlemi ise " ‖ " sembolü ile gösterilmiştir.

$$F(r.P) = \mathcal{H}\left(\kappa \parallel \mathcal{H}\big(\kappa \parallel (r.P)\big)\right) \tag{8}$$

(8) ifadesinde verilen fonksiyonun üreteceği değer r ilişkisinde yer alan kaydın o anki birincil anahtar değeri P ve özel anahtar değeri κ ile doğrudan ilişkilidir. Bu sayede damgalanmak üzere seçilen satırlar kullanıcı tarafından belirlenen gizli anahtar değeri olmadan saldırganlar tarafından belirlenemeyecektir.

2.2.1. Damga Yerleştirme Algoritması

Veri tabanından damgalama için kullanılacak kayıtların seçiminde (8) ifadesinden yararlanılmıştır. İfadenin κ tabanındaki karşılığı 0'a eşit olan satırlar damgalanmak üzere seçilmiştir. Damgalanacak satırın birden çok sayısal alanı olması durumunda alanlar arasından tek bir alanın seçilmesi gerekmektedir. Sayısal alan sayısının ν olduğu varsayıldığında damgalanacak alan indisi (9) ifadesi ile bulunmaktadır.

$$F(r.P)\ mod\ \nu \tag{9}$$

Damgalama sürecinde üzerinde dönüşüm gerçekleştirilecek sayısal değerler, damgalama işlemi için (8) ifadesi ile seçilen satırın (9) ifadesi ile seçilen alanından başlanmak üzere n adet ardışık satırın aynı alanının değerlerinden oluşacaktır. Damgalama algoritmasına ait yalancı kod Şekil 2.3.'de verilmiştir. Damgalama işlemi başlamadan önce veri seti birincil anahtar değeri dikkate alınarak artan sırada sıralanmıştır. Ayrıca veri setinin alanları da alfabetik olarak sıralı olacak şekilde yer değiştirilmiştir. Bu yer değiştirme veri tabanı üzerinde fiziksel bir değişikliğe sebep olmayan, algoritma içerisinde alan indislerini belirleyen sanal bir sıralamadır. Bu sayede ekleme ve alan değiştirme saldırılarına karşı önerilen yöntemin dayanıklılığı artırılmıştır.

Şekil 2.3.'de yer alan $SatırSayısı$ metodu damgalanmak üzere seçilen veri setinin kayıt sayısını, mod metodu matematiksel modüle işlemini gerçekleştirmektedir. t değeri damgalanmak üzere seçilen, üzerinde nümerik dönüşüm yapılarak n adet satıra saklanacak olan damga verisinin indisini belirlemektedir. N_p değeri kullanıcı tarafından belirlenen grup sayısıdır. Bu gruplama sayesinde aynı damga verisi farklı dönüşüm gruplarına da saklanarak damga çıkarım aşamasında saldırılara karşı dayanıklılık artırılmıştır. $Damgala$ metodu ise kendisine parametre olarak gönderilen dizi üzerinde nümerik dönüşüm algoritmasını uygulayarak t indisindeki damga verisini $(\mathrm{W_t})$ saklamaktadır.

Giriş : R veri kümesi, gizli anahtar κ, veri alanları $\{A_1, A_2, \dots, A_v\}$, damga verisi W, dönüşüm yapılacak grup eleman sayısı n, bölüm sayısı N_p

Çıkış : Damgalanmış veri R_w

for i=l *to* $SatırSayısı(R)$
  if $F(R_i.P)\ mod\ \kappa = 0$
    // damgalamada kullanılacak alan indisi
    j = $F(R_i.P)\ mod\ v$
    for k=0 *to* n
      // dönüşümde kullanılacak alan değerleri
      $dizi.Add(R_{i+k}.A_j)$
    end
    $t = \left(F(R_i.P)\ mod\ N_p\right) + 1$
    $Damgala(dizi, W_t)$
  end
end

Şekil 2.3. Damga yerleştirme algoritması

2.2.2. Damga Çıkarım Algoritması

Önerilen yöntem damga çıkarım esnasında orijinal veri tabanına ihtiyaç duymadan damgalanmış veri tabanını, kullanıcı tarafından belirlenen gizli anahtar değeri ve grup eleman sayısını kullanarak doğrulamayı gerçekleştirecektir. Damga çıkarım işlemi başlamadan önce veri seti birincil anahtar değeri dikkate alınarak artan sırada sıralanır. Ayrıca veri setinin alanları da alfabetik olarak sıralı olacak şekilde yer değiştirilir. Bu yer değiştirme veri tabanı üzerinde fiziksel bir değişikliğe sebep olmayan, algoritma içerisinde alan indislerini belirleyen sanal bir sıralamadır.

Damgalama sürecinde olduğu gibi damga çıkarım sürecinde de (8) ifadesinden yararlanılarak damga çıkarılacak satır gruplarının başlangıç satırları belirlenecektir. İfadenin κ tabanındaki karşılığı 0'a eşit olan satırlar damga çıkarılmak üzere seçilecek satır gruplarının başlangıcıdır. Satır seçiminin ardından damganın çıkarılacağı sayısal alan

belirlenecektir. Sayısal alan sayısının birden çok olduğu durumda, damgalanacak alanın indisi damga yerleştirme sürecinde olduğu gibi (9) ifadesi ile bulunmaktadır.

Çıkarılan damga değerinin saldırı sonucu bozulma ihtimaline karşı her bir damga değeri için boş bir dizi oluşturulacaktır. $DamgaÇıkar$ metodu yardımıyla çıkarılan damga değeri karşılık düşen bölüm dizisine yerleştirilecektir. Veri tabanındaki bütün satırların kontrol edilerek damga çıkarım işlemi ardından bölümleri temsil eden diziler üzerinde değerlendirme yapılmaktadır. Dizideki değerlerin sıklığına göre bölüme saklanmış olan damga değeri belirlenmektedir. En fazla görülme sıklığı olan değer, ilgili gruptan çıkarılacak olan damga değerini belirler. Şekil 2.4.'de verilen damga çıkarım algoritmasına ait yalancı kodda bu işlemi $majority$ metodu gerçekleştirmektedir.

Giriş : Damgalanmış veri R_w , gizli anahtar κ, veri alanları $\{A_1, A_2, \dots, A_v\}$, bölüm sayısı N_p, dönüşüm yapılacak grup eleman sayısı n, Damga aday listesi $\{D_1, D_1, \dots D_{N_p}\}$

Çıkış : Damga verisi W

for i=l to $SatırSayısı(R)$

  if $F(R_i.P)\ mod\ \kappa = 0$

    // damga çıkarımında kullanılacak alan indisi

    j = $F(R_i.P)\ mod\ v$

    for k=0 *to* n

      // dönüşümde kullanılacak alan değerleri

      $dizi.Add(R_{i+k}.A_j)$

    end

    $t = \left(F(R_i.P)\ mod\ N_p\right) + 1$

    $w = DamgaÇıkar(dizi, W_t)$

    $D_t.Add(w)$

end

forall $x \in \{1,2, \dots N_p\}$ **do**

  $W(x) = majority(D_x)$

end

Şekil 2.4. Damga çıkarma algoritması

2.3. İlişkisel Veri Tabanlarında Homoglif Değişim Tabanlı XML Damgalama Yöntemi

Veri tabanı damgalama şemaları, amaçlarına bağlı olarak sağlam veya kırılgan şemalar olarak kategorize edilebilirler. Sağlam damgalama şemaları [16-18, 51-59] telif hakları korumayı sağlarken, kırılgan damgalama şemaları müdahale tespiti ve bütünlüğün kanıtı için kullanılırlar.

Sağlam damgalama şemalarında, veri tabanının kime ait olduğunu (kişi, kurum, organizasyon) kanıtlayabilmek için damga bilgisi çoğunlukla veri sahibinin bilgilerini taşımaktadır. Telif hakkı koruması sağlayan sağlam şemaların birçoğu [16-18, 51-59] veri bütünlüğünü ve kullanılabilirliğini etkileyen orijinal içeriğin bozulmasına sebep olmaktadır. Bu şemalar damgalarını yerleştirmek için ilişkisel veri tabanlarının numerik [51, 55-59] veya kategorik alanlarını [52, 54] kullanmaktadırlar. Kullanılan damga bilgisi anlamsız bir şekilde oluşmuş bit dizisi olabileceği gibi resim, konuşma veya sahip bilgisi gibi anlamlı bit dizisi de olabilir.

XML'in sahip olduğu iki özellik sebebiyle Web'in temelini oluşturduğu söylenebilir. Bu özellikler, hiyerarşik veriyi düz metin kullanarak kodlayabilmesi ve kullandığı ayrıntılı etiketler sayesinde özel bir okuyucu veya tercümana ihtiyaç duyulmadan belgenin anlaşılabilir oluşudur [60, 61]. XML bilgi alışverişinde, web servislerinde, şirket-tüketici veya şirketler arasındaki iletişimde ve hatta medikal uygulamalarda sıklıkla kullanılmaktadır.

XML farklı türdeki verileri ortak bir dil ile tanımlamayı sağlamaktadır. Ağ teknolojilerinin gelişmesiyle, daha fazla veri XML formatında tanımlanmakta ve yer değiştirmektedir. Günümüzde verinin farklı platformlar arasında paylaşımı için XML sıklıkla kullanılmaktadır. Metin olarak saklandıkları için yasadışı izinsiz olarak kopyalanma tehdidi altındadır. Bu sebeple XML verilerinin telif hakkı problemini çözmek önem arz etmektedir [62, 63].

Literatürde yapılmış örnek çalışmalar ve XML alanların spesifik özellikleri dikkate alındığında, önerilen yöntem XML alanlar üzerinde uygulanabilir, bozulmaya sebep olmayan, etiketlerin içerdiği verinin türüne bağımlı olmaksızın damga yerleştirebilen, damga verisini sertifikalandırıp dış veri olarak farklı bir konumda saklanmasına gerek kalmayan, verinin boyutunu değiştirmeyen, damga çıkarım sonrası orijinal veriye geri dönüşüm sağlayan, algılanması güç yeni bir damgalama şemasıdır.

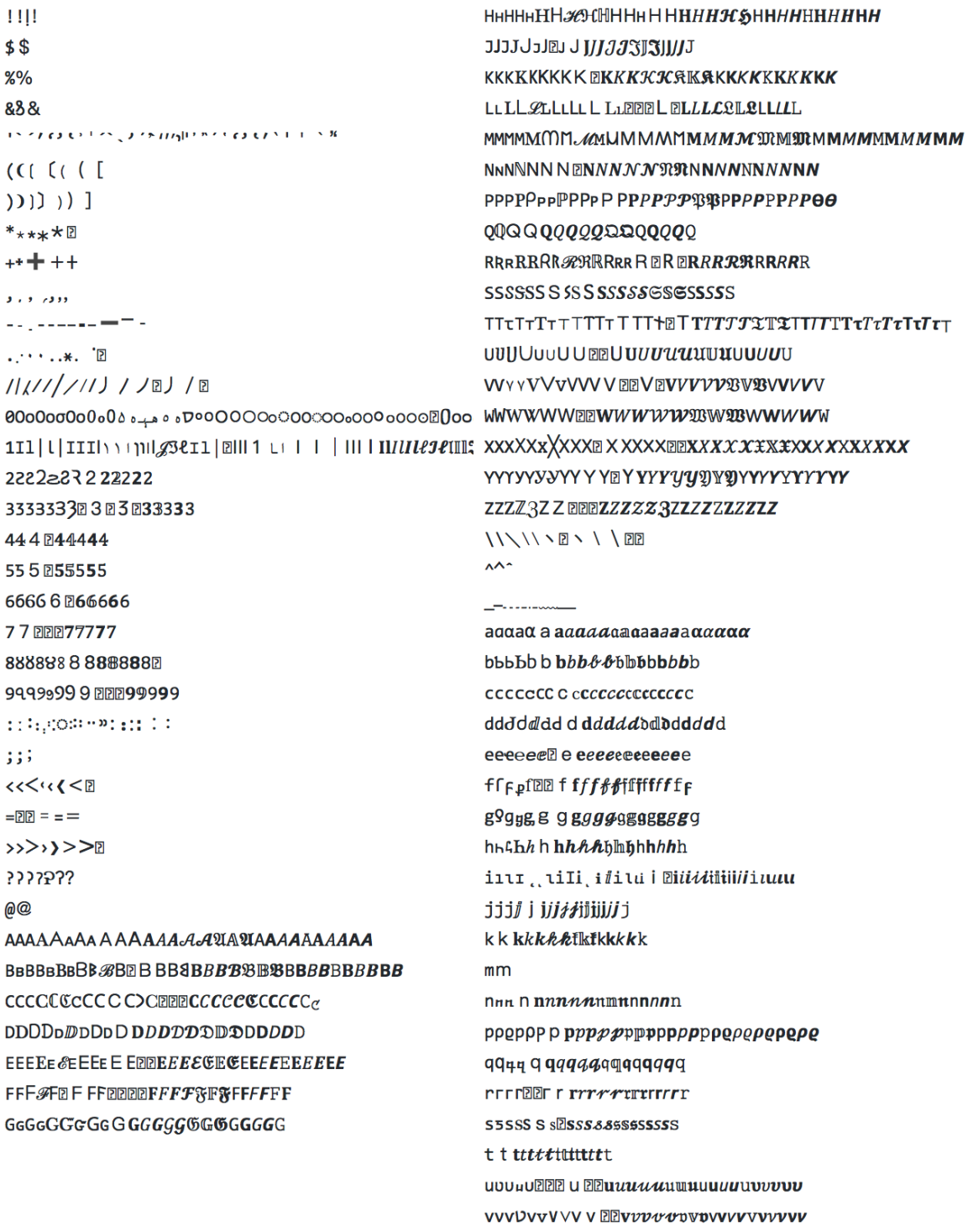
Gerçekleştirilen çalışmada veri tabanlarında XML veri alanlarına damga yerleştirme sürecinde damganın fark edilebilirliğini azaltmak, damgalama kapasitesini artırmak ve veri türünden bağımsız olarak damgalama işlemini gerçekleştirmek amacıyla Homoglif Dönüşüm yönteminden faydalanmaktadır [64]. Literatürdeki çalışmaların bir kısmı XML etiketlerin içeriğinde yer alan metin türlerine damgalama yapabilmektedir. Metinler ile beraber nümerik veriler üzerine de damga yerleştiren az sayıda çalışma ise nümerik verinin bozulmasına ve damga çıkarım sonrası orijinal verinin elde edilememesine sebep olmaktadır. Önerilen yöntem XML alanın içerdiği nümerik ve metin verileri ayırt etmeksizin orijinal veriyi bozmadan damga yerleştirebilmektedir. Yöntemin geri dönüşüm desteği sayesinde, damgalanmış XML verisi içerisinden, kullanıcının belirlediği özel anahtar dışında hiçbir ek bilgiye gerek kalmadan yerleştirilmiş damga verisi ve XML verinin orijinal hali başarıyla elde edilebilmektedir.

Literatürdeki çalışmalar XML alanların metin verileri üzerinde damga gerçekleştirirken metnin içerdiği kelimeleri sinonim havuzundan seçilen karşılıkları ile değiştirmekte, karakterleri büyültüp/küçülmekte veya boşluk eklemektedir. Bu işlemler damga verisinin saldırganlar tarafından belirlenebilmesini kolaylaştırmaktadır. Önerilen yöntem metin veriler üzerinde anlamsal bozulmaya sebep olmadığı için saldırganlar tarafından fark edilebilirliği düşüktür. Bu özelliği sayesinde sosyal ağ, forum, mesajlaşma gibi genel kullanıma açık veri tabanlarının yanı sıra tıbbi, askeri ve güvenlik alanındaki veri tabanları gibi hassas veriler üzerinde de uygulanabilirdir.

Önerilen damgalama yöntemi XML veri alanlarının etiket bilgisini, veri içeriğini ve boyutunu değiştirmeksizin, veri üzerinde Homoglif Dönüşüm yöntemini kullanarak damgalama gerçekleştirmektedir. Bu yöntemde, karakterler kendilerine benzer görünümlü, farklı Unicode [65] karakterler ile değiştirilmektedir. Damgalama sürecinde orijinal karakteri değiştirmesi durumunda “1” sayısal verisini, aynı bırakması durumunda ise “0” sayısal verisini damga olarak yerleştirmektedir. Algoritmanın bu özelliği sayesinde, damga verisinin tamamı veya büyük bir kısmı algılanamaz durumdadır. Orijinal verinin boyutu değişmemekle birlikte, verinin içeriğinde veya anlamsal bütünlüğünde de herhangi bir değişiklik yapılmamaktadır. Tüm bu özellikler sayesinde verinin bütünlüğü bozulmadan yüksek miktarda damga verisi gömülebilmektedir.

2017 yılı Haziran ayında duyurulan Unicode 10.0 versiyonu ile birlikte 1.200.000’e yaklaşan Unicode karakterler arasında birbiri ile benzerliği yüksek olan karakterler bulunmaktadır. Unicode Konsorsiyum görünüm olarak benzer karakterlerin listesini belirli

aralıklarla yayınlamaktadır [66]. Şekil 2.5.'de örnekleri verilen bu karakterler arasında orijinal karaktere benzerliği sayesinde gözle ayırt edilemeyecekler olduğu gibi genel görünüm itibariyle benzer, fakat damgalama algoritmasında kullanılamayacak şekilde fark edilebilir olanlar da yer almaktadır.



Şekil 2.5. Birbirine görünüşleri benzer Unicode karakter örnekleri

Değiştirilen Unicode karakterin fark edilebilirlik durumu, metnin sunulduğu ortamda kullanılan font ve karakter kodlama bilgisi ile de ilişkilidir. Belirli bir font ailesinde birbirine benzediği düşünülen ve gözle bakıldığında farklı oldukları belirlenemeyen 2 farklı Unicode karakter, başka bir font ailesinde gözle algılanabilir olabilmektedir. Homoglif dönüşümde kullanılacak karakter ile damga verisi yerleştirildiğinde yerine kullanılacak olan Unicode karakter seçilirken yapılan testlerde, web ortamında sıklıkla kullanılan kodlama türleri ile farklı font aileleri kullanılmıştır. Test sonucunda, insan gözü ile fark edilebilir karakter dönüşümleri kullanım dışı bırakılmıştır. Şekil 2.6.'da homoglif dönüşüm için kullanılan orijinal karakterler ile homoglif dönüşüm sonrası yerine kullanılacak karakterlerin listesi verilmiştir.

Orjinal Karakterler	!$%2345679ABCDEFGHJKLMNPQRSTUVXYZabcdeghijoprsuvxyz
Homoglif Karakterler	! $ % 2 3456 7 9 ABCDEFGHJKLMNPQRSTUVXYZabcdegһijoprsuvxyz

Şekil 2.6. Önerilen algoritmanın Homoglif Dönüşümde kullandığı karakterler ve Unicode karşılıkları

Önerilen yöntemde damga verisi ikilik tabana dönüştürülmektedir. Bu sayede damga verisinin türü (resim, metin, ses, vb.) algoritma açısından önemli değildir. Damga verisinin bit değerine göre homoglif dönüşümde karakter değişimi gerçekleştirilmektedir. Sadece damga bit değerinin "1" olduğu durumlarda homoglif karakter dönüşümü yapılmaktadır. Örneğin damga bit değeri "0" olduğu durumda veri içerisinde Şekil 2.6'da listelenen karakterlerden ilk karşılaşılan karakter değiştirilmeden bırakılacak ve "0" değerinin damgalanması sağlanacaktır. Damga bit değerinin "1" olması durumunda ise Şekil 2.6'da listelenen karakterlerden ilk karşılaşılan karakter homoglif eşi ile değiştirilerek "1" damga verisinin metne gömülmesi sağlanacaktır. Şekil 2.7'de "Bilgehan" örnek metni ve metnin içerisine "101001" damga verisinin gömülmesi sonucu metnin damgalanmış hali görülmektedir.

Orjinal Metin	Bilgehan
Homoglif Metin	Bilgehan

Şekil 2.7. Damga verisi olarak "101001" gömülen orijinal ve damgalanmış metin

“101001” damga verisinin ilk bit değeri “1” olduğundan, metnin ilk karakterinden başlamak üzere Şekil 2.6.’da verilen karakterler aranır. Bulunan ilk karakter “B” karakteridir ve homoglif karşılığı ile değiştirilir. Damga verisinin ikinci bit değeri “0” olduğundan “i” karakteri olduğu gibi bırakılır. “l” karakteri Şekil 2.6.’da verilen dönüşümde kullanılacak karakterlerden biri olmadığı için olduğu gibi bırakılır. Damga verisinin üçüncü bit değeri “1” olduğu için “g” karakteri homoglif karşılığı ile değiştirilir. Benzer şekilde “e” ve “h” karakterleri olduğu gibi bırakılırken “a” karakteri homoglif karşılığı ile değiştirilerek “1” değerindeki damga biti gömülür. 6 bitlik verinin gömülmesi sonucu metnin sadece 3 karakteri homoglif dönüşüme uğramıştır. Damga verisi içerisindeki “1” bitlerinin sayısı kadar homoglif dönüşümün gerçekleştirilecek olması, yüksek miktarda damga verisinin gömüldüğü durumlarda bile damganın fark edilme olasılığını düşürecektir.

Şekil 2.7’ye dikkatle bakıldığında damga verisi olarak “1” gömülen “B”, “g” ve “a” karakterlerinden “g” karakterinin homoglif karşılığı, aynı metinde orijinal “g” karakteri olduğu durumda gözle ayırt edilebilecek derecede farklılık göstermektedir. Damgalama kapasitesinden ödün vermeden, önerilen yöntemin sağlamlığının artırılması için homoglif dönüşümde kullanılacak karakterler “algılanabilir” ve “algılanamaz” olarak iki gruba ayrılır. Algılanamaz gruba dâhil edilen karakterler, orijinal karakter ile arasında minimum noktasal fark olan karakterlerdir. Algılanabilir gruba dâhil edilen karakterler ise orijinal karaktere benzemesinin yanında, aynı metin içerisinde orijinal karakter olduğunda insan gözü tarafından ayırt edilebilenlerdir. Homoglif dönüşümde kullanılacak karakterlerin fark edilebilirlik durumuna göre gruplanması sonucu ortaya çıkan karakter grupları Şekil 2.8.’de verilmiştir.

Karakter	Algılanabilir	Algılanamaz
Orjinal	$%2345679DPQRUVbghru	!ABCEFGHJKLMNSTXYZacdeijopsvxyz
Homoglif	$ % 2 345б 7 9 DPQRUVbgһru	!ABCEFGHJKLMNSTXYZacdeijopsvxyz

Şekil 2.8. Homoglif dönüşümde kullanılan karakterlerin algılanabilirlik durumuna göre gruplanması

Gruplama sonucu 31 karakter algılanamaz, 20 karakter ise algılanabilir olarak belirlenmiştir. Toplamda homoglif dönüşüm için kullanılabilecek 51 karakter ve dönüşümde yerlerine geçecek olan Unicode karşılıkları Tablo 2.1.’de verilmiştir.

Tablo 2.1. Homoglif Değiştirme sürecinde kullanılan benzer unicode karakterler

Sembol	*Bit 0*	*Bit 1*	*Sembol*	*Bit 0*	*Bit 1*
	Orijinal Kod	*Homoglif Kod*		*Orijinal Kod*	*Homoglif Kod*
$	0x24	0xff04	a	0x61	0x430
%	0x25	0xff05	b	0x62	0x15af
!	0x21	0x1c3	c	0x63	0x3f2
A	0x41	0x391	d	0x64	0x501
B	0x42	0x392	e	0x65	0x435
C	0x43	0x421	g	0x67	0x261
D	0x44	0x13a0	h	0x68	0x4bb
E	0x45	0x395	i	0x69	0x456
F	0x46	0x3dc	j	0x6a	0x3f3
G	0x47	0x13c0	o	0x6f	0x3bf
H	0x48	0x397	p	0x70	0x440
J	0x4a	0x408	r	0x72	0x433
K	0x4b	0x39a	s	0x73	0x455
L	0x4c	0x216c	u	0x75	0x1d1c
M	0x4d	0x39c	v	0x76	0x1d20
N	0x4e	0x39d	x	0x78	0x445
P	0x50	0x3a1	y	0x79	0x443
Q	0x51	0xff31	z	0x7a	0x1d22
R	0x52	0x13d2	2	0x32	0xff12
S	0x53	0x405	3	0x33	0x417
T	0x54	0x3a4	4	0x34	0x13ce
U	0x55	0x144c	5	0x35	0x1bc
V	0x56	0x474	6	0x36	0x431
X	0x58	0x2169	7	0x37	0xff17
Y	0x59	0x3a5	9	0x39	0xff19
Z	0x5a	0x396			

Tablo 2.1 incelendiğinde, alfa nümerik karakterler dışında özel karakterlerin de yer aldığı görülecektir. Bu sayede metin ve nümerik alanlar üzerinde damgalama yapılabildiği gibi, metin alanın içerdiği özel karakterler de damgalama aşamasında kullanılabilmektedir.

Homoglif dönüşümde kullanılacak karakterler fark edilebilirlik durumuna göre gruplandığı için, damgalama aşamasında kullanılacak karakterlerin algılanabilirlik oranı da değişim gösterebilmektedir. Kullanıcıdan alınacak [0-1] aralığındaki Algılanabilirlik Oranı ile homoglif dönüşümde kullanılacak karakterler belirlenmektedir. Örneğin, kullanıcı Algılanabilirlik Oranını "0" olarak belirlemişse, homoglif dönüşüm aşamasında Şekil 2.8.'de verilen karakterlerden sadece algılanamaz karakter grubu kullanılacaktır. Algılanabilirlik oranı 1 değerine yakınsadıkça, Şekil 2.8.'de verilen algılanabilir karakter listesindeki karakterler de homoglif dönüşüm sürecine dahil edilir. Kullanıcı Algılanabilirlik Oranını "0.5" olarak belirlemişse, homoglif dönüşüm için algılanamaz karakter grubunun tamamı ve algılanabilir karakter grubunun yarısı (ilk 10 tanesi) kullanılacaktır. Kullanıcının seçeceği Algılanabilirlik Oranına göre dönüşümde kullanılacak karakterlerin belirlenmesinde (10) ifadesinden yararlanılmaktadır.

$$P = \{\%, \$, D, P, Q, R, U, V, b, g, h, r, u, 2,3,4,5,6,7,9\}$$
$$IP = \{!, A, B, C, E, F, G, H, J, K, L, M, N, S, T, X, Y, Z, a, c, d, e, i, j, o, p, s, v, x, y, z\}$$
$$HG = IP \parallel First\left(P, \left\lceil size(P) * P_p \right\rceil\right) \quad (10)$$

P ifadesi algılanabilir karakter kümesi, IP ifadesi algılanamaz karakter kümesi, P_p ifadesi kullanıcı tarafından belirlenen Algılanabilirlik Oranı ve $\parallel$ sembolü birleştirme işlemidir. $First(X, Y)$ metodu X kümesinden ilk Y adet elemanı bulmaktadır. Homoglif dönüşümde kullanılacak karakter kümesini simgeleyen HG; algılanamayan karakter kümesi IP ile kullanıcının belirlediği P_p algılanabilirlik oranında algılanabilir karakter kümesi P'den alınan karakterlerden oluşturulmaktadır.

Damga verisinin çıkarımı aşamasında yetkili kullanıcı tarafından damga yerleştirme aşamasında da kullanılan gizli anahtar ve algılanabilirlik oranı kullanıldığından, damga verisine yetkisiz kullanıcıların erişimi söz konusu değildir. Yöntemin geri dönüşüm özelliği sayesinde damga olarak yerleştirilen veri çıkarıldıktan sonra, orijinal veri de kayıpsız olarak elde edilebilmektedir.

Veri tabanından damgalama için kullanılacak kayıtların seçiminde (11) ifadesinden faydalanılmaktadır. Bu eşitlik ifadesinde özüt fonksiyonu $\mathcal{H}$, gizli anahtar değeri κ, birleştirme işlemi ise ‖ sembolü ile gösterilmiştir.

$$F(r.P) = \mathcal{H}\left(\kappa \parallel \mathcal{H}\left(\kappa \parallel (r.P)\right)\right) \tag{11}$$

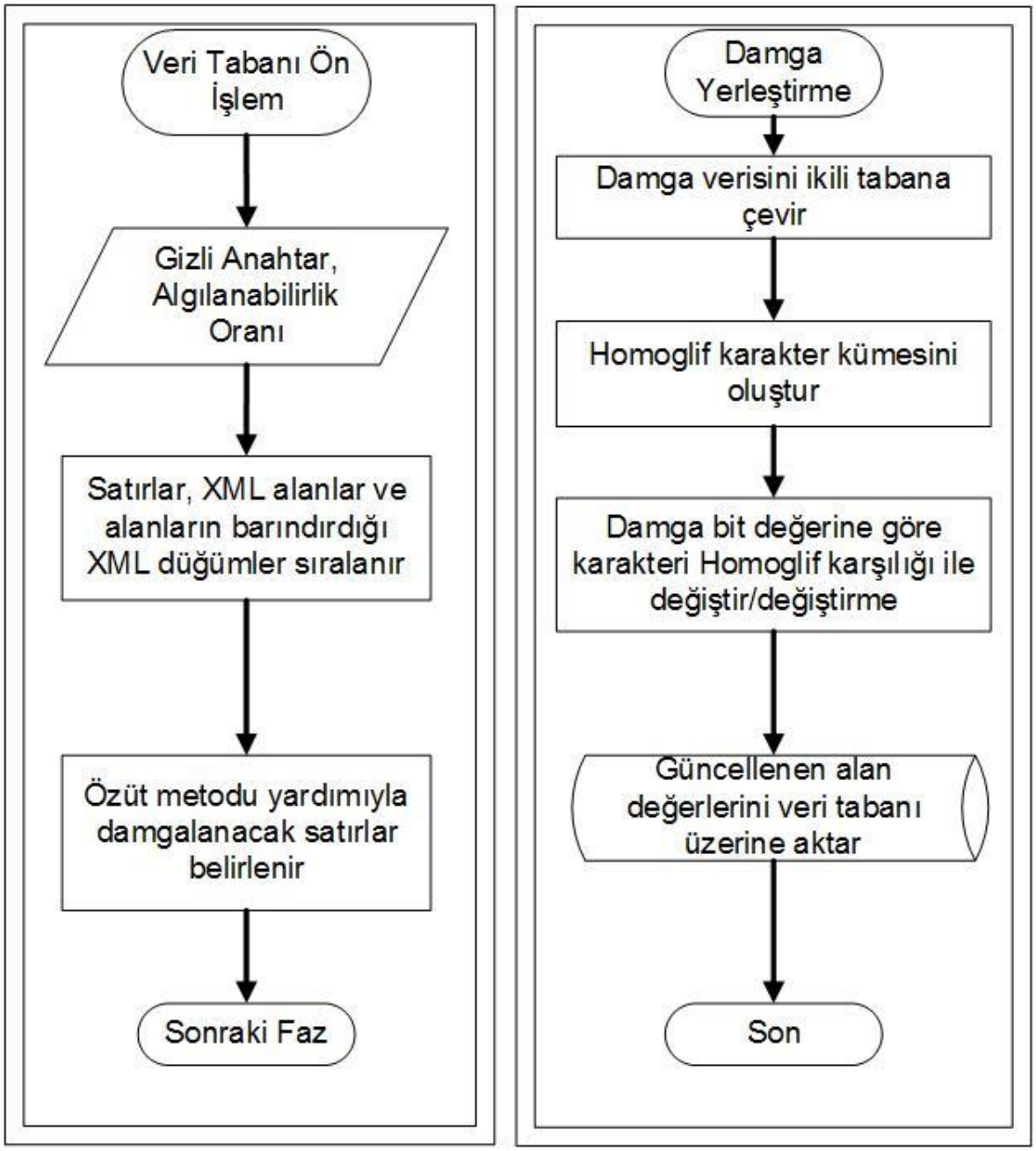
(11) ifadesindeki fonksiyonun üreteceği değer r ilişkisinde yer alan kaydın o anki birincil anahtar değeri P ve özel anahtar değeri κ ile doğrudan ilişkilidir. Bu sayede kullanıcı tarafından seçilen gizli anahtar değeri bilinmeden damganın bulunması mümkün değildir.

2.3.1. Damga Yerleştirme Algoritması

Damgalama sürecine ait akış şeması Şekil 2.9.'da verilmiştir. Süreç Ön İşlem ve Damga Yerleştirme fazlarından oluşmaktadır. Ön İşlem fazında ilk olarak kullanıcıdan damga yerleştirmede kullanılacak gizli anahtar ve algılanabilirlik oranı istenir. Gizliliğin önemli olduğu veri tabanlarında algılanabilirlik oranının 0'a yakınsaması, damgalama kapasitesinin yüksek olması istenen veri tabanlarında ise 1'e yakınsaması önerilmektedir. Bir sonraki adımda veri seti birincil anahtar değeri dikkate alınarak artan sırada sıralanmıştır. Ayrıca veri setinin XML alanları ve XML alanların düğümleri de alfabetik sıralı olacak şekilde yer değiştirilmiştir. Bu sayede veri setinin alanlarının veya XML verinin düğümlerine uygulanacak yer değiştirme saldırılarına karşı dayanıklılık sağlanmıştır. Bu yer değiştirme veri tabanı üzerinde fiziksel bir değişikliğe sebep olmayan, algoritma içerisinde alan indisini belirleyen sanal bir sıralamadır. Bu sayede ekleme ve alan değiştirme saldırılarına karşı önerilen yöntemin dayanıklılığı artırılmıştır.

Ön İşlem fazının bir sonraki adımında damga yerleştirilecek satırlar seçilecektir. Bu işlem için (12) ifadesinden yararlanılacaktır.

$$F(r.P)\ mod\ \kappa = 0 \tag{12}$$



Şekil 2.9. Damga yerleştirme algoritması

(12) ifadesindeki kriterin sağlanması durumunda, ilgili kayıtta veri gömme için kullanılacak XML alanların seçimi gerçekleştirilecektir. Veri tabanında birden çok XML alan olması durumunda, damgalamada kullanılacak alanın belirlenmesinde (13) ifadesi kullanılmaktadır.

$$F(r.P)\ mod\ v \tag{13}$$

(13) ifadesi ile $r.P$ birincil anahtarına sahip damgalanacak satır için damganın gömüleceği XML alan indisi bulunacaktır. ν simgesi satırın sahip olduğu XML alan sayısını belirtmektedir.

Damga yerleştirilecek XML alanların seçilmesiyle Ön İşlem fazı tamamlanmış olur. Damga Yerleştirme fazının ilk adımında damga verisi ikilik düzene dönüştürülür. Ardından kullanıcı tarafından belirlenen algılanabilirlik oranı P_p dikkate alınarak HG homoglif karakter kümesi elde edilir. Bir sonraki adımda damga yerleştirilecek XML alanın düğümleri gezilerek etiketler arasındaki metinler içerisinde Homoglif Dönüşüm için kullanılabilecek karakterler aranır. Damgalama algoritmasının yalancı kodunun verildiği Şekil 2.10'daki $Damgala$ metodu, kendisine parametre olarak verilen metin içerisinde HG homoglif karakter kümesinde yer alan karakterlerden herhangi birini bulduğunda, damga verisinin sıradaki bit değerini yerleştirmekle görevlidir. İncelenen veri HG karakter kümesindeki karakterlerden herhangi birini içeriyorsa, sıradaki damga biti dikkate alınarak karakter değeri Homoglif kodu ile değiştirilmekte (damga bit değeri "1" ise) veya değiştirilmeden aynı bırakılmaktadır (damga bit değeri "0" ise).

Giriş: R veri kümesi, gizli anahtar κ, veri alanları $\{A_1, A_2, \dots, A_v\}$, damga verisi W, algılanabilirlik oranı P_p
Çıkış: Damgalanmış veri R_w

for i=l to $SatırSayısı(R)$
  // damga yerleştirilecek satır seçimi
  if $F(R_i.P)\ mod\ \kappa = 0$
    // damgalamada kullanılacak XML alan indisi
    j = $F(R_i.P)\ mod\ v$
    // XML alanın tüm düğümleri için
    for k=0 to $DüğümSayısı(A_i)$
      $Damgala(A_i.Düğüm_k, W_t)$
    end
  end
end

Şekil 2.10. Damga yerleştirme algoritması

2.3.2. Damga Çıkarım Algoritması

Önerilen yöntem damga çıkarım esnasında damga bilgisinin yer aldığı veri tabanını, gizli anahtar değerini ve algılanabilirlik oranını kullanarak doğrulamayı gerçekleştirmektedir. Damga yerleştirme aşamasında damgalanacak satırların seçiminde kullanılan (12) ifadesi damga çıkarılacak satırların belirlenmesinde de kullanılacaktır.

Damga çıkarılacak satırların seçiminin ardından damganın yerleştirildiği XML veri alanı, damga yerleştirme sürecinde kullanılan (13) ifadesi yardımıyla belirlenmektedir. Bir sonraki aşamada kullanıcının belirlediği algılanabilirlik oranı dikkate alınarak Homoglif dönüşümde kullanılacak karakter kümesi oluşturulur. İlgili XML alanın tüm düğümlerindeki verinin içerdiği karakterler, oluşturulan Homoglif dönüşüm karakter kümesi ile karşılaştırılır. Homoglif dönüşüm karakter kümesinde yer alan karakterlerden herhangi birinin orijinal karakter değeri bulunduğunda çıkarılan damga biti "0", homoglif karşılığında yer alan karakter değeri bulunduğunda ise çıkarılan damga biti "1" olarak değerlendirilmektedir. Damga çıkarım sürecinin yalancı kodunun verildiği Şekil 2.11'de yer alan DamgaÇıkar metodunun gerçekleştirdiği bu işlem sonucunda çıkarılan damga bitleri birleştirilerek damgalama aşamasında yerleştirilen damga verisi elde edilmektedir.

Giriş: Damgalanmış veri R_w, gizli anahtar κ, veri alanları $\{A_1, A_2, \dots, A_v\}$, algılanabilirlik oranı P_p
Çıkış: Damga verisi W
for i=l to $SatırSayısı(R)$
  if $F(R_i.P)\ mod\ \kappa = 0$
    // damga çıkarımında kullanılacak alan indisi
    j = $F(R_i.P)\ mod\ v$
    // XML alanın tüm düğümleri için
    for k=0 to $DüğümSayısı(A_i)$
      $W.Add(DamgaÇıkar(A_i.Düğüm_k))$
    end
  end
end

Şekil 2.11. Damga çıkarma algoritması

2.4. Ateş Böceği Algoritması ile İlişkisel Veri Tabanı Damgalamada Yeni Bir Yaklaşım

Günümüzde Gezgin Satıcı Problemi gibi optimizasyon problemlerinin çözümünde biyolojik olarak esinlenen algoritmaların kullanımı popüler hale gelmiştir. Birbirinden etkilenen birden çok ajanın davranışı bu algoritmaların temelini oluşturmaktadır. Balık, kuş, karınca gibi sürülerin davranışlarını taklit ettiklerinden bazen Sürü Zekâsı tabanlı algoritmalar olarak isimlendirilmektedirler. 1995 yılında önerilen Parçacık Sürüsü Optimizasyonu balık ve kuşların davranışlarını kullanmıştır [67]. Karınca kolonisi optimizasyonu [68] ve yapay arı kolonisi optimizasyonları [69] da bu alandaki örneklerdendir.

Yang vd. 2008 yılında NP-Hard zorluğundaki problemleri çözebilmek için Ateş Böceği algoritmasını önermişlerdir [70]. Algoritmaları ateş böceklerinin davranışları ve ışık yaymaları üzerine kuruludur. Doğada yaklaşık 200 farklı çeşitte bulunan ateş böceklerinin her bir türü farklı ışık yayma karakteristiğine sahiptir. Bu karakteristik aşağıdaki 3 kural ile özetlenebilir,

Kural-1: Her bir ateş böceği, cinsiyetinden bağımsız olarak diğer ateş böceklerini çekebilir. Bu durum ateş böceklerinin unisex olduğu anlamına gelir.

Kural-2: Ateş böceğinin yaydığı ışığın parlaklığı aynı zamanda onun çekiciliğini belirlemektedir. Eğer iki ateş böceği ışık yayıyorsa, az ışık yayan ateş böceği kendisine nazaran daha fazla ışık yayan ateş böceğine doğru hareket edecektir. En parlak ateş böceğinin hareketi ise rastgele olacaktır.

Kural-3: Algoritmanın amaç fonksiyonu ateş böceğinin parlaklığını belirlemektedir.

Belirli bir x konumundaki ateş böceğinin I parlaklık değeri $I(x) \propto f(x)$ olarak seçilir. Ateş böceğinin β çekiciliği diğer ateş böcekleri tarafından değerlendirilir. Işık yoğunluğunun kaynağa olan uzaklıkla azalmasından dolayı ateş böcekleri arasındaki uzaklık çekiciliği değiştirmektedir. Işık yoğunluğu $I(r)$ en basit biçimde (14)'de verilmiştir. Işık kaynağındaki yoğunluğunun I_s ile gösterildiği (14) ifadesinde, mesafenin karesi ile ışık yoğunluğu ters orantılıdır.

$$I(r) = \frac{I_s}{r^2} \tag{14}$$

Başlangıç ışık yoğunluğu I_0, ortam için sabit ışık absorve katsayısı γ ile gösterildiğinde, ışık şiddeti (15) ifadesinde verildiği şekilde değişecektir.

$$I = I_0 e^{-\gamma r^2} \tag{15}$$

Işık yoğunluğu diğer ateş böcekleri için çekiciliği belirlemektedir. Ateş böceğinin çekiciliği (16) ifadesinde β ile verilmiştir. β_0 başlangıç pozisyonundaki çekiciliktir.

$$\beta = \beta_0 e^{-\gamma r^2} \tag{16}$$

Her bir ateş böceğinin d boyutlu uzayda bir noktada bulunduğu varsayıldığında, i ve j sıralı iki farklı ateş böceğinin koordinatlı x_i ve x_j ile ifade edilir. Bu iki ateş böceğinin arasındaki r_{ij} Kartezyen uzaklığı (17) ifadesindeki şekilde hesaplanır.

$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{17}$$

d boyutlu uzayda bir ateş böceğinin daha çekici başka bir ateş böceğine doğru hareketi (18) ifadesinde verilmektedir.

$$x_i = x_i + \beta_0 e^{-\gamma} r_{ij}^2 (x_j - x_i) + \alpha \varepsilon_i \tag{18}$$

(18) ifadesindeki ikinci kısım çekiciliği, üçüncü kısım ise rastgele belirlenen hareket özelliğini göstermektedir. Rastgeleliği sağlayan faktör α ile, Gauss dağılımdan seçilen rastgele sayıların oluşturduğu vektör ise ε_i ile gösterilmiştir. (18) ifadesinden ateş böceklerinin basit rastgele hareketler gerçekleştirdiği de anlaşılabilmektedir.

Çalışmada geliştirilen damgalama yöntemine Ateş Böceği Algoritmasını kullandığı için FFADEW ismi verilmiştir. Yöntem Ateş Böceği Algoritmasını kullanarak bozulmayı minimize ederken sağlamlığı artıracak yeni bir geri dönüşümlü damgalama algoritması önerilmektedir. 2008 yılında Yang [70] tarafından önerilen ateş böceği algoritması damganın yerleştirileceği kayıtların en doğru alanlarını belirlemekte kullanılmaktadır. Ateş böceği popülasyonundaki her bir birey damga yerleştirmede kullanılabilecek bir aday

çözümü göstermektedir. Popülasyonun en parlak ışığına sahip ateş böceği ise iterasyonun en iyi çözümüdür.

Işık ateş böcekleri için çekim sebebi olduğundan, diğer ateş böcekleri en parlak ışığa sahip olan ateş böceğine doğru yöneleceklerdir. Algoritmanın sonlanma kriterlerinden herhangi biri gerçekleştiğinde, elde edilen en parlak ateş böceği damgalama işlemine yardımcı olacak bilgiye sahiptir.

Ateş böcekleri damgalamada kullanılacak aday satırların damga yerleştirilecek alan indis değerleri ile oluşturulan noktalı sayılardan oluşmaktadır. Deneysel sonuçlar, önerilen yöntemin GADEW [30] ile karşılaştırıldığında literatürde sıklıkla kullanılan ataklara karşı daha dayanıklı, damgalama sonrası orijinal veri üzerinde ise en düşük düzeyde değişikliğe sebep olduğunu göstermiştir. Çalışma süreleri de incelendiğinde önerilen çalışmanın karmaşıklığının tüm algoritmalardan daha düşük olduğu görülmektedir.

Önerilen yöntem damga yerleştirme ve damga çıkarım algoritmalarından oluşmaktadır. Damga yerleştirme algoritması özel olarak seçilmiş satırlara Fark Genişleme Tabanlı Damgalama yöntemini kullanarak damga bilgisini yerleştirmektedir. Ateş böceği algoritması, seçilmiş satırların hangi alanlarının damgalama için daha uygun olduğunu belirlemektedir. Damga çıkarım algoritması ise, veri tabanından yerleştirilmiş damga verisini çıkararak orijinal damga verisi ile karşılaştırmaktadır. Bu algoritma aynı zamanda damga çıkarım sonrası veri tabanını damgalama öncesindeki orijinal haline geri döndürmektedir.

2.4.1. Damga Yerleştirme Algoritması

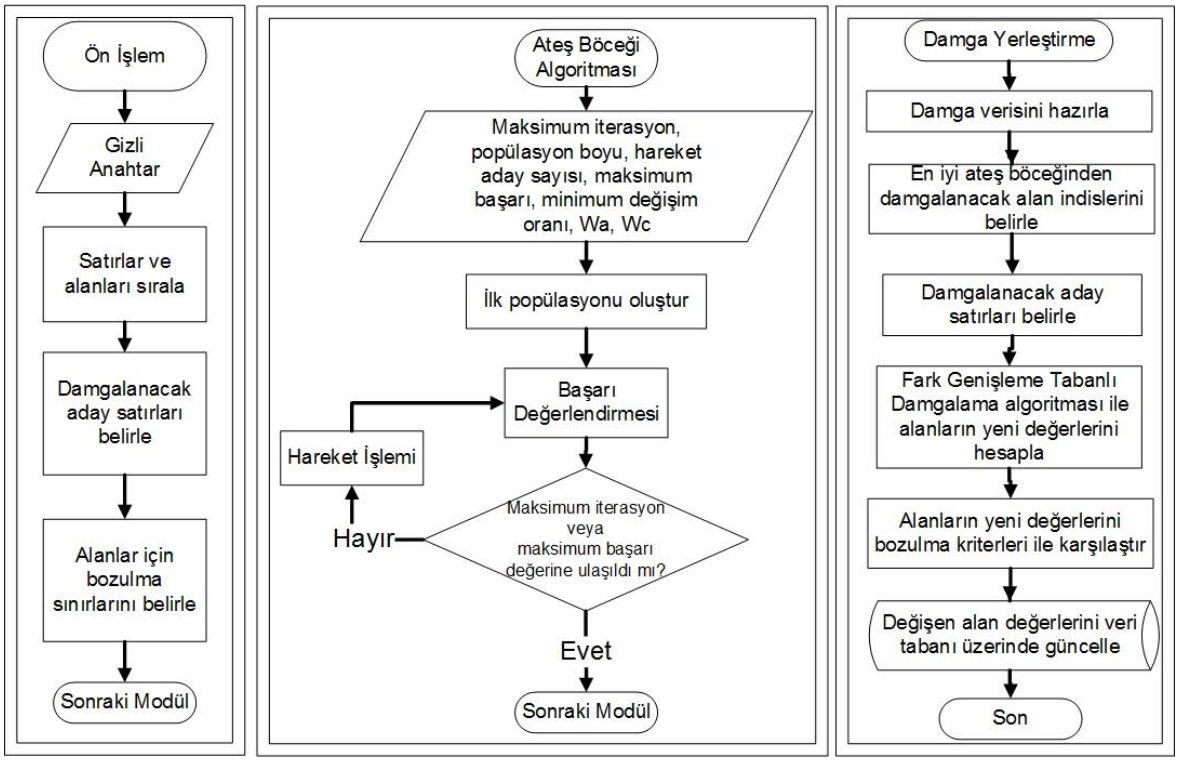
Damga yerleştirme algoritmasını oluşturan Ön İşlem, Ateş Böceği Belirleme, Damga ekleme modülleri Şekil-2.12'de verilmiştir.

Ön İşlem Modülü: Bu modül veri tabanının damgalanma öncesi hazırlık süreçlerini içerir. Veri tabanı kayıtları birincil anahtar dikkate alınarak sıralanır. Alanlar ise benzer şekilde alan adları dikkate alınarak alfabetik olarak sıralanır. Bu sıralama adımları sayesinde algoritma satır/sütun sıra değiştirme saldırılarına karşı dayanıklı olacaktır.

Bir sonraki adım damgalanacak aday satırların belirlenmesidir. Bu işlem için özüt metodu kullanılacaktır. Damgalanmak üzere seçilen aday satırlardan oluşan veri seti DS olarak ifade edilecektir. Seçim işleminde kullanılan kriterin verildiği (19) ifadesinde kullanıcı tarafından belirlenen gizli anahtar κ, ilgili kaydın birinci anahtarı $r.P$, metin

birleştirme operatörü || ve 512 bit özüt metodu (SHA-512) ise $\mathcal{H}$ ile temsil edilmektedir. γ değerinin seçilecek kayıt sayısını belirlediği (19) ifadesinde verilen şartı sağlayan kayıtlar DS veri setini oluşturacaktır.

$$mod(\mathcal{H}(\kappa||\mathcal{H}(r.P||\kappa),\gamma) \equiv 0 \tag{19}$$



Şekil 2.12. Damga yerleştirme algoritması (a) Ön İşlem, (b) Ateş Böceği Belirleme, (c) Damga Yerleştirme

(19) ifadesine uyan kayıtlardan oluşan DS veri setinin M adet satır ve A adet alandan oluştuğu varsayıldığında, damga yerleştirme sürecinde her bir alan için uyulması gereken bozulma sınır aralığı (20) ifadesinde verilmiştir.

$$\left(DT_x^{min}, DT_x^{max}\right), x = 1 \ldots A \tag{20}$$

FFADEW, kullanıcı tarafından bozulma sınır aralıkları belirlenmediği durumlarda orijinal veri tabanındaki her bir alanın sahip olduğu en büyük ve en küçük değeri ilgili alan için bozulma sınır aralığı olarak kabul etmektedir.

Ateş Böceği Belirleme Modülü: Bu adımda, Ön İşlem Modülünde seçilen aday satırların damga yerleştirilecek en iyi alan çiftlerini belirlemek için Ateş Böceği Algoritması kullanılır. Ön İşlem Modülünden veri setini girdi olarak alan bu modül, seçilen satırların damgalamada kullanılacak alan indis bilgilerini içeren ateş böceğini çıktı olarak vermektedir. Ateş Böceği Belirleme Modülü aşağıdaki detayları verilen 2 fazdan oluşmaktadır.

Faz 1 (Başlangıç Popülasyonun Oluşturulması) Algoritma $\mathrm{F}^1 \ldots \mathrm{F}^{\mathrm{P}}$ ifadesi ile temsil edilen P adet ateş böceği barındıran ilk popülasyonu oluşturmaktadır. Popülasyondaki her bir ateş böceği, DS veri seti için örnek bir çözüm oluşturan noktalı sayı dizisidir. Dizinin her bir değeri damgalanmak üzere seçilen satırın damga yerleştirilecek alan değerlerinin indis bilgisini içermektedir. Noktalı sayının tam kısmı seçilen ilk alan indisini, ondalık kısmı ise seçilen ikinci alan indisini göstermektedir. Örneğin 10 alandan oluşan bir veri tabanı için optimizasyon algoritması tarafından belirlenen çözümü barındıran ateş böceğinin ilk elemanı 7.2 değerine sahipse, damga yerleştirmek üzere seçilen ilk satırın 7 ve 2 indeksli alan değerleri kullanılarak damgalama yapılacak demektir. M adet noktalı sayı içeren j. aday çözümün içerdiği değerler (21) ifadesinde verilmiştir.

$$F^j = \left\{ F_i^j \middle| F_i^j = x.y, x \in [1 \ldots A], y \in [1 \ldots A], x \neq y, i \in [1 \ldots M] \right\} \quad (21)$$

Ateş böceğinin her bir elemanın sahip olabileceği en büyük değer $\mathrm{A}.(\mathrm{A}-1)$ şeklindedir. Bu değer ilgili satırın A ve A-1 indeksli alanlarının damga yerleştirmek üzere seçileceğini belirtir.

İlk popülasyonun oluşturulmasını sağlayan işlem adımları aşağıdaki gibidir.

Adım 1 $j \in [1 \ldots P]$ değerleri için Adım 2'yi tekrarla

Adım 2 $i \in [1 \ldots M]$ değerleri için Adım 2.1'den Adım 2.6'ya kadar tekrarla

Adım 2.1 DS veri setinin k. satırının rastgele bir alanını seç, At_x

Adım 2.2. $y \in [1 \ldots A]$ değerleri için Adım 2.3 ve Adım 2.4 tekrarla

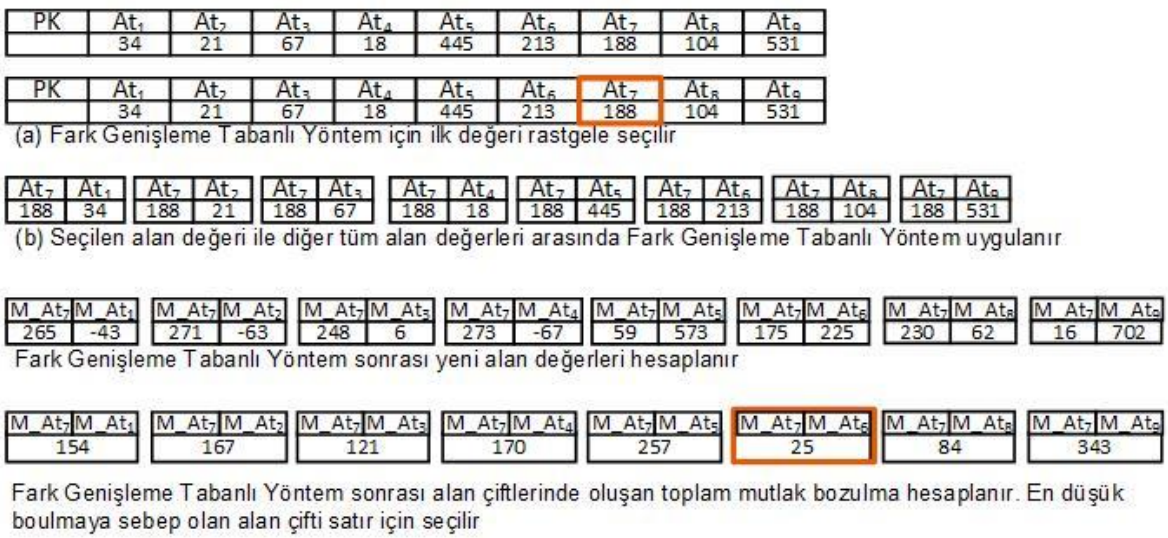
Adım 2.3 DS veri setinin k. satırının başka bir alanını seç, $At_y, x \neq y$

Adım 2.4 At_x ve At_y alan değerlerine DEW algoritmasını uygula $\left(M\_At_x, M\_At_y\right) = DEW\left(At_x, At_y, w\right)$

Adım 2.5 DEW ile damga yerleştirme sonrasında $|M\_At_x - At_x| + |M\_At_y - At_y|$ ifadesini kullanarak en az bozulmaya sebep olan (At_x, At_y) çiftini seç

Adım 2.6 Bu çiftin indeks değerleri, ilk popülasyonun j. bireyinin i. elemanına karşılık gelen $F_i^j = (x.y)$ değeri oluşturur. Bu ondalık sayının tam kısmını x, ondalık kısmını da y değeri oluşturur.

İlk popülasyonu oluştururken kullanılan uygunluk fonksiyonu, alan değerlerine damga yerleştirmede kullanılan Fark Genişleme Tabanlı Damgalama yönteminden dolayı alan değerinde oluşacak bozulmayı minimize etmeyi ve damgalama kapasitesini artırmayı hedeflemektedir. Bu sebeple, her bir satır için Fark Genişleme Tabanlı Damgalama sonrası en az değişikliğe sebep olacak alan çiftlerini seçer. Şekil2.13 örnek bir aday satır için ilk popülasyonda kullanılacak alan çiftlerinin seçimini göstermektedir.



Şekil 2.13. Örnek ilk popülasyon oluşumu

Şekil 2.13'de DS veri setinin i. satırı görülmektedir. Algoritmanın rastgele seçtiği (At_7) alanı ile diğer alan değerleri arasında Fark Genişleme Tabanlı Damgalama algoritması uygulanmıştır. Algoritma sonrası alanların sahip olacağı yeni değerleri ile algoritma öncesi değerleri arasındaki mutlak fark ilgili alan çifti seçildiğinde sebep olunacak olan bozulmayı göstermektedir. $(\mathrm{At}_7, \mathrm{At}_6)$ alan çiftleri arasındaki Fark Genişleme Tabanlı Damgalama işlemi en düşük değişime sebep olduğundan ilgili ateş böceğinin i. elemanındaki değer 7.6 olacaktır.

Faz 2 En iyi Ateş Böceğinin Belirlenmesi. Bu fazda, FFADEW algoritması bir önceki fazda oluşturulan ilk popülasyondaki ateş böcekleri arasından en parlağını belirlemektedir. Popülasyondaki diğer ateş böcekleri en parlak ateş böceğine doğru hareket ederken, en parlak ateş böceği ise rastgele olarak konumunu değiştirecektir. En parlak ateş böceğinin belirlenmesini sağlayan kod parçacığı aşağıdaki gibidir.

*En iyi ateş böceğinin belirlenmesi

Adım 1. $k = 1 \ldots \max_{iteration}$ değerleri için Adım-2'den Adım-3'e kadar tekrar et

Adım 2. $i = 1 \ldots P$ değerleri için Adım 2.1'den Adım 2.3'e kadar tekrar et

Adım 2.1. $j = 1 \ldots M$ değerleri için Adım 2.1.1'den Adım 2.1.5'e kadar tekrar et

Adım 2.1.1. *DS* veri setinden j. satırı çıkar, R^j.

Adım 2.1.2. $x = TamKısım\left(F_j^i\right), y = OndalıkKısım\left(F_j^i\right)$

Adım 2.1.3. $At_x = R_x^j, At_y = R_y^j$

Adım 2.1.4. $\left(M\_At_x, M\_At_y\right) = DEW\left(At_x, At_y, w\right)$

Adım 2.1.5. $If\left(DT_x^{min} \leq M\_At_x \leq DT_x^{max}\right) \wedge \left(DT_y^{min} \leq M\_At_y \leq DT_y^{max}\right)$

$$mdf_x = mdf_x + \left(M_{At_x} - At_x\right)$$

$$mdf_y = mdf_y + \left(M\_At_y - At_y\right)$$

$$row_w = row_w + 1$$

Adım 2.2. $Br^i = ((\sum_{t=1}^{A} |mdf_t|)/tuple_{count})w_a + (row_w/M)w_c$

Adım 2.3. $if\ Br^i \geq t_{end}$

İncelenen ateş böceği F^i'yi en iyi ateş böceği olarak ata F^{best} ve Adım 4'e geri dön.

Adım 3. *Hareket İşlemi'*ni gerçekleştir

Adım 4. En iti ateş böceğini F^{best} bir sonraki faza gönder (Damga Yerleştirme).

Ateş böceği belirleme modülünün ikinci fazının 2 sonlandırma kriteri kod parçacığının Adım 1 ve Adım 2.3'de verilmiştir. Adım 1'de belirtilen sonlandırma şartı iterasyon sayısını kontrol etmektedir. Algoritma belirlenen maksimum iterasyon $\max_{\text{iteration}}$ sayısına ulaştığında sonlanmakta ve bulduğu en iyi ateş böceğini bir sonraki aşamaya göndermektedir. Adım 2.3'de verilen ikinci sonlandırma kriteri damgalanmış satır sayısı ve veri tabanı üzerinde oluşan bozulmayı dikkate almaktadır. İncelenen ateş böceğinin parlaklık değeri, daha önce belirlenen sınır değeri $\mathrm{t_{end}}$'den büyük olması durumunda algoritma

sonlanarak incelenen ateş böceğini en parlak ateş böceği olarak bir sonraki aşamaya taşımaktadır.

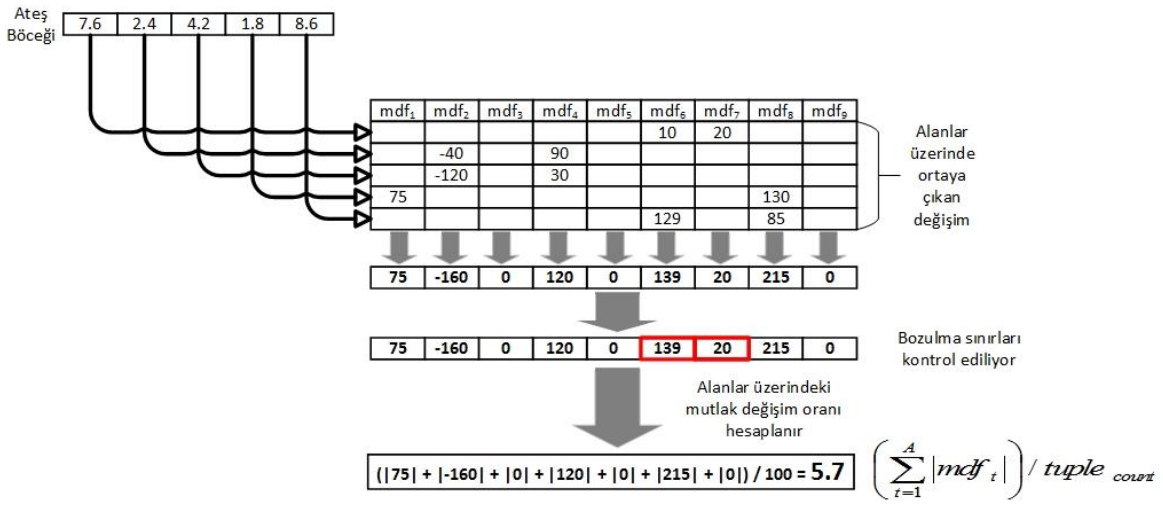
Kod parçacığında görüldüğü üzere bu faz parlaklık değerinin belirlenmesini ve ateş böceği hareketini sağlayan adımlara sahiptir. F^i ateş böceğinin Br^i parlaklık değerinin belirlenmesine ait işlem adımlarının detayları aşağıda verilmiştir.

$\text{TamKısım}(F_j^i)$ ve $\text{OndalıkKısım}(F_j^i)$ metotları i. ateş böceğinin j. elemanını parametre alarak bu değerin tam sayı kısmını ve ondalık sayı kısmını geri döndürmektedir. Elde edilen bu değerler, popülasyon oluşturma aşamasında belirlenen damgalamada kullanılan alan indisleridir. Adım 2.1.4'de verilen $\text{DEW}(At_x, At_y, w)$ metodu x ve y indislerinde bulunan At_x ve At_y alan değerlerine Fark Genişleme Tabanlı Damgalama algoritmasını kullanarak 1 bit damga verisini gömmekte ve yeni alan değerleri olan (M_At_x, M_At_y) değerlerini elde etmektedir.

Adım 2.1.5'de Fark Genişleme Tabanlı Damgalama sonrası alan değerlerinde oluşan bozulma kontrol edilmektedir. Fark Genişleme Tabanlı Damgalama sonrası alanın yeni değeri ilgili alan için belirlenen bozulma sınırları arasında kalıyorsa, veri tabanının alan bazında oluşan toplam bozulmayı tutan $mdf_x, x = 1 \ldots A$ değerlerine işlem gören alandaki değişim eklenmektedir.

Başarıyla damga yerleştirilmiş satır sayısını row_w değeri tutmaktadır. Fark Genişleme Tabanlı Damgalama sonrası yeni alan değerleri belirlenen sınırlar içerisinde kalmıyorsa, ateş böceğinin bu elemanına damgalama yapılmayacaktır. Dolayısıyla bu elemanın ateş böceğinin parlaklık değerinde herhangi bir katkısı olmayacaktır. Bu durumda algoritma Adım 2.1'e dönerek bir sonraki eleman için adımlarını tekrar edecektir.

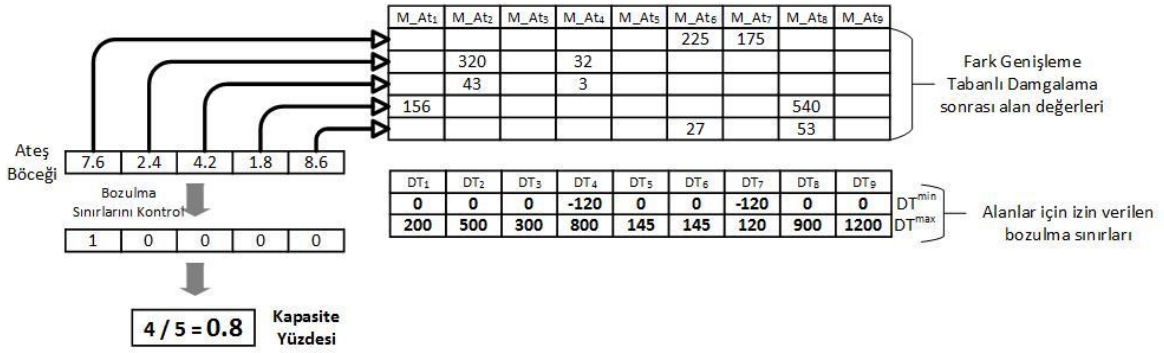
Adım 2.2'de algoritma her bir ateş böceğine bir parlaklık değeri hesaplamaktadır. Hesaplanan parlaklık değeri 2 faktörden etkilenmektedir; alan değerleri üzerinde damgalama sonrası oluşan toplam değişim ve veri setinde damgalanabilmiş satır sayısı. İlk faktör alanlar üzerinde toplam değişimi tutan $mdf_x, x = 1 \ldots A$ değeri ile veri tabanındaki toplam kayıt sayısı $tuple_{count}$ arasındaki orandır. İkinci faktör ise damgalanabilmiş satır sayısı ile damgalanmak üzere seçilen aday satırlardan oluşan veri setinin satır sayısının oranıdır. Bu iki faktör w_a ve w_c değerleri ile ağırlıklandırılarak ilgili ateş böceğinin parlaklık değerinin hesaplanmasını sağlamaktadır. Ateş böceğinin parlaklık değerini etkileyen ilk faktörün $\left(\left(\sum_{t=1}^{A} |mdf_t|\right)/tuple_{count}\right)$ örnek bir ateş böceği için hesabı Şekil 2.14'de verilmiştir.



Şekil 2.14. Parlaklık fonksiyonunun ilk kısmının örnek hesabı

Şekil 2.14'deki örnek ateş böceğinin ilk elemanının belirlenmesine ait örnek işlem adımları Şekil 2.13'de verilmişti. Örnekte damgalanmak üzere seçilen veri setinin 5 satırdan oluştuğu varsayılmaktadır. Bu sebeple ateş böceğinin eleman sayısı 5'dir. Her bir elemanın tam ve ondalık kısımlarının indekslediği alanlar arasında Fark Genişleme Tabanlı Damgalama algoritması sonrası alanlar üzerinde oluşan toplam değişim Şekil 2.14'deki tabloda verilmiştir. Veri tabanı kayıt sayısının 100 olarak alındığı örnekte, toplam değer değişimi sınırları içerisinde kalan elemanlar için parlaklık değerinin hesaplanmasında kullanılan ilk faktör 5.7 olarak bulunacaktır.

Ateş böceğinin parlaklık değerini etkileyen ikinci faktörün (row_w/M) örnek hesabı Şekil 2.15'de verilmiştir. Ateş böceğinin eleman değerlerinin tam ve ondalık kısımlarının indislendiği alanların Fark Genişleme Tabanlı Damgalama sonrası değerleri Şekil 2.15'deki tabloda verilmiştir. Alanların yeni değerleri, ilgili alanlar için izin verilen değer aralıkları ile karşılaştırıldığında, ateş böceğinin gösterdiği 5 aday satırdan 4'nün damgalanabileceği görülmektedir. Sadece ilk aday satırın damgalanması sonrası yeni alan değerlerinin ilgili alanlar için izin verilen sınır değerlerin dışında olduğu görülmektedir. Bu sebeple ateş böceğinin "7.6" değer ile temsil ettiği aday satır damga yerleştirmede kullanılmayacaktır. Şekil 2.15'de verilen örnek çözümü gösteren ateş böceğinin parlaklık değerini etkileyen ikinci faktörün değeri 0.8 olacaktır.



Şekil 2.15. Parlaklık fonksiyonunun ikinci kısmının örnek hesabı

Popülasyondaki her bir ateş böceğinin parlaklık değeri hesaplandıktan sonra, popülasyonun en parlak ateş böceği belirlenmektedir. Diğer ateş böcekleri F^{best} ile ifade edilen popülasyonun en parlak ateş böceğine doğru hareket edeceklerdir. En parlek ateş böceğine doğru hareket edecek ateş böceğinin eleman değerleri ile en parlak ateş böceğinin eleman değerleri arasındaki farklılıklar hareket işleminin temelini oluşturur. Hareket işleminin temel görevi, iki ateş böceğine karşılık gelen vektörlerin arasındaki farklılıkları minimize etmektir.

i. ateş böceği F^i ile en iyi ateş böceği F^{best} arasındaki uzaklığın hesabı (22) ifadesinde verilmiştir. Her bir ateş böceğinin M elemandan oluştuğu varsayıldığında, iki ateş böceği arasındaki uzaklık değeri $D(F^i, F^{best})$, $[0 \dots M]$ aralığında bir değer olacaktır.

$$D(F^i, F^{best}) = \sum_{k=1}^{M} (F_k^i \neq F_k^{best}) \qquad (22)$$

Hareket işlemini gerçekleştirecek algoritma, F^i ateş böceğini popülasyonun en parlak ateş böceği F^{best}'e doğru hareket ettirmek için F^i'nin rastgele r adet elemanını seçer. Rastgele olarak belirlenen bu r değeri $r \in [1 \dots D(F^i, F^{best})]$ ve $r/M \geq t_m$ şartlarını sağlamalıdır. t_m değişiklik oranı hareket edecek ateş böceğinin en iyi ateş böceğine yaklaşma (benzeme) oranını belirten $[0 \dots 1]$ aralığında değerdir. Popülasyondaki ateş böceklerinin en iyi ateş böceğine doğru hareketini gerçekleştiren *Hareket İşlemi*'nin adımları aşağıda verilmiştir.

*Hareket İşlemi

Adım 1. $i = 1 \dots P, i \neq best$ değerleri için Adım 2'den Adım 4'e kadar tekrar et

Adım 2. İlgili ateş böceği ile en iyi ateş böceği arasındaki uzaklığı hesapla (F^i, F^{best}).

Adım 3. $j = 1 \ldots c_{move}$ değerleri için Adım 3.1'den Adım 3.4'e kadar tekrar et

Adım 3.1. İlgili ateş böceğini geçici ateş böceği olarak sakla $Ftemp^j$.

Adım 3.2. $r \in \left[1 \ldots D\left(Ftemp^j, F^{best}\right)\right]$ ve $r/M \geq \mathrm{t_m}$ şartlarına uyan rastgele bir sayı oluştur.

Adım 3.3. $Ftemp^j$ ateş böceği içerisindeki rastgele seçilen r elemanı en iyi ateş böceği F^{best}'in aynı indeksli elemanı ile değiştir.

Adım 3.4. $Ftemp^j$ ateş böceğinin parlaklığını hesapla ve TB^j dizisine sakla.

Adım 4. *TB* dizisindeki en parlak ateş böceğini F^i ile değiştir.

$\mathrm{t_m} = 0.4$ (%40 değişim oranı) değeri için örnek bir Hareket İşlemi Şekil 2.16'da verilmiştir. Popülasyonun 3 ateş böceğinden oluştuğu varsayılmaktadır. Ateş böceklerinin parlaklık değerleri hesaplandığında 4.431 değer ile üçüncü ateş böceğinin popülasyonun en parlak ateş böceği olduğu görülmektedir. Diğer 2 ateş böceğinin elemanları ile en parlak ateş böceğinin elemanları karşılaştırıldığında, ortak elemanları olmadığından, her 2 ateş böceğinin de en parlak ateş böceğine olan uzaklığı 5 olarak hesaplanır.

İlk ateş böceğinin en parlak ateş böceğine hareketi için rastgele oluşturulan r değeri 1, ikinci ateş böceği için rastgele oluşturulan r değeri ise 3'dür. Hareket operasyonunun Adım 3.2'deki $\mathrm{r/M} \geq \mathrm{t_m}$ şartı ilk ateş böceği için sağlanmadığından $(1/5 \not\geq 0.4)$, ilk ateş böceği için yeniden rastgele r değeri oluşturulur. Yeni oluşturulan $\mathrm{r} = 2$ değeri şartları sağladığından değişim işlemine geçilir. İlk ateş böceğinin rastgele belirlenen 2 eleman değeri en iyi ateş böceğinde yer alan karşılıkları ile, ikinci ateş böceğinin rastgele belirlenen 3 eleman değeri yine en iyi ateş böceğinde yer alan karşılıkları ile değiştirilerek ilgili ateş böceklerinin en iyi ateş böceğine doğru hareket etmesi sağlanmıştır.

Şekil 2.16. Örnek hareket operasyonu

Popülasyondaki tüm ateş böceklerinin en iyi ateş böceğine doğru hareketinden sonra, en iyi ateş böceği de kendi hareketini gerçekleştirecektir. En iyi ateş böceği için Hareket Operasyonuna ait işlem adımlarında sadece Adım 3.3 değişiklik gösterecektir. En iyi ateş böceğinin hareket operasyonuna ait Adım-3.3 aşağıdaki şekildedir.

Adım-3.3 En iyi ateş böceğinin elemanları arasından rastgele seçilen r adet eleman, rastgele olarak üretilen x.y formundaki değerler ile değiştirilir. Rastgele üretilen bu x.y değerleri $x \in [1 \ldots A-1], y \in [1 \ldots A-1], x \neq y$ şartlarını sağlamalıdır. Yeni üretilen ateş böceği $Ftemp^j$ olarak saklanır.

Damga Yerleştirme Modülü: Bu modülde Ateş Böceği Belirleme Modülü sonucunda elde edilen F^{best} en iyi ateş böceğinin eleman değerleri dikkate alınarak damga yerleştirme işlemi gerçekleştirilir. Sürece ait işlem adımları aşağıda verilmiştir.

Giriş: *DS*, $W = w_1 \ldots w_n$, F^{best}
Çıkış: Damga Yerleştirilmiş *DS, DS'*
Adım 1. $j = 1 \ldots M$ değerleri için aşağıdaki adımları tekrar et
Adım 1.1. *DS* veri setinin j. satırını oku, R^j
Adım 1.2. $x = TamKısım\left(F_j^{best}\right), y = OndalıkKısım\left(F_j^{best}\right)$
Adım 1.3. $At_x = R_x^j, At_y = R_y^j$
Adım 1.4. $\left(At_x, At_y\right) = DEW\left(At_x, At_y, w_i\right)$
Adım 1.5. $i = i + 1$

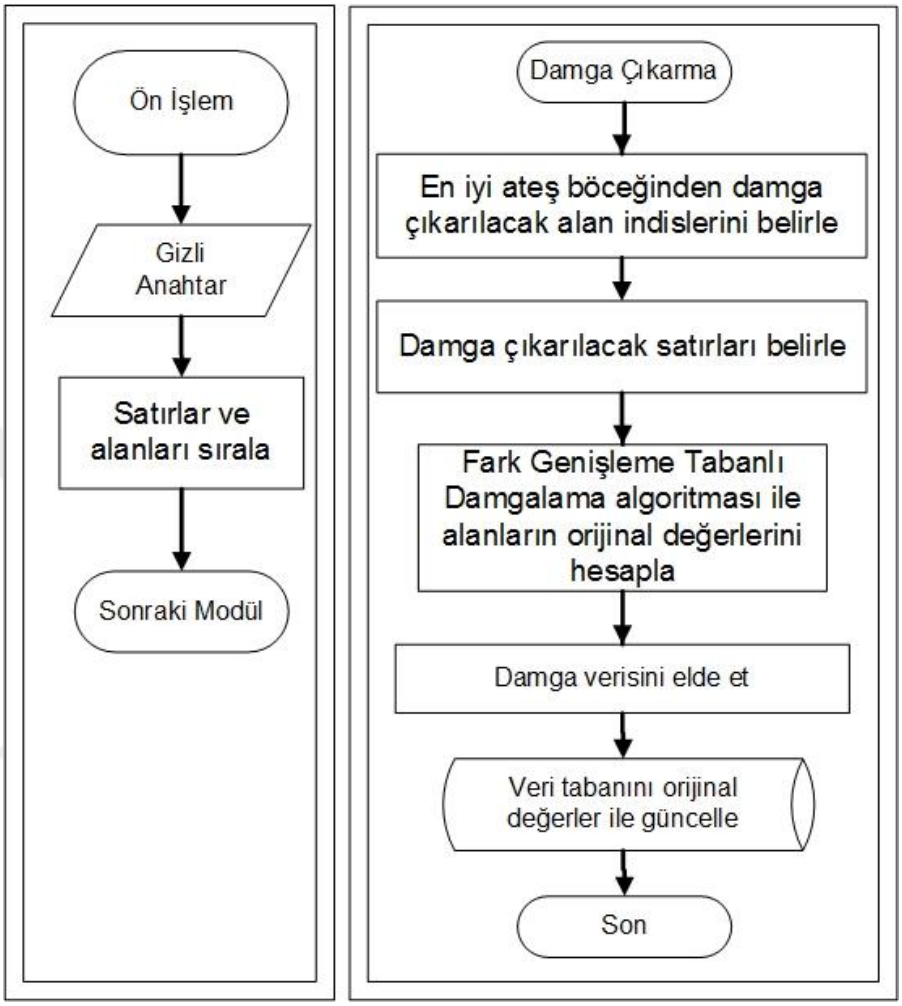
Adım 1.2'de ateş böceğinin eleman değerleri, seçili veri setindeki ilgili satırın damga yerleştirilirken kullanılan alan indislerini belirlemektedir. Gösterilen alan değerleri üzerine Fark Genişleme Tabanlı Damgalama algoritması kullanılarak damga biti yerleştirilmesi Adım 1.4'de gerçekleştirilmektedir.

2.4.2. Damga Çıkarım Algoritması

Damga Çıkarım Algoritması veri tabanından damga verisi çıkarmakta ve damgalama öncesi orijinal veri tabanı değerlerini Fark Genişleme Tabanlı Damgalama algoritmasını kullanarak yeniden elde etmektedir. Algoritmanın genel yapısı Şekil 2.17'de verilmiştir.

Damga çıkarım işleminin ilk adımı olarak veri tabanı satırları birincil anahtar değeri dikkate alınarak, alanlar ise isimlerine göre sıralanır. Damga yerleştirme aşamasında kullanılan DS veri setinin damga çıkarım sürecinde de aynı satırlardan oluşacak şekilde seçilmesi gereklidir. Bu işlem için, damga yerleştirme aşamasında kullanıcı tarafından

kullanılan gizli anahtar κ değeri ile (19) ifadesinden yararlanılır. Damga çıkarım işlemi için κ gizli anahtar değeri ile birlikte F^{best} en iyi ateş böceği bilgisine de ihtiyaç duyulmaktadır.



Şekil 2.17. Damga çıkarma algoritmasının genel yapısı

F^{best} en iyi ateş böceği M elemana sahip olduğu durumda, DS veri seti de M satıra sahip olacaktır. F^{best} en iyi ateş böceğinin her bir elemanındaki değer, damga yerleştirme aşamasında kullanılan ilgili satırların alan indislerini göstermektedir. Dolayısıyla, F^{best}'in eleman değerlerinin gösterdiği alan çiftleri ile Fark Genişleme Tabanlı Damgalama algoritması kullanılarak yerleştirilmiş damga biti elde edilirken, alan değerlerinin damgalama öncesi orijinal değerlerine de geri dönüşü sağlanmaktadır. Damga çıkarım sürecine ait işlem adımları aşağıdaki gibidir.

Giriş: *DS'*, F^{best}

Çıkış: Orijinal DS veri seti ve elde edilmiş damga verisi W'

Adım 1. $j = 1 \ldots M$ değerleri için aşağıdaki adımları tekrar et

Adım 1.1. *DS* veri setinin j. satırını oku, R^j

Adım 1.2. $x = TamKısım(F_j^{best}), y = OndalıkKısım(F_j^{best})$

Adım 1.3. $At_x' = R_x^j, At_y' = R_y^j$

Adım 1.4. $(At_x, At_y, w'_i) = Ters\_DEW(At_x', At_y')$

Adım 1.5. $i = i + 1$

$\mathrm{Ters\_DEW}(\mathrm{At_x}', \mathrm{At_y}')$ metodu parametre olarak aldığı değerlere Fark Genişleme Tabanlı Damgalama algoritmasını uygulayarak damga verisini ve alanların damga yerleştirme öncesi sahip olduğu orijinal değerleri geri döndürmektedir. Damga çıkarım sonucu elde edilen W' damga verisi yerleştirilen damga verisi ile tutarlıysa, veri tabanının özgün olduğu ve üzerinde izinsiz herhangi bir değişiklik yapılmadığı anlaşılmaktadır.

2.5. Yeni Bir Alfa Sayısal Veri Tabanı Damgalama Yaklaşımı

Literatürde sağlam damgalama yöntemleri çoğunlukla sayısal alanlar üzerinde damgalama gerçekleştirmektedir. Damgalama sürecinde sayısal veriler üzerinde farklı düzeylerde bozulmalara sebep olmakta, yöntemlerin sadece bir kısmı damga çıkarım sonrası orijinal veriye geri dönebilmektedir. Metin alanlar üzerinde damgalama gerçekleştiren sağlam damgalama yöntemleri ise boşluk ekleme, kelimeleri sinonimleri ile değiştirme gibi yöntemleri kullanmaktadırlar.

Kırılgan damgalama yöntemlerinin birçoğu için alanın barındırdığı veri türü önemli değildir. Veri tabanından çıkarılan özel bilginin sertifika ile güvenlik altına alınması ile damgalama yapılmaktadır. Damganın doğrulanacağı aşamada damgalama aşamasındaki yöntem kullanılarak veri tabanından çıkarılan yeni özel bilgi sertifika ile güvenlik altına alınmış bilgi ile karşılaştırılarak veri tabanının saldırıya uğrayıp uğramadığı belirlenir.

Bu çalışma ilişkisel veri tabanlarında sayısal ve metin alanlarına damgalama gerçekleştirirken sayısal alanlardaki bozulmayı minimize etmeyi ve metin alanlar üzerinde herhangi bir bozulmaya sebep olmadan maksimum damga kapasitesine ulaşmayı hedeflemektedir. Önerilen yöntem veri tabanının sayısal alan çiftlerine Fark Genişleme Tabanlı Damgalama yöntemi sayesinde damga verisini saklayabilmekte ve kayıpsız olarak

orijinal veriyi geri elde edebilmektedir. Metin alanlar üzerinde ise Homoglif Dönüşüm gerçekleştirerek, algılanabilirliği düşük ve yüksek damga kapasitesine ulaşmaktadır [64]. Damga yerleştirilecek kayıtların seçiminde SHA-256 özüt fonksiyonu kullanılarak kayıt seçimindeki güvenlik zafiyetleri giderilmiştir. Çalışma maksimum miktarda damga verisi saklarken, orijinal veri tabanı üzerinde minimum bozulmaya sebep olacak satırların seçilmesinde Ateş Böceği Algoritmasını kullanmaktadır. Bu sayede önerilen damgalama şeması nümerik ve metin alanları üzerinde kullanılabilen, orijinal veri üzerinde bozulmayı minimize edecek şekilde damgalanacak satırları seçebilen, algılanabilirliği düşük ve saldırılara karşı dayanıklı olması ile yenilikler içermektedir.

2.5.1. Damga Yerleştirme Algoritması

Damga yerleştirme algoritması Ön İşlem, Damga Yerleştirilecek Alanların Belirlenmesi ve Damga Yerleştirme süreçlerini içeren 3 aşamadan oluşmaktadır.

Ön İşlem Aşamasında; damgalanacak veri tabanının hazırlık işlemleri gerçekleştirilmektedir. Veri seti birincil anahtar değeri dikkate alınarak artan sırada sıralanmıştır. Ayrıca veri setinin alanları da alfabetik olarak sıralı olacak şekilde yer değiştirilmiştir. Bu yer değiştirme veri tabanı üzerinde fiziksel bir değişikliğe sebep olmayan, algoritma içerisinde alan indislerini belirleyen sanal bir sıralamadır. Bu sayede ekleme ve alan değiştirme saldırılarına karşı önerilen yöntemin dayanıklılığı artırılmıştır.

Ön İşlem Aşamasının bir sonraki adımı damgalanacak aday satırların belirlenmesidir. Bu işlem için özüt metodu kullanılacaktır. Damgalanmak üzere seçilen aday satırlardan oluşan veri seti DS olarak ifade edilirse seçim işleminde kullanılan kriterin verildiği (23) ifadesinde kullanıcı tarafından belirlenen gizli anahtar κ, ilgili kaydın birinci anahtarı $r.P$, metin birleştirme operatörü $||$ ve özüt metodu ise $\mathcal{H}$ ile temsil edilmektedir. γ değerinin seçilecek kayıt sayısını belirlediği (23) ifadesinde verilen şartı sağlayan kayıtlar DS veri setini oluşturacaktır.

$$mod(\mathcal{H}(\kappa||\mathcal{H}(r.P||\kappa),\gamma) \equiv 0 \tag{23}$$

(23) ifadesine uyan kayıtlardan oluşan DS veri setinin M adet satır ve A adet alandan oluştuğu varsayıldığında, damga yerleştirme sürecinde her bir alan için uyulması gereken bozulma sınır aralığı (24) ifadesinde verilmiştir.

$$\left(DT_x^{min}, DT_x^{max}\right), x = 1 \ldots A \quad (24)$$

Alan bazında bozulma sınır değerlerinin belirlenmesi sayesinde Fark Genişleme Tabanlı Damgalama yöntemi sonrası verinin kullanılamaz hale gelmesi önlenmektedir. Kullanıcı tarafından bozulma sınır aralıkları belirlenmediği durumlarda orijinal veri tabanındaki her bir alanın sahip olduğu en büyük ve en küçük değeri ilgili alan için bozulma sınır aralığı olarak kabul etmektedir.

Damga Yerleştirilecek Alanların Belirlenmesi Aşamasında; Ateş Böceği Algoritması yardımıyla damga yerleştirilecek satırların kriterlere uyan maksimum kazanım sağlayacak nümerik sütun çiftleri belirlenmektedir. Ön İşlem Aşamasında oluşturulan veri seti girdi olarak alınıp, nümerik alan çiftlerinin indis bilgilerini içeren ateş böceği çıktı olarak üretilmektedir. Yang vd. tarafından önerilen Ateş Böceği Algoritması, problem uzayının tamamının kabul edilebilir süre içerisinde aranmasının mümkün olmadığı problemlerin çözümünde kullanılan sezgisel algoritmalardan biridir [70]. Algoritma, optimizasyon sürecinde ateş böceklerinin davranışları ve yaydıkları ışıktan faydalanmaktadır. Ateş böceklerinin ışık karakteristikleri aşağıdaki 3 kural ile özetlenebilir;

1. Ateş böcekleri cins farkı gözetmeksizin bir diğerini yaydığı ışık sebebiyle çekebilir.
2. Ateş böceklerinin parlaklıkları çekim cazibesini gösterir. Parlaklığı düşük olan ateş böcekleri yüksek olana doğru hareket edecektir.
3. Optimizasyonu sağlayan amaç fonksiyonu ateş böceğinin parlaklığını belirlemektedir.

i. bireyin (ateş böceğinin) parlaklık değerini belirten Br^i hesabı (25) ifadesinde verilmiştir. Aday çözümü içeren bireyin parlaklık değerini belirten Br^i değeri (26) ifadesinde verilen R_a, (27) ifadesinde verilen R_c ve (28) ifadesinde verilen R_h değerlerinin, toplamları 1'e eşit olan w_a, w_c ve w_h değerleri ile ağırlıklandırılmasıyla hesaplanır.

$$Br^i = R_a w_a + R_c w_c + R_h w_h \quad (25)$$

$$R_a = \left(\sum \frac{|mdf|}{tuple_{count}}\right) \quad (26)$$

$$R_c = (row_w / M) \quad (27)$$

$$R_h = \sum hmgf(row_{text}) / \sum length(row_{text}) \quad (28)$$

R_a ifadesi sayısal alanlara damga verisi yerleştirildikten sonra oluşan mutlak ortalama bozulmanın toplamıdır. $|mdf|$ sayısal alan çiftlerine damga yerleştirilmesinden dolayı ortaya çıkan toplam değişimin mutlak değerini temsil etmektedir. Veri tabanının toplam satır sayısı ise $tuple_{count}$ ile gösterilmiştir.

R_c ifadesi damgalanmak üzere seçilen satırların damgalanabilme oranıdır. Damgalanmış satır sayısı row_w, damgalanmak üzere seçilen satır sayısı ise M ile gösterilmiştir.

R_h ifadesi damgalanmak üzere seçilen satırların metin alanlarına yerleştirilen damganın, metin alanlarındaki toplam karakter sayısına oranıdır. $hmgf$ metodu parametre olarak verilen satırın metin alan değeri row_{text} içerisine saklanabilecek damga bit sayısını hesaplarken, $length$ metodu metin alanın karakter uzunluğunu hesaplamaktadır.

İlk popülasyon bireylerinin oluşturulması ve en parlak ateş böceğine popülasyonu oluşturan diğer ateş böceklerinin yaklaşım hareketinde 2.4.1 bölümünde detayları açıklanan yöntem kullanılmıştır.

Damganın Yerleştirilmesi Aşamasında; aday satırların en parlak ateş böceğinin gösterdiği nümerik alan çiftlerine Fark Genişleme Tabanlı Damgalama Yöntemi, metin alanına ise önerilen Homoglif Dönüşüm Yöntemi kullanılarak damga verisi yerleştirilmektedir.

Homoglif Dönüşüm Yöntemi, karakterlerin kendilerine benzer görünümlü, farklı Unicode karakterler ile değiştirilmesi üzerine kuruludur. 2017 Haziran ayında duyurulan Unicode 10 versiyonu 1 milyonu aşkın birbirine benzerlik gösteren karakterler içermektedir [66]. Değiştirilen karakterin fark edilebilirlik durumu; yerine seçilen karakter, metnin sunulduğu font bilgisi ve karakter kodlamasına göre değişebilmektedir.

Homoglif dönüşümde kullanılacak karakterlerin belirlenmesinde "göz ile fark edilebilirlik" testleri gerçekleştirilmiş ve algoritmanın sağlamlığını sağlayacak karakter değişimleri kullanılmıştır. Yerleştirilecek olan damga değerinin "1" olması durumunda orijinal karakter homoglif karşılığı ile değiştirilirken, "0" olması durumunda ise orijinal hali ile bırakılmaktadır. Sadece damga bitinin "1" olduğu durumlarda homoglif karakter dönüşümü yapılacağından çalışmanın metin alanlar üzerindeki damga verisi düşük

algılanabilirliğe sahiptir. Dönüşüm sürecinde kullanılan toplamda 31 karakter ve yerlerine geçecek olan Unicode karşılıklar Tablo 2.2'de verilmiştir.

Damga yerleştirme sürecine ait yalancı kodun verildiği Şekil 2.18'de $TSBD$ metodu en iyi ateş böceği bireyinden Fark Genişlemede kullanılacak ilk nümerik değeri tutan alan indisini, OBD metodu ise ikinci nümerik değeri tutan alan indisini bulmaktadır. FGY metodu A_x ve A_y değerlerine W_j damga bitini saklayacak şekilde fark genişleme metodunu uygulamaktadır. HGY metodu ise A_t ifadesi ile gösterilen metin alanına Homoglif Dönüşüm Yöntemi ile damga yerleştirmektedir.

Tablo 2.2. Homoglif Dönüşümde kullanılan karakterler

Sembol	*Bit 0*	*Bit 1*	*Sembol*	*Bit 0*	*Bit 1*
	Orijinal Kod	*Homoglif Kod*		*Orijinal Kod*	*Homoglif Kod*
A	0x41	0x391	a	0x61	0x430
B	0x42	0x392	b	0x62	0x15af
C	0x43	0x421	c	0x63	0x3f2
D	0x44	0x13a0	d	0x64	0x501
E	0x45	0x395	r	0x72	0x433
G	0x47	0x13c0	v	0x76	0x1d20
H	0x48	0x397	x	0x78	0x445
J	0x4a	0x408	y	0x79	0x443
K	0x4b	0x39a	z	0x7a	0x1d22
L	0x4c	0x216c	2	0x32	0xff12
M	0x4d	0x39c	3	0x33	0x417
N	0x4e	0x39d	4	0x34	0x13ce
P	0x50	0x3a1	5	0x35	0x1bc
S	0x53	0x405	Y	0x59	0x3a5
T	0x54	0x3a4	Z	0x5a	0x396
X	0x58	0x2169			

Giriş: DS veri kümesi, damga verisi W, en iyi ateş böceği F^{best}

Çıkış : Damga yerleştirilmiş DS veri kümesi

for j=1 to $SatırSayısı(DS)$

$satır = DS_j$

$x = TSBD(F_j^{best})$

$y = OBD(F_j^{best})$

$A_x = satır.x$

$A_y = satır.y$

$(A_x, A_y) = FGY(A_x, A_y, W_j)$

$A_t = HGY(A_t, W)$

End

Şekil 2.18. Damga yerleştirme algoritması

2.5.2. Damga Çıkarım Algoritması

Damga Çıkarım Algoritması veri tabanından damga verisi çıkarmakta ve damgalama öncesi orijinal sayısal veri tabanı değerlerini Fark Genişleme Tabanlı Damgalama yöntemini kullanarak elde etmektedir. Metin alan değerleri üzerinde damga çıkarım sürecinde Homoglif Dönüşüm kullanılmıştır.

Damga çıkarım işleminin ilk adımı olarak veri setinin satırları birincil anahtar değeri dikkate alınarak artan sırada, alanları da alfabetik olarak sıralanacaktır. Bu yer değiştirme veri tabanı üzerinde fiziksel bir değişikliğe sebep olmayan, algoritma içerisinde alan indislerini belirleyen sanal bir sıralamadır. Damga yerleştirme aşamasında kullanılan DS veri setinin damga çıkarım sürecinde de aynı satırlardan oluşacak şekilde seçilmesi gerekmektedir. Bu işlem için, damga yerleştirme aşamasında kullanıcı tarafından kullanılan gizli anahtar κ değeri ile (23) ifadesinden yararlanılır. Damga çıkarım işlemi için κ gizli anahtar değeri ile birlikte F^{best} en iyi ateş böceği bilgisine de ihtiyaç duyulmaktadır.

Damga çıkarımda kullanılacak DS' veri seti oluşturulduktan sonra en parlak ateş böceği kullanılarak sayısal alan çiftlerinden, Homoglif Dönüşüm kullanılarak da metin alanlarından damga çıkarım işlemi gerçekleştirilir. Sürece ait yalancı kod Şekil 2.19'da verilmiştir.

Giriş: DS' veri kümesi, en iyi ateş böceği F^{best}

Çıkış: Orijinal DS veri seti ve damga verisi W

for j=1 to $SatırSayısı(DS')$

$satır = DS'_j$

$x = TSBD(F_j^{best})$

$y = OBD(F_j^{best})$

$A'_x = satır.x$

$A'_y = satır.y$

$(A_x, A_y, W_j) = FGYÇıkar(A'_x, A'_y)$

$(A_t, W) = HGYÇıkar(A'_t)$

End

Şekil 2.19. Damga çıkarma algoritması

Şekil 2.19'da $TSBD$ metodu en iyi ateş böceği bireyinden Fark Genişletmede kullanılacak ilk nümerik değeri tutan alan indisini, OBD metodu ise ikinci nümerik değeri tutan alan indisini bulmaktadır. $FGYÇıkar$ metodu A'_x ve A'_y değerlerine Fark Genişleme Tabanlı Damgalama yöntemi uygulayarak ilgili alanların damgalama öncesi sahip oldukları orijinal A_x ve A_y değerleri ile birlikte saklanmış damga verisini bulmaktadır. $HGYÇıkar$ metodu ise A'_t ifadesi ile gösterilen metin alanından Homoglif Dönüşüm Yöntemi ile damga verisini ve metnin damgalama öncesi orijinal halini bulmaktadır.

3. BULGULAR VE İRDELEME

Bu bölümde tez kapsamında gerçekleştirilen çalışmalara ait bulguların literatürde var olan benzer çalışmalara ait sonuçlarla kıyaslaması gerçekleştirilirken önerilen yöntemlerin üstünlükleri ve getirdikleri yenilikler irdelenecektir.

3.1. İlişkisel Veri Tabanları İçin Bölümleme Tabanlı Damgalama Yönteminin Değerlendirilmesi ve Elde Edilen Sonuçlar

Literatürdeki ilişkisel veri tabanları üzerinde sayısal damgalama alanındaki çalışmaların büyük bir kısmı sayısal alanlar üzerine yoğunlaşmıştır. Sayısal olmayan alanlara damga yerleştirme konusunda yapılan az sayıda çalışmalardan ikisi [48] ve [71]'dür.

[48] çalışmasında damga verisi olarak seçilen resim ikilik düzene döndürülerek, her bir bit değeri veri tabanının tarih alanında saklanmaya çalışılmıştır. Veri tabanlarında tarih verisini tutan alanlar gün ve saat bilgisi olarak 2 farklı bilgiyi aynı yerde saklamaktadırlar. Araştırmacılar tarih alanının saniye bilgisindeki küçük değişimlerin kabul edilebileceğini varsayarak damga bitini saniye verisine saklamışlardır. Damga yerleştirme sürecinde tarih alanının saniye verisinin üzerine damga verisi kaydedildiğinden, orijinal veri kaybedilecek, damga çıkarım aşamasında sadece damga verisi çıkarılabilecektir.

[71] çalışmasının [48] çalışmasından bazı farkları bulunmaktadır. Öncelikle çalışma sadece tarih alanının saniye değerine değil, aynı zamanda veri tabanının sayısal alanına da damga yerleştirmektedir. Algoritma başlangıcında diğer çoğu yöntemde olduğu kullanıcıdan tek bir gizli anahtar değeri değil, 2 farklı gizli anahtar değeri belirlemesi istenir. İlk anahtar değerinin 5 bitlik, ikinci anahtar değerinin ise 4 bitlik olması gereklidir. İlk anahtar değeri damga yerleştirilecek satırları belirlemektedir. Damgalanmak üzere seçilen satırların tüm sayısal alan değerlerinin en anlamsız bit değerine damga bit değeri yerleştirilir. Aynı bit değeri damgalanmak üzere seçilen satırın tarih alanının saniye değerine de yerleştirilecektir. Bu damga biti kullanıcının belirlediği 4 bitlik ikinci anahtar değeri ile birleştirilip oluşan 5 bitlik değerin sayısal karşılığı damgalanmak üzere seçilen satırın tarih alanına yerleştirilir. Algoritmanın saldırılara karşı dayanıklı olması için izlenen bu yöntem tüm veri tabanında değişikliğe sebep olurken kapasiteyi de düşürmektedir. [48] çalışmasında olduğu gibi [71]

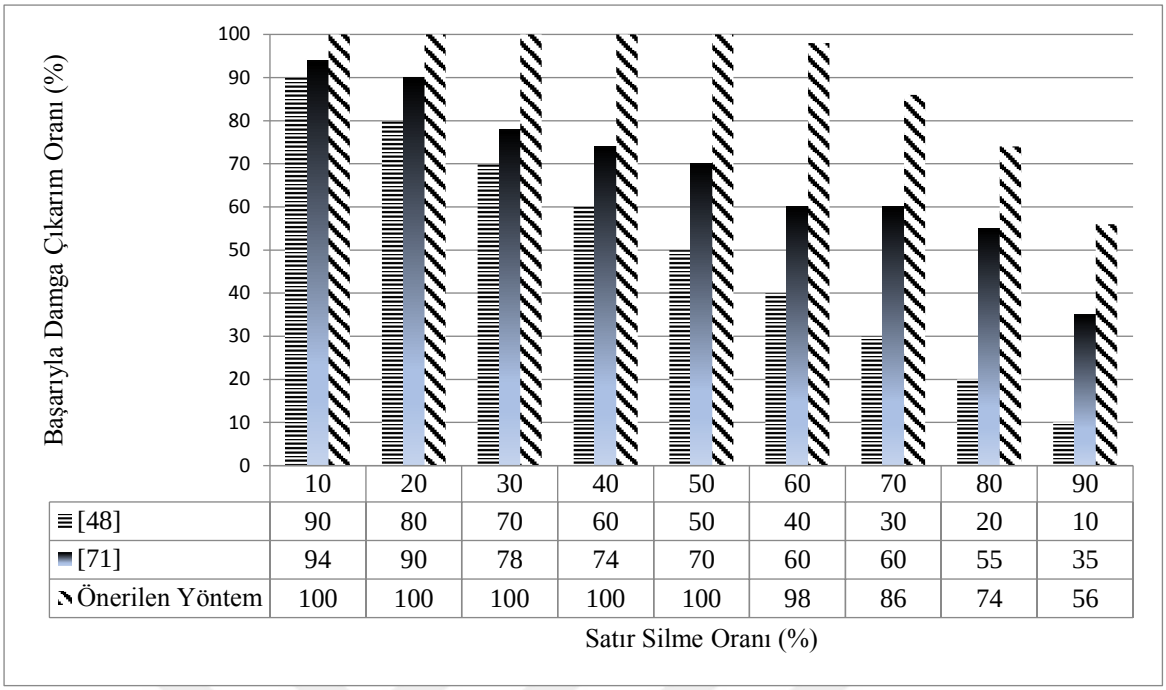
çalışması da damga çıkarım süreci sonrası orijinal veri tabanı değerlerine dönüşüm sağlamamaktadır.

Tez çalışması kapsamında gerçekleştirilen yöntem de [48] ve [71] çalışmalarına benzer şekilde tarih tipindeki verileri tutan alanlar üzerinde damgalama işlemini gerçekleştirmektedir. Damgalanacak satırların seçiminde özüt metodu ile kullanıcı tarafından belirlenen gizli anahtar kullanılmıştır. Saldırılara karşı dayanıklı olması amacıyla damgalanmak üzere seçilen aday satırlar arasında alt bölümleme uygulaması yapılmıştır. Damga verisinin her 5 bitlik değeri ilgili grubun içerisindeki tüm aday satırların tarih verisi barındıran alan veya alanlarından seçilenin barındırdığı değerin saniye bilgisine saklanmaktadır. Saldırıya maruz kalmış veri tabanından damga çıkarım sürecinde aynı gruptaki damgalanmış satırlardan en az 1 tanesinin kurtarılması durumunda, ilgili grubun damga verisi kayıpsız olarak kurtarılmış olacaktır. Saldırganlar damgalanan kaydı yok etmenin yanında, damga verisini de değiştirmeye çalışmaktadırlar. Bu sebeple damga çıkarım sonrası elde edilen damga verisinin doğruluğu için çoğunluk oylaması yapılarak saldırılar ile değiştirilmesi mümkün olan damga verilerinin kayıpsız kurtarılması amaçlanmıştır.

Önerilen yöntemin test veri kümesi ilerleyen bölümlerde sonuçlarının karşılaştırılacağı [48] ve [71] çalışmalarınki gibi rastgele oluşturulmuştur. Örnek veri kümesi bir sayısal ve bir tarih alanı olan 1000 satırdan oluşmaktadır. Veri tabanı yönetim sistemi olarak Microsoft SQL Server kullanılmıştır. Her bir saldırı testi 10 farklı veri kümesi için gerçekleştirilmiş ve sonuçların ortalaması alınmıştır. Test sonuçları [48] ve [71]'de yapılan çalışmalar ile karşılaştırılmıştır. Elde edilen sonuçlar önerilen algoritmanın diğer çalışmalara kıyasla daha yüksek başarı elde ettiği ve saldırılara karşı daha dayanıklı olduğunu göstermektedir.

3.1.1. Alt Küme Silme Saldırısı

Bu saldırı türünde, saldırgan verinin rastgele bir kısmını silerek, damga değerini ortadan kaldırmayı amaçlamaktadır. Önerilen yöntemin Altküme Silme Saldırısı karşısında, gömülen damgayı belirleme oranları açısından literatürdeki diğer çalışmalar ile karşılaştırılması Şekil 3.1.'de verilmektedir. Önerilen yöntemin damga yerleştirme sürecinde gruplama kullanarak damga verisini koruması sayesinde veri kümesine yapılan %90 oranındaki silme saldırısına karşı gömülen damga verisinin %56'sını çıkarabildiği görülmektedir.



	10	20	30	40	50	60	70	80	90
[48]	90	80	70	60	50	40	30	20	10
[71]	94	90	78	74	70	60	60	55	35
Önerilen Yöntem	100	100	100	100	100	98	86	74	56

Şekil 3.1. Alt küme silme saldırısına karşı dayanıklılık sonuçları

[48] çalışmasının saldırıdan en yüksek oranda etkilendiği görülmektedir. Çalışmanın damga verisini 5'er bitlik parçalara ayırıp her bir parçayı ayrı bir satıra sakladığı için, damgalanmış satırın silinmesi ile birlikte 5 bitlik damga verisi yok olmaktadır. Aynı damga verisinin kurtarılabileceği bir çözüm ilgili çalışmada sunulmadığından saldırı sonucu kaybedilen damga oranı yüksektir.

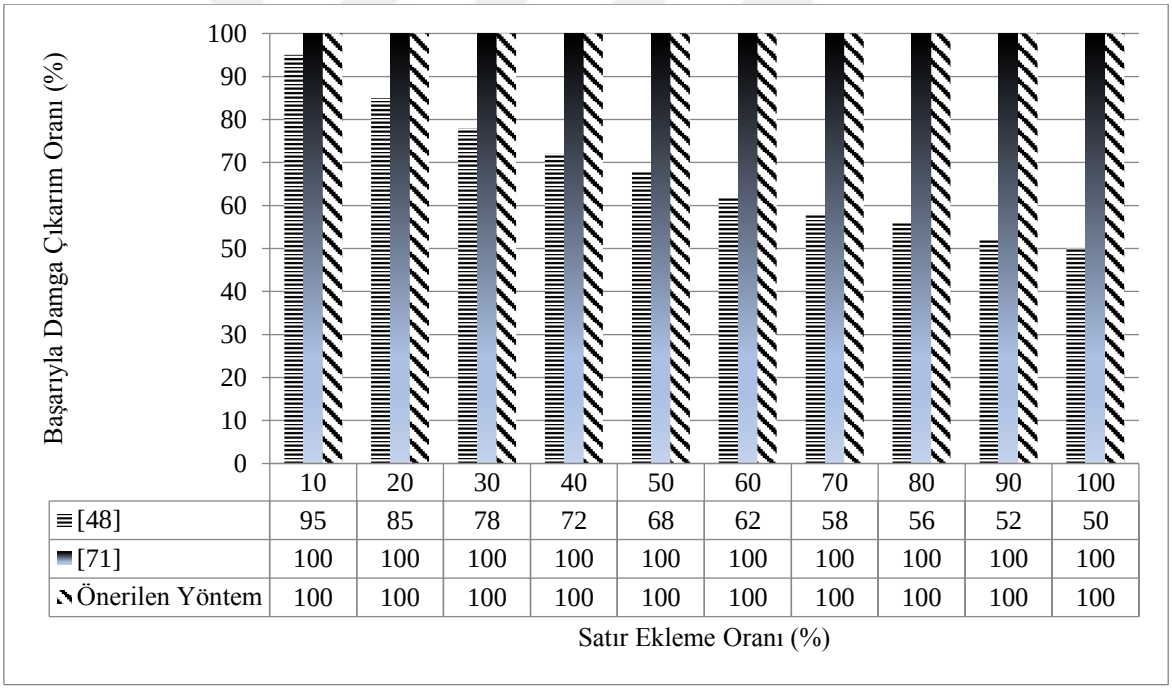
[71] çalışmasının da [48] çalışması gibi Alt Küme Silme Saldırısına karşı almış olduğu bir önlem bulunmamaktadır. Damgalama sürecinde her bir damga bitini damgalanmak üzere seçilen satırın hem sayısal alanın en anlamsız bit değerine, hem de tarih alanının saniye değerine sakladığı için ilgili satırın silinmesi durumunda damga verisini kaybedecektir. Kaybedilen damga verisinin kurtarılabileceği bir çözüm sağlanmadığından kayıp telafi edilemeyecektir. Saldırı sonucu damga çıkarım oranının [48] çalışmasına göre daha yüksek olmasının sebebi, silinen satır ile kaybedilen damga verisinin az oluşudur. [48] çalışması her bir damga satırına 5 bitlik damga verisi saklarken, [71] çalışması 1 bitlik damga verisi saklamaktadır.

Önerilen çalışma ise damga verisini 5'er bitlik parçalara ayırıp, her bir parçanın sayısal karşılığını kullanıcının belirlediği grup sayısınca veri tabanına damga verisi olarak sakladığından Alt Küme Silme Saldırısına karşı daha dayanıklıdır. Saldırı sonucu her bir

grubun tek bir bireyi olan damgalanmış satırın kurtarılması damga verisinin kurtarılmasını sağlayacaktır.

3.1.2. Alt Küme Ekleme Saldırısı

Saldırganın rastgele oluşturduğu veri kümesini damgalanmış veri kümesine eklediği saldırı türüdür. Saldırganın amacı damgalanmamış kayıtlardan da damga verisinin çıkarılıp damgalama yönteminin başarısını düşürmektir. Damga yerleştirilen satırların kullanıcı tarafından belirlenen gizli anahtar ile seçilmesi ve damgalama aşamasında gruplama kullanılması sayesinde bu saldırının önerilen yöntem üzerinde herhangi bir etkisi olmamıştır. Şekil 3.2'de verilen sonuçlar incelendiğinde, veri kümesinin boyutu kadar rastgele oluşturulmuş veri ile saldırıldığında bile damga çıkarım başarısı %100'dür.



	10	20	30	40	50	60	70	80	90	100
[48]	95	85	78	72	68	62	58	56	52	50
[71]	100	100	100	100	100	100	100	100	100	100
Önerilen Yöntem	100	100	100	100	100	100	100	100	100	100

Şekil 3.2. Alt küme ekleme saldırısına karşı dayanıklılık sonuçları

[48] çalışması Alt Küme Ekleme Saldırısı ile veri tabanına saldırgan tarafından eklenen satırlardan yerleştirmediği damga verisini çıkaracağından, damga çıkarım sonucu elde ettiği damga verisi eksik olacaktır.

[71] çalışması damga çıkarım sürecinde kullanıcı tarafından belirlenen gizli anahtarı kullandığı ve aynı satırın sayısal ve tarih alanından çıkardığı damga verisi üzerinde kontrol yaptığı için bu saldırıdan etkilenmemektedir.

Önerilen çalışma damga verisinin saklanacağı satırların seçiminde kullandığı kullanıcı tarafından belirlenen gizli anahtarı damga çıkarım sürecinde de kullanmaktadır. Aynı zamanda çıkarılan damga verisini aynı grup için çıkarılan damga verileri arasında çoğunluk oylaması yaparak karar vermektedir. Tüm bu özellikleri Alt Küme Ekleme Saldırısından etkilenmemesini sağlamıştır.

3.1.3. Alt Küme Değiştirme Saldırısı

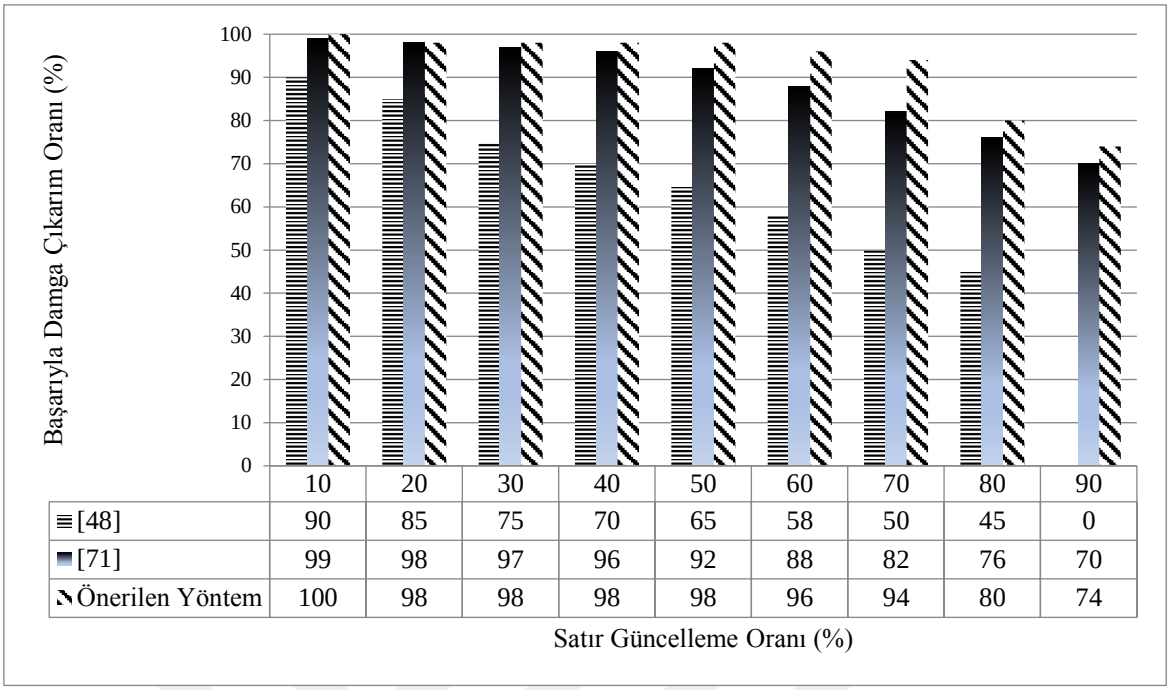
Alt Küme Değiştirme Saldırında, saldırgan veri kümesi içerisinden belirlediği satırların alan değerlerini rastgele olarak değiştirir. Saldırının amacı damgalanmış alan değerini değiştirerek damga çıkarım başarısını düşürmektir. Alt küme silme saldırısından farklı olması amacıyla alanların yarısının değeri değiştirilmiştir.

Şekil 3.3'de önerilen yöntemin, Altküme Değiştirme Saldırısı karşısında gömülen damgayı belirleme oranı açısından, benzer yöntemlerle karşılaştırması verilmiştir. Verinin %90'ının değiştirilmesi durumunda dahi önerilen algoritma gruplama özelliği sayesinde damgayı %74 doğrulukla belirlemektedir.

[48] çalışmasının bozulan damga değerini kurtarabilme özelliği olmadığından ve değişime uğrayan alanın tarih alanı olması durumunda değiştirilen her bir alan için 5 bitlik damga verisi kaybettiğinden saldırıdan en yüksek oranda etkilenmektedir.

[71] çalışmasının [48] çalışmasına göre saldırıdan daha az etkilenmesinin sebebi değiştirilme sonucu kaybedilen her bir alanda saklanan damga verisinin 1 bit olmasının yanında bazı durumlarda sayısal ve tarih alanından çıkarılan damga verileri arasında kontrol yaparak damga verisini bulabilme olasılığıdır.

Önerilen yöntem bu saldırıdan da diğer saldırılarda olduğu gibi [48] ve [71] saldırılarına nazaran daha az etkilenmiştir. Damga verisinin 5'er bitlik parçalarını veri tabanı içerisinde birden çok kez saklıyor oluşu saldırıya karşı dayanıklı olmasını sağlamıştır. Her bir grubun tek bir satırın saldırıdan etkilenmemesi durumunda ilgili grubun damga verisinin kurtarılması mümkün olabilmektedir.



	10	20	30	40	50	60	70	80	90
[48]	90	85	75	70	65	58	50	45	0
[71]	99	98	97	96	92	88	82	76	70
Önerilen Yöntem	100	98	98	98	98	96	94	80	74

Şekil 3.3. Alt küme değiştirme saldırısına karşı dayanıklılık sonuçları

3.2. İlişkisel Veri Tabanları İçin Nümerik Dönüşüm Tabanlı Damgalama Yönteminin Değerlendirilmesi ve Elde Edilen Sonuçlar

Literatürde gerçekleştirilen ilk çalışmalar geri dönüşüm desteğine sahip değillerdi. Geri dönüşüm özelliği, damga çıkarım işlemi sonrası orijinal veri tabanını elde edebilme olarak tanımlanmaktadır. Damgalama algoritmasının geri dönüşüm yapamaması durumunda, damgalanan alanların değerlerindeki bozulmadan dolayı farklı oranlarda veri kaybı oluşmaktadır. Araştırmacılar veri kaybını ortadan kaldırmak için geri dönüşüm desteği barındıran yöntemler geliştirmişlerdir. Bu yöntemlerde kullanıcı tarafından belirlenen gizli anahtar ile gizlenen damga verisi elde edilirken, veri tabanı değerleri damgalama öncesi orijinal değerlerine de döndürülmektedir.

İlk olarak 2004 yılında Alattar tarafından önerilen Fark Genişleme Tabanlı Damgalama yöntemi komşu iki pikselin ortalama değerini genişleterek damga verisinin saklanmasını sağlamaktadır [36]. 2008 yılında Gupta vd. Fark Genişleme Tabanlı Damgalama yöntemini kullanarak geri dönüşüm özelliğine sahip sayısal damgalama gerçekleştirmişlerdir [29]. 2013 yılında Jawad vd. Genetik Algoritma destekli GADEW ismini verdikleri bir diğer geri dönüşüm destekli damgalama algoritması önermişlerdir [30]. [30] çalışması damga yerleştirme aşamasında [29] çalışmasında olduğu gibi Fark Genişleme Tabanlı Damgalama yöntemini kullanmaktadır. Farklı olarak saldırganların damgayı fark

edebilme ihtimalini azaltmak için damgalama sonrası orijinal veri tabanında oluşan bozulmayı minimize etmek amacıyla damga yerleştirilecek alan değerlerini belirlerken Genetik Algoritmanın optimizasyon desteğini kullanmaktadır [72-76].

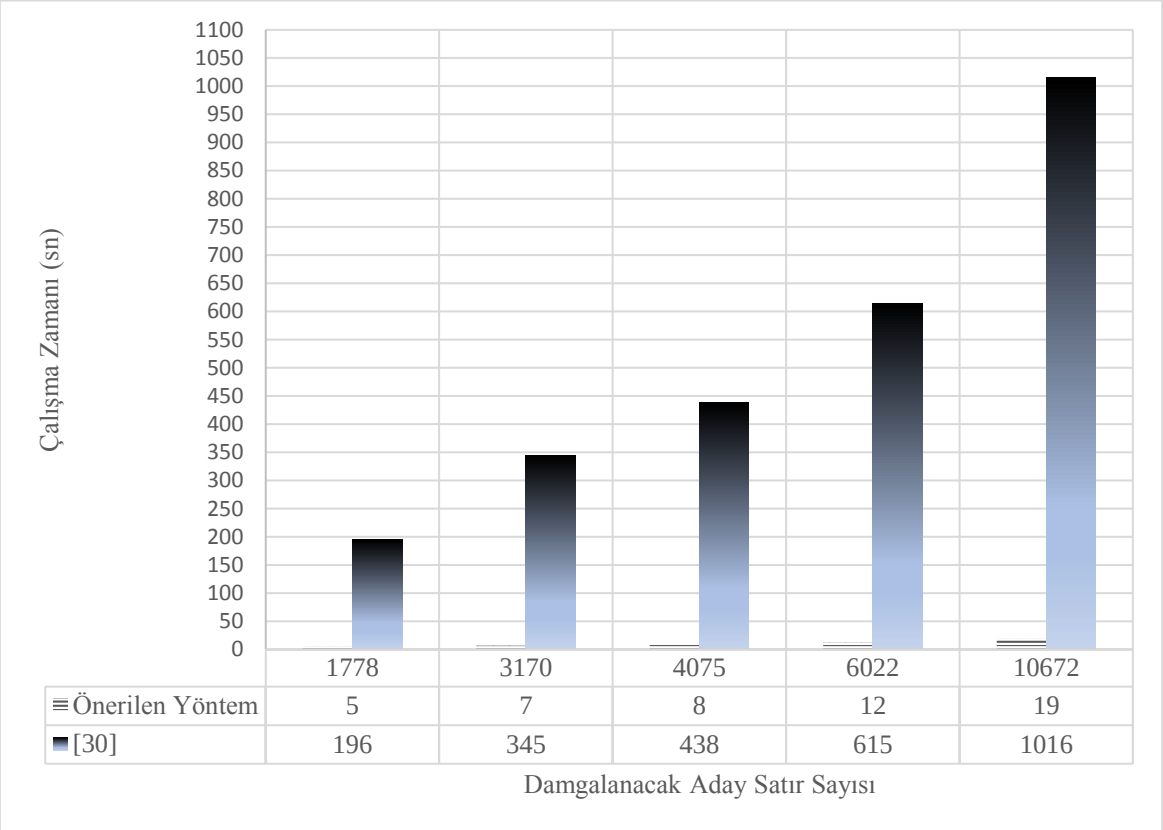
Fark Genişleme Tabanlı Damgalama yöntemi ortalama değeri genişlettiği için değerler üzerinde bozulmaya sebep olmaktadır. [29] çalışması bu bozulmayı sınırlar içerisinde tutmak için damga yerleştirme aşamasında şart kullanırken, [30] çalışması ise en az bozulmaya sebep olacak alan değerlerini belirlemeye çalışıp, bu alanlar üzerinde damga yerleştirme yapmaktadır.

Önerilen yöntemin başarısını ölçen testler [30] çalışması da dahil olmak üzere sayısal alanlar üzerindeki birçok çalışma tarafından kullanılan Forest Cover Type (FCT) veri tabanı üzerinde gerçekleştirilmiştir. FCT veri tabanı Kaliforniya Üniversitesi tarafından sunulmaktadır. 581.012 satıra ve 54 sayısal alana sahiptir. Çalışmada bu alanların 10 tanesi kullanılmıştır. Simülasyonlar MicroSoft SQL Server veri tabanı yönetim sistemi üzerinde MicroSoft Visual Studio IDE ortamında C# programlama dili ile kodlanmıştır. Her bir saldırı testi 10 kez test edilmiş ve sonuçların ortalaması alınmıştır.

3.2.1. Performans Analizi

Veri tabanları özelliklerinden dolayı güncellemeye açık durumdadırlar. Yeni satırların eklenmesi, var olan satırların silinmesi veya güncellenmesi veri tabanı uygulamalarında normal bir süreçtir. Bu sebeple damgalama işleminin gerçek zamanlı çalışmaya uygun olması büyük bir avantajdır. FCT veri tabanından rastgele seçilen 1778, 3170, 4075 ve 6022 satırdan oluşan 4 farklı veri seti üzerinde [30] çalışmasının ve önerilen yöntemin damgalama sürecine ait çalışma süreleri incelenmiştir. Her bir veri seti üzerinde 10'ar kez yapılan testlerin çalışma sürelerinin ortalaması alınarak elde edilen sonuçlar Şekil 3.4'de verilmiştir.

Sonuçlar dikkate alındığında önerilen yöntemin çalışma süresinin [30] ile kıyaslandığında çok daha düşük olduğu gözükmektedir. Örneğin, 1778 satırlık veri setini [30] çalışması 196 saniyede damgalarken, önerilen yöntem sadece 5 saniyede damgalamaktadır. Önerilen yöntemin damgalama sürecinde ihtiyaç duyduğu işlem sayısının az, sürecin ise basit matematiksel işlemlerden oluşuyor olması çalışma süresinin düşük olmasını sağlamıştır. Ayrıca [30] çalışmasının Fark Genişleme Tabanlı Damgalama yönteminde kullanacağı alanları belirlemek için destek aldığı Genetik Algoritma çalışma süresini uzatmaktadır.

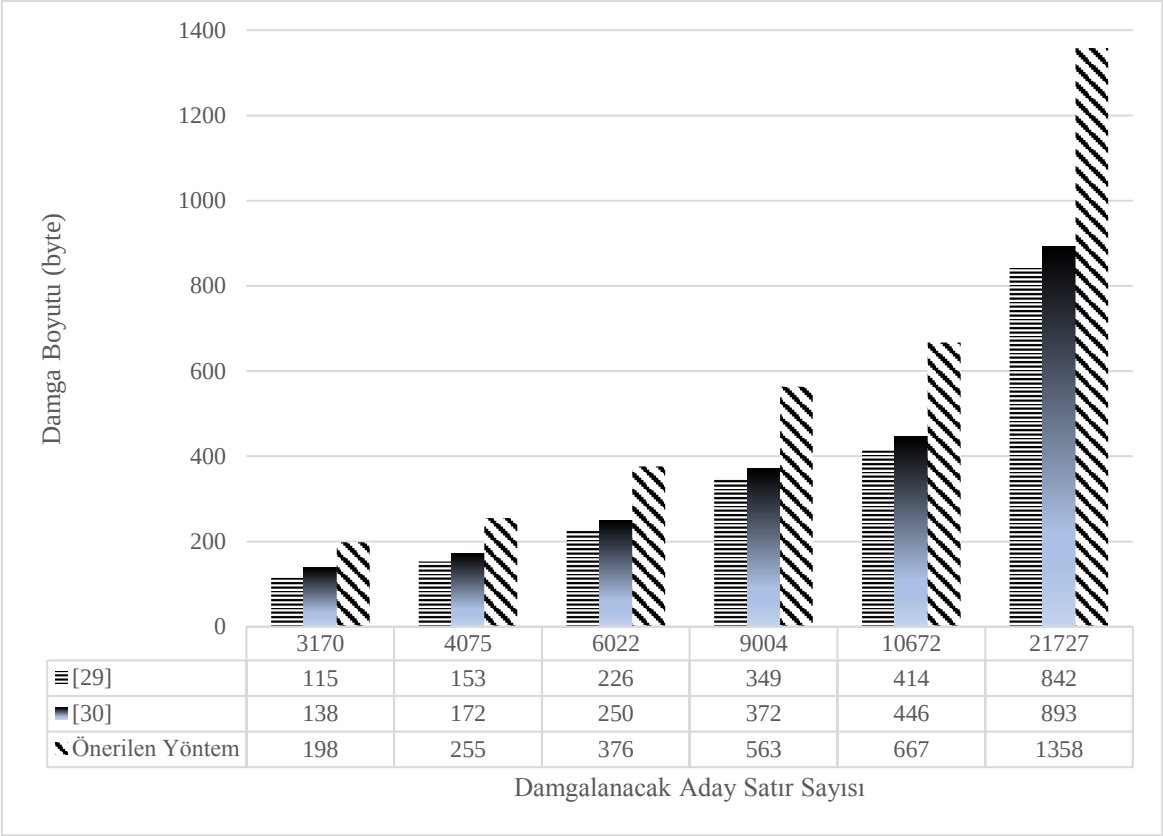


Şekil 3.4. Algoritma çalışma süreleri

3.2.2. Kapasite Analizi

Damgalama algoritmalarının başarısını belirleyen kriterlerden bir diğeri de damgalama kapasiteleridir. Damgalama kapasitesinin yüksek olması algoritmanın uygulanabileceği veri tabanı türü (medya, askeri, tıbbi, vb.) ve damga veri türü (metin, resim, ses, vb.) çeşitliliğini artıracaktır.

FCT veri tabanından rastgele seçilen 3170, 4075, 6022, 9004, 10672 ve 21727 satırdan oluşan 6 farklı veri seti üzerinde önerilen yöntem, [29] ve [30] çalışmaları test edilmiştir. Her bir veri seti üzerinde 10'ar kez yapılan testlerin ortalama çalışma süreleri Şekil 3.5'de verilmiştir. Sonuçlar incelendiğinde önerilen çalışmanın diğer algoritmalara göre daha yüksek damgalama kapasitesine sahip olduğu görülmektedir. Örneğin 21727 satırdan oluşan veri setinde [29] çalışması 842 byte, [30] çalışması 893 byte damga verisi saklayabilirken, önerilen yöntem 1358 byte'lık damgalama kapasitesine sahiptir.



	3170	4075	6022	9004	10672	21727
[29]	115	153	226	349	414	842
[30]	138	172	250	372	446	893
Önerilen Yöntem	198	255	376	563	667	1358

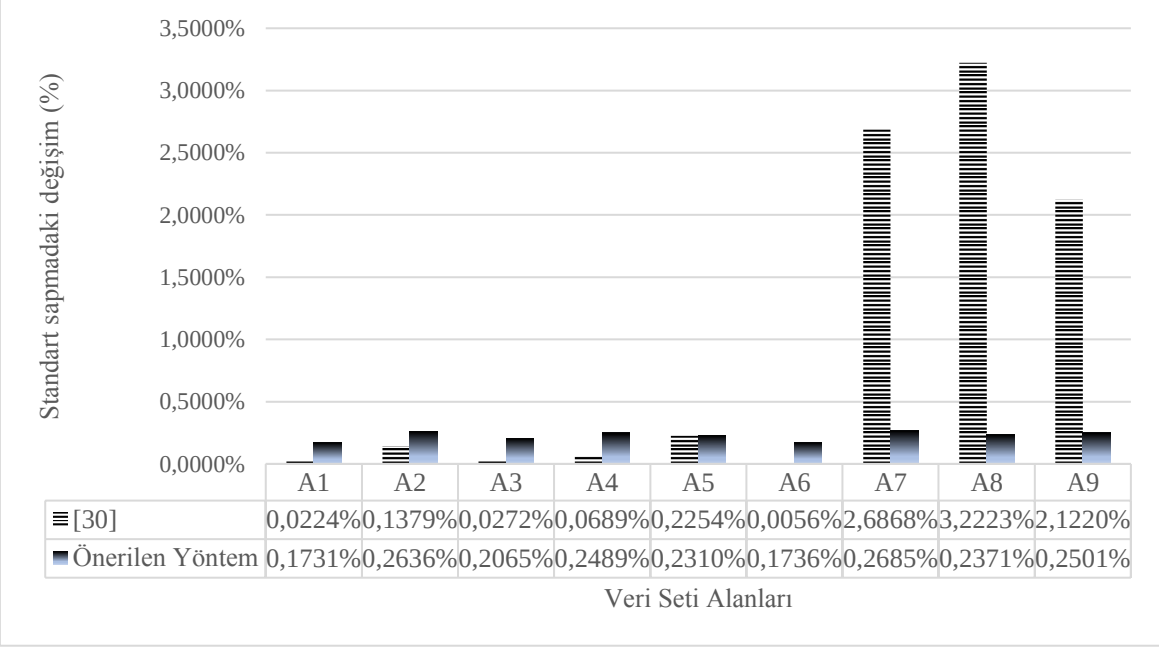
Şekil 3.5. Algoritma damgalama kapasiteleri

[29] ve [30] çalışmaları damgalama sürecinde Fark Genişleme Tabanlı Damgalama yöntemini kullandıkları için damgalamada kullanılan alan değerinin seçimine bağlı olarak damgalama sonrası elde edilen yeni değerler izin verilen bozulma sınırlarını aşabilmektedirler. Örneğin damga yerleştirilmek üzere seçilen satırın Fark Genişleme Tabanlı Damgalama yönteminde kullanılacak olan alan değerleri $A_1 = 70$, $A_2 = 91$ ve damga verisinin 0 olduğu varsayıldığında, damgalama sonrası alanların yeni değerleri $A'_1 = 59$ ve $A'_2 = 101$ olacaktır. Yeni değerlerin veri tabanının kullanım amacına (örneğin öğrenci notları) uygun olmadığı durumlarda bu alanlar damgalama işleminde kullanılmayacaktır. Bu durum [29] ve [30] çalışmalarının damgalama kapasitelerini sınırlandırabilmektedir.

Önerilen yöntem nümerik dönüşümden dolayı sebep olduğu değişimin az olması ve dönüşümde kullanacağı değer vektör eleman sayısı değişken olabilmesinden dolayı aynı veri tabanı üzerinde daha yüksek oranda damga verisi saklayabilmektedir.

3.2.3. Güvenlik Analizi

Saldırganlar damga verisini bozmak veya yok etmek amacıyla gerçekleştirecekleri saldırıların başarısını artırmak amacıyla damganın yerleştirilmiş olma ihtimali olan kayıt veya alanları belirlemeye çalışırlar. Bu nedenle damganın gizlendiği alan üzerinde gerçekleştirilen bozulmanın algılanamaması gerekir. Önerilen yöntem saldırılara karşı dayanıklılığı artırmak için bazı önlemler almıştır. Damgalanacak alanın seçiminde kullanıcı tarafından belirlenen gizli anahtar kullanılmıştır. Bu sayede saldırganın belirli bir kural dahilinde damgayı bulabilmesi önlenmiştir. Damga verisinin gömülürken alt bölümleme yapılması ve çıkarım aşamasında değerlendirme yönteminin kullanımı da saldırılar karşısında kaybedilen damga verisinin yeniden elde edilebilmesini sağlamıştır. Gerçekleştirilen testlerde [30] çalışması ve önerilen yöntemin damgalama sonrası veri seti alanlarının standart sapmasındaki değişim Şekil 3.6'da verilmiştir.



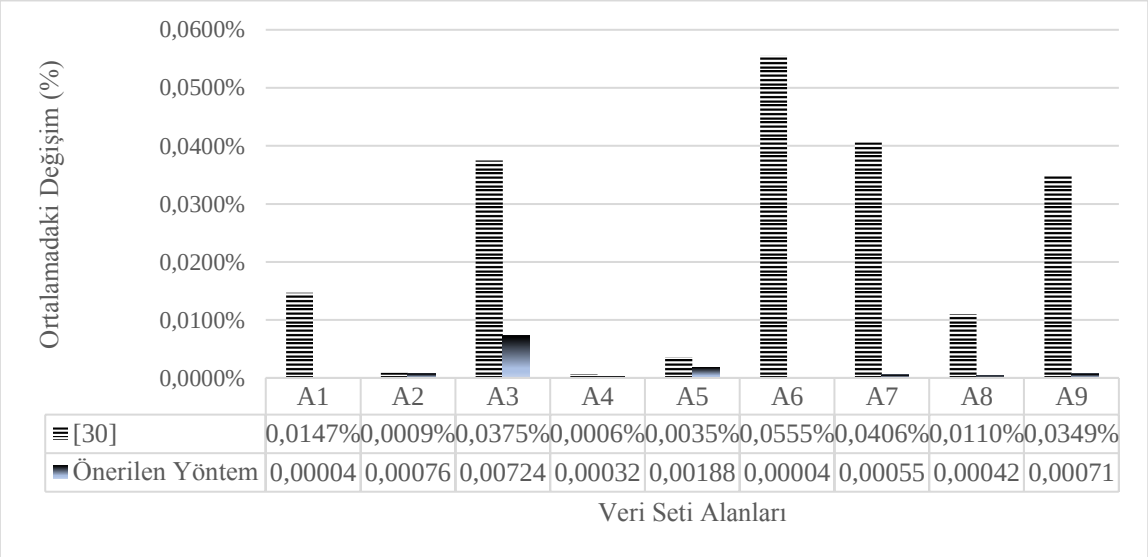
	A1	A2	A3	A4	A5	A6	A7	A8	A9
[30]	0,0224%	0,1379%	0,0272%	0,0689%	0,2254%	0,0056%	2,6868%	3,2223%	2,1220%
Önerilen Yöntem	0,1731%	0,2636%	0,2065%	0,2489%	0,2310%	0,1736%	0,2685%	0,2371%	0,2501%

Şekil 3.6. Damgalama sonrası veri seti alanlarının standart sapma değerindeki değişim oranları

Veri tabanından seçilen 3170 aday satırdan oluşan veri seti üzerinde yapılan damgalama sonrasında önerilen yöntemin sebep olduğu standart sapmadaki değişimin, tüm alanlar üzerine benzer oranlarda yansıtıldığı görülmektedir. [30] çalışmasının A7, A8 ve A9 alanlarında sebep olduğu standart sapma değişim oranı sırasıyla %2.6, %3.2 ve %2.1 iken,

önerilen yöntemin aynı alanlar üzerindeki sebep olduğu standart sapma değişim oranları %0.26, %0.23 ve %0.25 oranlarında kalmıştır. Diğer alanlardaki değişim GADEW algoritmasına göre yüksek olmasına rağmen aradaki farkın çok düşük olduğu gözlemlenmektedir.

Gerçekleştirilen bir başka inceleme ise damgalama sonrası veri seti alan değerlerinin ortalamasındaki değişimdir. 3170 aday satırdan oluşan veri seti üzerinde gerçekleştirilen simülasyon sonuçları Şekil 3.7'de verilmiştir. Sonuçlar incelendiğinde önerilen algoritmanın veri setindeki alan değerlerinin ortalaması üzerinde sebep olduğu değişimimin daha düşük olduğu görülmektedir. Bu da damganın saldırgan tarafından algılanmasını zorlaştırmaktadır.



Şekil 3.7. Damgalama sonrası veri seti alanlarının ortalama değerindeki değişim oranları

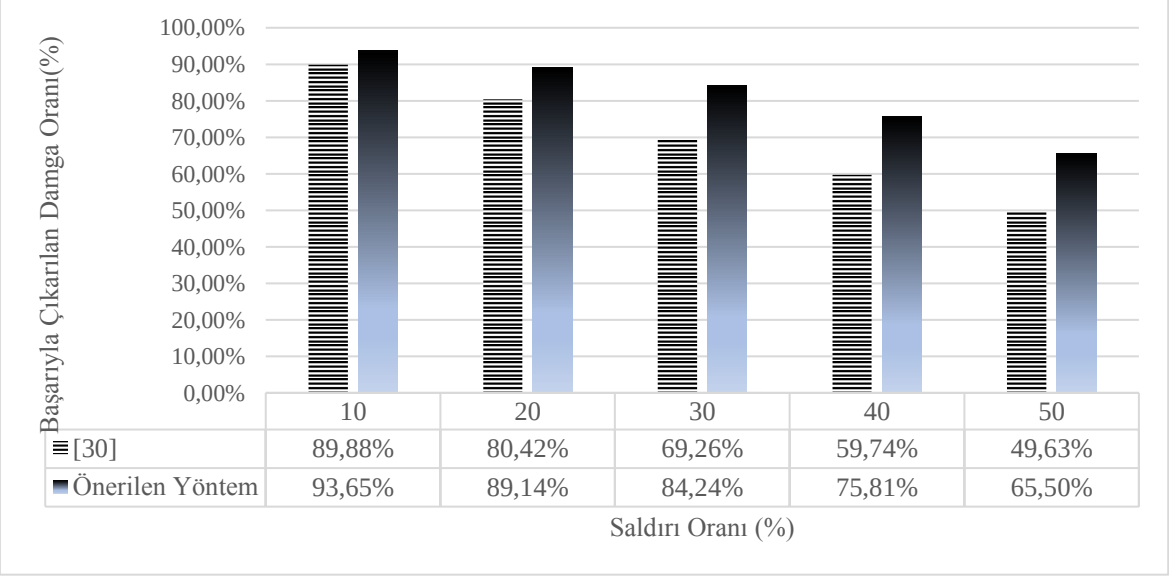
[30] çalışması damgalama sürecinde kullanılan alanların seçim oranlarını birbirinden farklı olabilmektedir. Bu da Fark Genişleme Tabanlı Damgalama yönteminden kaynaklanan bozulmanın damgalanan veri tabanı alanları üzerine eşit oranda dağıtılmamasına sebep olmaktadır. Önerilen yöntem ise dönüşümde kullanılan değerleri aynı alanın farklı satırlarından elde ettiği için, veri tabanı alan değerlerinin ortalama değerleri üzerinde sebep olduğu değişim daha düşük, standart sapmadaki sebep olduğu değişim ise orantılıdır.

3.2.4. Saldırılar

Saldırganlar damgalanmış veri üzerindeki damgayı yok etmek veya kendi damgalarını eklemek için farklı saldırılar düzenlemektedirler. Literatürde sayısal alanlara damga yerleştirme yapan çalışmaların sağlamlık analizinde sıklıkla kullanılan Alt Küme Silme, Bit Değiştirme, Satır Bazlı Komplike ve Alan Bazlı Komplike Saldırılar önerilen yöntem ve [20] çalışması üzerinde test edilmiştir.

3.2.4.1. Alt Küme Silme Saldırısı

Alt küme silme saldırısı damgalanan veriyi ortadan kaldırmak amacıyla veri setinde satırların silinmesi yöntemiyle gerçekleştirilir. Saldırgan, rastgele olarak belirlediği satırları veri setinden silerek damgalanmış verinin bütünlüğünü bozmaya veya tamamen ortadan kaldırmaya çalışmaktadır. FCT veri tabanından rastgele seçilmiş 1175 satırdan oluşan veri setine uygulanan Alt Küme Silme Saldırısı karşısında gömülen damgayı başarıyla belirleme oranları açısından önerilen çalışma ve [30] çalışmasının karşılaştırılması Şekil 3.8'de verilmiştir. Önerilen yöntemin saldırılara karşı dayanıklı olması amacıyla damgalanmak üzere seçilen aday satırlar arasında alt bölümleme uygulaması sayesinde veri setinin %50'sinin silinmesi durumunda %65.5 oranında damgayı çıkarım oranına ulaştığı görülmektedir.

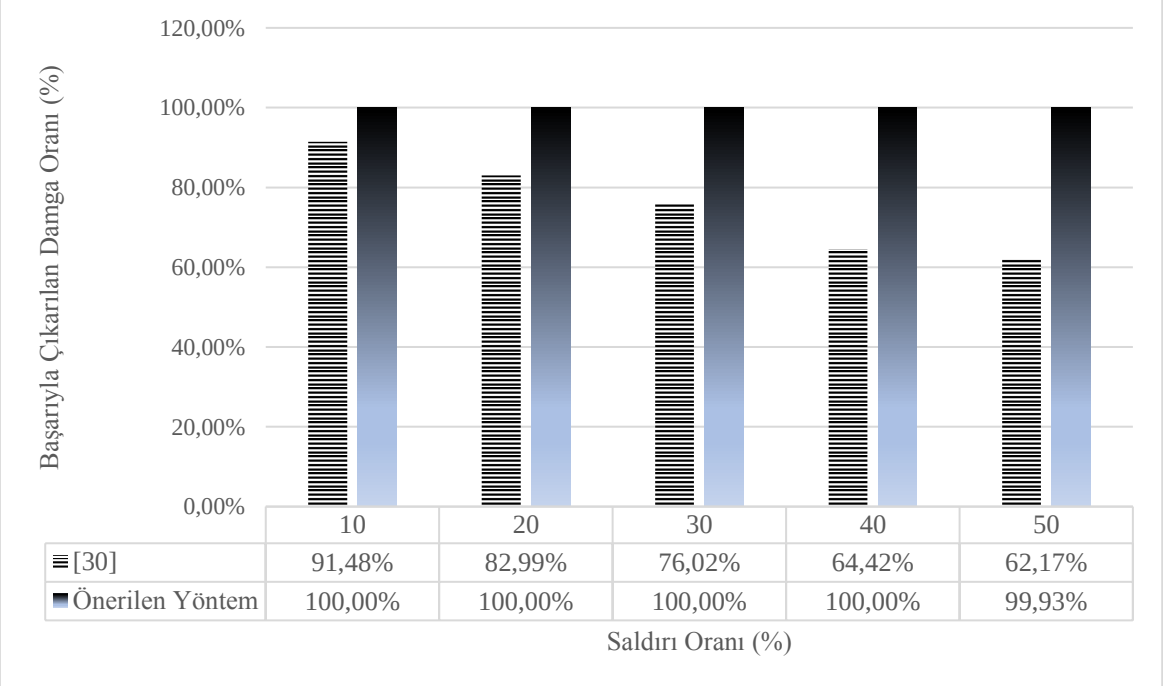


Şekil 3.8. Alt Küme Silme Saldırısı karşısında dayanıklılık sonuçları

[30] çalışması Alt Küme Silme Saldırısına karşı almış olduğu bir önlem bulunmamaktadır. Kaybedilen damga verisinin kurtarılabileceği bir çözüm sağlanmadığından kayıp telafi edilemeyecektir. Önerilen çalışma ise damga saklama sürecinde aday satırlar arasında bölümleme yapmaktadır. Bu sayede aynı damga parçasının farklı satır gruplarına saklanması sağlanmıştır. Saldırı sonucu aynı bölümdeki satır gruplarından sadece birinin kurtarılması damga parçasının kurtarılmasını sağlayacaktır. Bu sebeple Alt Küme Silme Saldırısına karşı daha dayanıklıdır.

3.2.4.2. Bit Değiştirme Saldırısı

Bu saldırı türünde rastgele olarak seçilen satırların tüm alanlarının en anlamsız bitleri üzerinde değiştirme işlemi gerçekleştirilir. Silme atağından farklı olması amacıyla saldırı yapılan satır alanlarının %50'si üzerinde bit değiştirme işlemi gerçekleştirilmiştir. Önerilen yöntem ve [30] çalışmasının saldırıya karşı dayanıklılık sonuçları Şekil 3.9'da verilmiştir.



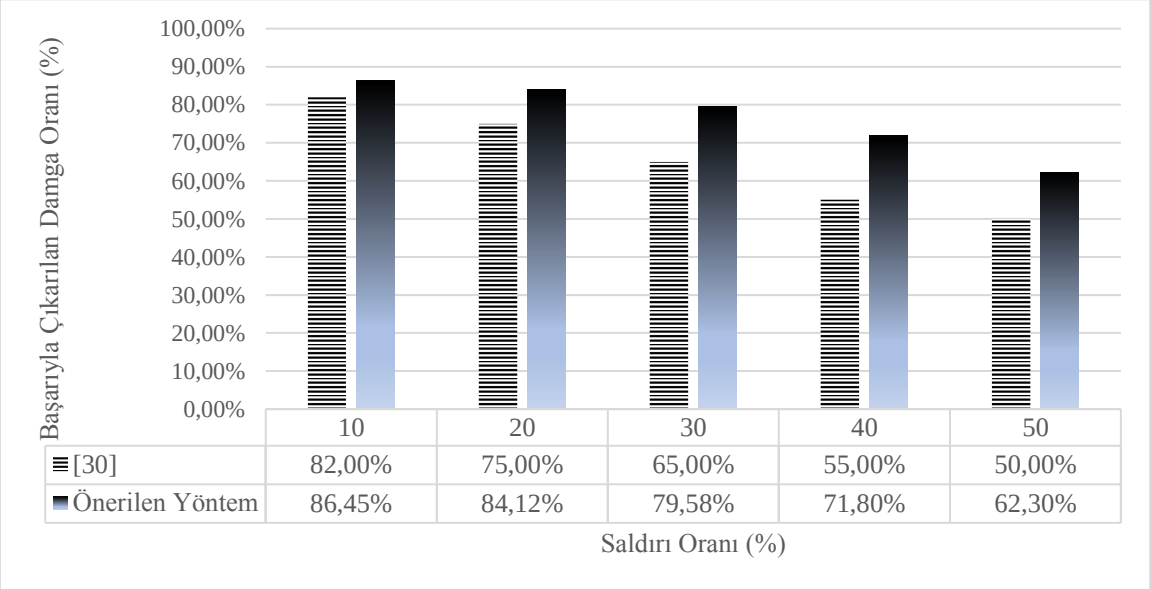
Şekil 3.9. Bit Değiştirme Saldırısı karşısında dayanıklılık sonuçları

Önerilen algoritmanın alt bölümleme özelliği sayesinde değiştirilen alan değerinin bölümün diğer üyelerinden çıkarılabildiği durumlarda damga verisinin tamamının geri elde edilebildiği görülmüştür. Damgalanacak aday satırların %50'sine bit değiştirme atağı

uygulandığında bile, önerilen algoritmanın damgalanmış verinin %99'unu başarıyla geri döndürebildiği görülmektedir.

3.2.4.3. Satır Bazlı Çoklu Saldırı

Tek tür saldırının başarılı olmamasına karşın saldırganların birden çok saldırıyı birlikte uyguladığı saldırı türüdür. Damgalanmış veriye 3 farklı saldırı art arda uygulanır. Yanlış damganın çıkarılması amacıyla ilk olarak Alt Küme Ekleme saldırısı gerçekleştirilir. Oluşan yeni veri tabanına damga yerleştirilmiş satırların yok edilmesi için Alt Küme Silme saldırısı uygulanır. Son olarak Bit Değiştirme saldırısı uygulanarak damga çıkarım oranının düşürülmesi amaçlanmıştır. Saldırı sonrası önerilen algoritmanın ve [30] çalışmasının dayanıklılık oranları Şekil 3.10'da verilmiştir. Saldırı oranının %50 olduğu durumda bile önerilen algoritmanın damgalanan verinin %62'sini başarıyla elde ettiği görülmüştür.



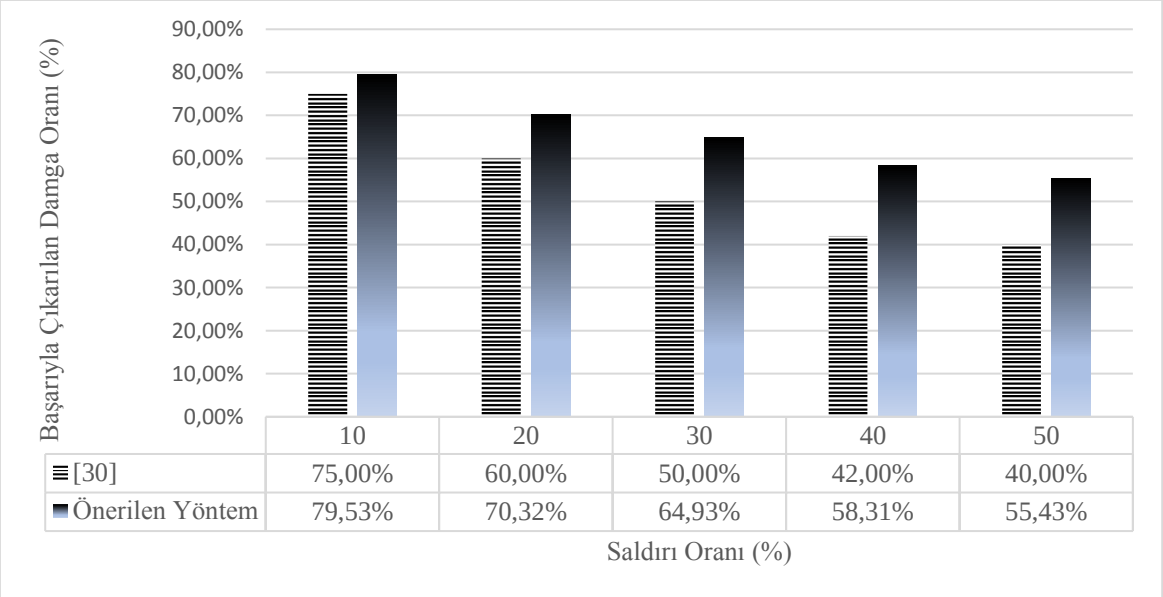
	10	20	30	40	50
[30]	82,00%	75,00%	65,00%	55,00%	50,00%
Önerilen Yöntem	86,45%	84,12%	79,58%	71,80%	62,30%

Şekil 3.10. Satır Bazlı Çoklu Saldırı karşısında dayanıklılık sonuçları

3.2.4.4. Alan Bazlı Çoklu Saldırı

Bu saldırı türünde saldırı yapılacak veriye 2 farklı saldırı art arda uygulanır. Saldırı oranı belirlendikten sonra ilk olarak bit değiştirme saldırısı veri setine uygulanır. Ardından aynı oranda rastgele belirlenmiş satırın rastgele belirlenen alanına güncelleme saldırısı yapılarak değerleri değiştirilir. Saldırı sonrası önerilen yöntemin ve [30] çalışmasının

dayanıklılık oranları Şekil 3.11'de verilmiştir. Saldırı oranının %50 olarak alındığı durumda bile önerilen algoritmanın damgalanan verinin %55'sini başarıyla elde ettiği görülmüştür.



Şekil 3.11. Alan Bazlı Çoklu Saldırı karşısında dayanıklılık sonuçları

3.3. İlişkisel Veri Tabanlarında Homoglif Değişim Tabanlı XML Damgalama Yönteminin Değerlendirilmesi ve Elde Edilen Sonuçlar

Deneylerin üzerinde yapıldığı veri tabanı, "1998statics.xml" isimli benzer çalışmalarda da kullanılan istatistiksel verileri içeren XML dosyasından oluşturulmuştur [77-79]. 1226 düğümden oluşan XML dosyanın her bir düğümü veri setinde birer XML veri alanı haline getirilmiştir. Örnek XML düğüm yapısı Şekil 3.12'de verilmiştir.

Simülasyonlar MicroSoft SQL Server veri tabanı yönetim sistemi üzerinde MicroSoft Visual Studio IDE ortamında C# programlama dili ile kodlanmıştır. Bu bölümde önerilen yöntem ve literatürdeki benzer çalışmalar farklı kriterler dikkate alınarak karşılaştırılmıştır. Damgalama algoritmalarının en önemli özelliklerinden kapasite ve algılanabilirlik analizi gerçekleştirilmiştir. XML alanlar üzerinde damgalama gerçekleştiren algoritmalara yapılan saldırılar, metin ve nümerik alanlar üzerinde damgalama gerçekleştiren algoritmalara yapılan saldırılara göre farklılık göstermektedir. Bu sebeple XML verinin bütünlüğünü ve kullanılabilirliğini bozmadan damgayı yok etmeye yönelik literatürde gerçekleştirilen saldırılar ile önerilen algoritmanın sağlamlığı test edilmiştir.

```
<PLAYER>
  <NUMBER>2989</NUMBER>
  <SURNAME> Martinez</SURNAME>
  <GIVEN_NAME>Dennis </GIVEN_NAME>
  <THROWS />
  <POSITION>Relief Pitcher</POSITION>
  <WINS>4</WINS>
  <LOSSES>6</LOSSES>
  <SAVES>2</SAVES>
  <GAMES>53</GAMES>
  <GAMES_STARTED>5</GAMES_STARTED>
  <COMPLETE_GAMES>1</COMPLETE_GAMES>
  <SHUT_OUTS>1</SHUT_OUTS>
  <ERA>4.45</ERA>
  <INNINGS>91</INNINGS>
  <HOME_RUNS>109</HOME_RUNS>
  <RUNS>8</RUNS>
  <EARNED_RUNS>53</EARNED_RUNS>
  <HIT_BATTER>45</HIT_BATTER>
  <WILD_PITCHES>3</WILD_PITCHES>
  <BALK>2</BALK>
  <WALKED_BATTER>0</WALKED_BATTER>
  <STRUCK_OUT_BATTER>19</STRUCK_OUT_BATTER>
</PLAYER>
```

Şekil 3.12. Örnek XML veri yapısı

3.3.1. Kapasite ve Algılanabilirlik Analizi

Saldırganlar damgalama sonrası veri setindeki değişimi belirleyerek, damgayı bozmak veya ortadan kaldırmak için saldırılar yapmaktadırlar. Bu sebeple damga verisinin fark edilebilirliği düşük olması çok önemlidir. Önerilen çalışmada Homoglif Dönüşüm yöntemi kullanımı sayesinde, damga verisi metinler üzerine eklenmesine rağmen fark edilebilirliği minimum düzeydedir. Homoglif Dönüşümde kullanılacak karakterlerin algılanabilirlik

oranının kullanıcı tarafından belirlenebilmesi, kullanıcının damgalama kapasitesi ile algılanabilirlik arasındaki dengeyi gereksinimine göre belirleyebilmesini sağlamaktadır. Yüksek damgalama kapasitesi yerine damganın gizliliğinin önemli olduğu tıbbi, güvenlik ve askeri veri tabanlarındaki uygulamalarda kullanıcı saldırganlar tarafından fark edilebilirliği azaltmak için algoritmanın algılanabilirlik oranının 0'a yakın seçebilecektir. Web ortamında bilginin sunulduğu veri tabanları gibi algılanabilirlikten daha çok kapasitenin önem kazandığı durumlarda algılanabilirlik oranının 1'e yakın olması ile homoglif dönüşümde kullanılacak karakter kümesinin büyümesi sağlanarak damgalama kapasitesi maksimize edilebilmektedir. Şekil 3.13'de örnek bir XML kaydın orijinal ve farklı algılanabilirlik oranları ile damgalanmış halinin HTML çıktıları sunulmaktadır.

Pp	Data
Original	Trabzon Milletvekili Mustafa Reşit Tarakçıoğlu ve 28 arkadaşının verdiği teklifin TBMM'de 20 Mayıs 1955 tarih ve 6594 sayılı kanunla kabul edilmesi ile kurulmuş olan Karadeniz Teknik Üniversitesi İstanbul ve Ankara illeri dışında kurulan ilk üniversitedir.
0	Trabzon Milletvekili Mustafa Reşit Tarakçıoğlu ve 28 arkadaşının verdiği teklifin TBMM'de 20 Mayıs 1955 tarih ve 6594 sayılı kanunla kabul edilmesi ile kurulmuş olan Karadeniz Teknik Üniversitesi İstanbul ve Ankara illeri dışında kurulan ilk üniversitedir.
0.5	Trabzon Milletvekili Mustafa Reşit Tarakçıoğlu ve 28 arkadaşının verdiği teklifin TBMM'de 20 Mayıs 1955 tarih ve 6594 sayılı kanunla kabul edilmesi ile kurulmuş olan Karadeniz Teknik Üniversitesi İstanbul ve Ankara illeri dışında kurulan ilk üniversitedir.
1	Trabzon Milletvekili Mustafa Reşit Tarakçıoğlu ve 28 arkadaşının verdiği teklifin TBMM'de 20 Mayıs 1955 tarih ve 6594 sayılı kanunla kabul edilmesi ile kurulmuş olan Karadeniz Teknik Üniversitesi İstanbul ve Ankara illeri dışında kurulan ilk üniversitedir.

Şekil 3.13. Örnek XML verisine farklı P_p oranları ile damgalama sonuçları

P_p ifadesinin algılanabilirlik oranını temsil ettiği şekilde, ilk satırdaki veri orijinal XML verisinin HTML çıktısını göstermektedir. P_p değerinin 0 alınarak algılanabilirliğin minimum olduğu durumda damgalama sonrası elde edilen metin ikinci satırda verilmiştir. 106 karakter üzerinde homoglif dönüşüm yapılmasına rağmen damgalanan karakterlerin göz ile algılanabilmesi çok zordur. P_p değerinin 0.5 alınarak gerçekleştirilen damgalama sonrası elde edilen metin üçüncü satırda verilmiştir. Algılanabirlik oranının artması, homoglif dönüşümde kullanılacak karakter kümesinin büyümesini sağladığından 110 karakterlik damgalama kapasitesine ulaşılmıştır. Damgalama kapasitesinin önemli olduğu son satırda, P_p parametresi için 1 değeri seçilmiştir. Bu durumda damga olarak yerleştirilmiş bit sayısı

143'e yükselmiştir. Önerilen algoritmada damga verisinin 0 bit değerleri için homoglif dönüşüm yapılmayacağından, damga verisi içerisindeki 0 bit sayısının artması damgalanan verinin gizliliğini artıracaktır. Şekil 3.13'de örnek XML verisine damgalanacak veri bitlerinin tamamı 1 seçilerek damga yerleştirilecek her bir karakter homoglif karşılığı ile değiştirilmiştir. Bu senaryo damga verisinin güvenliğinin en düşük olduğu durumdur. Buna rağmen P_p değerinin 0 seçildiği durumda damgalanan karakterlerin algılanabilmesi mümkün değildir.

Literatürde XML verilere damgalama gerçekleştiren diğer çalışmaların saldırılara karşı fark edilemezlik durum karşılaştırmaları Tablo 3.1.'de verilmiştir. Tabloda algoritmaların fark edilemezlik durumları, örnek XML verisi ve bu örnek veriye damga verisi yerleştirmelerinden sonra elde edilen damgalanmış veri sunulmuştur.

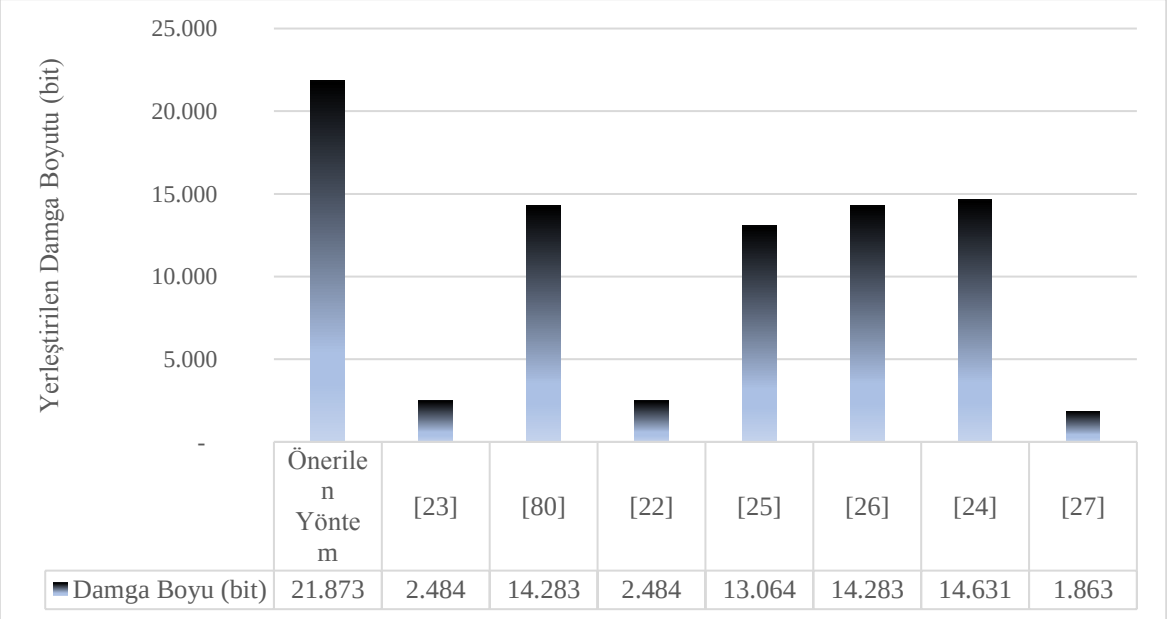
Tablo 3.1. Literatürde XML damgalama yapan bazı algoritmaların fark edilebilirlik durumları

Algoritmalar	*Fark Edilememe*	*Orijinal Veri*	*Damga Yerleştirilmiş Veri*
Önerilen Yöntem	✓	<city>Los Angeles</city>	<city>Los Angeles</city>
[23]	✕	<city>Los Angeles</city>	<city>L.A.</city>
[80]	✕	<city>Los Angeles</city>	<city >Los Angeles</city >
[22]	✕	<city>Los Angeles</city>	<city>L.A.</city>
[25]	✕	<city>Los Angeles</city>	<city>LoS AnGelEs</city>
[26]	✕	<city>Los Angeles</city>	<city>Los Angeles</city >
[24]	✕	<city>Los Angeles</city>	<CiTy>Los Angeles</CiTy>
[27]	✕	<city>Los Angeles</city>	<city>Los An gele s</city>

[23] ve [22] çalışmalarında damgalama sırasında kelimelerin sinonimleri ile değiştirilmesinden dolayı orijinal dosya boyutu değişecek, metinlerde oluşabilecek anlamsal problemlerin sonucu damga verisinin algılanması kolaylaşacaktır. [26, 27, 80] çalışmalarında [23] ve [22] çalışmalarına benzer şekilde sinonim kullanımı yanı sıra "boşluk" ekleme yaklaşımından dolayı oluşacak ekstra boşluklar saldırganların dikkatini

çekecektir. [25] çalışmasında damga verisinin eklenmesi aşamasında büyültülen karakterler sebebiyle aynı kelimede yazım kurallarına uymayan büyük-küçük harf olması damganın fark edilebilmesine sebep olacaktır. [24] çalışmasında [25] çalışmasında veriler üzerinde gerçekleştirilen karakter büyültme işlemi etiket karakterleri üzerinde yapıldığından, damga verisinin fark edilebilme ihtimali yüksek olacaktır. Literatürdeki XML verilere damgalama gerçekleştiren diğer çalışmalar ile karşılaştırıldığında, önerilen algoritmanın damga verisi fark edilemezdir.

Kapasite testlerinde veri tabanında bulunan 1226 satırdan kullanıcının seçmiş olduğu gizli anahtar vasıtasıyla belirlenen 621 aday satıra damga verisi gömülmüştür. Önerilen yöntem ve literatürde XML verilerin damgalanmasını sağlayan algoritmaların damgalama kapasitelerinin karşılaştırılması Şekil 3.14'de verilmiştir. Önerilen yöntem kullandığı Homoglif Dönüşüm sayesinde metinler üzerinde yüksek damgalama kapasitesine sahiptir. Örnek veri seti için 621 aday satır üzerinde 21.873 bit veriyi saklayabilmiştir. Bu damgalama kapasitesine en yakın çalışma etiket büyütme yöntemini kullanarak 14.631 bit veri gömebilen [24] çalışmasıdır.



Şekil 3.14. Literatürde XML damgalama yapan algoritmaların damgalama kapasitelerinin örnek veri setinde karşılaştırılması

3.3.2. Sağlamlık Analizi

Sağlamlık analizi damgalama yapılmış veri tabanına saldırı yapıldıktan sonra damga çıkarım algoritması ile elde edilebilen damga verisinin doğruluk oranı dikkate alınarak yapılmaktadır.

XML içeren veri tabanlarına yapılan saldırılar, sadece nümerik veya metin değerlerden oluşan veri tabanlarına yapılan saldırılara göre farklılık göstermektedir. Bu saldırılar Element Sıra Karıştırma, Etiket Ad Karıştırma, İçerik Silme, Alan Silme veya Karıştırma saldırılardır.

XML verinin yapısal özelliğinden dolayı, bir etiketin içerdiği alt etiketlerin sıralarının değişimi verinin bozulmasına sebep olmaz. Element Sıra Karıştırma saldırısı XML'in bu özelliği kullanılarak yapılmaktadır. Saldırganlar XML verinin yapısı üzerinde değişiklik yaparak veriyi bozmadan damgayı ortadan kaldırmaya çalışmaktadırlar.

Veriyi barındıran XML açma ve kapama etiketlerinin büyük/küçük harf duyarlı bir şekilde birbirinin aynı olması gerekir. Fakat etiket karakterlerinin büyük veya küçük olması veriyi etkilememektedir. Etiket Ad Karıştırma saldırı türünde, XML etiket karakterlerinin büyültülüp/küçültülmesi ile etikete yerleştirilen damganın zarar görmesi sağlanabilir.

Alan Silme veya Karıştırma saldırı türünde ise XML etiket özelliklerinin yerlerinin değiştirilmesi veya etiket özelliklerinin silinmesi, etiket özellikleri kullanılarak yerleştirilen damga verisinin kaybolmasına yol açabilir.

XML etiketlerinin içerdiği verinin bir kısmının veya tamamının silinmesi yöntemiyle gerçekleştirilen İçerik Silme saldırısı damga verisinin eklendiği içerikleri hedef aldığında damganın kaybolmasına sebep olacaktır.

Tüm bu saldırı türleri örnek veri seti üzerinde gerçekleştirilerek sonuçları Tablo 3.2.'de sunulmuştur. [23], [80], [22] ve [24] çalışmaları Element Sıra Karıştırma saldırısından etkilenecek ve yerleştirdikleri damganın tamamını çıkaramayacaklardır. [80] ve [26] çalışmalarında damga yerleştirme sürecinde XML düğümünün kapama etiketine boşluk eklendiğinden Element Ad Karıştırma saldırısından etkilenecek ve damganın bütünlüğü bozulacaktır. Benzer şekilde [24] çalışmasında damgalama işleminde etiketlerin karakterleri büyük/küçük karakter karşılıkları ile değiştirildiğinden Element Ad Karıştırma saldırısı bu çalışmada da damga verisinin bozulmasına sebep olacaktır. Alan Silme veya Karıştırma saldırısı ise [80] çalışmasını etkileyecek ve yerleştirilen damganın kaybolmasına sebep olacaktır.

Tablo 3.2. Literatürde XML damgalama yapan bazı algoritmaların saldırılara karşı dayanıklılıkları

Algoritma	*Saldırı*			
	Element Sıra Karıştırma	*Etiket Ad Karıştırma*	*İçerik Silme*	*Alan Silme veya Karıştırma*
Önerilen Yöntem	Etkilenmez	Etkilenmez	Etkilenir	Etkilenmez
[23]	Etkilenir	Etkilenmez	Etkilenir	Etkilenmez
[80]	Etkilenir	Etkilenir	Etkilenir	Etkilenir
[22]	Etkilenir	Etkilenmez	Etkilenir	Etkilenmez
[25]	Etkilenmez	Etkilenmez	Etkilenir	Etkilenmez
[26]	Etkilenmez	Etkilenir	Etkilenir	Etkilenmez
[24]	Etkilenir	Etkilenir	Etkilenir	Etkilenmez
[27]	Etkilenmez	Etkilenmez	Etkilenir	Etkilenmez

Önerilen yöntem damga yerleştirme ve damga çıkarma işlemi öncesinde XML düğümlerini alfabetik olarak sıraladığından Element Sıra Karıştırma saldırısından etkilenmeyecektir. Veri saklama işlemi etiketler içerisine yapılmadığından Etiket Ad Karıştırma saldırısı da önerilen algoritmanın damgalama başarını etkilemeyecektir. Benzer şekilde etiket alanları üzerinde yapılan değişiklik veya silme saldırısı önerilen çalışmanın damgalama sağlamlığını olumsuz yönde etkilemeyecektir. İncelenen saldırılar içerisinde önerilen çalışmanın sağlamlığını etkileyen tek saldırı literatürdeki tüm çalışmaları da etkileyen, veri kaybın kaçınılmaz olduğu İçerik Silme saldırısıdır.

Gerçekleştirilen saldırılar sonrasında önerilen çalışmanın karşılaştırılan diğer algoritmalara göre XML verinin silinmesi dışındaki tüm saldırılara karşı dayanaklı ve saldırganlar tarafından fark edilemez olduğu görülmüştür.

3.3.3. Önerilen Yöntemin Literatürdeki Benzer Yöntemler ile Karşılaştırılması

Bu bölümde literatürde XML veri damgalama üzerine yapılan çalışmalar ile önerilen yöntemin özellikleri karşılaştırılmıştır. Çalışmaların destekledikleri veri türü, damga

çıkarımı için ihtiyaç duyulan ekstra bilgilerin saklanma ihtiyacı, damgalama sonrası veri boyutunun değişimi, damga yerleştirme yöntemi, geri dönüşüm desteği, veri üzerinde bozulmaya sebep olup olmadığı gibi özellikler dikkate alınarak oluşturulan karşılaştırma tablosu Tablo 3.3.'de verilmiştir.

Tablo 3.3. Literatürde XML damgalama yapan bazı algoritmaların özellik karşılaştırmaları

Çalışma	Yıl	Damga Veri Türü (S: Sayısal M: Metin)		Damga Çıkarım İçin Ekstra Bilgi Saklama Gerekliliği	Veri Boyutunu Değiştirme	Damgalama Metodu (S: Sayısal M: Metin)		Orijinal Veriye Geri Dönme	Bozulmaya Sebep Olma
		S	M			S	M		
Önerilen Yöntem	2017	✓	✓	✕	✕	Homo glif	Homoglif	✓	✕
[23]	2006	✓	✓	✓	✓	LSB	Sinonim	✕	✓
[80]	2001	✕	✓	✕	✓	✕	Sinonim, Boşluk, Yeniden Sıralama	✕	✓
[22]	2005	✓	✓	✓	✓	LSB	Sinonim	✕	✓
[25]	2010	✕	✓	✓	✕	✕	Büyük/ Küçük Harf	✕	✓
[26]	2012	✕	✓	✓	✓	✕	Boşluk	✓	✕
[24]	2006	✕	✓	✕	✕	✕	Büyük/ Küçük Kodlama	✓	✕
[27]	2012	✕	✓	✓	✓	✕	Boşluk	✓	✓

XML veri alanı içerisinde tutulan verinin türü metin veya nümerik olabilmektedir. Önerilen yöntem XML alanın içerdiği veri türünden bağımsız olduğu için XML etiketlerin içerdiği metin ve nümerik veri türlerinin her ikisine de damgalama desteğine sahiptir. [23] ve [22] çalışmaları metin ve nümerik veri türlerinin her ikisini de desteklerken [80], [25], [26], [24] ve [27] çalışmaları sadece metin veriler üzerinde damgalama yapabilmektedirler.

Önerilen algoritmada yerleştirilen damga verisinin çıkarımı için gerekli olan bilginin farklı bir ortamda saklanması gerekmemektedir. Damga çıkarım süreci için damgalama aşamasında kullanıcı tarafından belirlenen gizli anahtar ve algılanabilirlik oranı yeterlidir. [80] ve [24] çalışmaları damga çıkarım işlemi için farklı bir ortamda saklanan veriye ihtiyaç duymazken [23], [22], [25], [26] ve [27] çalışmaları damganın geri dönüşümü sürecinde gerekli olan veriyi farklı bir ortamda saklamak zorundadırlar.

Orijinal veri tabanına yerleştirilen damga verisinin boyutu ne kadar fazla olursa olsun, önerilen algoritma XML verinin veya dosya boyutunun değişmesine sebep olmamaktadır. Literatürde incelenen benzer uygulamalardan [25] ve [24] çalışmaları da bu özelliğe sahiptir. Fakat [23], [80], [22], [26] ve [27] çalışmaları damgalama sonrası orijinal verinin boyutunda değişime sebep olarak saldırılara karşı zaaf göstermektedirler.

Damgalama algoritmalarının yerleştirdiği damga verisini çıkardıktan sonra damgalama öncesi orijinal veriyi hatasız olarak elde edebilmesi istenir. Geri dönüşüm özelliği olarak bilinen bu özellik önerilen algoritma ve literatürde incelenen algoritmalar arasında [26], [24] ve [27] çalışmalarında sağlanmaktadır. Buna karşın [23], [80], [22] ve [25] çalışmaları damga çıkarım sonrası orijinal veriyi elde edememektedirler.

Damgalama işleminin orijinal veri üzerinde olabildiğince az miktarda değişime sebep olması istenir. Bu sayede verinin kullanılabilirliği devam edecek ve saldırganlar tarafından damga verisinin saklandığı değerlerin belirlenmesi de zorlaşacaktır. Önerilen yöntem kullanmış olduğu Homoglif Dönüşüm sayesinde damgalama gerçekleştirdiği verilerin içerdiği nümerik değerler üzerinde bir değişim, metinler üzerinde farklı bir kelime kullanımı veya fazladan boşluk ekleme gibi işlemler gerçekleştirmediğinden orijinal veri üzerinde tahribata neden olmamaktadır. Damga bit değerinin “1” olduğu durumda orijinal karakterlerin yerlerine koyulan Unicode karakterlerin birbirlerine benzerliği sayesinde değişimin gözle belirlenmesi zorlaşmıştır. Damga bit değerinin “0” olduğu durumda karakter değişimi yapılmayacağından bozulma olmayacaktır. Literatürde incelenen çalışmalardan [26] ve [24] çalışmaları orijinal metin üzerinde değişikliğe sebep olmazken [23], [80], [22], [25] ve [27] çalışmaları damga yerleştirme sürecinde metinler üzerinde sinonim kelime kullanımı, boşluk ekleme, karakter büyültme/küçültme gibi işlemlerden dolayı orijinal verinin bozulmasına sebep olmaktadırlar.

Önerilen algoritma karakterler üzerinde Homoglif dönüşüm gerçekleştirdiğinden fark edilemezliği yüksektir. Damga verisini çıkardıktan sonra damgalama öncesi orijinal veriyi elde edebilmektedir. XML verinin barındırdığı nümerik ve metin veri türlerine damga yerleştirebilirken, damga çıkarım işlemi için farklı bir ortamda saklanan veriye ihtiyacı yoktur ve damgalama sonrası orijinal verinin bozulmasına sebep olmamaktadır. Tüm bu özellikleri bünyesinde barındıran tek çalışma olarak gerçekleştirilen saldırılara karşı dayanıklılığı üst düzeydedir.

3.4. Ateş Böceği Algoritması ile İlişkisel Veri Tabanı Damgalamada Yeni Bir Yaklaşımın Değerlendirilmesi ve Elde Edilen Sonuçlar

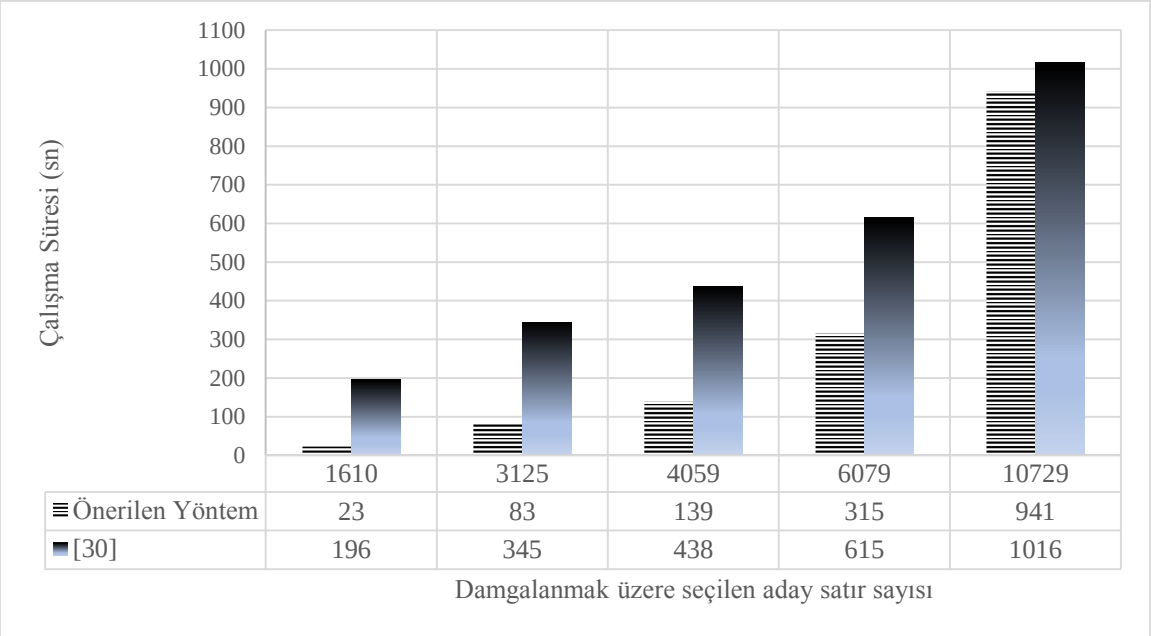
Deneysel metotlar MicroSoft Visual Studio yazılım geliştirme ortamında C# programlama dili kullanılarak geliştirilmiştir. Veri tabanı sistemi olarak MicroSoft SQL Server kullanılmıştır. Kaliforniya Üniversitesi tarafından sağlanan FCT veri tabanı damgalanmak üzere seçilmiştir. Veri seti 54 nümerik alan ve 581.012 satırdan oluşmaktadır. Gerçekleştirilen testlerde 54 alanın 9 tanesi kullanılmıştır. Test sonuçları 3 alt başlıkta incelenmiştir. İlk kısımda benzer çalışmalardan [29] ve [30] çalışmaları ile önerilen yöntem karşılaştırılmıştır. İkinci kısımda literatürde metotların sağlamlıklarının testinde kullanılan saldırılar karşısında çalışmanın sağlamlığı ve performansı test edilmiştir. Üçüncü kısımda ise yöntemde kullanılan bazı parametrelerin çalışmanın performansı üzerine etkisi irdelenmiştir.

3.4.1. Performans Analizi

Bu bölümde, gerçekleştirilen çalışmanın [30] çalışması ile çalışma süresi, damgalama kapasitesi ve sebep olunan bozulma miktarı kriterleri üzerinden karşılaştırılması yapılmıştır.

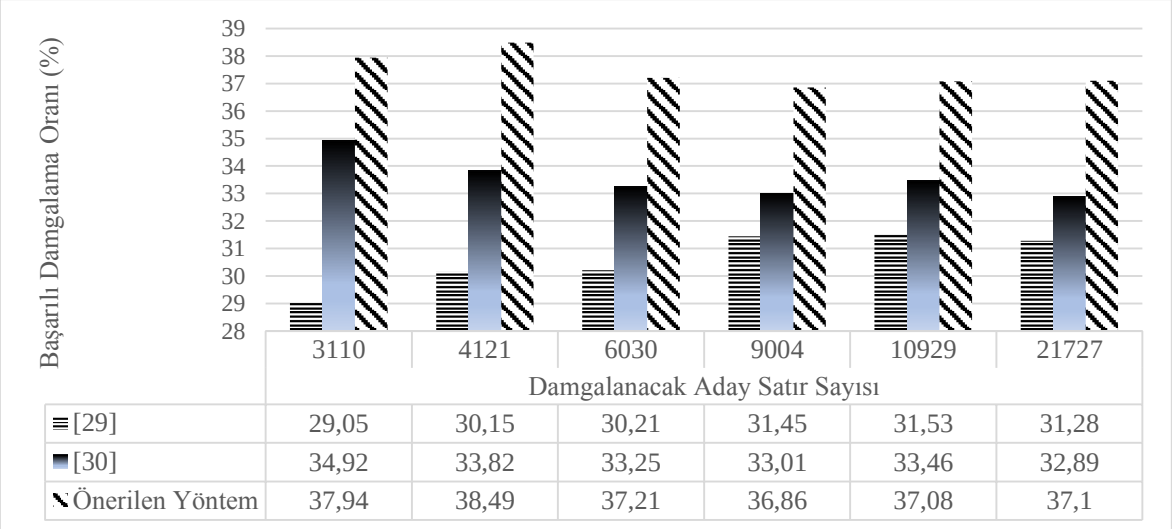
İlk olarak [30] çalışması ile yöntemin çalışma süresi karşılaştırılmıştır. Her iki algoritma da 1610, 3125, 4059 ve 6079 satırlık veri setleri üzerinde 10'ar kez koşulmuştur. Algoritmaların farklı veri setleri üzerinde ortalama çalışma süreleri Şekil 3.15'de verilmiştir. Sonuçlar incelendiğinde önerilen yöntemin aynı veri seti üzerinde [30] çalışmasından daha hızlı olduğu açıkça görülmektedir. Örneğin, 1610 satırlık veri setinde önerilen yöntem en iyi çözümü 23 saniyede bulurken, [30] çalışması en iyi çözümü 196 saniyede bulabilmektedir. Şekil 3.15'den ayrıca önerilen yöntemin yüksek sayıda kayıt tutan veri setleri üzerinde de çalışma hızı olarak performanslı olduğu anlaşılmaktadır.

Çalışma süresinin düşüklüğü yanında damgalama kapasitesinin yüksek olması da sistemin kullanılabilirlik alanlarını artırmaktadır. İdeal damgalama sistemlerinin büyük miktarda damga verisini, orijinal verinin kullanılabilirliğini bozmadan yerleştiriyor olması beklenir. Yüksek damgalama kapasitesi, aynı veri setine daha yüksek miktarda veri gömülebileceği anlamına gelmektedir.



Şekil 3.15. Farklı sayıda kayıt barındıran veri setlerinde önerilen yöntem ve [20] çalışmalarının ortalama çalışma süreleri

Önerilen yöntemin [29] ve [30] çalışmaları ile damgalama kapasiteleri üzerinden karşılaştırılması için rastgele seçilen 3110, 4121, 6030, 9004, 10929 ve 21727 satırlık 6 farklı veri seti oluşturulmuştur. Test sonuçları Şekil 3.16'da verilmiştir.

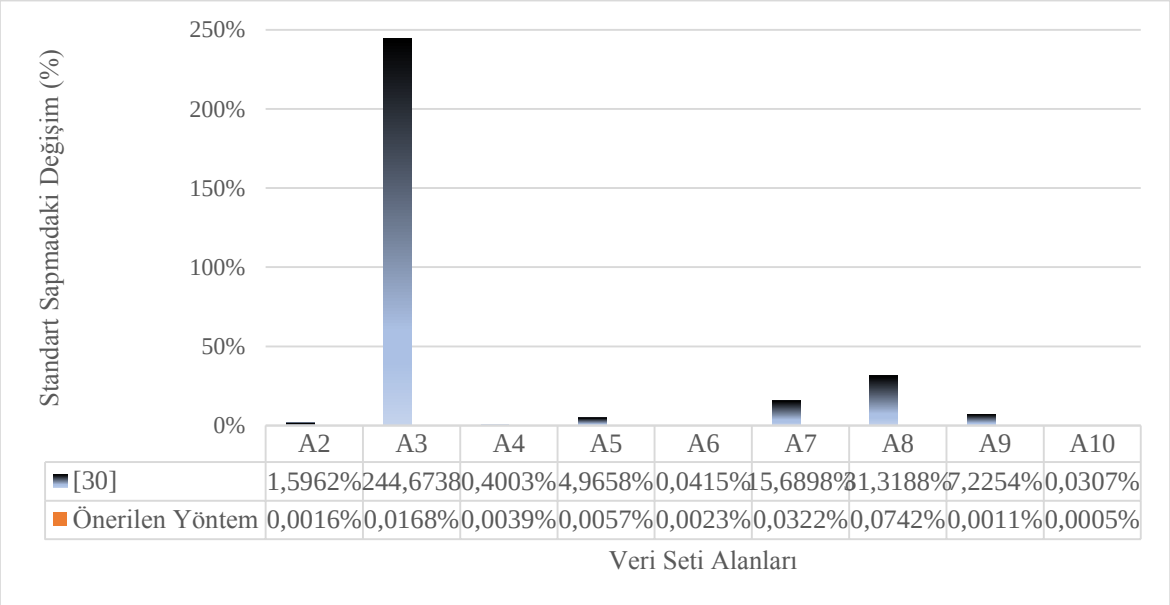


Şekil 3.16. Önerilen Yöntem, [29] ve [30] çalışmalarının damgalama kapasiteleri

Sonuçlar incelendiğinde önerilen yöntemin damgalama kapasitesinin diğer algoritmaların üstünde olduğu görülmektedir. Örneğin 21727 kayıtlık veri setinde [29] ve [30] algoritmaları %31 ve %32 oranında damgalama kapasitesine çıkabilirken, önerilen

çalışma %37 oranına ulaşabilmiştir. Ayrıca Şekil 3.16'dan önerilen yöntemin farklı büyüklükteki veri setlerinde de benzer şekilde yüksek damgalama kapasitesine eriştiği görülmektedir.

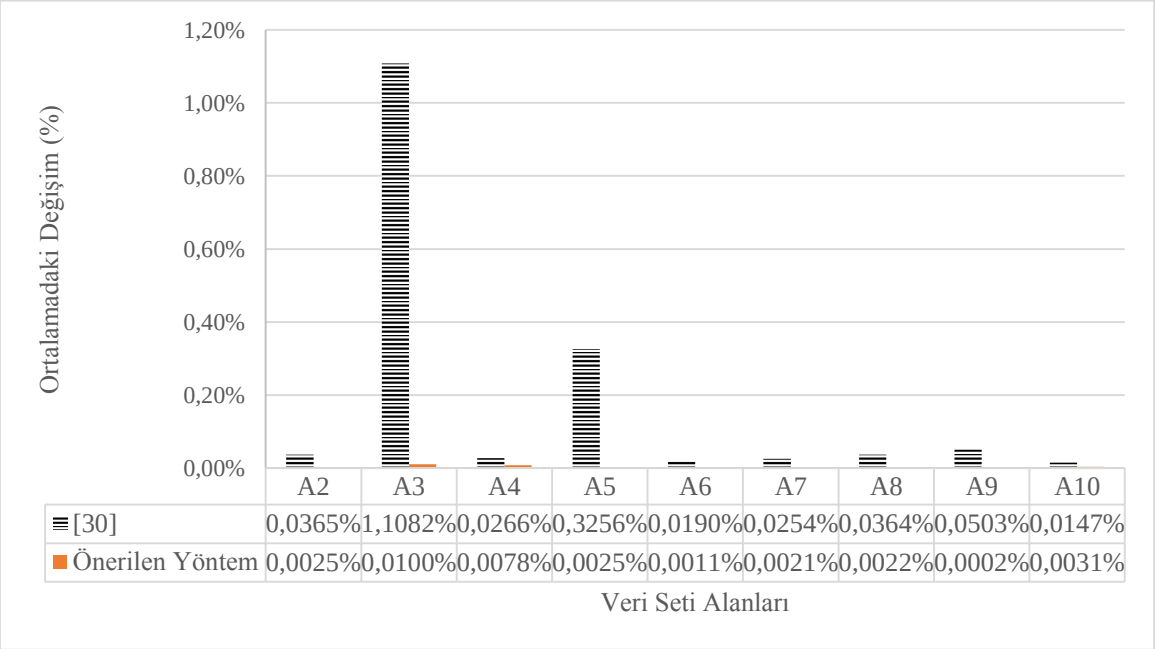
Üçüncü performans kriteri olarak damgalama işleminin orijinal veri tabanı üzerinde sebep olduğu bozulma incelenmiştir. Saldırganlar damgayı yok etmeden önce damganın bulunduğu kayıtları belirlemeye çalışmaktadırlar. Bu noktada damganın veri üzerinde sebep olduğu bozulma saldırganlara yol göstermektedir. Kayıtlar üzerinde ortaya çıkacak belirlenebilir veya anlamsız veri değişimleri, damga konumunun saldırganlar tarafından belirlenmesini ve damganın yok edilmesine sebep olacaktır. Damgalama algoritmasının sebep olduğu bozulmanın ölçülmesinde standart sapma değeri ve alan değerlerinin ortalamalarındaki değişim olarak 2 metrik dikkate alınmıştır. Veri tabanındaki alanların damgalama sonrası standart sapma değerlerindeki değişim [30] ve önerilen yöntem için karşılaştırmalı olarak Şekil 3.17'de verilmiştir. 3110 kayıt içeren veri setinde gerçekleştirilen testlerde [30] çalışmasının damgalama sonrası A3 ve A8 alan değerlerinin standart sapmasında sebep olduğu değişiklik %244 ve %31 oranında iken, önerilen yöntemin sebep olduğu değişiklikler %0.016 ve %0.07 oranında kalmıştır.



Şekil 3.17. Veri seti alan değerlerinin standart sapmaları üzerinde damgalama algoritmalarının sebep olduğu değişim

Damgalama işleminin alan bazında ortalama değerler üzerinde sebep olduğu değişim Şekil 3.18'de verilmiştir. Şekilden de görüldüğü üzere önerilen yöntemin orijinal veri tabanı

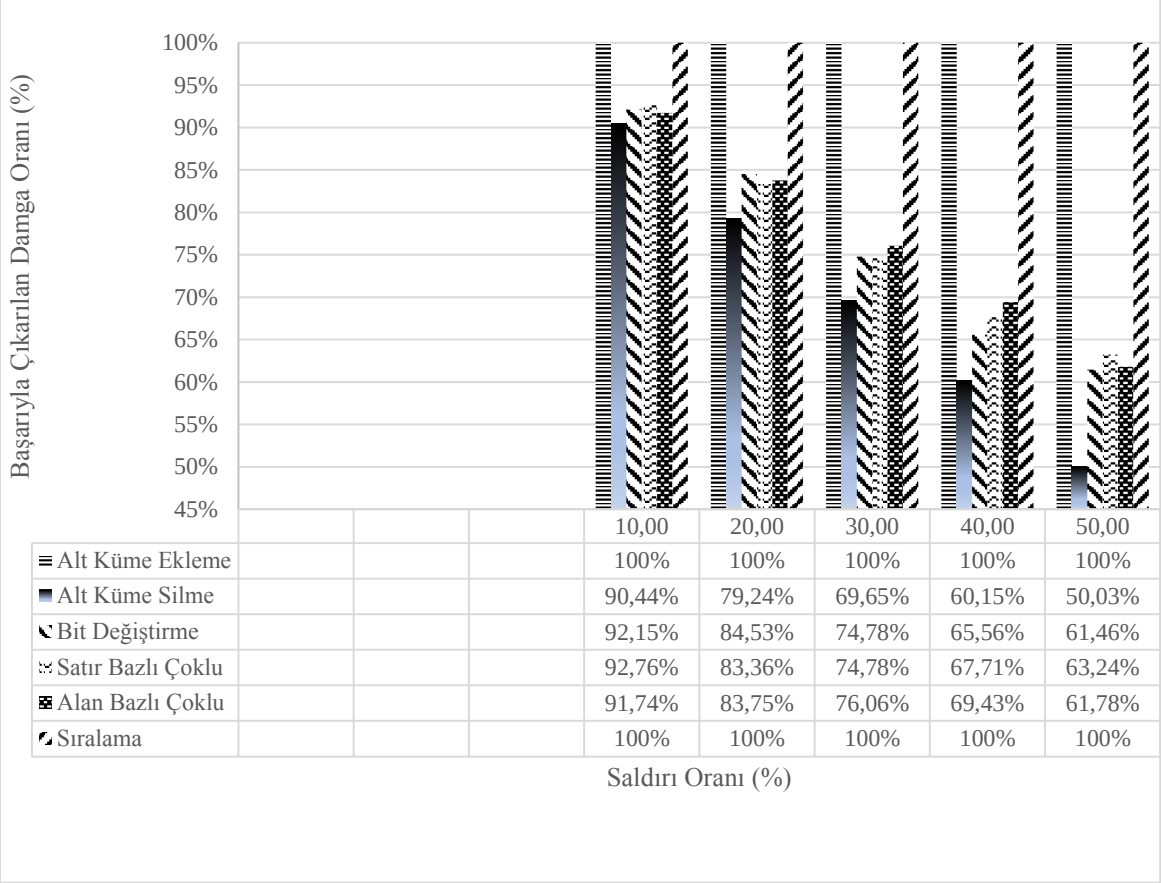
üzerinde sebep olduğu değişim minimum düzeydedir. [30] çalışmasının A3 ve A5 alan değerlerinin ortalaması üzerinde sebep olduğu değişim %1.1 ve %0.3 oranında iken, önerilen yöntemin sebep olduğu değişim %0.01 ve %0.0025 oranında kalmıştır.



Şekil 3.18. Veri seti alan değerlerinin aritmetik ortalamaları üzerinde damgalama algoritmalarının sebep olduğu değişim

3.4.2. Saldırılar

Bu bölümde önerilen yöntemin literatürde sayısal alanlar üzerine damga yerleştiren çalışmaların sağlamlık analizlerinde kullanılan saldırılara karşı dayanıklılığı test edilmiştir. Saldırıların amacı veri içerisine saklanan damganın tamamını veya kullanılamaz hale gelmesini sağlayacak oranını yok etmek veya kendi damga bilgisini veri içerisine yerleştirmektir. FCT veri setinin 3125 satırına damga verisi saklandıktan sonra veri seti üzerinde farklı oranlarda gerçekleştirilen saldırılara karşı önerilen yöntemin damga çıkarım başarı oranları Şekil 3.19'da verilmiştir. İlerleyen bölümler de bu saldırılar detaylarıyla incelenecektir.



	10,00	20,00	30,00	40,00	50,00
Alt Küme Ekleme	100%	100%	100%	100%	100%
Alt Küme Silme	90,44%	79,24%	69,65%	60,15%	50,03%
Bit Değiştirme	92,15%	84,53%	74,78%	65,56%	61,46%
Satır Bazlı Çoklu	92,76%	83,36%	74,78%	67,71%	63,24%
Alan Bazlı Çoklu	91,74%	83,75%	76,06%	69,43%	61,78%
Sıralama	100%	100%	100%	100%	100%

Şekil 3.19. Önerilen yöntemin farklı saldırılara karşı damga çıkarım başarı oranları

3.4.2.1. Alt Küme Ekleme Saldırısı

Alt Küme Ekleme saldırısı saldırgan tarafından rastgele üretilen satırların veri setine eklenmesi ile gerçekleştirilir. Saldırının amacı, damga çıkarım sürecinde eklenmemiş damga verisinin de çıkarılarak orijinal damga verisi ile eşleşme oranını düşürmektir. Farklı oranda saldırılara karşı önerilen yöntemin başarıyla çıkardığı damga verisi oranlarının verildiği Şekil 3.19'dan görüldüğü üzere, Alt Küme Ekleme saldırısının önerilen yöntemin başarısı üzerinde etkisi yoktur. Damga yerleştirme sürecinden önce satırların birincil anahtar değeri dikkate alınarak sıralanması ve damga yerleştirilecek aday satırların kullanıcı tarafından belirlenen gizli anahtara bağlı olarak seçilmesi sayesinde, veri tabanına yeni eklenen satırlar damga çıkarım başarısını etkilemeyecektir.

3.4.2.2. Alt Küme Silme Saldırısı

Alt Küme Silme Atağı damgalama algoritmalarının performansını ölçmede kullanılan bir diğer saldırı türüdür. Bu atak türünde damganın yok edilmesi amacıyla rastgele belirlenen satırlar veri setinden silinmektedir. Şekil 3.19'da atağın önerilen çalışma üzerindeki etkileri incelendiğinde, atak oranı (silinecek satır sayısı) arttıkça başarıyla çıkarılabilen damga boyutunun da düştüğü gözlemlenmektedir. Saldırı sonucu damga barındıran satırın silinmesinin onarımı söz konusu olamayacağından, diğer saldırı türleri ile karşılaştırıldığında en etkili ve savunması en zor saldırı türü Alt Küme Silme saldırısıdır.

3.4.2.3. Bit Değiştirme Saldırısı

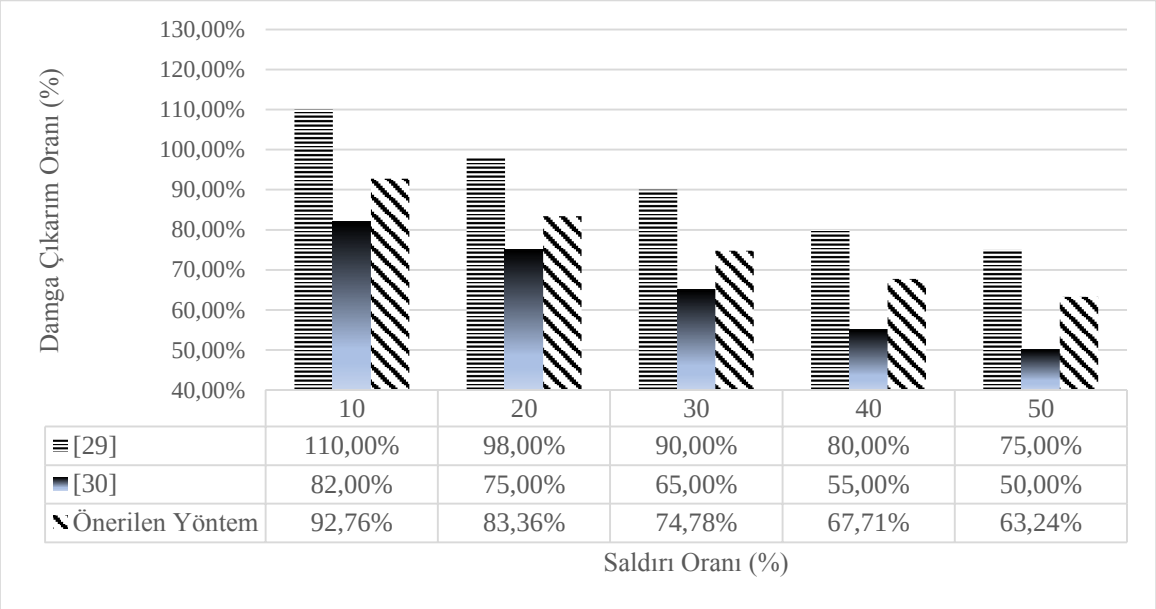
Bit Değiştirme saldırısında rastgele seçilen kayıtların alan değerlerinin son bit değeri değiştirilerek damga verisinin yok edilmesi amaçlanmaktadır. Saldırı yapılan kaydın herhangi bir alanında damga verisi varsa, bit değiştirme atağı bu verinin kayıp olmasına sebep olacaktır. Alt küme silme saldırısından farklı olması amacıyla, bit değiştirme saldırısında rastgele seçilen satırların alan değerlerinin rastgele belirlenen yarı alanına saldırı yapılmıştır. Şekil 3.19'da verilen sonuçlar incelendiğinde, Alt Küme Silme saldırısına göre Bit Değiştirme saldırısına daha dayanıklı olmasına karşın, saldırı oranı arttıkça doğal olarak kaybedilen damga verisinin de arttığı görülmektedir.

3.4.2.4. Sıralama Saldırısı

Veri tabanındaki alanların konumlarının değiştirilmesi ile gerçekleştirilen Sıralama saldırısının önerilen yönteme bir etkisi yoktur. Damga yerleştirme ve damga çıkarım süreçlerinin başında veri tabanı alanları alfabetik olarak sıralandığı için, bu saldırıdan önerilen yöntem etkilenmeyecektir. Şekil 3.19'dan da görüldüğü gibi saldırı oranının değişimi başarıyla çıkarılan damga oranını değiştirmemekte, saldırı oranı ne olursa olsun yerleştirilen damga verisinin tamamı çıkarılabilmektedir.

3.4.2.5. Satır Bazlı Çoklu Saldırı

Satır Bazlı Çoklu saldırı 3 farklı saldırının art arda uygulanması ile gerçekleştirilmektedir. Saldırı oranı belirlendikten sonra saldırı yapılacak veri tabanına sırasıyla Alt Küme Ekleme, Alt Küme Silme ve Bit Değiştirme saldırıları uygulanır. Şekil 3.19'da önerilen yöntemin Satır Bazlı Çoklu saldırıya karşı dayanıklılığı görülmektedir. Önerilen yöntem ile [29] ve [30] çalışmalarının aynı atak türüne karşı dayanıklılık karşılaştırmaları Şekil 3.20'de verilmiştir.



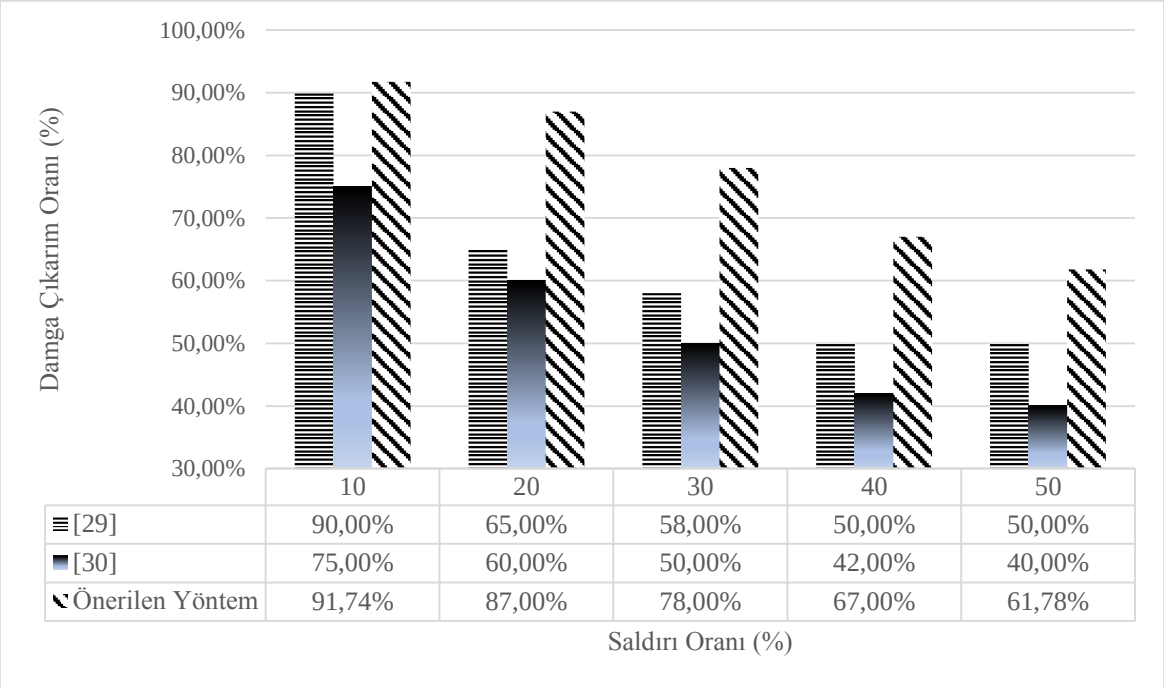
	10	20	30	40	50
[29]	110,00%	98,00%	90,00%	80,00%	75,00%
[30]	82,00%	75,00%	65,00%	55,00%	50,00%
Önerilen Yöntem	92,76%	83,36%	74,78%	67,71%	63,24%

Şekil 3.20. Önerilen yöntem, [29] ve [30] çalışmalarının Satır Bazlı Çoklu saldırıya karşı dayanıklılık karşılaştırması

Şekil 3.20 incelendiğinde [29] çalışmasının 100%'ün üstünde bir damga çıkarım oranı olduğu görülmektedir. Alt Küme Ekleme atağından dolayı yerleştirilmemiş damga verisini çıkardığı için damga çıkarım oranı yüksek görülmektedir. Saldırı oranının %50 olduğu durumda [30] çalışmasının damga çıkarım başarı oranı da %50 oranında kalmaktadır. Aynı şartlarda önerilen yöntemin başarıyla çıkardığı damga oranı ise %63'dür.

3.4.2.6. Alan Bazlı Çoklu Saldırı

Alan Bazlı Çoklu Saldırı, Satır Bazlı Çoklu saldırıya benzer şekilde 2 farklı saldırının art arda gerçekleştirilmesiyle uygulanır. Saldırı oranı belirlendikten sonra ilk olarak veri setine Bit Değiştirme Saldırısı uygulanır. Ardından saldırı oranında rastgele seçilmiş kayıtların, yine rastgele seçilen alan değeri değiştirilir. Önerilen yöntem, [29] ve [30] çalışmalarının Alan Bazlı Çoklu saldırıya karşı dayanıklılık sonuçları Şekil 3.21'de verilmiştir. Saldırı oranının %50 olarak belirlendiği durumda [30] algoritması sadece %40 oranında damgayı saldırıya uğramış veri tabanından elde edebilirken, önerilen yöntemin damga çıkarım oranı %62 düzeyindedir.



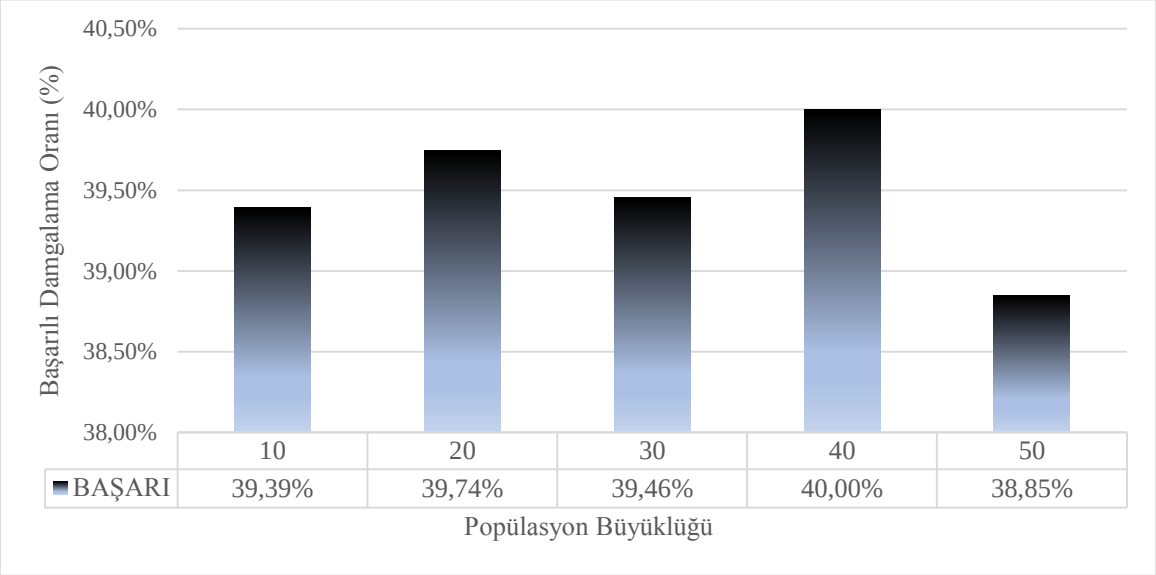
	10	20	30	40	50
[29]	90,00%	65,00%	58,00%	50,00%	50,00%
[30]	75,00%	60,00%	50,00%	42,00%	40,00%
Önerilen Yöntem	91,74%	87,00%	78,00%	67,00%	61,78%

Şekil 3.21. Önerilen yöntem, [29] ve [30] çalışmalarının Alan Bazlı Çoklu saldırıya karşı dayanıklılık karşılaştırması

3.4.3. Parametre Analizi

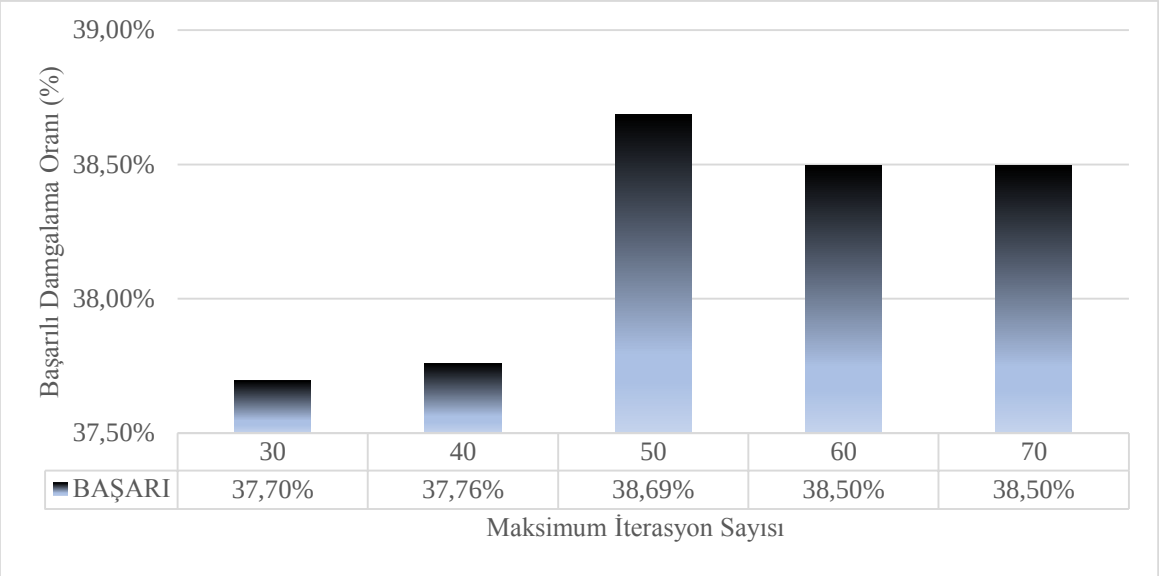
Önerilen yöntemde kullanılan parametrelerin algoritma başarısı üzerindeki etkisi bu bölümde irdelenmiştir. Gerçekleştirilen testlerde popülasyon boyutu P, maksimum adım sayısı $max_{iteration}$ ve hareket işleminde kullanılan c_{move} parametrelerinin farklı değerleri için damga çıkarım başarısı incelenmiştir.

FCT veri tabanında rastgele seçilen 3125 satırdan test veri seti oluşturulmuştur. Ateş böceği popülasyon birey sayısı olarak 10, 20, 30, 40 ve 50 seçildiğinde önerilen yöntemin test veri setini damgalayabilme oranları Şekil 3.22'de verilmiştir. Sonuçlar irdelendiğinde çalışmanın farklı popülasyon sayılarında benzer oranlarda başarıyı elde ettiği görülmektedir. Düşük popülasyon sayılarında bile yöntem %38-40 aralığında damga yerleştirebilme oranını yakalamıştır. Sonuçlar popülasyon birey sayısının önerilen yöntem üzerinde fazla etkisi olmadığı göstermektedir.



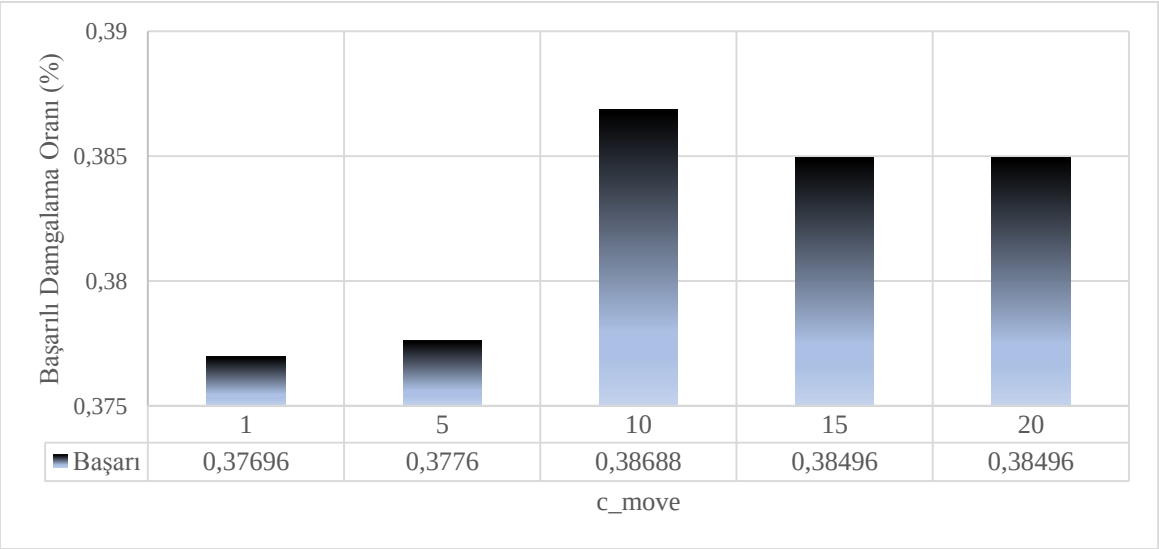
Şekil 3.22. Farklı popülasyon boyutları için önerilen yöntemin damga yerleştirme oranları

Damgalama oranı üzerinde etkisi incelenen ikinci parametre önerilen yöntemin sonlanma kriterlerinden biri olan maksimum adım sayısı $max_{iteration}$'dır. Şekil 3.23'de sonuçları verilen testlerde maksimum adım sayısı olarak 30, 40, 50, 60 ve 70 alındığında algoritmanın eriştiği damgalama oranları görülmektedir. Düşük adım sayılarında bile yüksek damgalama oranı elde ettiği, adım sayısının belirli bir değere ulaşana kadar damgalama oranı üzerinde artışa sebep olduğu, devam eden artışlarda doyuma uğrayıp stabil olduğu görülmektedir. Bu sonuçlar, algoritmanın düşük adım sayısında yakalayacağı maksimum başarıyı yakaladığını göstermektedir.



Şekil 3.23. Farklı maksimum adım değerleri için önerilen yöntemin damga yerleştirme oranları

İncelenen son parametre c_{move}, hareket işleminde ateş böceğinin en iyi ateş böceğine yaklaşım hareketi aşamasında oluşturulacak olan aday konumların sayısını belirlemektedir. Parametre değer adedince aday konum oluşturulup, arasından ateş böceğinin en parlak olacağı (en yüksek başarı) konum seçilmektedir. Farklı c_{move} değerleri için önerilen yöntemin elde ettiği damgalama oranları Şekil 3.24'de verilmiştir. Parametre değeri olarak 15'in üzerindeki değerlerin damgalama oranı üzerinde etkili olmadığı görülmektedir. Bu sonuç algoritmanın düşük işlem adımlarında bile yüksek başarı elde ettiğini göstermektedir.



Şekil 3.24. Farklı c_move değerleri için önerilen yöntemin damga yerleştirme oranları

3.5. Yeni Bir Alfa Sayısal Veri Tabanı Damgalama Yaklaşımının Değerlendirilmesi ve Elde Edilen Sonuçlar

Deneysel metotlar MicroSoft Visual Studio yazılım geliştirme ortamında C# programlama dili kullanılarak geliştirilmiştir. Veri tabanı sistemi olarak MicroSoft SQL Server kullanılmıştır. Kaliforniya Üniversitesi'nin Forest Cover Type (FCT) isimli 581.012 satırdan oluşan 54 nümerik alan içeren veri tabanı kullanılmıştır. Veri tabanı metin alan içermediği için veriye uygun şekilde yeni bir metin alanı eklenip içeriği meşcere tip bilgisi ile doldurulmuştur.

Meşcere, ilk olarak farklı özellikleri içeren orman parçaları olarak tanımlanmıştır [81]. Meşcere sıklığının hacim ve hacim artımı üzerinde önemli etkide bulunduğu görülerek; ağaç sayısı, meşcere tipi ayrımında bir özellik olarak kabul edilmiştir [82, 83]. Sıklık etkenine, ağaç türü ve yaş da katılarak; saf, karışık, aynı yaşlı ve değişik yaşlı meşcereler ayrılmıştır. Çalışmada kullanılan meşcere bilgisi ağaç tür veya türleri, gelişme çağları ve kapalılık değerlerini barındırmaktadır [84-87]. Meşcere oluşumunda kullanılan ağaç türlerine örnekler Tablo 3.4.'de verilmiştir.

Tablo 3.4. Meşcere bilgisini oluşturan ağaç tür örnekleri

Sembol	*Ağaç Türü*	*Sembol*	*Ağaç Türü*
Cs	Sarıçam	M	Meşe
Ck	Karaçam	Cz	Kızılçam
Kn	Kayın	Kz	Kızılağaç
Kv	Kavak	Ak	Akağaç
Ks	Kestane	Df	Defne
S	Sedir	Cv	Ceviz

Meşcere bilgisini oluşturan bir diğer alt bilgi Gelişme Çağları'dır [66]. Gelişme çağları meşcerenin oluşum sürecinde gösterdiği aşamalardır. Çalışmamızda kullanılan gelişme çağları Tablo 3.5.'de verilmiştir.

Tablo 3.5. Meşcere bilgisini oluşturan gelişme çağları

Gelişme Çağı	*Simgesi*
Gençlik ve Sıklık	A
Sırıklık ve Direklik	B
İnce Ağaçlık	C
Orta Ağaçlık	D
Kalın Ağaçlık	E

Meşcere bilgisini oluşturan son alt bilgi Kapalılık değeridir. Meşcere içinde komşu ağaç tepelerinin birbirine yanaşmalarına Meşcere Kapalılığı adı verilir. Çalışmamızda kullanılan kapalılıklar Tablo 3.6'da verilmiştir.

Tablo 3.6. Meşcere bilgisini oluşturan tepe kapalılıkları

Tepe Kapalılığı	*Sembol*
%1-10	B
%11-40	1
%41-70	2
%71-100	3

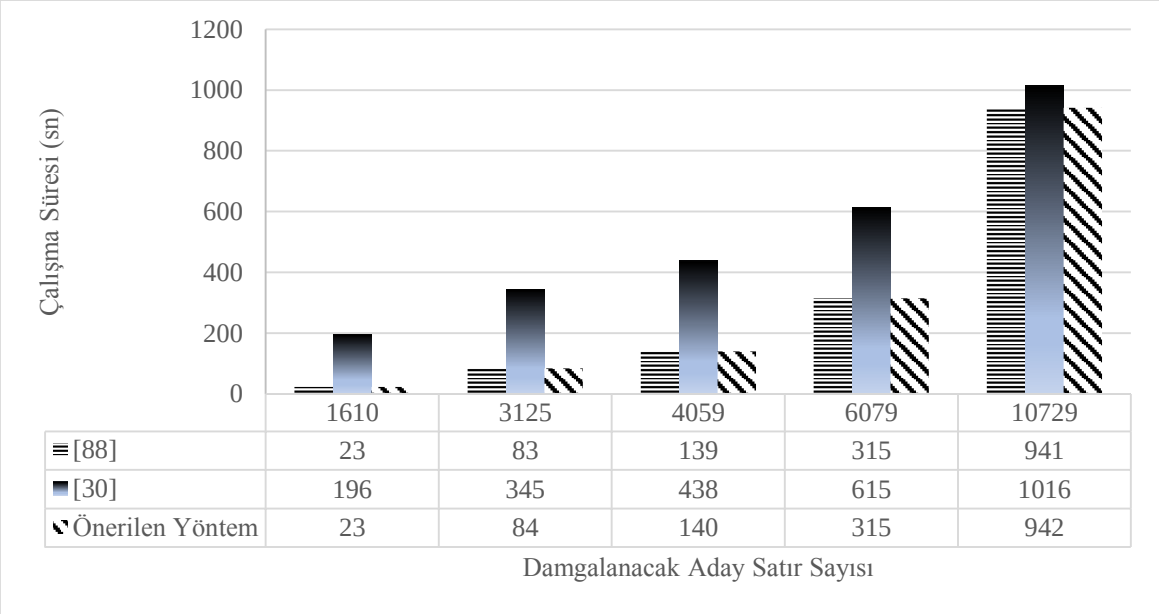
FCT veri tabanına eklediğimiz metin alanı doldururken kullanılan bazı meşcereler Tablo 3.7'de verilmiştir.

Tablo 3.7. Veri tabanına eklenen metin alanında kullanılan meşcere örnekleri

Meşcere Bilgisi	
CsKnab1	KvCsab3
CkCsa2	KsKnab1
KnMab3	KnAkcd1

3.5.1. Performans Analizi

Önerilen yöntem, [30] ve [88] çalışmaları ile damga yerleştirme süresi ve damgalama kapasitesi açısından karşılaştırılmıştır. Çalışma süresi dikkate alınarak yapılan testlerin sonuçları Şekil 3.25'de verilmiştir. Testler FCT veri tabanından rastgele seçilen 1610, 3125, 4059, 6079 ve 10729 satırlık veri setleri üzerinde gerçekleştirilmiştir. Her bir veri seti üzerinde çalışmalar 10'ar kez test edilmiş ve çalışma sürelerinin ortalaması alınmıştır. Sonuçlardan görüldüğü üzere önerilen yöntemin çalışma süresi [30] çalışmasından daha düşük, [88] çalışması ile çok yakındır. Sayısal alanların damgalanması aşamasında [88] çalışmasında da kullanılan Ateş Böceği algoritmasının kullanılması düşük iterasyon adımlarında çözüme ulaşılması sağlamış ve çalışma süresini düşürmüştür.

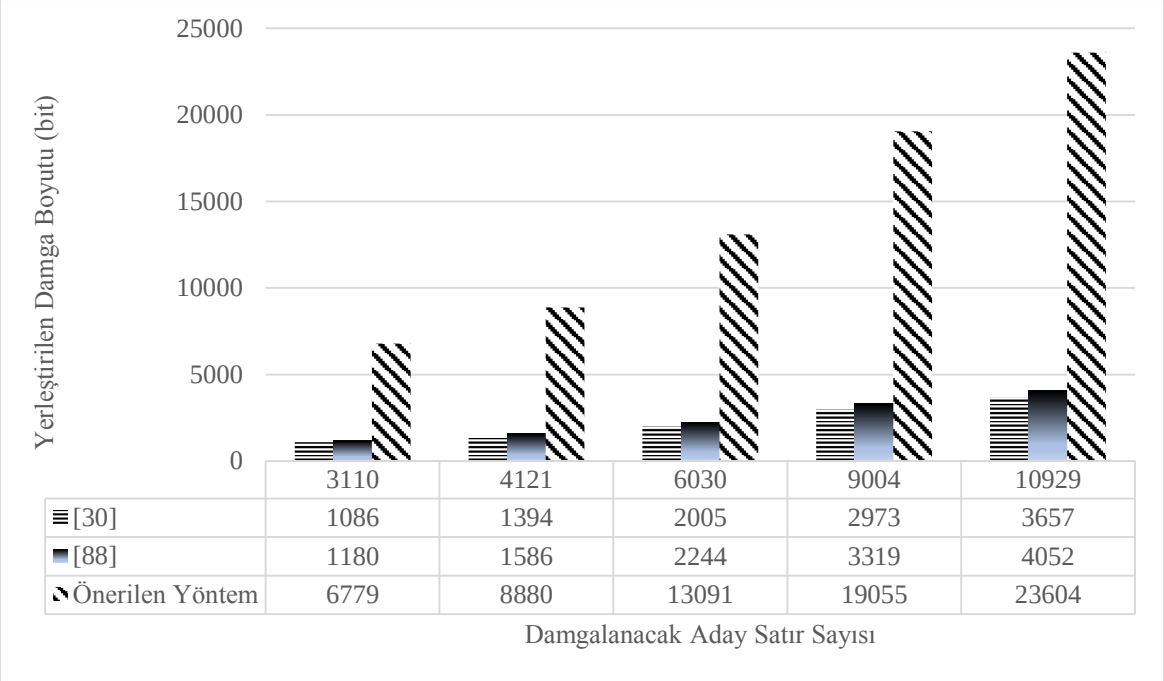


	1610	3125	4059	6079	10729
[88]	23	83	139	315	941
[30]	196	345	438	615	1016
Önerilen Yöntem	23	84	140	315	942

Şekil 3.25. Farklı veri setleri üzerinde önerilen yöntem, [30] ve [88] çalışmalarının ortalama çalışma sürelerinin karşılaştırılması

FCT veri tabanından rastgele seçilen 3110, 4121, 6030, 9004 ve 10929 satırlık veri setleriyle yapılan damgalama kapasite testlerinin sonuçları Şekil 3.26'da verilmiştir. [30] ve [88] çalışmaları veri tabanındaki sadece sayısal alanlar üzerinde damga yerleştirme yapabilmektedir. Önerilen yöntem ise homoglif dönüşüm sayesinde sayısal alanlar ile birlikte metin alanların üzerinde de damgalama yapabilmektedir. Şekil 3.26 incelendiğinde veri türünden bağımsız olarak damgalama yapabilme özelliği sayesinde önerilen yöntemin damgalama kapasitesinin [30] ve [88] çalışmalarının ortalama 6 katı olduğu görülmektedir.

3110 aday satırlık veri setinde [30] çalışması 1086 bit, [88] çalışması 1180 bit damga verisi saklayabilirken önerilen çalışma 6779 bit veriyi başarıyla saklayabilmiştir. FCT veri tabanına eklenen meşcere bilgisini barındıran metin alanının Tablo 3.7'de örnekleri sunulan verilerin uzunluğunun 5 ile 7 karakter arasında değiştiği göz önüne alındığında, daha uzun metin içeren veri tabanlarında damgalama kapasitesi artış gösterecektir.



	3110	4121	6030	9004	10929
[30]	1086	1394	2005	2973	3657
[88]	1180	1586	2244	3319	4052
Önerilen Yöntem	6779	8880	13091	19055	23604

Şekil 3.26. Farklı veri setlerine yerleştirilebilen damga veri büyüklüklerinin karşılaştırılması

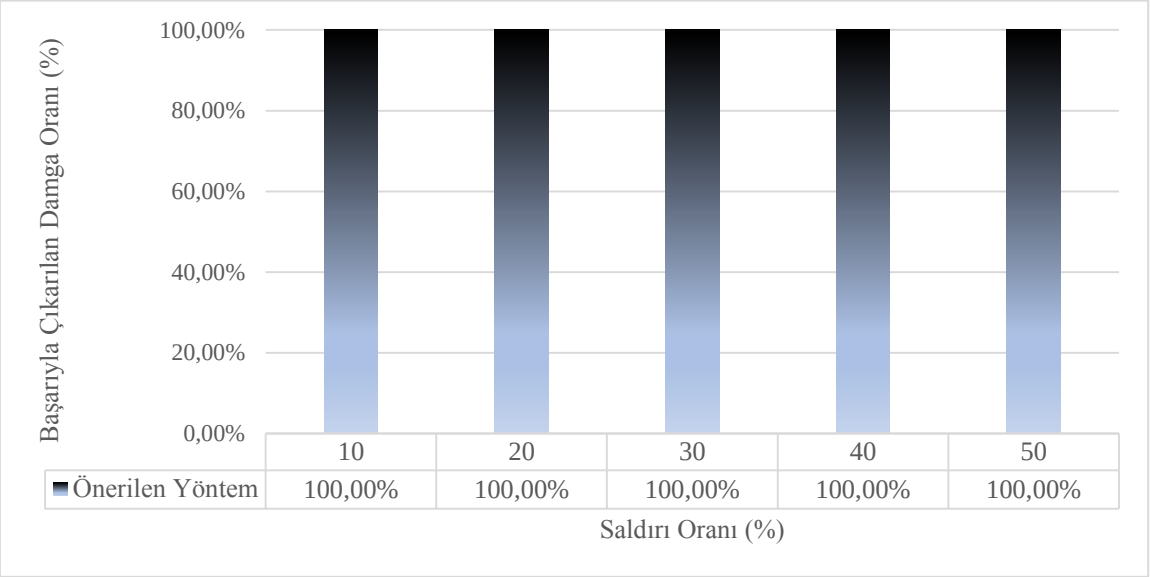
3.5.2. Saldırılar

Veri tabanı damgalama şemalarının sağlamlık analizinde sıklıkla kullanılan saldırı testlerinin önerilen çalışma için sonuçları bu bölümde verilmiştir. Gerçekleştirilen saldırıların amacı veri tabanına yerleştirilmiş damga verisini yok etmek veya kendi damga bilgisini yerleştirmektir.

3.5.2.1. Alt Küme Ekleme Saldırısı

Altküme Ekleme Atağı olarak isimlendirilen saldırı türünde saldırgan rastgele verilerden oluşturulan veri kümesini orijinal veri kümesi içerisine yerleştirmektedir. Saldırının amacı damga çıkarım aşamasında saklanmayan damganın çıkarımına sebep olarak

damgalama başarımını düşürmektir. Önerilen yöntem damga yerleştireceği satırları özüt fonksiyon yardımıyla belirlediği için Şekil 3.27'de görüldüğü üzere Altküme Ekleme Saldırısından etkilenmemektedir.

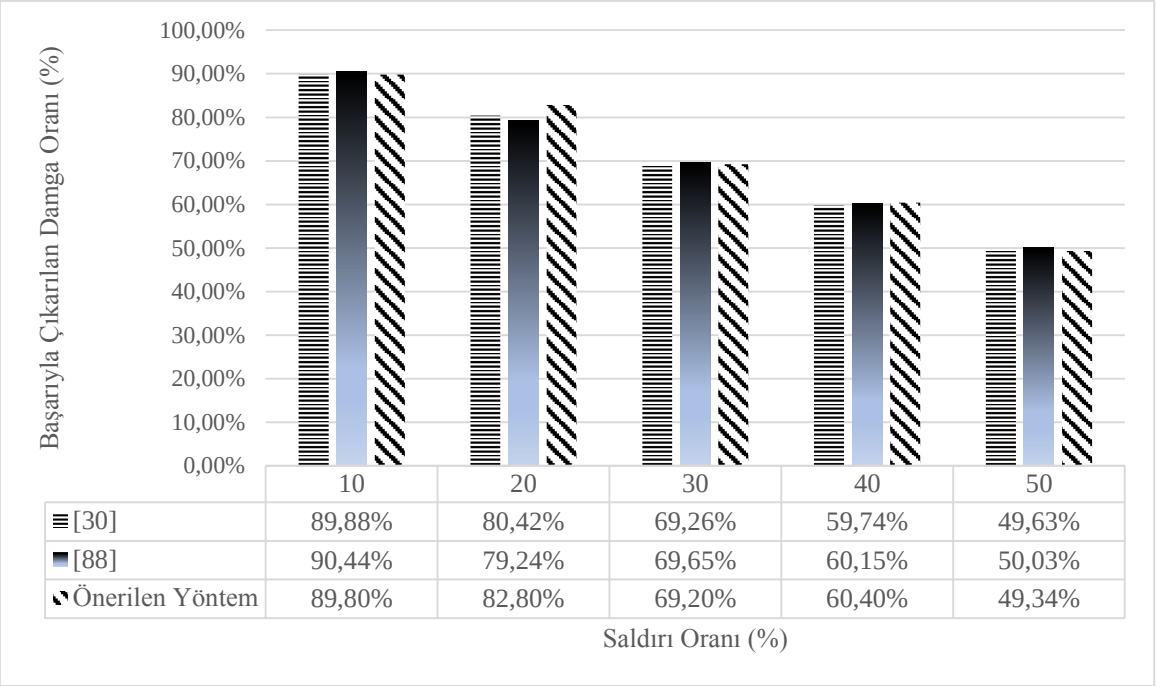


Şekil 3.27. Önerilen yöntemin Alt Küme Ekleme saldırısına karşı dayanıklılığı

3.5.2.2. Alt Küme Silme Saldırısı

Saldırılar arasındaki en etkili saldırı türü olan Altküme Silme saldırısında rastgele olarak belirlenen satırlar veri tabanından silinmektedir. Silinen kayıtlar ile damga verisi de ortadan kaybolduğu için saldırı oranı arttıkça elde edilebilen damga oranı da aynı oranda düşmektedir. Önerilen yöntem, [30] ve [88] çalışmalarının Alt Küme Silme saldırısına karşı başarıyla çıkardıkları damga oranları Şekil 3.28'de verilmiştir.

1201 aday satırdan oluşan örnek veri seti üzerinde yapılan Alt Küme Silme saldırılarında önerilen yöntemin [30] ve [88] çalışmalarıyla benzer oranlarda damga çıkarım başarısına ulaştığı görülmüştür. Çıkarılan damga oranı aynı olmasına rağmen, önerilen yöntemin metin alan üzerinde de damga yerleştirme yapabilmesi sebebiyle çıkarılan damganın boyutu yüksek olmuştur.



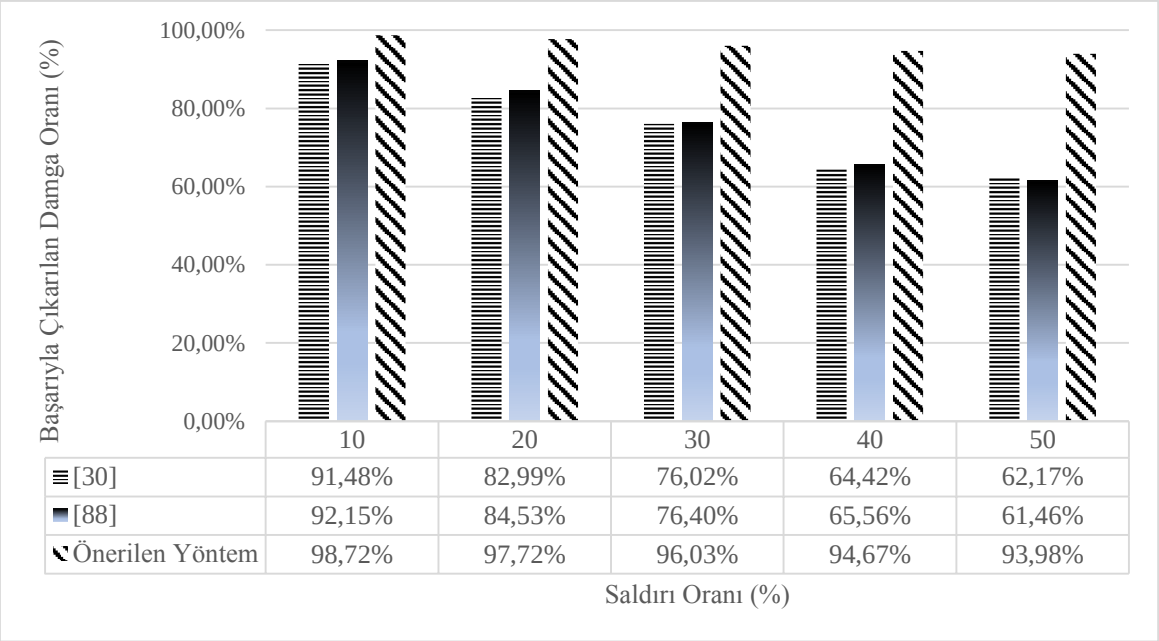
	10	20	30	40	50
[30]	89,88%	80,42%	69,26%	59,74%	49,63%
[88]	90,44%	79,24%	69,65%	60,15%	50,03%
Önerilen Yöntem	89,80%	82,80%	69,20%	60,40%	49,34%

Şekil 3.28. Önerilen yöntem, [30] ve [88] çalışmalarının Alt Küme Silme saldırısına karşı dayanıklılık sonuçları

3.5.2.3. Bit Değiştirme Saldırısı

Bit-Değiştirme saldırısı rastgele belirlenmiş satırların alan değerlerinin en anlamsız bitinin terslenmesi ile gerçekleştirilmektedir. Alt Küme Silme saldırısından farklı olması amacıyla seçilen satırların alanlarının tamamında değil, yarısında bu atak uygulanmıştır. Önerilen yöntem, [30] ve [88] çalışmalarının Bit Değiştirme Silme saldırısına karşı başarıyla çıkardıkları damga oranları Şekil 3.29'da verilmiştir.

Önerilen yöntemin sayısal alanlar ile beraber metin alanlara da damga yerleştirme özelliğinden dolayı Bit Değiştirme saldırısından minimum düzeyde etkilenmiştir. [30] ve [88] çalışmaları yerleştirdikleri damga verisini saldırı ile kaybederken, önerilen yöntemin metin alanlarına yerleştirdiği damga veri boyutunun büyük oluşu ve saldırının metin alanlar üzerinde etkili olmaması sebebiyle %50 saldırı oranında bile başarıyla çıkarabildiği damga oranı %94'ü bulmuştur.



	10	20	30	40	50
[30]	91,48%	82,99%	76,02%	64,42%	62,17%
[88]	92,15%	84,53%	76,40%	65,56%	61,46%
Önerilen Yöntem	98,72%	97,72%	96,03%	94,67%	93,98%

Şekil 3.29. Önerilen yöntem, [30] ve [88] çalışmalarının Bit Değiştirme saldırısına karşı dayanıklılık sonuçları

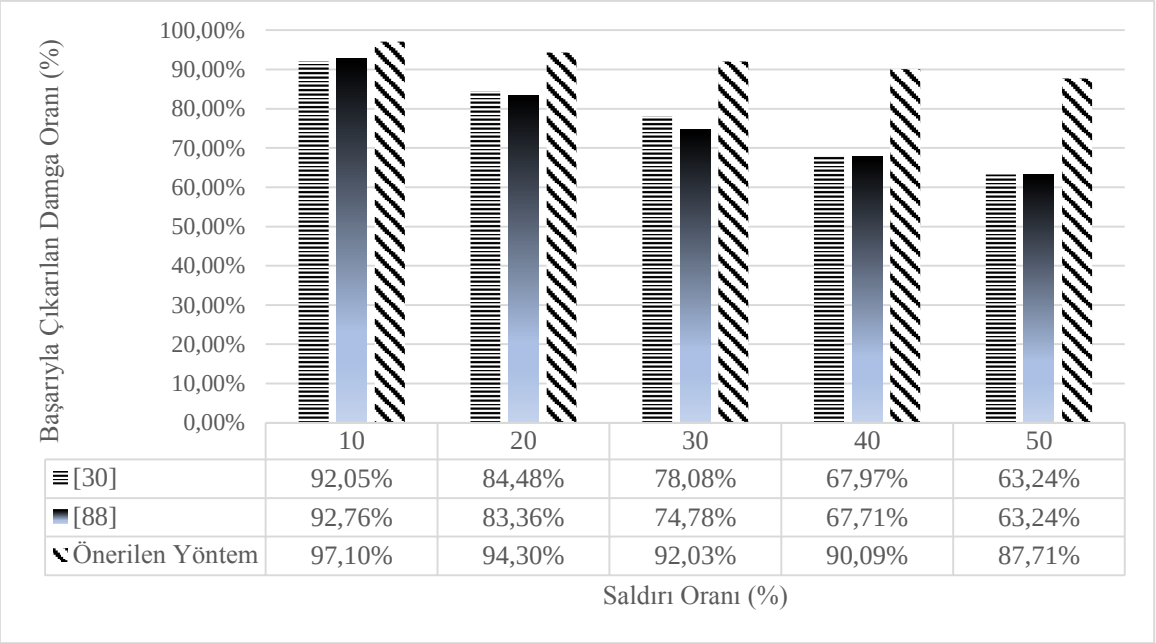
3.5.2.4. Sıralama Saldırısı

Önerilen yöntemde damgalama sürecinin ilk aşamasında satırların birincil anahtar, sütunların alfabetik sıralanmasından dolayı veri tabanındaki satır ve sütunların yerlerinin değiştirilmesi ile gerçekleştirilen Sıralama saldırısının herhangi bir etkisi yoktur.

3.5.2.5. Satır Bazlı Çoklu Saldırı

Satır Bazlı Çoklu Saldırıda birden çok saldırı art arda uygulanmaktadır. Altküme Ekleme, Altküme Silme ve Bit-Değiştirme saldırılarının sırasıyla uygulanması ile Satır Bazlı Çoklu Saldırı gerçekleştirilmiş olur. Önerilen yöntem, [30] ve [88] çalışmalarının Satır Bazlı Çoklu saldırıya karşı başarıyla çıkardıkları damga oranları Şekil 3.30'da verilmiştir.

Önerilen yöntemin sayısal alanlar ile beraber metin alanlara da damga yerleştirme özelliğinden dolayı Bit Değiştirme saldırısında olduğu gibi Satır Bazlı Çoklu saldırıdan da minimum düzeyde etkilenmiştir. [30] ve [88] çalışmaları yerleştirdikleri damga verisini saldırı adımlarında kaybederken, önerilen yöntemin metin alanlarına yerleştirdiği damga veri boyutunun büyük oluşu ve saldırının metin alanlar üzerinde etkili olmaması sebebiyle %50 saldırı oranında bile başarıyla çıkarabildiği damga oranı %88'i bulmuştur.



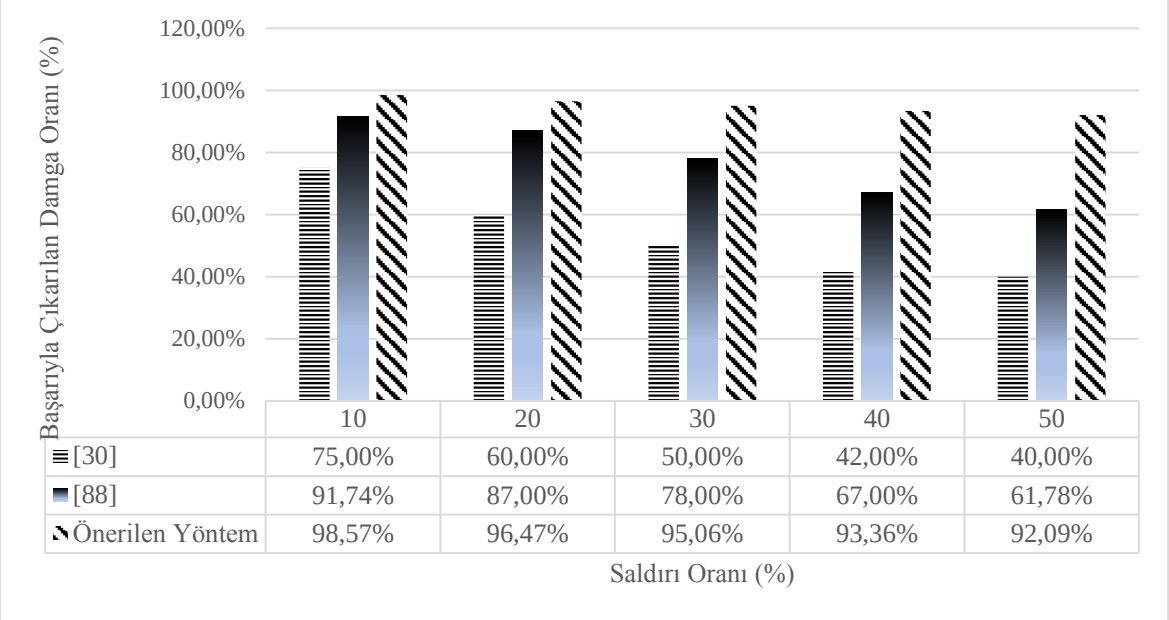
Şekil 3.30. Önerilen yöntem, [30] ve [88] çalışmalarının Satır Bazlı Çoklu saldırıya karşı dayanıklılık sonuçları

3.5.2.6. Alan Bazlı Çoklu Saldırı

Alan Bazlı Çoklu Saldırı, Satır Bazlı Çoklu saldırıya benzer şekilde 2 farklı saldırının art arda gerçekleştirilmesiyle uygulanır. Saldırı oranı belirlendikten sonra ilk olarak veri setine Bit Değiştirme Saldırısı uygulanır. Ardından saldırı oranında rastgele seçilmiş kayıtların, yine rastgele seçilen alan değeri değiştirilir. Önerilen yöntem, [30] ve [88] çalışmalarının Alan Bazlı Çoklu saldırıya karşı dayanıklılık sonuçları Şekil 3.31'de verilmiştir.

Önerilen yöntemin sayısal alanlar ile beraber metin alanlara da damga yerleştirme özelliğinden dolayı Bit Değiştirme, Satır Bazlı Çoklu saldırılarında olduğu gibi Alan Bazlı Çoklu saldırıdan da minimum düzeyde etkilenmiştir. [30] ve [88] çalışmaları yerleştirdikleri damga verisini saldırı adımlarında kaybederken, önerilen yöntemin metin alanlarına yerleştirdiği damga veri boyutunun büyük oluşu ve saldırının metin alanlar üzerinde etkili olmaması sebebiyle %50 saldırı oranında bile başarıyla çıkarabildiği damga oranı %92'yi bulmuştur.

Atak testlerinin sonuçları dikkatle incelendiğinde önerilen yöntemin, verinin yok edilmesine sebep olan Alt Küme Silme Atağı dışındaki 5 atağın 2'sinden hiçbir şekilde etkilenmediği, diğer 3 atağın %50 oranında yapılması durumunda bile damganın yeniden elde edilebilme oranının %87'nin üzerinde olduğu görülmektedir.



Şekil 3.31. Önerilen yöntem, [30] ve [88] çalışmalarının Alan Bazlı Çoklu saldırıya karşı dayanıklılık sonuçları

4. SONUÇLAR

Veri paylaşımındaki artış ile beraber farklı formatlardaki verilerin yetkiler dahilinde kullanımının sağlanması gerekliliği ortaya çıkmıştır. Dijital Hak Koruma mekanizmaları dijital içerik üzerinde sahipliğin kanıtlanması ve yasal olmayan yollarla paylaşımını sınırlandırmak için önemli rol oynamaktadır. Kriptografik teknikler genellikle kriptografik bilgileri dijital içerikle ilişkilendiremediklerinden sahiplik bilgisinin kanıtlanmasında eksik kalmaktadırlar. Diğer taraftan sayısal damgalama sahiplik bilgisi ile dijital veriyi birleştirebilmesi sayesinde verinin sahipliğinin kanıtlanmasında kullanılabilmektedir.

Veri tabanları üzerinde gerçekleştirilen sayısal damgalama algoritmaları saldırılara karşı dayanıklılık durumlarına göre Sağlam ve Kırılgan olarak sınıflandırılmaktadırlar. Sağlam yöntemler saldırılara karşı dayanıklıdırlar ve saldırı yapılmış veri tabanından damga verisini başarıyla çıkarabilmeleri beklenir. Tez kapsamında gerçekleştirilen çalışmalar 1.4 bölümünde detayları açıklanan sayısal damgalama yöntemlerinin sahip olması gereken ortak özellikleri iyileştirmek doğrultusunda tasarlanan damgalama yöntemleri ve önerilen Ateş Böceği Algoritması ile yeni bir veri tabanı damgalama yaklaşımından elde edilen sonuçlar kısaca aşağıdaki şekilde özetlenebilir.

Çoğu veri tabanında bulunan fakat literatür çalışmalarında nadiren kullanılan tarih verisinin saniye bilgisine damga verisi gizlenmiştir. Yöntem damga verisini ikilik düzene çevirerek sakladığı için damga verisinin türünden (resim, ses, metin, vb.) bağımsızdır. Saldırılara karşı dayanıklı olması amacıyla damgalanmak üzere seçilen aday satırlar arasında alt bölümleme uygulaması yapılmıştır. Literatürde gerçekleştirilen saldırılar sayısal üzerine odaklandığı için önerilen yöntemin dayanıklılığı yüksektir. Alt Küme Ekleme Saldırısından etkilenmeyen yöntem, veri tabanının %50'sinin silindiği Alt Küme Silme Saldırısında damga çıkarım oranları %50 ve %70 olan [48] ve [71] çalışmalarına kıyasla tamamını kurtararak damga çıkarımında artış sağlamıştır. Alt Küme Değiştirme saldırısının %90 oranında yapıldığı test sonuçlarında [48] çalışmasının damga verisini kaybettiği, [71] çalışmasının ise %70 oranında damgayı kurtarabildiği durumda önerilen yöntem damga çıkarım oranında %4'lük artış sağlamıştır.

Sayısal alanlar üzerinde damga saklayan çoğu yöntemin 2 alana ihtiyaç duymasına karşılık tek sayısal alanın olduğu veri setlerine de damga yerleştirilebilmesini sağlayan yöntem, kullandığı nümerik dönüşüm işlemlerinden dolayı çalışma zamanını düşürmüştür.

Damga çıkarım işlemi sonrası orijinal veri tabanı değerlerini de elde edebilen yöntem, damga verisinin türünden bağımsız olarak veriyi saklayabilmekte ve saldırılara karşı dayanıklılığı sağlamak için aday satırlar arasında alt bölümleme uygulaması yapmaktadır. 6000 satırlık örnek veri setinde [30] çalışmasının damga yerleştirme süresi 615 saniye iken önerilen yöntem bu süreyi 12 saniyeye düşürmüştür. Damgalama yöntemlerinin başarısını belirleyen kriterlerden biri olan damgalama kapasitesinde de önerilen yöntem artış sağlamıştır. 21727 satırlık örnek veri setinde [29] ve [30] çalışmaları 842 ve 893 byte veri saklayabilirken önerilen yöntem damga boyutunu 1358 byte'a çıkarabilmiştir. Yüksek kapasitede damga verisini saklarken damga verisinin saldırganlar tarafından algılanamaz olması gerekir. Yapılan güvenlik testlerinde damgalama sonrası alanların standart sapma ve ortalama değerlerindeki değişimler incelenmiştir. Önerilen yöntemin damga yerleştirdiği 9 alanın 6'sının standart sapmasında sebep olduğu değişim [30] yönteminden yüksek iken, 3 alanda sebep olduğu değişim daha düşüktür. Fakat önerilen yöntemin damga yerleştirdiği alanların ortalama değerleri üzerinde sebep olduğu değişimin [30] çalışmasından düşük olmuştur. Ayrıca tüm veri tabanı dikkate alındığında önerilen yöntemin sebep olduğu değişim alanlar üzerine dağıtılmıştır. Yapılan testlerde önerilen yöntemin saldırılara karşı daha dayanıklı olduğu görülmüştür. Alt Küme Silme saldırısının %50 oranında yapıldığı testlerde [30] çalışmasının kurtarabildiği damga oranı %50 iken, önerilen yöntemin %66'dır. Aynı oranda yapılan Bit Değiştirme Saldırısında [30] çalışması saklanan verinin %62'sini kurtarabilirken, önerilen yöntem oranı %99'a çıkarmıştır. Birden çok saldırının art arda yapıldığı Satır ve Alan Bazlı Çoklu Saldırılarda da önerilen yöntem [30] çalışmasına kıyasla damga çıkarımı %12 ve %15 oranlarında artırmıştır.

XML veri saklayan alanlar üzerinde uygulanabilen yeni bir damgalama yöntemi önerilmiştir. Kullanıcının belirlediği algılanabilirlik oranı ile gizliliğinin önemli olduğu veri tabanlarında algılanamazlığın, kapasitenin önemli olduğu veri tabanlarında damgalamanın yüksek olması sağlanmıştır. Literatürde XML alanlar üzerine damga yerleştiren [22], [23], [24], [25], [26], [27] ve [80] çalışmaları ile damgalama sonrası veri üzerinde oluşan değişim yönünden karşılaştırıldığında önerilen yöntem fark edilemez ve orijinal veri üzerinde değişime sebep olmamaktadır. Damgalama kapasiteleri karşılaştırıldığında önerilen yöntemin benzer çalışmalar arasındaki en yüksek damga kapasitesine sahip [24] çalışmasının 1.5 katı, en düşük damga kapasitesine sahip [27] çalışmasının ortalama 12 katı damga kapasitesine sahip olduğu görülmüştür. XML alanlara damga yerleştiren çalışmalarda sağlamlık analizinde kullanılan saldırılardan Element Sıra Karıştırma, Etiket Ad Karıştırma

ve Alan Silme/Karıştırma saldırılarından etkilenmediği, sadece verinin yok edilmesine sebep olan Etiket Silme saldırısından etkilendiği görülmüştür. Önerilen yöntem karakterler üzerinde Homoglif dönüşüm gerçekleştirdiğinden fark edilemezliği yüksektir. Damga verisini çıkardıktan sonra damgalama öncesi orijinal veriyi elde edebilmektedir. XML verinin barındırdığı nümerik ve metin veri türlerine damga yerleştirebilirken, damga çıkarım işlemi için farklı bir ortamda saklanan veriye ihtiyacı yoktur ve damgalama sonrası orijinal verinin bozulmasına sebep olmamaktadır. Tüm bu özellikleri bünyesinde barındıran tek çalışma olarak gerçekleştirilen saldırılara karşı dayanıklılığı üst düzeydedir.

Ateş Böceği algoritmasını kullanarak damgalama sonrası bozulmayı minimize ederken saldırılara karşı sağlamlığı artıracak geri dönüşümlü yeni bir veri tabanı damgalama yöntemi önerilmiştir. 10729 satırlık örnek veri setinde [30] çalışmasının damga yerleştirme süresi 1016 saniye iken önerilen yöntem bu süreyi 941 saniyeye düşürmüştür. Damgalama yöntemlerinin başarısını belirleyen kriterlerden bir diğeri olan damgalama kapasitesinde de önerilen yöntem artış sağlamıştır. 21727 satırlık örnek veri setinde [29] ve [30] çalışmalarının saklayabildiği damga veri oranı %31 ve %33 iken önerilen yöntemin oranı %37 olmuştur. Yüksek kapasitede damga verisini saklarken damga verisinin saldırganlar tarafından algılanamaz olması gerekir. Yapılan güvenlik testlerinde damgalama sonrası alanların standart sapma ve ortalama değerlerindeki değişimler incelenmiştir. Önerilen yöntemin damga yerleştirdiği tüm alanların standart sapmasında ve ortalama değerlerinde sebep olduğu değişim [30] yönteminden daha düşük olmuştur. Yapılan testlerde önerilen yöntemin saldırılara karşı dayanıklı olduğu görülmüştür. Alt Küme Ekleme ve Sıralama saldırılarından etkilenmeyen yöntem, Alt Küme Silme saldırısının %50 oranında yapıldığı testlerde damga verisinin %50'sini kurtarabilmiştir. Aynı oranda yapılan Bit Değiştirme Saldırısında kurtarılan damga oranı %61'e çıkmıştır. Birden çok saldırının art arda yapıldığı Satır Bazlı Çoklu Saldırıda önerilen yöntem [30] çalışmasına kıyasla damga çıkarımı %13 oranında artırmıştır. Alan Bazlı Çoklu Saldırıda ise önerilen yöntem [29] ve [30] çalışmalarına kıyasla damga çıkarımı %11 ve %21 oranında artırmıştır. Ateş Böceği Algoritmasının popülasyon birey ve maksimum adım sayısı değişiminin önerilen yöntem başarısı üzerine etkisi incelendiğinde, düşük popülasyonlar ve adım sayılarında bile damgalama oranının %40'lara yükseldiği görülmüştür.

Sayısal alanlar ile birlikte metin alanlarına da damga yerleştirirken sayısal alanlardaki bozulmayı minimize eden, metin alanlar üzerinde herhangi bir bozulmaya sebep olmadan maksimum damgalama kapasitesine ulaşan yeni bir damgalama yöntemi önerilmiştir. Farklı

kayıt sayısına sahip veri setlerinde gerçekleştirilen testlerde önerilen yöntemin çalışma süresinin [30] çalışmasından düşük, [88] çalışmasına yakın olduğu görülmüştür. Damgalama yöntemlerinin başarısını belirleyen kriterlerden bir diğeri olan damgalama kapasitesi de [30] ve [88] çalışmalarının ortalama 6 katı olmuştur. Yapılan saldırı testlerinde önerilen yöntemin Alt Küme Ekleme ve Sıralama saldırılarından etkilenmediği görülmüştür. Alt Küme Silme saldırısında [30] ve [88] çalışmalarına benzer başarılar elde eden yöntem, Bit Değiştirme saldırısında metin alanlara damga yerleştirebilme özelliğinden dolayı yüksek başarı sağlamıştır. %50 oranında Bit Değiştirme saldırısı yapılan test veri kümesinde [30] ve [88] çalışmaları %61 ve %62 oranında veri kurtarabilirken, önerilen yöntem %93 oranında damga verisini kurtarabilmiştir. Birden çok saldırının art arda yapıldığı Satır Bazlı Çoklu Saldırıda önerilen yöntem [30] ve [88] çalışmalarına kıyasla damga çıkarımı %13 oranında artırarak %88 yapmıştır. Alan Bazlı Çoklu Saldırıda [30] ve [88] çalışmaları %40 ve %62 oranında damgayı başarıyla çıkarabilirken, önerilen yöntemin oranı %92'dir.

5. ÖNERİLER

Sayısal damgalama veri tabanları üzerinde ilk kez 2002 yılında uygulanmış ve zaman içerisinde araştırmacılar tarafından ilgi görmüştür. Konunun, oldukça yeni olmasına rağmen yapılan çalışmalar, geliştirmelere ve yeniliklere açık olduğunun göstergesidir. Araştırmacılar çoğunlukla sayısal alanlar üzerine damga yerleştirmeyi tercih etmektedir. Veri tabanlarındaki tarih ve ikili verilerin saklandığı alanlar üzerinde damga saklanmasını sağlayan yöntemler geliştirilebilir.

Veri tabanlarında aynı veri seti içerisinde farklı veri türleri tutan alanlar bulunmaktadır. Damgalama yönteminin tek bir veri türüne odaklanması yerine, birden çok veri türünü desteklemesi yöntemin farklı veri tabanlarına uygulanabilir olmasını sağlayacaktır. Bu sebeple farklı veri türlerine beraber damga saklayabilen yöntemler geliştirilebilir.

Farklı veri türlerine damga saklayabilen yöntemlerin damga verisini türler arasında yedeklemesi sağlanabilir. Örneğin sayısal alan veya alanlar kullanılarak saklanan verinin aynı satırın farklı türdeki alanında da saklanması sayesinde, sayısal alanlara yapılacak saldırı ile kaybedilen damga verisinin metin alan ile kurtarılabilmesi sağlanabilir.

Damgalama sonrası sayısal alanlardaki bozulmanın olabildiğince az olması amaçlanmaktadır. Bu sayede damga verisinin saldırganlar tarafından algılanması zor olacaktır. Bu amaçla sayısal değerler üzerinde daha düşük değişimlere sebep olacak dönüşüm algoritmaları veri tabanı damgalamaya uyarlanarak saldırılara karşı dayanıklı yöntemler geliştirilebilir.

Metin verisi tutan alanlar üzerinde gerçekleştirilen veri tabanı damgalama çalışmaları boşluk ekleme, kelimeyi sinonim havuzundaki karşılığı ile değiştirme veya metin içerisine göz ile belirlenemeyen gizli karakter ekleme yöntemlerini kullanmaktadırlar. Homoglif dönüşümde kullanılacak karakterlerin veri tabanının kullanıldığı alan dikkate alınarak seçilmesi ile algılanması zor, damgalama kapasitesi yüksek yöntemler geliştirilebilir.

Veri tabanı damgalama yönteminin kullanımda olan bir veri tabanı üzerinde uygulandığı düşünüldüğünde, yeni eklenen satır ve güncellenen alan değerleri üzerine damganın gerçek zamanlı gizlenmesi gerekecektir. Veri tabanını kullanan uygulamanın performansının etkilenmemesi için damgalama yöntemlerinin çalışma sürelerinin olabildiğinde düşürülmesi gerekmektedir.

Literatürdeki çalışmaların sağlamlığının test edildiği saldırıların daha zeki ve çevik olması sağlanabilir. Değiştirilecek ve güncellenecek satırların veri setindeki değerler incelenerek damga olması muhtemel satır ve alanlar üzerinde saldırıların yapılması bu alandaki çalışmaların saldırılara karşı başarısını yeniden sorgulamasını sağlayacaktır.

6. KAYNAKLAR

1. Coron, J.-S., What is cryptography?, IEEE security & privacy, 4,1 (2006) 70-73.
2. O'Ruanaidh, J.J., Dowling, W. ve Boland, F., Watermarking digital images for copyright protection, IEE Proceedings-Vision, Image and Signal Processing, 143,4 (1996) 250-256.
3. Hartung, F. ve Girod, B., Watermarking of uncompressed and compressed video, Signal processing, 66,3 (1998) 283-301.
4. Boney, L., Tewfik, A.H. ve Hamdy, K.N., Digital watermarks for audio signals, Multimedia Computing and Systems, 1996., Proceedings of the Third IEEE International Conference on 1996: 473-480.
5. Czerwinski, S., Fromm, R. ve Hodes, T., Digital music distribution and audio watermarking, UCB IS, 219 (2007).
6. Collberg, C.S. ve Thomborson, C., Watermarking, tamper-proofing, and obfuscation-tools for software protection, IEEE Transactions on software engineering, 28,8 (2002) 735-746.
7. Collberg, C., Carter, E., Debray, S., Huntwork, A., Kececioglu, J., Linn, C. ve Stepp, M., Dynamic path-based software watermarking, ACM Sigplan Notices, 39,6 (2004) 107-118.
8. Cousot, P. ve Cousot, R., An abstract interpretation-based framework for software watermarking, ACM Sigplan Notices 2004, 39: 173-185.
9. Gupta, G. ve Pieprzyk, J., Software watermarking resilient to debugging attacks, Journal of multimedia, 2,2 (2007) 10-16.
10. Honsinger, C., Digital watermarking, Journal of Electronic Imaging, 11,3 (2002) 414.
11. Cox, I.J. ve Miller, M.L., Review of watermarking and the importance of perceptual modeling, Human Vision and Electronic Imaging II 1997, 3016: 92-100.
12. Johnson, N.F., Duric, Z. ve Jajodia, S., Information Hiding: Steganography and Watermarking-Attacks and Countermeasures: Steganography and Watermarking: Attacks and Countermeasures, 1, Springer Science & Business Media, 2001.
13. Petitcolas, F.A. ve Katzenbeisser, S., Information Hiding Techniques for Steganography and Digital Watermarking (Artech House computer security series), Artech House, 2000.
14. Agrawal, R. ve Kiernan, J., Watermarking relational databases, VLDB'02: Proceedings of the 28th International Conference on Very Large Databases 2002, 155-166.
15. Hu, Z., Cao, Z. ve Sun, J., An image based algorithm for watermarking relational databases, Measuring Technology and Mechatronics Automation, 2009. ICMTMA'09. International Conference on 2009, 1, 425-428.
16. Zhou, X., Huang, M. ve Peng, Z., An additive-attack-proof watermarking mechanism for databases' copyrights protection using image, Proceedings of the 2007 ACM symposium on Applied computing 2007, 254-258.
17. Al-Haj, A. ve Odeh, A., Robust and blind watermarking of relational database systems, (2008).
18. Sion, R., Atallah, M. ve Prabhakar, S., Rights protection for relational data, IEEE transactions on knowledge and data engineering, 16,12 (2004) 1509-1525.

19. Pournaghshband, V., A new watermarking approach for relational data, Proceedings of the 46th annual southeast regional conference on XX 2008, 127-131.
20. Gross-Amblard, D., Query-preserving watermarking of relational databases and XML documents, Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems 2003, 191-201.
21. Kamran, M. ve Farooq, M., A formal usability constraints model for watermarking of outsourced datasets, IEEE transactions on information forensics and security, 8,6 (2013) 1061-1072.
22. Ng, W. ve Lau, H.-L., Effective approaches for watermarking xml data, International Conference on Database Systems for Advanced Applications 2005, 68-80.
23. Guo, F., Wang, J., Zhang, Z. ve Li, D., Knowledge Discovery from XML Documents, A new scheme to fingerprint XML data, Springer, 85-94, 2006.
24. Yao, R., Zhao, Q. ve Lu, H., A novel watermark algorithm for integrity protection of XML documents, International Journal of Computer Science and Network Security, 6,2B (2006) 202-207.
25. Darwish, S.M., An optimized fingerprinting system for tamper proof and copy control of XML documents, Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of 2010, 454-457.
26. Darwish, S., New system to fingerprint extensible markup language documents using winnowing theory, IET signal processing, 6,4 (2012) 348-357.
27. Romaric, T., Damiani, E. ve Bennani, N., Robust XML watermarking using fuzzy queries, Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual 2012, 433-438.
28. Wen, Q., Wang, Y. ve Li, P., Two Zero-Watermark methods for XML documents, Journal of Real-Time Image Processing, 14,1 (2018) 183-192.
29. Gupta, G. ve Pieprzyk, J., Reversible and blind database watermarking using difference expansion, Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop 2008, 24.
30. Jawad, K. ve Khan, A., Genetic algorithm and difference expansion based reversible watermarking for relational databases, Journal of Systems and Software, 86,11 (2013) 2742-2753.
31. Bhattacharya, S. ve Cortesi, A., A Distortion Free Watermark Framework for Relational Databases, ICSOFT (2) 2009, 229-234.
32. Hanyurwimfura, D., Liu, Y. ve Liu, Z., Text format based relational database watermarking for non-numeric data, Computer Design and Applications (ICCDA), 2010 International Conference on 2010, 4: V4-312-V314-316.
33. Farfoura, M.E., Horng, S.-J., Lai, J.-L., Run, R.-S., Chen, R.-J. ve Khan, M.K., A blind reversible method for watermarking relational databases based on a time-stamping protocol, Expert Systems with Applications, 39,3 (2012) 3185-3196.
34. Petitcolas, F.A., Anderson, R.J. ve Kuhn, M.G., Information hiding-a survey, Proceedings of the IEEE, 87,7 (1999) 1062-1078.
35. Miller, G. ve Fellbaum, C., Wordnet: An electronic lexical database, 1998.
36. Alattar, A.M., Reversible watermark using the difference expansion of a generalized integer transform, IEEE transactions on image processing, 13,8 (2004) 1147-1156.
37. Thodi, D.M. ve Rodriguez, J.J., Prediction-error based reversible watermarking, Image Processing, 2004. ICIP'04. 2004 International Conference on 2004, 3: 1549-1552.

38. Sion, R., Atallah, M. ve Prabhakar, S., On watermarking numeric sets, International Workshop on Digital Watermarking 2002, 130-146.
39. Li, Y., Guo, H. ve Jajodia, S., Tamper detection and localization for categorical data using fragile watermarks, Proceedings of the 4th ACM workshop on Digital rights management 2004, 73-82.
40. Guo, H., Li, Y., Liu, A. ve Jajodia, S., A fragile watermarking scheme for detecting malicious modifications of database relations, Information Sciences, 176,10 (2006) 1350-1378.
41. Kamel, I., A schema for protecting the integrity of databases, computers & security, 28,7 (2009) 698-709.
42. Rashidi, H., A novel watermarking scheme for detecting and recovering distortions in database tables, arXiv preprint arXiv:1009.0827, (2010).
43. Hamadou, A., Sun, X., Gao, L. ve Shah, S.A., A fragile zero-watermarking technique for authentication of relational databases, International Journal of Digital Content Technology and its Applications, 5,5 (2011).
44. Khan, A. ve Husain, S.A., A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations, The Scientific World Journal, 2013 (2013).
45. Iqbal, S., Rauf, A., Mahfooz, S., Khusro, S. ve Shah, S.H., Self-constructing fragile watermark algorithm for. relational database integrity proof, World Applied Sciences Journal, 19,9 (2012) 1273-1277.
46. Prasannakumari, V., A robust tamperproof watermarking for data integrity in relational databases, Research Journal of Information Technology, 1,3 (2009) 115-121.
47. Camara, L., Li, J., Li, R. ve Xie, W., Distortion-free watermarking approach for relational database integrity checking, Mathematical problems in engineering, 2014 (2014).
48. Odeh, A. ve Al-Haj, A., Watermarking relational database systems, Applications of Digital Information and Web Technologies, ICADIWT 2008. First International Conference on the 2008, 270-274.
49. Choi, K.-C. ve Pun, C.-M., Difference Expansion Based Robust Reversible Watermarking with Region Filtering, Computer Graphics, Imaging and Visualization (CGiV), 2016 13th International Conference on 2016, 278-282.
50. Wang, S.-H. ve Lin, Y.-P., Wavelet tree quantization for copyright protection watermarking, IEEE transactions on image processing, 13,2 (2004) 154-165.
51. Agrawal, R., Haas, P.J. ve Kiernan, J., Watermarking relational data: framework, algorithms and analysis, The VLDB Journal—The International Journal on Very Large Data Bases, 12,2 (2003) 157-169.
52. Wang, H., Cui, X. ve Cao, Z., A speech based algorithm for watermarking relational databases, Information Processing (ISIP), 2008 International Symposiums on 2008, 603-606.
53. Wang, C., Wang, J., Zhou, M., Chen, G. ve Li, D., Atbam: An arnold transform based method on watermarking relational data, Multimedia and Ubiquitous Engineering, 2008, MUE 2008, International Conference on 2008, 263-270.
54. Sion, R., Proving ownership over categorical data, Data Engineering, 2004. Proceedings. 20th International Conference on 2004, 584-595.
55. Hu, T.-L., Chen, G., Chen, K. ve Dong, J.-X., Garwm: Towards a generalized and adaptive watermark scheme for relational data, International Conference on Web-Age Information Management 2005, 380-391.

56. Huang, M., Cao, J., Peng, Z. ve Fang, Y., A new watermark mechanism for relational data, Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on 2004: 946-950.
57. Guo, F., Wang, J., Zhang, Z., Ye, X. ve Li, D., An improved algorithm to watermark numeric relational data, International Workshop on Information Security Applications 2005: 138-149.
58. Zhang, Y., Niu, X. ve Zhao, D., A method of protecting relational databases copyright with cloud watermark, International Journal of Information and Communication Engineering, 1,7 (2005) 337-341.
59. Gupta, G. ve Pieprzyk, J., Database relation watermarking resilient against secondary watermarking attacks, International Conference on Information Systems Security 2009: 222-236.
60. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E. ve Yergeau, F., Extensible markup language (XML), World Wide Web Journal, 2,4 (1997) 27-66.
61. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E. ve Yergeau, F., Extensible markup language (XML) 1.0. (2008).
62. Dournaee, B. ve Dournee, B., XML security, Mcgraw-hill New York, 2002.
63. Ekelhart, A., Fenz, S., Goluch, G., Steinkellner, M. ve Weippl, E., XML security–A comparative literature review, Journal of Systems and Software, 81,10 (2008) 1715-1724.
64. Rizzo, S.G., Bertini, F. ve Montesi, D., Content-preserving text watermarking through unicode homoglyph substitution, Proceedings of the 20th International Database Engineering & Applications Symposium 2016, 97-104.
65. Unicode Staff, C., The Unicode Standard: Worldwide Character Encoding, Addison-Wesley Longman Publishing Co., Inc., 1991.
66. http://www.ktu.edu.tr/dosyalar/silvikultur_c9745.pdf 20/02/2018
67. Kennedy, J. ve Eberhart, R., Particle swarm optimization, Neural Networks, 1995. Proceedings., IEEE International Conference on, Nov/Dec 1995 1995, 4, 1942-1948 1944.
68. Dorigo, M., Maniezzo, V. ve Colorni, A., The ant system: An autocatalytic optimizing process, 1991.
69. Karaboga, D. An idea based on honey bee swarm for numerical optimization2005.
70. Yang, X.-S., Nature-inspired metaheuristic algorithms, Luniver press, 2010.
71. Mehta, B.B. ve Rao, U.P., A Novel approach as Multi-place Watermarking for Security in Database, arXiv preprint arXiv:1402.7341, (2014).
72. Afridi, T.H., Khan, A. ve Lee, Y.S., Mito-GSAAC: mitochondria prediction using genetic ensemble classifier and split amino acid composition, Amino Acids, 42,4 (2012) 1443-1454.
73. Hayat, M. ve Khan, A., MemHyb: predicting membrane protein types by hybridizing SAAC and PSSM, Journal of theoretical biology, 292 (2012) 93-102.
74. Khan, A. ve Mirza, A.M., Genetic perceptual shaping: Utilizing cover image and conceivable attack information during watermark embedding, Information fusion, 8,4 (2007) 354-365.
75. Naveed, M. ve Khan, A.U., GPCR-MPredictor: multi-level prediction of G protein-coupled receptors using genetic ensemble, Amino Acids, 42,5 (2012) 1809-1823.
76. Tahir, M., Khan, A. ve Majid, A., Protein subcellular localization of fluorescence imagery using spatial and transform domain features, Bioinformatics, 28,1 (2011) 91-97.

77. Bhowmick, S.S., Wee, T.K., Leonardi, E. ve Madria, S., Storing dtd-conscious xml data in xedy, International Conference on Electronic Commerce and Web Technologies 2003, 270-280.
78. Maneth, S., Mihaylov, N. ve Sakr, S., XML tree structure compression, Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on 2008, 243-247.
79. Lohrey, M., Maneth, S. ve Mennicke, R., XML tree structure compression using RePair, Information Systems, 38,8 (2013) 1150-1167.
80. Inoue, S., Makino, K., Murase, I., Takizawa, O., Matsumoto, T. ve Nakagawa, H., A proposal on information hiding methods using XML, The 1st Workshop on NLP and XML 2001, 707-710.
81. Eler, Ü., Ağaç Serveti Envanterinin Yapılması Amacıyla Meşcere Tipi Ayırımı Üzerine Araştırmalar, İÜ Orman Fakültesi Dergisi, Seri A, 28 (1978) 293-324.
82. Özçelik, R., Kizilçamda (Pinus brutia Ten.) Çağ Siniflarinin Meşcere Tipi Ayrimi Üzerine Etkileri.
83. Anlar, H.C., Günlü, A., Keleş, S. Ve Bulut, S., Spot-4 Uydu Görüntüsü Yardımıyla Bazı Meşcere Parametreleri (Gelişim Çağı ve Kapalılık) ve Arazi Kullanım Sınıflarının Belirlenmesi; Devrez Planlama Birimi Örneği, Anadolu Orman Araştırmaları Dergisi, 1,1-2 (2016) 33-40.
84. Karahalil, U., Başkent, E.Z., Sivrikaya, F. ve Kılıç, B., Analyzing deadwood volume of Calabrian pine (Pinus brutia Ten.) in relation to stand and site parameters: a case study in Köprülü Canyon National Park, Environmental monitoring and assessment, 189,3 (2017) 112.
85. Özkan, U.Y. ve Yeşil, A., Forest stand delineation using Ikonos image and object based image analysis, Journal of the Faculty of Forestry Istanbul University| İstanbul Üniversitesi Orman Fakültesi Dergisi, 66,2 (2016) 600-612.
86. Asan, Ü., Orman kaynaklarının çok amaçlı kullanım ve fonksiyonel planlama, Journal of the Faculty of Forestry Istanbul University| İstanbul Üniversitesi Orman Fakültesi Dergisi, 40,3 (1990) 67-84.
87. Avcı, Z.D.U., Kuşak, B. ve Kuşak, L., Meşcere Tiplerinin Uydu Verileri ile Belirlenmesinde Farklı Doku Ölçütlerinin Değerlendirilmesi.
88. Imamoglu, M.B., Ulutas, M. ve Ulutas, G., A New Reversible Database Watermarking Approach with Firefly Optimization Algorithm, Mathematical problems in engineering, 2017 (2017).

ÖZGEÇMİŞ

Mustafa Bilgehan İMAMOĞLU; 1980 yılında Trabzon'da doğdu. İlk ve orta öğrenimini Trabzon'da tamamladı. 1998 yılında Trabzon Süper Lisesi'nden mezun oldu. Aynı yıl ÖSYS ile yerleştirildiği Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü'nü 2002 yılında bitirdi. 2005 yılında aynı bölümde yüksek lisansını tamamladı. 2008 yılında Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı'nda doktora çalışmalarına başladı. 2002-2013 yılları arasında Karadeniz Teknik Üniversitesi Enformatik Bölümü'nde Öğretim Görevlisi olarak görev yaptı. 2013 yılında Mühendis olarak atandığı Karadeniz Teknik Üniversitesi Bilgi İşlem Daire Başkanlığı'nda halen çalışmaya devam etmektedir. Evli ve iki çocuk babasıdır. Başlıca yayınları aşağıda verilmiştir.

Imamoglu, M.B., Ulutas, M. ve Ulutas, G., A New Reversible Database Watermarking Approach with Firefly Optimization Algorithm, Mathematical problems in engineering, 2017 (2017).

İmamoğlu, M. B., Ulutaş, M., & Ulutaş, G. (2015, May). A watermarking technique for relational databases based on partitioning. In Signal Processing and Communications Applications Conference (SIU), 2015 23th (pp. 2094-2097). IEEE.

Imamoglu M.B., Ulutaş G., Ulutaş M., Numeric Transform Based Digital Watermarking Technique for Relational Databases, ISCTURKEY 2016, ANKARA, TÜRKIYE, 2-4 Kasım 2016, pp.1-4