T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM GÜVENLİK AÇIKLARININ SKORLANMASI VE KATEGORİLERİNİN BELİRLENMESİNDE YENİ BİR YÖNTEM

Hakan KEKÜL

Doktora Tezi

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Donanım

HAZİRAN 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

YAZILIM GÜVENLİK AÇIKLARININ SKORLANMASI VE KATEGORİLERİNİN BELİRLENMESİNDE YENİ BİR YÖNTEM

Tez Yazarı
Hakan KEKÜL

Danışman
Prof. Dr. Burhan ERGEN

HAZİRAN 2022
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Başlığı:	Yazılım Güvenlik Açıklarının Skorlanması ve Kategorilerinin Belirlenmesinde Yeni Bir Yöntem
Yazarı:	Hakan KEKÜL
İlk Teslim	11.05.2022
Savunma	20.06.2022

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

		İmza
Danışman:	Prof. Dr. Burhan ERGEN Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Başkan:	Prof. Dr. Bilal ALATAŞ Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Prof. Dr. Mehmet KARAKÖSE Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Doç. Dr. Zafer CÖMERT Samsun Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Doç. Dr. Ahmet Gürkan YÜKSEK Sivas Cumhuriyet Üniversitesi, Mühendislik Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Yazılım Güvenlik Açıklarının Skorlanması ve Kategorilerinin Belirlenmesinde Yeni Bir Yöntem" Başlıklı Doktora Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

20.06.2022

Hakan KEKÜL

ÖNSÖZ

Günümüzde siber güvenlik, bilişim ve yazılım dünyasının en önemli araştırma alanlarından biridir. Yazılımlar modern günlük yaşantımızın içinde çok önemli bir yere sahiptir. Birçok durumda, yazılım sisteminde oluşabilecek sorunlar kötü sonuçlara sebep olabilmektedir. Bu nedenle, yazılım güvenliğini belirleyebilmek önemli bir gereksinim halini almıştır. Araştırmaların pek çoğunda yazılım güvenlik açıklarının yer aldığı, farklı araştırma grupları tarafından oluşturulan güvenlik açığı veri tabanları kullanılmaktadır. Yazılım güvenlik açıkları ve bunların yer aldığı veri tabanları, uzman kişiler tarafından manuel tespit edilerek sınıflandırılmakta, kategorize edilmekte ve skorlanmaktadır. Manuel olarak uzmanlar tarafından gerçekleştirilen bu işlemler zaman gecikmelerine ve insan doğasından kaynaklanan hatalara neden olmaktadır.

Yayınlanan güvenlik açığı verilerinin kötüye kullanım riski olsa da bu alanın araştırmacıları için önemli bir kaynak oluşturmaktadır. Bu nokta da güvenlik açığı verilerini kullanarak pek çok çalışma yapılmaktadır. Yazılım kalitesi, güvenlik açığı özelliklerinin analizi, keşfi ve skorlarını tahmin ederek sınıflandıracak çalışmalar son yıllarda giderek artmaktadır. İlk çalışmalarda geleneksel yaklaşımlar uygulanmış olsa da başarı istenen seviyede olamamıştır. Makine öğrenmesi ve veri madenciliği teknikleri birçok alanda başarılar elde etmesi, yazılım bileşen kalitesi ve güvenlik açıklarının belirlenmesinde de kullanılabileceğini düşündürmüştür. Çalışmamızda, yazılım güvenlik açıklarının analizi ve keşif problemi için makine öğrenme ve veri madenciliği tekniklerini kullanan hibrit bir model geliştirilmiştir.

Doktora çalışmalarım kapsamında kendisi ile çalışma imkânı sağlayan, akademik ve insani destekleri ile bu zorlu süreci atlatmamda büyük katkısı olan danışman hocam Prof. Dr. Burhan ERGEN'e sonsuz teşekkürlerimi sunarım. Doktora çalışmalarımın gerçekleştirilmesinde desteklerini esirgemeyen Dr. Öğr. Üyesi Halil ARSLAN'a teşekkür ederim. Bu günlere gelmemde ve başarımda en büyük destekçilerim annem ve babama, tüm destek ve sabırlarıyla arkamda olan eşim Yasemin ve oğullarım İbrahim Kerem ve Yunus Emre'ye sonsuz teşekkürlerimi sunarım.

Doktora süreci boyunca ihmal ettiğim fedakâr eşim YASEMİN KEKÜL'e ithafen…

Bu tez çalışması, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından **121E298** protokol numaralı proje ile desteklenmiştir.

Hakan KEKÜL
ELAZIĞ, 2022

İÇİNDEKİLER

Sayfa

ÖZET

Yazılım Güvenlik Açıklarının Skorlanması ve Kategorilerinin Belirlenmesinde Yeni Bir Yöntem

Hakan KEKÜL

Doktora Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı
Haziran 2022, Sayfa: xii + 89

Yazılım güvenlik açıkları, şahıslar, şirketler ve ülkeler için finansal risklere ve itibar kayıplarına neden olabilmektedir. Yazılım güvenlik açıklarının giderilmesi, test kaynaklarının yetersizliği ve uzman personel eksikleri nedeni ile istenilen seviyede değildir. Kurumların itibarlarını korumak ve güvenli yazılımlar geliştirmek adına sınırlı kaynaklarını doğru kullanarak, test ve düzeltmeleri planlamaları gerekmektedir. Ancak güvenlik vektörlerinin sahip olduğu metrik değerlerinin tespit edilmesi manuel bir işlem olarak insanlar tarafından yapılmaktadır. Bu nedenle bu süreç, zaman almakta ve insanın doğasından kaynaklanan hatalar barındırabilmektedir. Bu metrikler güvenlik açığı önem derecelerinin hesaplanmasında kullanılmasından dolayı önemlidir. Güvenlik açığı analizi ve keşfi işlemlerinin kalitesini artırmak ve süreçleri hızlandırmak için makine öğrenmesi algoritmalarının ve veri madenciliği tekniklerinin kullanılması gerekmektedir. Ancak bu alanda yapılan çalışmalar hala sınırlıdır. Bu çalışmada doğal dil işleme tekniklerinden Bag of Words, Term Frequency Inverse Document Frequency, Ngram, Word2Vec, Doc2Vec ve FastText özellik çıkarımı yöntemleri kullanılarak güvenlik açığı vektörlerinin farklı çok sınıflı sınıflandırılma algoritmaları ile tahmini gerçekleştirilmiştir. Elde edilen metrik değerleri ile güvenlik açığı önem skorları hesaplanmıştır. Sınıflandırma aşamasında Naive Bayes, Desicion Tree, K-Nearest Neighbors, Multi-layer Perceptron ve Random Forest algoritmaları kullanılmıştır. Kamuya açık büyük bir veri setini kullandığımız deneyler, değerlendirmeyi kolaylaştırır ve yazılım güvenlik açığı vektörlerinin sınıflandırılmasında standartlara uygun bir tahmin modeli sunar. Elde edilen sonuçlar çok olasılıklı ve tahmini zor bir problemde farklı tekniklerin ve sınıflandırma algoritmalarının birlikte kullanımının umut verici olduğunu göstermektedir. Ayrıca çalışmamız, kullandığı veri boyutu ve henüz üzerinde pek çalışılmamış güvenlik açığı skorlama sistemi versiyonlarını kapsaması bakımından alanında önemli bir boşluğu doldurmaktadır.

Anahtar Kelimeler: Yazılım güvenliği, Yazılım güvenlik açıkları, Bilgi güvenliği, Siber güvenlik, Metin analizi, Çok sınıflı sınıflandırma

ABSTRACT

A New Method to Determine Scoring and Category of Software Vulnerabilities

Hakan KEKÜL

Ph.D. Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Computer Engineering
June 2022, Pages: xii + 89

Software vulnerabilities can cause financial and reputational losses for individuals, companies, and countries. Software vulnerabilities have not been eliminated to the desired level due to insufficient test resources and lack of expert personnel. In order to protect their reputations and develop secure software, organizations need to accurately plan tests and implement corrections using limited resources. However, the metric values of security vectors are manually determined by humans, which takes time and may introduce errors stemming from human nature. These metrics are important because of their role in the calculation of vulnerability severity. It is necessary to use machine learning algorithms and data mining techniques to improve the quality and speed of vulnerability analysis and discovery processes. However, studies in this area are still limited. In this study, vulnerability vectors were estimated using the natural language processing techniques bag of words, term frequency–inverse document frequency, Ngram, Word2Vec, Doc2Vec, and FastText for feature extraction together with various multiclass classification algorithms, namely Naive Bayes, decision tree, k-nearest neighbors, multilayer perceptron, and random forest. Our experiments using a large public dataset facilitate assessment and provide a standard-compliant prediction model for classifying software vulnerability vectors. The results show that the joint use of different techniques and classification algorithms is a promising solution to a multi-probability and difficult-to-predict problem. In addition, our study fills an important gap in its field in terms of the size of the dataset used and because it covers a vulnerability scoring system version that has not yet been extensively studied.

Keywords: Software security, Software vulnerability, Information security, Cyber security, Text analysis, Multiclass classification

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

KISALTMALAR

Kısaltmalar

AC	: Access Complexity / Attack Complexity
AI	: Availability Impact
AR	: Availability Requirement
Au	: Authentication
AV	: Access/Attack Vector
BoW	: Bag of Words
CBOW	: Continous Bag of Words
CDP	: Collateral Damage Potential
CERT	: Computer Emergency Response Team
CI	: Confidentiality Impact
CR	: Confidentiality Requirement
CVE	: Common Vulnerabilities and Exposures
CVSS	: Common Vulnerability Scoring System
CWE	: Common Weakness Enumeration
E	: Exploitability / Exploit Code Maturity
II	: Integrity Impact
IR	: Integrity Requirement
NIST	: National Institute of Standards and Technology
NIST	: National Institute of Standards and Technology
NVD	: National Vulnerability Database
PoC	: Proof of Concept Code
PR	: Privileges Required
RC	: Report Confidence
RL	: Remediation Level
SARD	: Software Assurance Reference Dataset
SRD	: Standard Reference Dataset
TD	: Target Distribution
TF-IDF	: Term Frequency Inverse Document Frequency
UI	: User Interaction
WIVSS	: Weighted Impact Vulnerability Scoring System

1. GİRİŞ

Bilişim Teknolojilerinin temel araştırma alanlarından birini, siber güvenlik çalışmaları oluşturmaktadır. Bireylerin, kurumların ve devletlerin günlük işlemlerinin büyük bir çoğunluğunu yazılım ve bilişim sistemleri aracılığı ile gerçekleştirdiği günümüzde, siber güvenlik büyük bir öneme sahiptir. Siber güvenliğin alanı içerisinde yazılım güvenlik açıkları önemli bir yer tutmaktadır. Çünkü yazılımlardaki tasarımsal hatalar, çalışma zamanında istenmeyen güvenlik ihlallerine neden olabilmektedir.

Güvenlik açıklarının kötüye kullanımı olarak ortaya çıkan siber suçlar büyük zararlara neden olabilmektedir. Siber suçlar sonucu meydana gelecek zararın trilyonlarca dolara mal olacağı tahmin edilmektedir [1]. Tespit edilen güvenlik açıklarının güvenlik politikasını ne ölçüde ihlal ettiğinin belirlenmesi çok önemlidir. Son zamanlarda güvenlik açıklarının kötüye kullanımı ile ilgili endişelerin artığı görülmektedir. Bu nedenle istismar edilmeden açıkların sınıflandırılması gerekmektedir [2]. Güvenlik açıklarının sayısı büyük bir hızla artmaktadır. Yapılan çalışmalar oluşan büyük boyutlu arşivin istatistiki tekniklerle değerlendirilemeyeceğini ve bu sorunun giderek artığını göstermektedir. Seksen binin üzerinde arşivlenmiş güvenlik açığının düzenli regresyon analizine dayanan ampirik sonuçlara göre, güvenlik açıklarının skorlarının hesaplanması için harcanan çaba zaman gecikmelerini istatistiksel olarak negatif yönde etkilemektedir [3]. Bu boyutta verinin kapsamlı analizinin yapılamaması keşfedilemeyen noktaların varlığına işaret etmektedir [4].

Mevcut çalışmanın ele aldığı temel problem, yazılım güvenlik açığı vektörlerinin metrik değerlerini belirmek ve önem skorlarını hesaplamaktır. Güvenlik açıklarını değerlendirebilmek için vektörler ve metrikler çok önemli bilgiler sağlar. Ayrıca güvenlik açığının ciddiyeti hakkında en önemli ve ilk bilgi önem skorudur. Bu skor, güvenlik metriklerinin sonucunda oluşturulan vektörler üzerinden hesaplanmaktadır. Alanın uzmanları, şirketler ve güvenlik açığından etkilenenler için durumun önemi noktasında bir göstergedir. Önem skorlarını değerlendirerek sorumlular yazılımın her aşamasında önemli kararlar alabilirler. Örneğin bu skorlar, test, yama, güncelleme öncelikleri ve geliştirme esnasında belirli kütüphanelerin kullanılması veya bir bileşenin uygulamadan çıkarılması gibi kararlar alınmasına yardımcı olabilirler.

1.1. Literatür Araştırması

Son yıllarda yapılan çalışmalar, yapılandırılmamış güvenlik açığı verilerinin analizi için makine öğrenimi tekniklerinin kullanımının hızla artığını göstermektedir [5]. Bu çalışmaların başarılarının artması için iyi yapılandırılmış verilere ve özellik çıkarma tekniklerine ihtiyaç duyulduğu görülmektedir [4,6–8]. Güvenlik açığı değerlendirme faaliyetleri, hangi güvenlik

açıklarına öncelik verileceğinin belirlenmesi açısından önemli bir işlemdir. Bu işlem uzmanlar tarafından manuel olarak yapılmaktadır. Bu durum zaman alıcı ve kesin değildir. Ayrıca güvenlik açıklarının uzman kişiler tarafından manuel tespit edilerek sınıflandırılması maliyetlidir ve insanın doğasından kaynaklanan zaaflar barındırmaktadır [9]. İstismarların yarısı güvenlik açıklarının ilanından sadece iki hafta içerisinde gerçekleşmektedir. Bu nedenle yazılım güvenlik açıklarının, güvenlik metrikleri ile önem skorlarının doğru ve hızlı belirlenebilmesi çok önemlidir [10].

Bu duruma çözüm getirmek isteyen uzmanlar tarafından pek çok yazılım güvenlik açığı puanlama sistemi önerilmiştir. Bunun sonucu olarak farklı özellikleri ile ön plana çıkan puanlama yöntemleri bulunmaktadır [11]. Ulusal Güvenlik Açığı Veri Tabanı (National Vulnerability Database - NVD), en büyük güvenlik açığı veri tabanlarından biridir ve herkese açıktır [12]. Ortak Güvenlik Açığı Skorlama Sistemi (Common Vulnerability Scoring System - CVSS) güvenlik uzmanları tarafından bir standart olarak kabul edilen puanlama yöntemidir. Bunun nedeni NVD tarafından resmi olarak kullanılan sistem olmasıdır. CVSS skorlama sistemi sürekli güncellenen ve yeni sürümleri çıkan bir sistemdir. Şuan da resmi olarak NVD tarafından desteklenen sürümleri CVSS 2.0 ve CVSS 3.1'dir.

Yazılım güvenlik açıklarının analizi ve keşfi son yıllarda yoğun ilgi gören akademik bir çalışma alanıdır. Bunun sonucu olarak yayınlaşmış pek çok çalışma bulunmaktadır. Spanos vd. [9] çalışmalarında, amaçlarını güvenlik açığı karakteristik atamasının manuel prosedürünün geliştirilmesi, hızlandırılması ve desteklenmesi şeklinde ifade etmişlerdir. Bu amaca ulaşmak için, metin analizi ve çoklu hedef sınıflandırma tekniklerini birleştiren bir model geliştirilmişlerdir. Önerdikleri model, güvenlik açığı güvenlik metriklerini tahmin ederek, önem puanlarını tahmin ettikleri metrikler üzerinden hesaplamaktadır. Mevcut araştırmayı gerçekleştirmek için, halka açık NVD veri tabanından 99.091 kayıt içeren bir veri kümesi kullanmışlardır.

Russo vd. [13], güvenlik açıklarının özetini otomatik olarak oluşturabilen ve bunları sınıflarına göre kategorize edebilen bir yaklaşım olan CVErizer'ı sunmuşlardır. Yaklaşımın sınıflandırma yeteneklerini 3369 önceden etiketlenmiş Ortak Güvenlik Açıkları ve Etkilenmeleri (Common Vulnerabilities and Exposures - CVE) kayıt seti üzerinde ampirik olarak değerlendirmişlerdir. On beş siber güvenlik uzmanı öğrenci ve dört profesyonel güvenlik uzmanı CVErizer özetlerinin uçtan uca değerlendirmesini yapmıştır. Çalışmada önerilen yaklaşım CVE açıklamalarından doğru bilgi çıkarılması ve sınıflandırılmasında yüksek performans göstermektedir. Özetler ayrıca, güvenlik açığı değerlendirme süreçlerinde analistlere yardımcı olmak için son derece yararlı kabul edilmektedir.

Yasasin vd. [14], işletim sistemleri, tarayıcılar ve ofis çözümleri de dâhil olmak üzere farklı sistem ve yazılım paketlerinin yayınlanması sonrası güvenlik açığı sayısını tahmin etme sorununu ele almaktadır. Hassasiyetlerin zaman serisi metodolojileri ile analizi literatürde yükselen bir akımdır ve çalışmaları tahmin metodolojilerinin kapsamlı bir analizine katkıda bulunmaktadır.

Çalışmalarında hata metriklerinin sistematik olarak karşılaştırmışlarıdır. Hata metriklerinin avantaj ve dezavantajlarını ortaya koymaktadırlar.

Sharma vd. [15], kelime gömme ve evrişimsel sinir ağları (CNN) kullanarak yazılım güvenlik açıklarını önceliklendirmişlerdir. Veri kümelerini Linux, Microsoft ve Google satıcıları ve karışık satıcıların güvenlik açıklarından oluşturmuşlardır. Veri setlerinde CVE listelerindeki 10.000 güvenlik açığını kullanmışlardır. CVSS puanlarını üç kategoriye ayırarak tahmin etmişlerdir. Beş gruba ayırdıkları veri setlerinin ortalama doğruluğu %87'dir.

Malhotra vd. [16], Apache Tomcat güvenlik açıklarının metinsel tanımlarını girdi olarak kullanarak yazılım güvenlik açıklarının önem derecelerini tahmin etmeye çalışmışlardır. Veri hacimlerini azaltmak için ki-kare ve bilgi kazanımı yöntemlerini kullanmışlardır. Modellerinde sınıflandırma algoritmaları olarak Bagging Technique, Random Forest, Naive Bayes, Support Vector Machine kullanmışlardır. Bilgi kazanım tekniği ile birlikte Naive Bayes algoritmasının en iyi sonuçlar ürettiğini ve yaklaşık %92 başarım elde ettiklerini bildirmişlerdir.

Aota vd. [17], güvenlik açığı raporlarının teknik açıklamalarını kullanarak hangi Yaygın Zayıflık Numaralandırmasına (Common Weakness Enumeration - CWE) ait olduğunu tespit etmeye çalışmışlardır. Çalışmalarında güvenlik raporlarının etiketlenmesinin manuel yapılmasından kaynaklanan hataları düzelmek için metin madenciliği tekniklerini kullanmışlardır. Sınıflandırmaları sonucunda pek çok yanlış sınıflandırılmış rapor olduğunu tespit etmişlerdir. Önerdikleri model yaklaşık %96 oranında doğrulukla CWE etiketlemesi gerçekleştirmiştir.

Wu vd. [6], makine öğrenimi tabanlı güvenlik hata raporları tahmini için büyük ölçekli veri kümeleri oluşturma yaklaşımını önermektedir. Bu çalışmalarında yaklaşık 80 bin hata raporu içeren OpenStack veri kümesinin başlangıç sürümünü oluşturmuşlardır. Sonuç olarak, oluşturulmuş veri setlerinin kalitesini artırmak için diğer yöntemleri (özellik seçimi, derin öğrenme gibi) dahil ederek veri seti oluşturma yaklaşımını geliştirmeyi önermektedirler.

Williams vd. [4], çalışmalarında yıllardır biriken güvenlik açığı verilerinin büyük bir yapılandırılmamış veri grubu haline geldiğini belirtmektedirler. Bu durumun verilerin kapsamlı analizini yapmak için gerekli araçların ve algoritmaların denenmesindeki eksiklikler yüzünden çoğunlukla keşfedilemeyen noktalar olduğunu vurgulamaktadırlar. Çalışmalarının sonucunda güvenlik açığı eğilimleri, evrimi ve etkileşimleri ile güvenlik açıklarına karşı genel ürün duyarlılığı konusunda önemli bir boşluk bulunduğunu ortaya koymuşlardır. Böylece güvenlik açığı verilerinin önemli özelliklerini anlamanın, araştırmacıların ve endüstri uzmanlarının gelecekte güvenli sistemler geliştirmelerinde, güvenlik açıklarından kaynaklanan sorunların azaltılmasında ve yeni akademik çalışma alanlarının ortaya konulmasında önemli faydalar sağlayacağını ifade etmişlerdir.

Fang vd. [18] çalışmalarında, güvenlik açıklarının sadece küçük bir bölümünün saldırganlar tarafından istismar edildiğini belirtmişlerdir. Bu nedenle istismar edilemez güvenlik açıkları ile diğerlerinin ayırt edilmesinin sınırlı kaynakların verimli kullanımı açısından önemini

vurgulamışlardır. Çalışmalarında belirlenen güvenlik açıklarının sistemde yayınlanmasının zaman alması ve NVD'nin kurumsal yapısından kaynaklanan eksikliklerden dolayı yetersiz kaldığı, bu nedenle farklı toplulukların oluşturduğu veri tabanlarının daha verimli özellikler içerdiği ifade edilmiştir.

Yang vd. [10], istismar edilen güvenlik açıklarının, yaklaşık yarısının güvenlik açığının ilanından sonraki iki hafta içerisinde istismar edildiğini belirtmişlerdir. Bunun yanında ilan edilen açıkların sadece %20'sinin istismara maruz kaldığı belirtilmektedir. Bu nedenle güvenlik açıklarının skorlarının doğru bir şekilde tahmin edilmesi ve önceliklendirilmesinin önemini vurgulanmışlardır.

Raducu vd. [7], güvenlik açıklarını tespit etmek için farklı makine öğrenimi tekniklerinin ortaya çıktığını ve geliştirildiğini vurgulamaktadırlar. Ancak, bu algoritmaların performanslarının ve başarımlarının veri kümeleri olarak bilinen çok miktarda verinin işlenmesine dayanan veri güdümlü motorlara ihtiyaç duyduğunu belirtmektedirler.

Şahin vd. [19], yazılım geliştirme aşamalarında artan karmaşıklık ve çeşitlilik sonucu, yazılım güvenlik açıklarının yönetilmesinin zorluğuna vurgu yapmaktadırlar. Yazılım güvenlik açıklarını tespit etmenin zorluğuna vurgu yaptıkları çalışmalarında metasezgisel optimizasyon yöntemlerinin kullanımını önermektedirler. Derin öğrenme ve SYNbiyotik Genetik algoritmaları kullanan yeni bir yazılım güvenlik açığı yaklaşımı önermektedirler. Sonuçlarından yazılım kalitesini ve yazılım güvenlik açıklarının tahminini geliştiren bulgular elde edildiğini öne sürmektedirler.

İncelenen çalışmalarda görüldüğü üzere yazılım güvenlik açıklarının raporlanmaya başlanması ile birlikte akademik camianın bu alana ilgisi artmıştır. Makine öğrenmesi algoritmalarının pek çok problemde başarı ile uygulanması sonrası özellikle 2012 yılından itibaren bu problem özelinde de kullanıldığı görülmektedir [5]. Son yıllarda yapılan çalışmalarda makine öğrenmesi temelli yaklaşımların kullanılması tavsiye edilmektedir. Ancak literatür incelemelerinden de anlaşılacağı üzere yüksek başarım elde edilebilmesi için çalışmalarda yapılandırılmış ve özellikleri çıkarılmış veri setlerinin kullanılmasına ihtiyaç olduğu açıktır [4–8]. Veri setlerinin temel sorunu doğal dille ve uzmanların anlayacağı yapılar olarak oluşturulmuş olmalarıdır. Makine öğrenimi algoritmalarında doğrudan kullanımları uygun değildir.

1.2. Tezin Amacı ve Organizasyonu

Tezin amacı, yazılım güvenlik raporlarının hızlı ve doğru bir şekilde analiz edilebilmesi için farklı veri madenciliği ve makine öğrenmesi tekniklerinin kullanılabileceğini göstermektir. Daha açık bir ifadeyle, güvenlik açığı raporlarının uzmanlar tarafından oluşturulan güvenlik vektörlerini, metin madenciliği yöntemleri ve çok sınıflı sınıflandırma algoritmaları kullanarak en doğru şekilde tahmin eden bir model geliştirmektir. Güvenlik açığı vektörleri çok önemli bilgiler sağlamaktadır.

Ayrıca güvenlik açığı skorlarının hesaplanmasında da doğrudan kullanılmaktadırlar. Bu bilgiler alanın uzmanları ve sektör temsilcileri açısından büyük önem arz etmektedir. Ayrıca güvenlik açıklarının önceliklendirilmesi de kaynakların doğru planlanmasına yardımcı olacaktır. Önerilen yöntem raporların sıfırıncı gün (zero-day) ataklarda dâhil henüz güvenlik puanı atanmamış raporlarının değerlendirmesinde de temel teşkil edecektir. Ayrıca çalışmamızda, şuana kadar gerçekleştirilmiş benzer çalışmalarda kullanılan en geniş veri seti kullanılmıştır. Bu da özellikle geçmiş tecrübeleri kullanan yeni güvenlik açıklarının analizinde faydalı olacaktır. Ayrıca çalışmamızın en önemli yeniliklerinden biri skorlama sistemlerinin en yeni versiyonu olan CVSS 3.1 ile yapılan ilk çalışmalardan biri olmasıdır. Dahası farklı istatistiki yöntemleri ve derin öğrenme tabanlı kelime gömme yaklaşımlarını bir arada kullanan ender çalışmalardan biridir. Ayrıca, farklı veri tabanlarındaki verileri inceleyerek her veri tabanının farklılıklarını belirten ve alanın uzmanlarına çalışmalarında rehberlik edecek şekilde yapılandırılmıştır.

Tez çalışması altı bölümden oluşmaktadır. Birinci bölümde, tezin önemi, amacı ve güvenlik açığı analizi ve sınıflandırması çalışmalarının yer aldığı literatür ayrıntılı bir şekilde sunulmuştur. İkinci bölümde yazılım güvenlik açığı kavramı açıklanmış ve veri tabanları sistematik olarak karşılaştırılmıştır. Üçüncü bölümde yazılım güvenlik açıklamalarının sınıflandırmasında ve puanlamasında kullanılan sistemler açıklanmıştır. Dördüncü bölümde önerilen yöntem tüm ayrıntıları ile açıkça ele alınarak sunulmuştur. Beşinci bölümde deneysel çalışmalardan elde edilen bulgular verilmiştir. Altıncı bölüm sonuçlar ve önerilerden oluşmaktadır.

1.3. Tezin Literatüre Katkıları

Bu tez çalışmasının literatüre katkıları aşağıdaki gibi özetlenebilir.

- Farklı araştırma grupları tarafından oluşturulan veri tabanlarının avantaj ve dezavantajlarının ortaya çıkarılması,
- Araştırmacıların hangi veri tabanını kullanacaklarına karar vermelerine yardımcı olacak bir rehber olması,
- Literatürdeki en büyük boyuta sahip veri setinin oluşturulması, kullanılması ve literatürde ifade edilen farklı veri setlerini kullanarak karşılaştırılabilir başarılı sonuçlar üretilmiş olması.
- Yazılım güvenlik açıklarının sınıflandırılması için yayınlanan resmi skorlama sisteminin son versiyonu üzerindeki ilk çalışmalardan biri olması,
- Kullandığı farklı metin madenciliği ve çok sınıflı sınıflandırma algoritmaları ile alanında en geniş kapsamlı ve orijinal bir çalışma olması.

Ayrıca bu tez çalışması kapsamında, bir adet Türkiye Bilimsel ve Teknolojik Araştırma Kurumu projesi (TÜBİTAK - 1002) gerçekleştirilmiş, bir adet uluslararası özet bildiri, bir adet TR

dizin dergi makalesi, üç adet uluslararası alan indeksine sahip dergi makalesi, bir adet Scopus indeksli makale ve iki adet SCI-E makale yayımlanmıştır. İlgili çalışmalar hakkında bilgiler özgeçmiş bölümünde yer almaktadır.

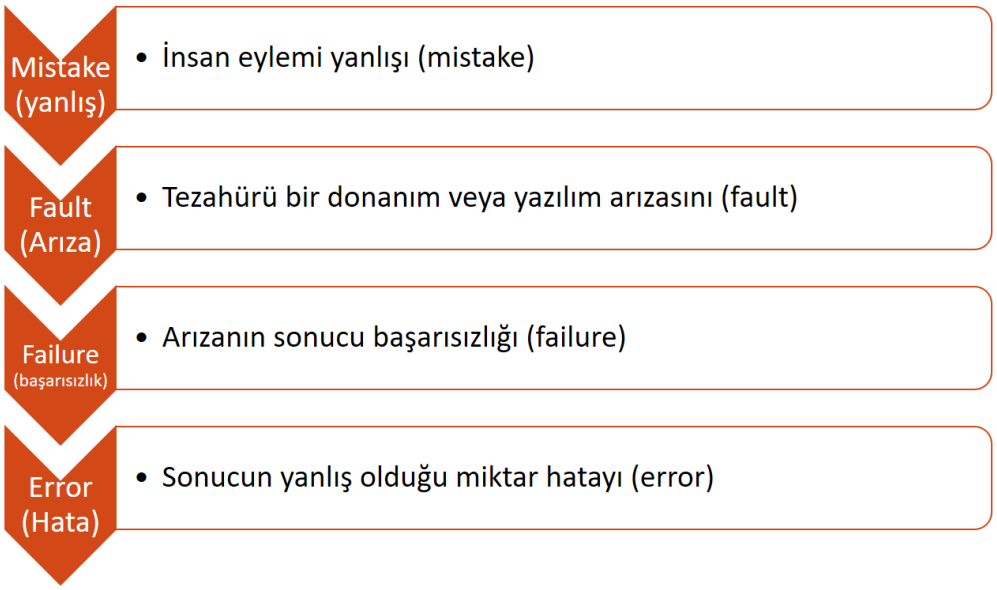
2. YAZILIM GÜVENLİK AÇIĞI

Yazılım güvenlik açıklarının belirlenmesi ve sınıflandırılması, yazılım geliştirme sürecinde doğru karar verme noktasında geliştiricilere yardımcı olacaktır. Bu sebeple yazılımlarda tespit edilen açıklar ve zafiyetler uzun zamandır veri tabanlarına kaydedilmektedir. Farklı araştırma grupları tarafından pek çok veri tabanı oluşturulmuştur. Bu çeşitlilik her veri tabanına kendine özgü avantajlar sağlamasının yanında bir takım dezavantajlar da ortaya çıkarmıştır. Çalışmamızın bu bölümünde farklı veri tabanlarının oluşturulmasının arkasında yatan nedenlere odaklanılmıştır. Ayrıca avantaj ve dezavantajları açıkça ortaya konulmuştur. İlk olarak Yazılım güvenlik açıkları alanında yapılan akademik çalışmalar incelenerek kullanılan veri tabanları tespit edilmiştir. Literatürde kullanılan on iki farklı veri tabanı tespit edilmiştir. Ancak bunlar arasında güncel ve araştırmacıların erişimine açık olanlar seçilmiştir. Bu eleme işlemi sonucunda yedi farklı veri tabanı bu çalışma kapsamına alınmıştır. Belirlenen veri tabanları ayrıntılı bir şekilde incelenmiştir ve açıklanmıştır. Daha sonra belirli ölçütlere göre veri tabanları karşılaştırılmıştır. Karşılaştırma sonucu elde edilen veriler ayrıntılı bir şekilde sunulmuştur. Araştırmacıların hangi veri tabanını kullanacaklarına karar vermelerine yardımcı olmak amacıyla literatürde yaygın kullanım alanı bulan güncel ve erişime açık olan güvenlik açığı veri tabanları sistematik olarak incelenmiştir.

2.1. Yazılım Güvenlik Açığı Kavramı

Güvenlik açığı ifadesini birçok araştırmacı tanımlamıştır. Ancak bu terimin standart olarak kabul edilmiş bir tanımı yoktur. Bu nedenle güvenlik açığı kavramına giren durumları kesin bir sınırla ayırmak, zorluk olmaya devam etmektedir [14]. Yazılım güvenlik açığı kavramı tanımlanırken IEEE Standart Yazılım Mühendisliği Terminolojisi Sözlüğü temel alındığında Krsul ve Ozment'in tanımlarının kabul edildiği görülmektedir [18]. Bu tanımlar, Krsul tarafından "Yazılımın tanımlanmasında, geliştirilmesinde veya yapılandırılmasında, yazılımın çalıştırılmasında güvenlik politikasını ihlal edebilecek bir hata örneğidir" [20] ve Ozment tarafından "Bir yazılım güvenlik açığı, uygulamanın örtük veya açık güvenlik politikasını ihlal edebilmesi için yazılımın teknik özelliklerinde, geliştirilmesinde veya yapılandırılmasında yapılan yanlışın bir örneğidir" [21] şeklinde ifade edilmiştir. Bu iki tanımdaki temel fark hata kelimesinin yanlış olarak değiştirmiş olmasıdır. IEEE Standart Yazılım Mühendisliği Terminolojisi Sözlüğünde bu tanımlar benimsenmiştir [22].

IEEE Yazılım Mühendisliği Terminolojisi Sözlüğüne bakıldığında Şekil 2.1'deki dört anahtar terimin önemli olduğu anlaşılmaktadır. Bu terimlerin ilişkisinin bir özeti olarak; insan eylemi yanlışı (mistake), tezahürü bir donanım veya yazılım arızasını (fault), arızanın sonucu başarısızlığı (failure) ve sonucun yanlış olduğu miktar hatayı (error) ifade etmektedir [22].



Şekil 0.1. Yazılım güvenlik açıklarının tanımlanmasında kulllanılan terimler ve ilişkieri

Bu tanımlamalardan yola çıkarak bir yazılım güvenlik açığı tanımında kullanılacak uygun anahtar terimin arıza (fault) (ayrıca kusur veya bug) olabileceği belirtilmektedir [5]. Genel kabul gören tanıma göre yazılım güvenlik açığı şu şekilde tanımlanmaktadır; "Bir yazılım güvenlik açığı, bazı açık veya örtülü güvenlik politikasını ihlal etmek için kullanılabilecek şekilde yazılımın tasarımında, geliştirilmesinde veya yapılandırılmasındaki bir hatanın neden olduğu bir kusur örneğidir." [5]. Bu açıdan ele alındığında Şekil 2.1, güvenlik açığı terimlerini anlamlandırmak için yararlı bir bakış açısı sunar.

Bu tanımlama dikkate alındığında, herhangi bir yazılım bileşeninin kusur (bug) içermemesinin mümkün olamayacağı anlaşılmaktadır. Dolayısıyla yazılım ürünlerinin güvenlik protokollerini ihlal eden kusurlar tespit edildiğinde bunların raporlanması ve ilan edilmesi rutin bir işlem halini almıştır. Bu raporların yer aldığı temel veri tabanı olarak Ortak Güvenlik Açıkları ve Etkilenmeleri (Common Vulnerabilities and Exposures - CVE) ve bunlar üzerinden geliştirilen diğer veri tabanlarının incelenmesi araştırmacılar için önemlidir.

2.2. Yazılım Güvenlik Açığı Veri Tabanları

Güvenlik açıkları bilişim teknolojileri ile donatılmış günümüz toplumunda önemli bir risk oluşturmaktadır. Bu riskin önemi son dönemde fark edilmeye başlanmıştır [23]. Bu nedenle konunun paydaşları arasında etkili bir bilgi paylaşımı ve koordinasyon şarttır. Alınacak önleyici tedbirler sayesinde önemli sorunların önüne geçilebilir. Yazılım güvenlik açıklarının yönetimi de bu nedenle çok önemlidir [24]. Hangi verilerin toplandığı ve nasıl raporlandığını bilmek önemlidir.

Herhangi bir yazılımda bir kusurun güvenlik politikalarını ihlal ettiğini tespit eden bireyler, şirketler veya diğer kurumlar bu durumu bildirmek için bir güvenlik raporu doldurarak durumu bildiriler. Bu bildirim, herhangi bir güvenlik açığı veri tabanı sağlayıcısı üzerinden yapılabilir. Ancak bir güvenlik açığı tespit edildiği zaman resmi yollar ile ilan dilmesi için uluslararası bir standart olan ve ABD İç Güvenlik Bakanlığı tarafından finanse edilen bir prosedür uygulanmaktadır. Bu işlem için yetkilendirilmiş kurum kar amacı gütmeyen MITRE şirketidir [25]. Bu organizasyona üye olan pek çok ülkeye bağlı bilgisayar acil müdahale ekipleri (Computer Emergency Response Team – CERT) tarafından CVE veri tabanına tespit edilen güvenlik açıkları kaydedilerek resmi süreç başlatılmış olmaktadır.

2.2.1. CVE - Ortak Güvenlik Açıkları ve Etkilenmeleri Sözlüğü

Tespit edilen güvenlik açıklarına uluslararası bir standart getirmek üzere MITRE tarafından bir araya getirilen büyük güvenlik organizasyonları ile birlikte 1999 yılında kurulmuştur. Genel olarak bilinen siber güvenlik açıkları için bir tanımlayıcı listedir. CVE girişlerinin kullanımı yazılımların güvenliği hakkında uluslararası güvenirliği olan benzersiz bir güven sağlamaktadır [26]. CVE girişleri birçok deneysel makale, Adobe, Apple, IBM veya Microsoft gibi çok sayıda bilgi güvenliği ürün ve hizmet satıcısı tarafından kullanılmaktadır [27,28].

Tablo 0.1. CVE veri tabanı özellikleri

Özellik	Açıklaması
CVE-ID	CVE tarafından atanan benzersiz ID
Description	Güvenlik açığına dair teknik uzman görüşü
References	Güvenlik açığı ile ilgili bilgilerin yer aldığı dış bağlantılar
Assigning CNA	Bildirimi yapan yetkili veya Yazar bilgisi
Date Entry Created	Girişin oluşturulduğu tarih

CVE temelde, bir güvenlik açığı için tanımlayıcı ve standartlaştırılmış bir açıklama sunmasının yanında endüstri tarafından onaylanmış herkese açık ve ücretsiz veri sağlamayı amaç edinmiş bir güvenlik açığı listesidir. Temel misyonu farklı veri tabanları ile araçlar için aynı standartları oluşturmak, birlikte çalışabilirlik ve bilişim ekosisteminin güvenlik kapsamını iyileştirmektir. Bir veri tabanından çok bir sözlük olarak kendisini tanımlamaktadır. Bilinen tüm büyük veri tabanları temelde CVE'de yayınlanan listeleri baz alarak oluşturulmaktadır [26]. Standart ve güvenilir bilgi sağlamasının yanında listelerinin ham bilgiler barındırması ve ek bilgiler içermemesi önemli bir eksiklik olarak görülmektedir. CVE veri tabanı güvenlik raporunda bulunan özellikler ve açıklamaları Tablo 2.1'de sunulmuştur.

2.2.2. NVD - Ulusal Güvenlik Açığı Veri Tabanı

ABD Ulusal Standartlar ve Teknoloji Enstitüsüne (National Institute of Standards and Technology - NIST) bağlı olarak 2000 yılında oluşturulmuştur. Güvenlik açığı verilerinin yönetimini, skorlamasını ve uyumluluğunu içeren bir veri tabanıdır. NVD, güvenlik kontrol listesi referansları, güvenlikle ilgili yazılım kusurları, yanlış yapılandırmaları, ürün adları ve etki metrikleri bilgilerini içermektedir. ABD Ulusal Güvenlik Bakanlığı'nın Ulusal Siber Güvenlik Bölümü tarafından desteklenmektedir [12].

NVD çalışanlarının temel görevi CVE sözlüğünde yayınlanan güvenlik açığı listeleri üzerinde analizler yapmaktır. Bu aşamada CVE'de bulunan açıklamaları, referansları toplayabildikleri tüm ek verileri kullanmaktadırlar. NVD veri tabanında yayınlanan veriler için temel olarak, ilişkili etki metrikleri (CVSS), güvenlik açığı türleri (CWE) ve uygulanabilirlik ifadeleri (CPE) ve diğer ilgili meta veriler eklenmektedir. Ancak NVD atadığı öznitelikler için güvenlik açığı testi yapmamaktadır. Yeni bilgilere göre verilerin CVSS puanları ve uygulanabilirlik ifadeleri değişebilmektedir [12]. Veri tabanında bulunan verilerdeki açıklamaların açığı yeterince ifade edemediği eleştirisi alanın uzmanları tarafından ifade edilmektedir [18]. NVD veri tabanında bulunan özellikler ve açıklamaları Tablo 2.2'de sunulmuştur.

Tablo 0.2. NVD veri tabanı özellikleri

Özellik	Açıklaması
CVE-ID	CVE tarafından atanan benzersiz ID
Current Description	Güvenlik açığı mevcut açıklaması
Analysis Description	Güvenlik açığı analiz sonrası açıklaması
References to Advisories, Solutions, Tools	Önerilen çözüm yöntemleri, dış bağlantılar ve araçlar
Severity	CVSS V.3.X CVSS V2.0 puanları ve güvenlik açık vektörleri.
Weakness Enumeration	Açık kategorisi, numarası ve kaynağı. (CWE-ID - CWE Name, Source)
Known Affected Software Configurations	Etkilendiği bilinen yazılım ve versiyonları
Change History	Açığa dair önemli işlemlerin tarihsel geçmişi

2.2.3. Exploit-DB

Offensive Security topluluğu tarafından 2004 yılında kamu hizmeti ve kar amacı gütmeyen bir proje olarak doğmuştur. CVE sözlüğünde yayınlanan listeler ile uyumludur. Temel amacı, en kapsamlı istismar arşivine hizmet etmek ve bunları serbestçe erişilebilen ve gezinmesi kolay bir veri tabanında sunmaktır. Exploit veri tabanı, tanımlardan ziyade CVE listelerinde yayınlanan açıkların istismar edilebilirliğini gösteren PoC kodları (Proof of Concept Code) ve kavram kanıtları sağlamaktadır. PoC, bir saldırganın güvenlik açığını nasıl istismar edebileceğini açıklayan basit bir

kod parçasıdır. Bu özelliği veri tabanını hemen eyleme geçirilebilir verilere ihtiyaç duyanlar için değerli bir kaynak haline getirmektedir. Ancak PoC kodu bulunmayan veriler ihmal edilmektedir [29]. Exploit-DB veri tabanında bulunan özellikler ve açıklamaları Tablo 2.3'te sunulmuştur.

Tablo 0.3. Exploit-DB veri tabanı özellikleri

Özellik	Açıklaması
CVE-ID	CVE tarafından atanan benzersiz ID
Exploit Title	Etkilen yazılım ismi ve açık türü
Exploit Author	Bildirimi yapan yetkili veya yazar bilgisi
Date	Girişin oluşturulduğu tarih
Version	Etkilendiği bilinen yazılım ve versiyonları
Poc Code	İstismar Kodu
Vendor Homepage	Güvenlik açığı ile ilgili bilgilerin yer aldığı dış bağlantılar
Tested On	Açığın test edildiği işletim sistemi
Author Contact	Bildirimi yapan yetkili veya yazar iletişim bilgisi

2.2.4. Rapid7

Birleşik güvenlik yönetimi çözümleri sağlayan bir güvenlik şirketi olan Rapid7, 2000 yılında kurulmuştur. Güvenlik uzmanları ve araştırmacıların incelemesi için güvenlik açığı ve istismar için teknik ayrıntılar içeren bir veri tabanıdır. CVE listeleri ile uyumludur. Veri tabanında yayınlanan tüm istismar kodları Metasploit (ticari sızma testi çerçevesi) çerçevesine dâhil edilmiştir. Güvenlik açığı ve istismarları sık sık güncellenir. Veri tabanı en son güvenlik araştırmalarını içerir. Kamu politikası olarak, tüketicilere fayda sağlayan ve sorumlu siber güvenlik uygulayıcılarını savunan politikaları, standartları ve mevzuatı şekillendirmek için hükümetler, şirketler, kar amacı gütmeyen kuruluşlar ve uzmanlarla birlikte çalışmayı benimsemiştir [30]. Rapid7 veri tabanında bulunan özellikler ve anlamları Tablo 2.4'te sunulmuştur.

Tablo 0.4. Rapid7 veri tabanı özellikleri

Özellik	Açıklaması
CVE-ID	CVE tarafından atanan benzersiz ID
Title	Etkilen yazılım ismi ve açık türü
Description	Güvenlik açığına dair teknik uzman görüşü
References	Güvenlik açığı ile ilgili bilgilerin yer aldığı dış bağlantılar
Solution(s)	Açık için çözüm önerileri
Severity	CVSS V2.0 puanları ve güvenlik açığı vektörü
Dates	Yayınlanma, Oluşturulma, Eklenme ve Değiştirilme tarihleri

2.2.5. SecurityFocus

Bağımsız güvenlik uzmanları tarafından oluşturulan bir topluluk tarafından 1999'da kurulmuştur. CVE listelerini temel alan SecurityFocus Güvenlik Açığı Veri Tabanı, güvenlik profesyonellerine tüm platformlar ve hizmetler için güvenlik açıkları hakkında en güncel bilgileri sağlamayı amaçlamaktadır. Bunun için haber bültenleri, teknik makaleler ve yazılar yayınlamaktadır. Posta listeleri sayesinde dünyanın her yerindeki üyeleri ile güvenlik sorunlarını tartışmaya olanak sağlamaktadır [31]. SecurityFocus en önemli ve en saygın güvenlik açığı veri tabanlarından biridir. NVD veri tabanındaki tanımlamalara göre SecurityFocus listelerindeki tanımlamalar güvenlik açığının etkisini ve sömürülebilirliğini daha spesifik olarak açıklamaktadır [18]. SecurityFocus veri tabanında bulunan özellikler ve açıklamaları Tablo 2.5'te sunulmuştur.

Tablo 0.5. SecurityFocus veri tabanı özellikleri

Özellik	Açıklaması
CVE-ID	CVE tarafından atanan ID
BUGTRAQ ID	SecurityFocus tarafından tanımlanan ID
Info	Güvenlik açığı genel bilgiler
Discussion	Güvenlik açığına dair detaylı bilgi
References	Güvenlik açığı ile ilgili bilgilerin yer aldığı dış bağlantılar
Solution	Açık için çözüm önerileri
Exploit	İstismar Kodu
Dates	Yayınlanma ve Güncelleme tarihleri
Credit	Bildirimi yapan yetkili veya yazar bilgisi
Vulnerable	Etkilendiği bilinen yazılım ve versiyonları
Class	Açık kategorisi ismi

2.2.6. Snyk

Snyk veri tabanı, açık kaynak projeleri için ücretsiz olan kod değerlendirme araçları sağlayan kar amaçlı bir şirket tarafından oluşturulmuştur. Açık kaynak projelerinin geliştirilmesini desteklemek ve güvende kalmalarını sağlamaya yardımcı olmayı misyon edinmiştir. Snyk, 800.000'den fazla açık kaynak paketindeki güvenlik açıklarını izleyerek 25.000'den fazla uygulamanın korunmasına yardımcı olmaktadır. Snyk kullanıcılarının %83'ü uygulamalarında güvenlik açıkları bulduğunu belirtmiştir. Yeni güvenlik açıkları düzenli olarak ifşa edilmektedir. Snyk veri tabanı dört temel ilke üzerine yapılandırılmıştır. Bunlar açığı bul, düzelt, önle ve sürekli izledir [32]. Snyk veri tabanında bulunan özellikler ve açıklamaları Tablo 2.6'da sunulmuştur.

Tablo 0.6. Snyk veri tabanı özellikleri

Özellik	Açıklaması
CVE-ID	CVE tarafından atanan ID
SNYK ID	Snyk firması tarafından tanımlanan ID
Title	Etkilen yazılım ismi ve açık türü
Overview	Güvenlik açığı özeti
Details	Güvenlik açığına dair detaylı bilgi
References	Güvenlik açığı ile ilgili bilgilerin yer aldığı dış bağlantılar
Remediation	Açık için çözüm önerileri
Severity	CVSS V3.1 puanları ve güvenlik açığı vektörü
CWE	Açık kategorisi numarası
Dates	İfşa ve yayınlanma tarihleri
Credit	Bildirimi yapan yetkili veya yazar bilgisi

2.2.7. SARD – Yazılım Güvencesi Referans Veri Kümesi

Koleksiyon 2005 yılında NIST tarafından oluşturulmaya başlanmıştır. İlk duyurulduğunda Standart Referans Veri Kümesi (Standard Reference Dataset - SRD) olarak adlandırılmıştır. Bu ad 2014 yılında Yazılım Güvencesi Referans Veri Kümesi (Software Assurance Reference Dataset - SARD) olarak değiştirilmiştir. Bir dizi bilinen güvenlik açıklarını sağlayarak kullanıcıların, araştırmacıların ve yazılım geliştiricilerin güvenlik araçları geliştirmelerine yardımcı olmayı amaçlamaktadır.

Tablo 0.7. SARD veri tabanı özellikleri

Özellik	Açıklaması
Test Case ID(up)	SARD tarafından tanımlanan ID
Description	Güvenlik açığına dair detaylı bilgi
Language	Desteklenen Programlama Dili
Type of Artifact	Test kodunun oluşturulma Yöntemi
Status	Durum Bilgisi
Weakness CWE	Açık kategorisi numarası
Submission Date	Yayınlanma tarihi

Ayrıca test senaryoları tasarımları, kaynak kodları, ikili dosyalar gibi verileri de sağlayarak yazılım yaşam döngüsünün tüm aşamalarını barındıran bir arşiv sunmaktadır. Bu durum son kullanıcıların geliştirdikleri araçları ve araç geliştirme yöntemlerini test etmelerini ve değerlendirmelerini sağlamaktadır. Veri kümesi, "gerçek" (üretim), "suni" (test etmek için yazılmış) ve "akademik" (öğrencilerden) test senaryolarını içermektedir. Bu veri tabanı aynı

zamanda bilinen hatalara ve güvenlik açıklarına sahip gerçek bir yazılım uygulaması içermektedir. Veri kümesi, çok çeşitli olası güvenlik açıklarını, dilleri, platformları ve derleyicileri kapsamaktadır. Veri kümesi, birçok katılımcıdan test senaryoları toplayarak büyümektedir [33]. SARD veri tabanında bulunan özellikler ve açıklamaları Tablo 2.7'de sunulmuştur.

2.3. Veri Tabanlarının Karşılaştırılması

Güvenlik açığı veri tabanları genellikle benzer özelliklere sahiptir. Tespit ettiğimiz özellikler incelendiğinde temel olarak numaralandırma, açıklama, yazar bilgileri, referanslar, önem puanı, çözüm yöntemleri, istismar kodları, açık kategorisi ve tarih bilgileri gibi ortak özellikler görülmektedir. Ancak Tablo 2.8'de ayrıntıları görülebileceği gibi her veri tabanı sağlayıcısı bu özelliklerin yanında kendine özgü farklı özellikler sağlamaktadır. Ayrıca özelliklerin içerdiği değerlerin önemli farklılıklar arz ettiği görülmektedir.

Yukarıda tanımlamaları, avantaj ve dezavantajları ortaya konulan veri tabanları sistematik bir kıyaslama yöntemiyle karşılaştırılmıştır. Burada incelenen veri tabanlarının güvenlik açığı skorlaması olup/olmadığı, güvenlik açığı için çözüm yöntemi içerip/içermediği, istismar kodlarının bulunup/bulunmadığı, güvenlik açıkları için test yapılıp/yapılmadığı, kimlerin raporladığı, referans bilgilerinin varlığı, yazar bilgilerinin verilip/verilmediği, açığın kategorisi hakkında bilgi içerip/içermediği, veri besleme teknolojilerini destekleyip/desteklemedikleri, iş modeli ve veri boyutu kriterlerine göre değerlendirilmiştir. Tablo 2.9'da seçilen veri tabanları ve bunların belirlenen değerlendirme kriterlerine göre karşılaştırmaları sunulmuştur.

Bir güvenlik raporu yayınlanırken bir ID ataması gerçekleştirilmektedir. Genellikle CVE tarafından verilen CVE-ID değeri kullanılır. Ancak bazı veri tabanları CVE-ID ile birlikte kendi ID değerlerini vermektedir. SARD veri tabanı ise sadece kendi TEST CASE ID değerini kullanmayı tercih etmektedir. Bunun nedeni SARD veri tabanının diğer sağlayıcılardan tamamen farklı bir içerik listesine sahip olmasıdır. Ayrıca SecurityFocus ve Snyk veri tabanları CVE-ID değerinin yanında kendi numaralandırma sistemlerini de birlikte kullanmaktadır. Bunlar; SecurityFocus veri tabanında Bugtrag-ID iken Snyk için Snyk-ID değerleridir.

Veri tabanlarının tamamında aynı olan diğer bir özellik ise raporlarında açıklama ve tarih bilgilerinin yer almasıdır. Bu durumun tek istisnası olarak Exploit-DB'nin raporlarının bazılarında açıklama bilgisinin yer almamasıdır. Ayrıca bu bilgilerin içeriği her veri tabanına özgü olmakla birlikte farklı başlıklar ile ifade edilebilmektedir. Ayrıca SARD veri tabanları haricinde tüm veri tabanları raporlarında referans bilgilerini sunmaktadır. Ancak Exploit-DB veri tabanının sunduğu bazı bilgilerin içerisinde referanslara rastlanmamıştır.

Tablo 0.8. Güvenlik açığı veri tabanlarının güvenlik raporu bilgilerinin eşleşmesi

Veri Tabanı	Numaralandırma	Açıklama	Referans	Yazar	Tarihler	CVSS Versiyon	Çözüm	İstismar Kodu	Etkilenen Yazılım	CWE	Bağımsız Alanlar
CVE	CVE-ID	Description	References	Assigning CNA	Date Entry Created						Phase, Votes, Comments, Proposed
NVD	CVE-ID	Current Desc. Analysis Desc.	References to Advisories		Change History	CVSS 3.x, CVSS 2.0	Solutions	References to Advisories	Known Affected Software Configurations	Weakness Enumeration CWE-ID CWE Name	Tools
Exploit-DB	CVE-ID	Vuln. Details	Exploit Author, Vendor, Homepage, Author Contact		Date			Poc Code	Exploit Title, Version		Tested On
Rapid7	CVE-ID	Description	References		Published, Created, Added, Modified	CVSS 2.0	Solution		Title		
Snyk	CVE - ID SNYK - ID	Overview, Details	References	Credit	Disclosed, Published	CVSS 3.1	Remediation	Details	Title	CWE	
SecurityFocus	CVE-ID Bugtraq -ID	Info, Discussion	References	Credit	Dates		Solution	Exploit	Vulnerable	Class	Remote, Local, Not Vulnerable
SARD	Test Case-ID	Description			Sub. Date					Weakness CWE	TypeofArtifact, Language, Status

Güvenlik açıklarının etki değerini ifade etmek için güvenlik skorları kullanılmaktadır. Tablo 2.9 incelendiğinde bu skor bilgilerinin elde edilebildiği veri tabanlarının NVD, Rapid7 ve Snyk olduğu görülmektedir. Ancak resmi olarak kullanılan skorlama sürümlerinden CVSS 2.0 ve CVSS

3.x değerlerinin birlikte verildiği tek veri tabanı NVD olarak görülmektedir. Rapid7 veri tabanında sadece 2.0 sürümünün değerleri verilmektedir. Snyk veri tabanında ise sadece 3.1 versiyonun skor değerleri bulunmaktadır.

Tablo 0.9. Güvenlik açığı veri tabanlarının karşılaştırılması

Veri Tabanı	Numaralandırma	Açıklama	Referans	Yazar	CVSS Versiyon	Çözüm	İstismar Kodu	Test	CWE	Json / XML/RSS	Raporlama	İş Modeli	Veri Boyutu*
CVE	✓	✓	✓	(✓)	×	×	×	×	×	✓	Herkes	Kamu	177.042
NVD	✓	✓	✓	(✓)	2.0-3.X	✓	(✓)	×	✓	✓	Üyeler	Kamu	187.584
Exploit-DB	✓	(✓)	(✓)	✓	×	×	✓	✓	×	×	Herkes	Kamu	42.962
SecurityFocus	✓	✓	✓	✓	×	✓	(✓)	✓	✓	×	Herkes	Kamu	102.330
Rapid7	✓	✓	✓	×	2.0	✓	×	×	×	×	Çalışanlar	Ticari	171.816
Snyk	✓	✓	✓	✓	3.1	✓	(✓)	×	✓	×	Çalışanlar	Ticari	6.012
SARD	✓	✓	×	×	×	×	×	✓	✓	✓	Herkes	Kamu	177.184

✓: Var - ×: Yok - (✓): Bazı veriler için var bazıları için yok. * 31.05.2022 tarihi itibari ile.

Güvenlik açıklarını halka açık bir platformda sunmanın bazı riskleri vardır. Kötü amaçlı kullanıma yol açabilecek böyle bir durumun önüne geçilmesi için açık ile birlikte çözüm yönteminin de sunulması önemlidir. Güvenlik açıklarının nasıl çözümleneceğine dair bilgi ise SecurityFocus, NVD, Rapid7 ve Snyk veri tabanlarından elde edilebilmektedir. Güvenlik açığının istismar edilebilirliğinin ispatı olan PoC kodlarının düzenli olarak yer aldığı tek veri tabanı Exploit-DB'dir. Ancak her veri için olmasa da NVD, Snyk ve SecurityFocus veri tabanlarında da PoC kodları yer alabilmektedir.

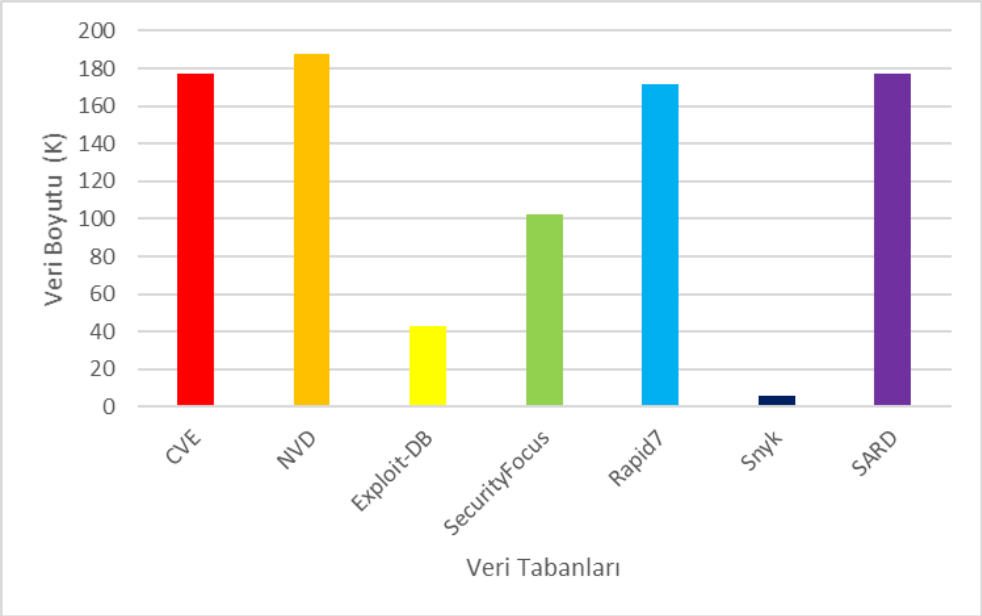
Bir açık tespit edildiğinde değerlendirmesi uzmanlar tarafından yapılmaktadır. Bu aşamada açığı doğrulayıcı testler yapılıp yapılmaması önemlidir. Rapid7 ve SARD veri tabanları haricindeki tüm veri tabanları bildirim yapan yazar bilgisini sunmaktadır. Ancak sadece Exploit-DB, SecurityFocus ve SARD veri tabanları veriler üzerinde test işlemi gerçekleştirmektedirler.

Güvenlik açıklarının sistemde raporlanması konusunda sadece üyelerinin raporlarını kabul eden NVD diğer veri tabanlarından ayrılmaktadır. Ayrıca ticari veri tabanları olan Rapid7 ve Snyk ise raporlama işlemlerini çalışanlarına yaptırmaktadır. Diğer veri tabanları ise bir güvenlik açığının raporlanmasında herkesin erişimine açıktır.

İş modeli kriterinden bakıldığında tüm veri tabanlarının kamu ve ticari olarak iki sınıfa ayrıldığı görülmektedir. Kamu iş modeli kar amacı gütmeden kamu yararına çalışmayı ifade ederken, ticari iş modelinin amacı sunduğu araçlar itibariyle kar elde etmek olsa da belli sayıda veriyi ücretsiz olarak araştırmacıların kullanımına sunmaktır. Rapid7 ve Snyk veri tabanları Ticari iş modelini benimsemektedir.

Veri boyutları dikkate alındığında Rapid7 yüksek boyutta veriyi ücretsiz sağlarken, Snyk ise en az veriyi sağlamaktadır. Şekil 2.2'de veri tabanlarının içerdikleri veri boyutları görülmektedir. Temelde CVE listelerini baz alan veri tabanlarının bu verileri yorumlama ve değerlendirme metodolojilerinden kaynaklı farklı boyutlarda veri setleri sağladıkları görülmektedir.

Veri tabanları listelerinin indirilmesi noktasında JSON veri beslemeleri sağlayan CVE, NVD ve toplu indirme özelliği olan SARD haricinde diğer veri tabanlarının böyle bir hizmeti yoktur. Bu veri tabanlarından verilerin tamamını indirebilmek için düzenli ifadeler kullanarak web sitelerinin taranması gerekmektedir.

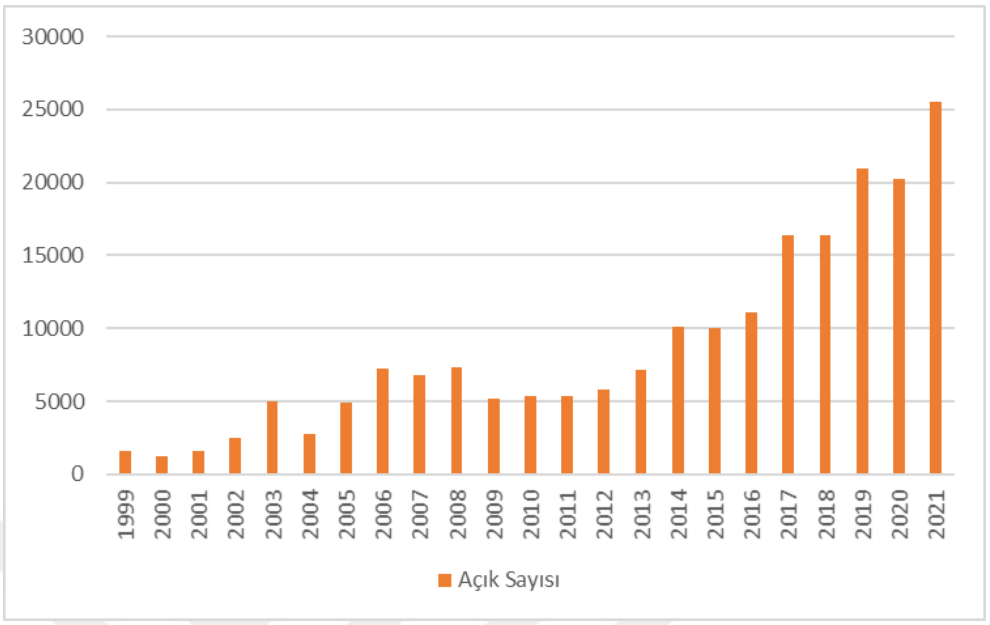


Şekil 0.2. Veri tabanlarının veri boyutlarına göre karşılaştırılması

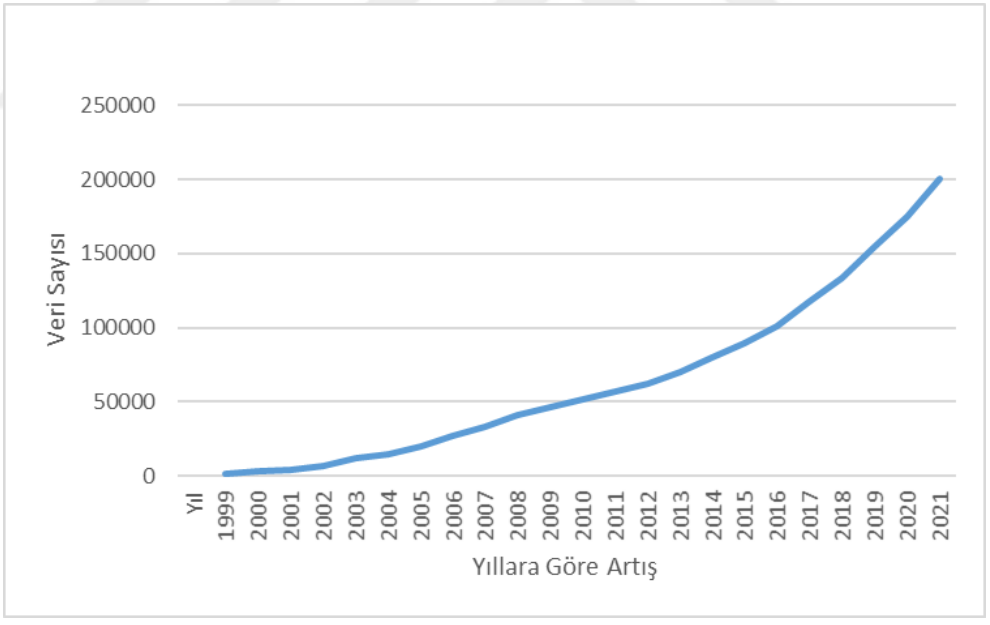
Güvenlik açığı listelerindeki artan verilerin işlenmesi aşaması uzmanlar tarafından manuel yapılan bir işlemdir [9]. Güvenlik açığı sayısı Şekil 2.3 ve 2.4'te açıkça görüleceği üzere hızla artmaktadır. NVD veri tabanı verileri incelendiğinde özellikle Şekil 2.4'te görüldüğü gibi güvenlik açığı bildirim artış hızı 2016'dan sonra yıllık yaklaşık %60 artış göstermektedir. Ayrıca Şekil 2.5'teki grafik incelendiğinde toplam açıkların sayısındaki artış trendinin devam edeceği anlaşılmaktadır.

Yayınlanan bir güvenlik açığının ilgili ürünün güvenlik politikasını ciddi bir şekilde ihlal edip etmediği hızlı bir şekilde belirlenmelidir. Ayrıca istismar amaçlı kullanılmadan önce derhal giderilmesi gerekmektedir. Son dönemde yapılan çalışmalar açıkların kötüye kullanılması ile ilgili son zamanlarda ortaya çıkan endişeleri tekrarlamaktadır [2]. Siber suçlar sonucu meydana gelecek maddi zararın çok yüksek meblağlara mal olacağı tahmin edilmektedir [1]. Bir güvenlik açığı bulunduğunda bu açık herkesin erişimine açık veri tabanlarında yayınlanmaktır. Ancak yayınlanan

bu raporlar doğal dille hazırlanmıştır ve makineler tarafından otomatik olarak yorumlanamazlar [13].



Şekil 0.3. NVD veri tabanı verilerinin yıllara göre dağılımı

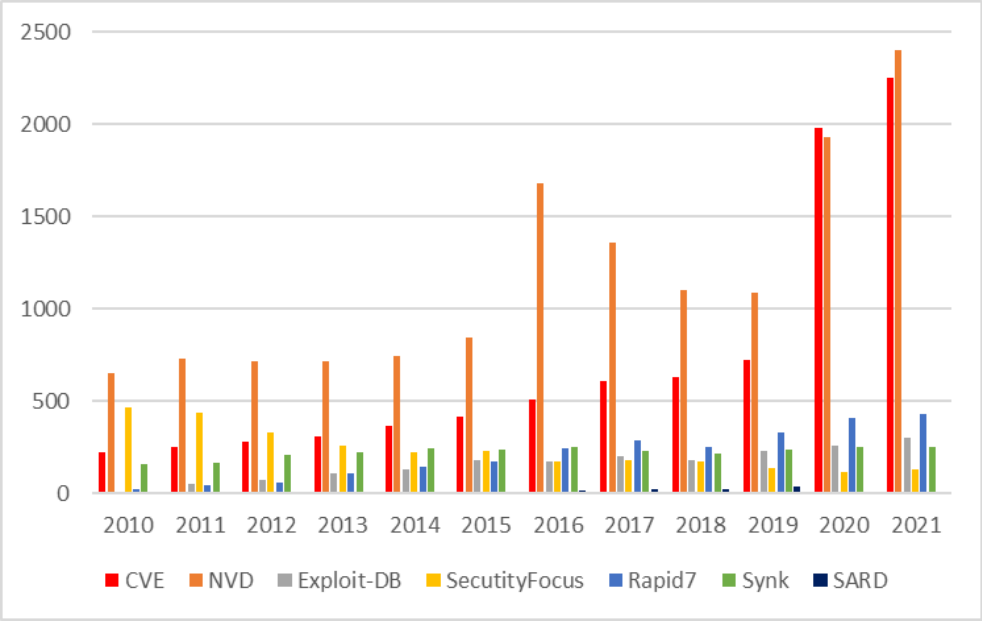


Şekil 0.4. NVD veri tabanı verilerinin yıllık artış grafiği

Yayınlanan güvenlik açığı sayısındaki durdurulamaz artış nedeni ile oluşan arşiv materyalindeki büyüklük uygulamalı araştırmalar için istatiksel önemini yitirmeye başladığı görülmektedir. Son yıllardaki artış oranı ile bu sorunun daha da artacağı anlamına gelmektedir [3]. Yıllardır biriken ve büyük bir yapılandırılmamış veri kümesi halini alan bu durum ancak yüksek hesaplama ve birçok problemde başarı ile uygulanmış makine öğrenmesi algoritmaları ile

çözülebilir. Bu alandaki eksiklikler yüzünden verilerin kapsamlı analizleri yapılamamakta ve farklı algoritmalar denenememektedir. Bu durum çoğunlukla keşfedilemeyen noktalar olduğu anlamına gelmektedir [4].

Şekil 2.5, tez çalışması kapsamında incelenen veri tabanlarının son on yıl içerisinde akademik çalışmalarda kullanımlarını göstermektedir. Dikkat edilirse 2016 yılına kadar kullanım yoğunluğu belirli bir oranda devam etmektedir. Ancak 2016 yılında itibaren çalışmalarda ciddi bir artış olmaktadır. Bu artışın nedeni makine öğrenmesi algoritmalarının güvenlik açığı problemlerinde kullanılmaya başlanması ile ilgili doğru orantılıdır [5].

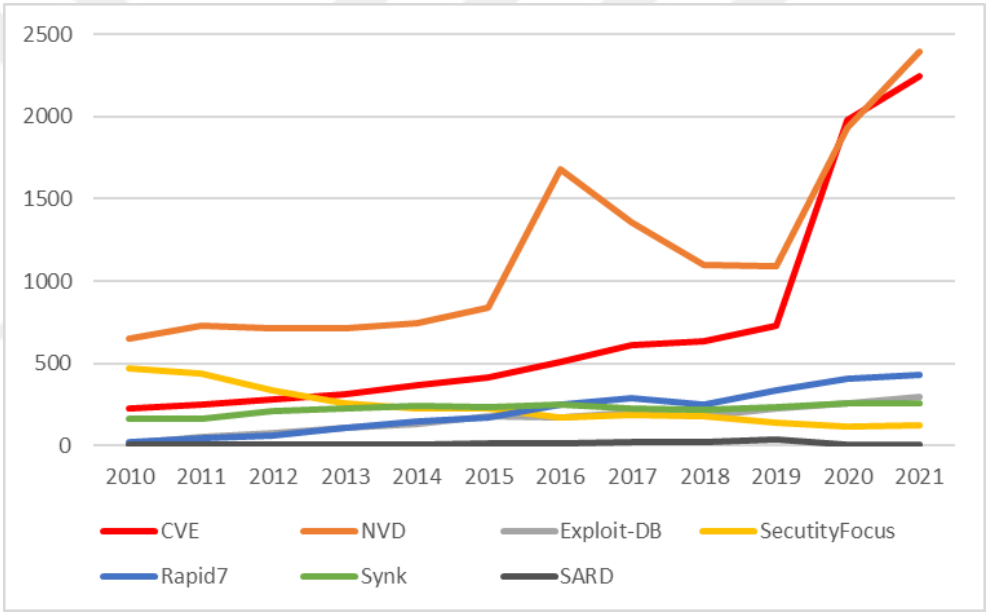


Şekil 0.5. Veri tabanlarının yıllara göre akademik çalışmalarda kullanımı

Ayrıca Şekil 2.6'da görüldüğü üzere bazı veri tabanlarında (rapid7, synk, securityfocus, sard) yapılan çalışmalar giderek azalma eğilimi göstermektedir. Bahsedilen durumun nedenini veri tabanlarının raporlama yapılarından kaynaklanmaktadır [3,4]. Şekil 2.6 incelendiğinde diğer önemli nokta ise çalışmalarda NVD ve CVE veri tabanlarının diğerlerinden bariz şekilde daha fazla kullanıldığıdır. Ancak kurumsal yapıları ve güvenlik açığı ekosistemine temel oluşturmalarına rağmen diğer veri tabanlarının da alternatif olarak çalışmalarda son dönemde kullanımları artmaktadır. Bu durum CVE ve NVD raporlarının içerdiği özelliklerin istenilen yeterlilikte olmamasından kaynaklanmaktadır [4,18]. Bu aşamada özellikle birikmiş ve sürekli gelmeye devam eden verilerin yeni bir yapıyla raporlanması ve arşivlenmesi ihtiyacının olduğunu göstermektedir [5,9].

Sonuç olarak CVE sözlüğü, ortak güvenilir bir standart ve alt yapı sağlamaktadır. Diğer veri tabanları CVE listelerini kullanarak kendi veri setlerini güncellemektedirler. NVD bu listelerdeki

ham verilere uzman görüşleri ile yeni özellikler ve açıklamalar eklemektedir. Ancak eklenen bu özellikler ve açıklamaların yeterliliği tartışma konusudur. Bu sorundan dolayı alanın uzmanlarından oluşan topluluklar bu listelere daha anlaşılır ve kullanışlı özellikler ekleyerek yeni veri setleri oluşturmuşlardır. SecurityFocus ve ExploitDB bunların en başında gelen veri tabanlarıdır. Bu veri tabanlarının NVD'den temel farkları daha anlaşılır bilgiler içeren açıklamaları ve istimrar kodlarını içeren yapılarıdır. Rapid7 ve Snyk gibi ticari veri tabanları da güvenlik açıklarının ticari değerlerini yasal yollarla değerlendirmektedirler. Ayrıca bu durum ticari güvenlik çerçevelerinin geliştirilmesini sağlayarak daha güvenli yazılım ürünlerinin çıkması için sektörü desteklemektedir. SARD veri tabanın sağladığı güvenlik test senaryoları bu veri tabanının en temel özelliğidir. Bu durum yazılım test mühendisliğinin gelişmesine önemli katkılar sunmaktadır. Alandaki her veri tabanı farklı bir özelliği ile öne çıkmaktadır.



Şekil 0.6. Veri tabanlarının yıllara göre kullanım frekansları

Veri tabanlarının mevcut yapılarında özellikle CVE ve NVD'nin çoğunlukla baz alınması avantajları yanında bazı dezavantajlarda getirmektedir. Kurumsal yapıları ve insan kaynağı kullanmaları nedeni ile güvenlik raporlarının yayınlanmasında gecikmeler olmakla birlikte önem puanlarının değerlerinin ilk hesaplamalarda olması gerekenden düşük tahmin edilmesi gibi durumlar yaşanmaktadır. İstismarların çoğunun açığın yayınlandığı ilk iki haftada meydana geldiği düşünülürse durumun önemi daha da anlaşılacaktır. Önem puanlarının makine öğrenmesi algoritmaları kullanılarak hesaplanması uzmanlara yol gösterici bir tahmin olacaktır. Ayrıca sistemdeki tüm veri tabanı sağlayıcılarının raporlarının veri besleme teknolojilerini ve ilişkisel veri tabanlarını kullanarak tasarlaması, araştırmacılar ile diğer veri tabanı sağlayıcılarının birlikte çalışma olanağını artırarak hata oranını azaltacaktır.

3. YAZILIM GÜVENLİK AÇIĞI SKORLAMA SİSTEMLERİ

Günümüz bilgi sistemleri büyük ölçüde yazılım sistemlerinin desteğiyle oluşturulmaktadır. Bu sistemlerde oluşabilecek zafiyetler istenmeyen kötü sonuçlar doğurabilir. Bunun önüne geçmek ve daha güvenli sistemler inşa edebilmek için yazılım zafiyetlerinin belirlenmesi ve standart bir ölçüm mekanizması ile sınıflandırılması önem arz etmektedir. Yazılım güvenlik açıklarının sıralanması ve puanlanması için bir sistem, güvenlik açığı satıcıları ve kar amacı gütmeyen kuruluşlar tarafından her zaman geliştirmek istenmiştir. Bu nedenle bilgi ve yazılım güvenliğini artıracak sistemler teşvik edilmiştir. Bu anlamda yapılan ilk çalışmalar olarak, Microsoft Tehdit Puanlama Sistemi, Symantec Tehdit Puanlama Sistemi, CERT Güvenlik Açığı Puanlaması ve SANS Kritik Güvenlik Açığı Analiz Ölçeği sayılabilir [11]. Sponas vd. [34] CVSS ile aynı güvenlik açığı şemasını kullanan Ağırlıklı Etki Güvenlik Açığı Puanlama Sistemi (Weighted Impact Vulnerability Scoring System –WIVSS) adını verdikleri önerilerini 2013'te tanıtmışlardır. WIVSS sisteminin temel farkı puan hesaplama esnasında daha önemli olduğunu düşündükleri güvenlik vektörlerini farklı ağırlık değerleri ile formüllerine katmalarıdır. Daha sonra optimize edilmiş ağırlıkları bulunan güncel WIVSS versiyonlarını 2015'te bildirmişlerdir [35].

Bu sistemlerin en belirgin eksiği, birbirinden bağımsız yapıları nedeni ile sistemler arasında uyum veya birlikte çalışabilirlik imkânı bulunmamasıdır. Ayrıca kapsamları açısından sınırlıdırlar. Bu sorunların çözümü ancak açık kaynak ve evrensel bir güvenlik açığı puanlama sistemi ile çözülebileceği anlayışı ile Ortak Güvenlik Açığı Puanlama Sistemi (Common Vulnerability Scoring System-CVSS) önerilmiştir. Önerilen sisteminin amacı güvenlik açıklarının ciddiyetini ve etkisini ortaya koymak için ortak bir dil oluşturmaktır [36].

NVD tarafından resmi puanlama sistemi olarak CVSS seçilmiştir [12]. Bunun sonucu olarak CVSS sistemi, bilişim teknolojileri güvenlik topluluğu tarafından hızla kabul görmüştür. Sistemin popülaritesi hızla artmış ve yaygın bir kabul ile bu alanda bir standart olmayı başarmıştır [9].

3.1. Ortak Güvenlik Açığı Puanlama Sistemi - CVSS

CVSS'in rakiplerinden temel farkı kişiselleştirmeye izin vermesi ve sunduğu açık çerçevesi sayesinde sıralama için kullanılabilecek tutarlı bir ortam oluşturmasıdır [11]. CVSS bu yapısı sayesinde geliştirilmeye açıktır. İlk versiyonunun tanıtıldığı yıl olan 2004 den bir yıl sonra CVSS 1.0 versiyonu resmen ilan edilmiştir [36]. İlan edilen versiyon düzenli iyileştirmeler ile güncellenmiştir. Bu güncelleme sonucu CVSS 2.0 versiyonu 2007 yılında tanıtılmıştır [37]. Uzun yıllar CVSS 2.0 versiyonu yaygın bir şekilde benimsenmiş ve kullanımda kalmıştır. Ancak tespit edilen güvenlik açıklarının sayısındaki artış ve güvenlik algısındaki değişiklikler yeni güncellemeleri zorunlu kılmıştır. Bunun sonucunda 2016 yılında CVSS 3.0 versiyonu

duyurulmuştur [38]. Günümüzde kullanımdaki CVSS 3.1 versiyonu ise 2019'da yayınlamıştır [6]. Günümüzde CVSS sürümlerinden CVSS 2.0 ve 3.1 birlikte kullanılmaktadır. CVSS 1.0 ve 3.0 versiyonları artık kullanılmamaktadır. Bundan dolayı tez çalışmamıza kullanımda olan versiyonlar dahil edilmiştir.

CVSS yayınlandığı günden itibaren genel kabul görmüş olmasının nedenleri vardır. Standartlaştırılmış bir güvenlik açığı puanlama sistemi sağlamaktadır. Bu sayede güvenlik politikaları oluşturulurken hangi açıklara öncelik verileceği belirlenebilir ve böylece bu açıklara yönelik acil ve hızlı düzeltmeler sağlanabilir. Karışıklığa neden olan hususları kaldırarak, hangi özelliklere göre puanlama yapıldığını açıkça gösteren bir çerçeve sunmaktadır. Dahası gerçek riskleri ortaya koyarak bir güvenlik açığının diğer açıklara göre ne derece öncelikli risk oluşturulduğunu göstermektedir [37].

Yazılım güvenlik açıklarının skorlarını belirlemek için önerilen bu sistemin iki versiyonu öne çıkmaktadır. Bunlar CVSS 2.0 ve CVSS 3.1'dir. Genellikle bu iki sistem ile hesaplama yapılmaktadır [9]. Güvenlik açıklarının hangisinin en yüksek önceliğe sahip olduğuna karar vermek amacıyla, güvenlik açığı değerlendirme faaliyetleri önemlidir. Bu işlem uzmanlar tarafından manüel olarak yapılmaktadır. Bu durum zaman alıcı ve kesin değildir. Ayrıca güvenlik açıklarının uzman kişiler tarafından manuel tespit edilerek sınıflandırılması maliyetlidir ve insanın doğasından kaynaklanan zaaflar barındırmaktadır. Güvenlik açığı skorlama sistemleri, güvenlik açığını sınıflandırmak için geliştirilmiş farklı güvenlik metriklerinden oluşan bir hesaplama sistemi kullanmaktadırlar. Bu metriklerden oluşan tanımlamalara güvenlik vektörü denilmektedir.

Yukarıda bahsedildiği gibi, puanlama sistemleri farklılık ve benzerlikler barındırmaktadır. Bunun sonucu olarak farklı özellikleri ile ön plana çıkan puanlama yöntemleri önerilmiştir. NVD, en büyük güvenlik açığı veri tabanlarından biridir ve herkese açıktır. CVSS skorlama sistemi sürekli güncellenen ve yeni sürümleri çıkan bir sistemdir. Şuan da resmi olarak NVD tarafından desteklenen sürümleri CVSS 2.0 ve CVSS 3.1'dir. Güvenlik açıkları incelendiğinde sürümler arasında ciddi farklılar görülebilmektedir.

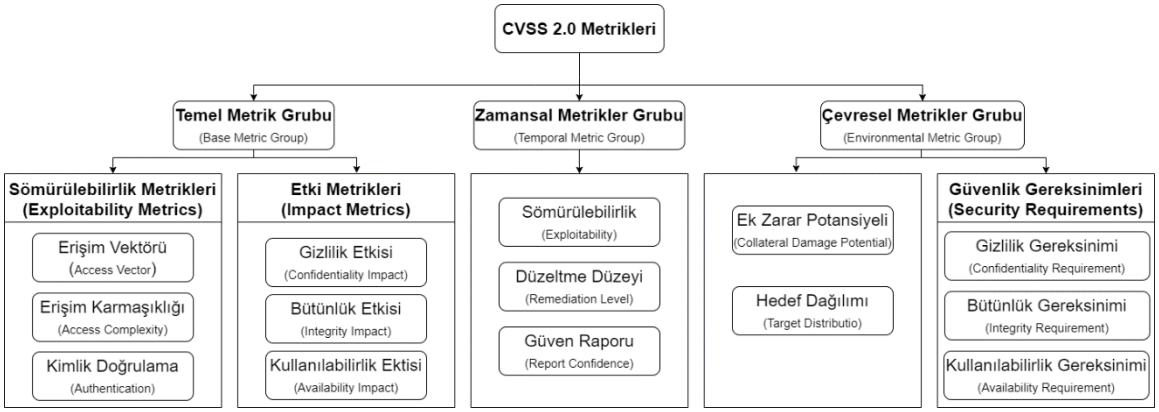
CVSS resmi olarak kullandığı her iki versiyonu içinde üç farklı metrik grubu kullanmaktadır. Bu metrik grupları Temel (Base), Zamansal (Temporal) ve Çevresel (Environmental) olarak isimlendirilmektedir. Her metrik grubunun bağımsız bir güvenlik vektörü ve hesaplaması vardır. Ancak Temel Skor zamanla değişim göstermez. Ayrıca bu skor genellikle yazılımın geliştirici firması veya onların yetkilendirildiği yazılım ürünün koruyan kuruluşlar tarafından oluşturulmaktadır. Bu durum Temel skorların zamanla değişmemesini ve tüm platformlar için standart bir değer üretmesini sağlar. Bu nedenle sadece Temel Metrik değerleri yayınlanmaktadır [39].

Her iki sistem için de puanlama 0.0 – 10.0 aralığında yapılmaktadır. Her iki sistemde temel skor değerini hesaplamak üzere iki kategoriye ayrılmış vektör kümesi kullanmaktadır. Bunlar

Exploitability ve Impact kategorileridir. Impact kategorisi iki sistemde de aynı isimlere sahip vektörler kullansa da içerdikleri değerler farklıdır. Exploitability kategorisinde ise sistemler tamamen farklı vektörler ve değerleri kullanarak birbirlerinden ayrılmaktadır. Tablo 3.1 ve 3.2'te sırayla CVSS 2.0 ve 3.1 sistemlerinin kullandığı temel güvenlik metrikleri ve değerleri açıklamıştır. Görüldüğü üzere temel skoru hesaplamak için CVSS 2.0'da 6 boyutlu bir güvenlik şeması kullanırken CVSS 3.1'de 8 boyutlu bir güvenlik şeması kullanılmaktadır.

3.2. CVSS 2.0

CVSS'nin ilk sürümü farklı sektör temsilcileri tarafından incelenmeden ilan edilmiştir. CVSS 1.0 kullanımından sonra gelen geri bildirimler sonucu önemli sorunları olduğu ortaya çıkmıştır. Toplanan geri bildirimlerden elde edilen tecrübeler ile sorunları ortadan kaldıracak ve CVSS'nin doğruluğunu artıracak çalışmalar yapılmaya başlanmıştır. Bu çalışmalar sonucu ilk versiyonun eksikleri giderilmiş hali 1 Haziran 2007 tarihinde onaylanmış ve resmi olarak CVSS 2.0 kabul edilmiştir. CVSS 2.0 en büyük avantajı gerçek risk değerleri ile karşılaştırılarak skorlamanın test edilmiş olmasıdır. Ayrıca geriye dönük eski güvenlik açıkları yeni versiyonun güvenlik şemasına göre puanlanmıştır. CVSS 2.0 yukarıda bahsedildiği gibi Temel, Zamansal ve Çevresel metrik gruplarına sahiptir [37]. Şekil 3.1'de metrik grupları ve alt metrikleri görülmektedir.



Şekil 0.1. CVSS 2.0 güvenlik metrikleri [37]

3.2.1. CVSS 2.0 Temel Metrikler

Bu metrik grubu ile bir yazılım güvenlik açığının, kullanıcı ortamının veya platformun değişmesi ile zamanla değişmeyecek özellikleri belirtilir. Bu metrik grubu altı bağımsız metrik grubuna sahiptir. Bu metrik grubunda herhangi bir alt ayrım açıkça belirtilmemiş olsa da güvenlik açığına erişimin nasıl sağlandığını ve bu açığın istismar edilmesi için ek şartlara ihtiyaç duyulup duyulmadığını belirleyen üç alt metrik Erişim Vektörü (Access Vector - AV), Erişim Karmaşıklığı

(Access Complexity - AC) ve Kimlik Doğrulama (Authentication - Au) olarak isimlendirilir. Diğer üç metrik güvenlik açığının olası etkilerini ve istismar edildiğinde verebileceği potansiyel zararı belirten etki (Impact) metrikleri olarak isimlendirilmektedir. Bunlar olası etkilerin farklı yönlerini birbirinden bağımsız olarak belirten Gizlilik Etkisi (Confidentiality Impact - CI), Bütünlük Etkisi (Integrity Impact - II) ve Kullanılabilirlik Etkisi (Availability Impact - AI) olarak isimlendirilmektedir. Bağımsız etki bir güvenlik açığının kullanılabilirlik kaybına neden olurken, herhangi bir gizlilik etkisine neden olamayabileceği anlamına gelmektedir [37]. Tablo 3.1'de CVSS 2.0'ın temel metriklerinin aldığı değerler gösterilmektedir.

Tablo 0.1. CVSS 2.0 güvenlik metrikleri [37]

Vektör	Açıklama	Değer	Ağırlık	Kategori
Access Vector	Güvenlik açığından yararlanabilme yöntemini belirtir. Saldırgan ne kadar uzaksa değer o kadar artar.	Local (L) Adjacent Network (A) Network (N)	0.395 0.646 1.0	Exploitability
Access Complexity	Açıktan yaralanmak için gereken saldırının karmaşıklık ölçüsüdür. Karmaşıklık ne kadar düşükse değer o kadar yüksek olur.	High (H) Medium (M) Low (L)	0.35 0.61 0.71	Exploitability
Authentication	Açıktan yararlanabilmek için gereken kimlik doğrulama değeridir. Değer ne kadar düşük olursa puan o derece yüksek olur.	Multiple (M) Single (S) None (N)	0.45 0.56 0.704	Exploitability
Confidentiality Impact	Açığın istismarının sistemin gizliliği üzerindeki etkisini ifade eder. Sistemde meydana gelecek bir sömürü sonucu gizliliğin ne ölçüde etkileneceğini puanlar. Etki arttıkça puan artar.	None (N) Partial (P) Complete (C)	0.0 0.275 0.660	Impact
Integrity Impact	Başarılı bir saldırının sistem bütünlüğü üzerinde oluşturacağı etkiyi ifade eder. Bütünlük etkisinin artması puanını artırır.	None (N) Partial (P) Complete (C)	0.0 0.275 0.660	Impact
Availability Impact	Sistemin kullandığı kaynakların olası bir saldırı durumunda nasıl etkilendiği belirtir. Etkinin artması güvenlik açığı puanının atmasını sağlar.	None (N) Partial (P) Complete (C)	0.0 0.275 0.660	Impact

Erişim Vektörü (Access Vector - AV)

Bir yazılım güvenlik açığından nasıl bir erişim üzerinden yararlanılabileceğini ifade eden metriktir. İstismarı gerçekleştiren saldırgan ana bilgisayara ne kadar uzaktan erişim sağlayabiliyorsa o kadar yüksek bir puan almaktadır. Alabileceği değerler: Local(L), Adjacent Network (A), Network (N)'dır [37].

Local (L): Bir güvenlik açığının yalnızca yerel bilgisayar üzerinden istismar edilebileceği anlamına gelmektedir. Bu saldırganın, istismar edeceği sistemler ile fiziksel olarak aynı ortamda olması gerektiği anlamına gelmektedir. Ayrıca sistem üzerinde oturum açma yetkisini de elde etmesi gerektiği durumlardır [37].

Adjacent Network (A): Saldırı altındaki sistem ile saldırganın aynı alt ağda bulunmasını gerektiren güvenlik açıklarını ifade etmektedir [37].

Network (N): Güvenlik açığına ağ üzerinden gerçekleştirilebilen bir saldırı durumunu ifade eder. Bu şekilde sınıflandırılan bir güvenlik açığına erişim için yerel ağ erişimi gerekmemektedir. Bu tarz güvenlik açıkları uzaktan yararlanılabilir açıklar olarak isimlendirilir [37].

Erişim Karmaşıklığı (Access Complexity - AC)

Güvenlik açığının bulunduğu sisteme erişim elde eden bir saldırganın sistem içerisindeki zafiyetten yararlanabilmesi için gereken karmaşıklık düzeyini belirten metriktir. Bu sistemdeki güvenlik açığının istismarı ne kadar yüksek bir karmaşıklığa sahip ise bu metriğin değeri o kadar yükselir. Bu değerler High (H), Medium (M) ve Low (L) olarak sınıflandırılır [37].

High (H): Özel gereksinimler gerektiren koşullara sahip açıkları ifade eder [37].

Medium (M): Sistem üzerinde saldırganın özelleşmiş prosedür uygulayabilmesi için biraz özelleşmiş durumların gereksinimini ifade eder [37].

Low (L): Herhangi bir özelleştirilmiş durumu bulunmayan durumları ifade etmektedir [37].

Kimlik Doğrulama (Authentication - Au)

Bir güvenlik açığının istismar edilebilmesi için saldırganın kimlik doğrulama işlemi yapma veya yapmama durumunu belirleyen metriktir. Bu metrik oturum açma sürecinin ne kadar güvenli olduğunu ölçmemektedir. Metrik değerleri Multiple (M), Single (S) ve None (N)'dur [37].

Multiple (M): İstismar edilmek istenen sisteme erişmek için saldırganın birden fazla kimlik doğrulama aşamasını gerçekleştirmesi gereklidir [37].

Single (S): Saldırganın tek seviyeli bir oturum açmanın yeterli olması durumudur [37].

None (N): Saldırganın güvenlik açığını istismar etmesi için herhangi bir kimlik doğrulamaya gereksinim duymaması durumudur [37].

Gizlilik (confidentiality Impact - CI)

Bu metrik bir saldırganın sistem üzerinde başarı ile gerçekleştirdiği bir istismar sonucunda meydana gelen bilgi ifşasının etkisini göstermektedir. İstismar sonucu sistemden bilgi sızmasının boyutuna göre değer almaktadır. Bu değerler None (N), Partial (P) ve Complete (C)'dir [37].

None (N): Sistemin gizliliğini etkileyen bir durum söz konusu değildir [37].

Partial (P): Sistemin gizliliğini kısmi olarak etkileyen, bazı verilere sınırlı erişimin mümkün olduğu durumlardır [37].

Complete (C): Saldırganın sistem üzerindeki tüm verilerine erişim yetkisi kazanması durumudur [37].

Bütünlük Etkisi (Integrity Impact - II)

Bir sistemdeki bilgilerin güvenilirliğinin ve doğruluğunun garanti olup olmaması üzerine saldırının etkisini ölçen metriktir. Saldırı sonucu meydana gelen etkinin bütünlüğüne etkisinin

artması skorlama sonucunu doğrudan artıracaktır. Metriğin sınıfları None (N), Partial (P) ve Complete (C)'dir [37].

None (N): Sistemin bütünlüğünü etkileyen bir durum söz konusu değildir [37].

Partial (P): Sistemdeki bazı dosyaların veya bilgilerin değiştirilmesi mümkündür [37].

Complete (C): Sistem saldırı sonucu tamamen korumasız kalmıştır ve sistem bütünlüğü tamamen risk altındadır [37].

Kullanılabilirlik Etkisi (Availability Impact - AI)

Bir sistemin kullandığı kaynakların (bilgi kaynaklarına erişim, ağ bant genişliği, işlemci döngüleri, disk kullanımı vb.) başarılı bir saldırı sonrası etkilenmesini ölçen metriktir. Bu metriğin alabileceği değerler, None (N), Partial (P) ve Complete (C)'dir [37].

None (N): Sistemin kullanılabilirliğini etkileyen bir durum söz konusu değildir [37].

Partial (P): Sistemdeki bazı kaynaklar için kısmı olarak kesintiler veya performans düşüklükleri mümkündür [37].

Complete (C): Sistemde saldırı sonucu, etkilenen kaynaklar tamamen işlevsiz kalmıştır. [37].

3.2.2. CVSS 2.0 Zamansal Metrikler

Zamansal metrikler, bildirilen güvenlik açığının bazı etkilerinin zamanla değişebileceği durumları ifade etmektedir. Bu durumlar iyileştirme durumları, istismar kodlarının varlığı ve kullanılabilir olup olmadığıdır. Bu metrik grubuna ait değerlerin bildirimi zorunlu değildir. İsteğe bağlı olarak bazı metrik değerleri verilebilir veya atlanabilir. Metrik değerleri Sömürülebilirlik (Exploitability - E), Düzeltme Düzeyi (Remediation Level - RL) ve Güven Raporu (Report Confidence - RC)'dur [37].

Sömürülebilirlik (Exploitability - E)

Bildirilen bir güvenlik açığının saldırganlar tarafından istismar edilebilirliğinin düzeyini gösteren metriktir. Bir güvenlik açığı teorik olarak gerçek dünyada her zaman sömürülebilir. Ancak bunun için ihtiyaç duyulan teknik ihtiyaçların ayrıntıları veya herhangi bir gereksinime ihtiyaç duyulmadan istismar edilmesi tehlike düzeyini belirlemektedir. Ayrıca başlarda bir kavramsal kanıt alarak belirtilen bildirim zamanla tutarlı bir istismar koduna dönüşebilir. Bu metriğin alabileceği değerler; Unproven (U), Proof-of-Concept (POC), Functional (F), High (H) ve Not Defined (ND)'dır [37].

Unproven (U): Bilinen herhangi bir istismar kodu henüz mevcut değildir [37].

Proof-of-Concept (POC): Kavramsal olarak istismar kodu olabilir ancak pratik olarak kullanılamaz [37].

Functional (F): Kullanılabilir bir istismar kodu vardır. Güvenlik açığının bazı durumlarında çalıştırılabilir.

High (H): Güvenlik açığı için istismar kodu otonom veya manuel olarak güvenlik açığının her durumunda çalışabilir [37].

Not Defined (ND): Herhangi bir tanımlama içermez. Hesaplama denklemi için bir işarettir [37].

Düzeltme Düzeyi (Remediation Level - RL)

Bir güvenlik açığı ilk bildirildiği anda hemen bir güvenlik yaması ile çoğu zaman düzeltilmez. Çoğu zaman ilk aşamada belirli geçici çözümler, düzeltmeler ve sonrasında bir resmi yama yayınlanmaktadır. Bazen de geçici çözümlerden sonra yama yerine resmi bir yeni sürüm yükselmesinde sorun çözülmektedir. Bu durum zamanla düzeltilen bir süreci işaret etmektedir. Bu metriğin alabileceği değerler, Official Fix (OF), Temporary Fix (TF), Workaround (W), Unavailable (U) ve Not Defined (ND)'dır [37].

Official Fix (OF): Satıcı tarafından resmi olarak yayınlanmış bir yama veya yükseltme ile güvenlik açığına dair tüm riskler ortadan kaldırılmıştır [37].

Temporary Fix (TF): Satıcı tarafından güvenlik açığının risklerini azaltmak için resmi bir geçici çözüm yayınlanmıştır [37].

Workaround (W): Güvenlik açığı için resmi olmayan bir çözüm mevcuttur [37].

Unavailable (U): Güvenlik açığı için herhangi bir çözüm mevcut değildir [37].

Not Defined (ND): Herhangi bir tanımlama içermez. Hesaplama denklemi için bir işarettir [37].

Güven Raporu (Report Confidence - RC)

Tespit edilen güvenlik açığı bildirilirken, açık hakkında rapor edilen teknik ayrıntı düzeyine göre açığın güven derecesini belirleyen metriktir. Bir güvenlik açığı tüm teknik ayrıntıları ile duyurulabileceği gibi hiçbir ek ayrıntı bilgi vermeden, sadece varlığı duyurulabilir. Eksik veriler ile bildirilen açıklar sonradan doğrulanabilir ve onaylanabilir. Bir güvenlik açığının resmi olarak onaylanması ciddiyetini artıracaktır. Ayrıca bu metrik saldırıda kullanılabilecek teknik bilgi düzeyini de göstermektedir. Bir güvenlik açığı resmi satıcı veya diğer kurumlar tarafından da doğrulanıp ve onaylanması puanını artıracaktır. Bu metriğin alabileceği değerler; Unconfirmed (UC), Uncorroborated (UR), Confirmed (C) ve Not Defined (ND)'dır [37].

Unconfirmed (UC): Bildirilen raporun geçerliğine çok düşük güven vardır. Güvenilir kaynaklardan doğrulama yapılamamıştır [37].

Uncorroborated (UR): Resmi olmayan bağımsız birkaç organizasyon tarafından onaylanmış bir güvenlik açığı raporudur. Ancak teknik ayrıntılarda çelişkiler veya belirsizlikler vardır [37].

Confirmed (C): Resmi satıcı tarafından güvenlik açığı onaylanmıştır [37].

Not Defined (ND): Herhangi bir tanımlama içermez. Hesaplama denklemi için bir işarettir [37].

3.2.3. CVSS 2.0 Çevresel Metrikler

Bir güvenlik açığının istismarı son kullanıcının, kuruluşların ve paydaşların kendi ortamlarına göre risk düzeyini etkilemektedir. Bağımsız güvenlik protokollerinin uygulandığı ortamlarda etkisiz kalan bir açık, daha güvensiz bir ortamda risk oluşturabilmektedir. Çevresel metrikler farklı ortamların güvenlik açığı üzerindeki etkisini göstermektedir. Bu metrik grubunu oluşturan metrikler; Ek Hasar Potansiyeli (Collateral Damage Potential - CDP), Hedef Dağılımı (Target Distribution - TD) ve Güvenlik Gereksinimleri (Security Requirements)'dir. Güvenlik gereksinimleri metriği kendi içerisinde üç farklı metrik grubuna ayrılmaktadır. Bunlar Gizlilik Gereksinimi (Confidentiality Requirement – CR), Bütünlük Gereksinimi (Integrity Requirement – IR) ve Kullanılabilirlik Gereksinimi (Availability Requirement – AR)'dir [37].

Ek Hasar Potansiyeli (Collateral Damage Potential - CDP)

Bir sistemin saldırısında meydana gelebilecek maddi ve can kaybı gibi potansiyel zararların etkisini ölçmektedir. Metrik değeri ile meydana gelebilecek gelir kaybı da ölçülebilir. Bu metrik güvenlik açığının ekonomik anlamda sistemin verimliliğine etkisinin de bir göstergesidir. Alabileceği değerler; None (N), Low (L), Low-Medium (LM), Medium-High (MH), High (H) ve Not Defined (ND)'dir. [37].

None (N): Herhangi bir maddi veya can kaybı riski yoktur [37].

Low (L): Güvenlik açığı küçük çaplı bir maddi veya fiziksel zarara neden olabilir [37].

Low-Medium (LM): Güvenlik açığı orta çaplı bir maddi veya fiziksel zarara neden olabilir [37].

Medium-High (MH): Güvenlik açığı önemli çapta bir maddi veya fiziksel zarara neden olabilir [37].

High (H): Güvenlik açığı telafisi mümkün olmayacak şekilde feci maddi veya fiziksel zararlara neden olabilir [37].

Not Defined (ND): Herhangi bir tanımlama içermez. Hesaplama denklemi için bir işarettir [37].

Hedef Dağılımı (Target Distribution - TD)

Bir bilişim sistemindeki savunmasız bileşenlerin oranını ölmektedir. Başarılı bir saldırı anında sistemdeki etkilenebilecek bileşenlerin yüzdesidir. Alabileceği değerler; None (N), Low (L), High (H) ve Not Defined (ND)'dir. [37].

None (N): Güvenlik açığından etkilenen hiçbir bileşen yoktur. Risk %0'dır [37].

Low (L): Güvenlik açığından etkilenen küçük ölçekte bileşenler var. Risk %1 - %25'dır [37].

Medium (M): Güvenlik açığından etkilenen orta ölçekte bileşenler var. Risk %26 - %75'dır [37].

High (H): Güvenlik açığından etkilenen önemli ölçekte bileşenler var. Risk %100'dır [37].

Not Defined (ND): Herhangi bir tanımlama içermez. Hesaplama denklemi için bir işarettir [37].

Güvenlik Gereksinimleri (Security Requirements)

Bir güvenlik açığına sahip sistemin kullanıcısı olan kurumlar açısından önemini ölçen metriktir. Kullanıcı ortamlarının sahip olduğu güvenlik düzeylerine göre temel metriklerdeki gizlilik, bütünlük ve kullanılabilirlik metriklerinin özelleştirilmesini sağlamaktadır. Bu açıdan üç alt metrik değerine sahiptir. Bunlar; Gizlilik Gereksinimi (Confidentiality Requirement – CR), Bütünlük Gereksinimi (Integrity Requirement – IR) ve Kullanılabilirlik Gereksinimi (Availability Requirement – AR). Alt metriklerin her birinin alabileceği değerler; High (H), Medium (M) ve Low (L) ve Not Defined (ND)'dır [37].

Low (L): Güvenlik açığının sınırlı olumsuz etkisini ifade eder.[37].

Medium (M): Güvenlik açığının potansiyel ciddi etkileri olabileceğini ifade eder.[37].

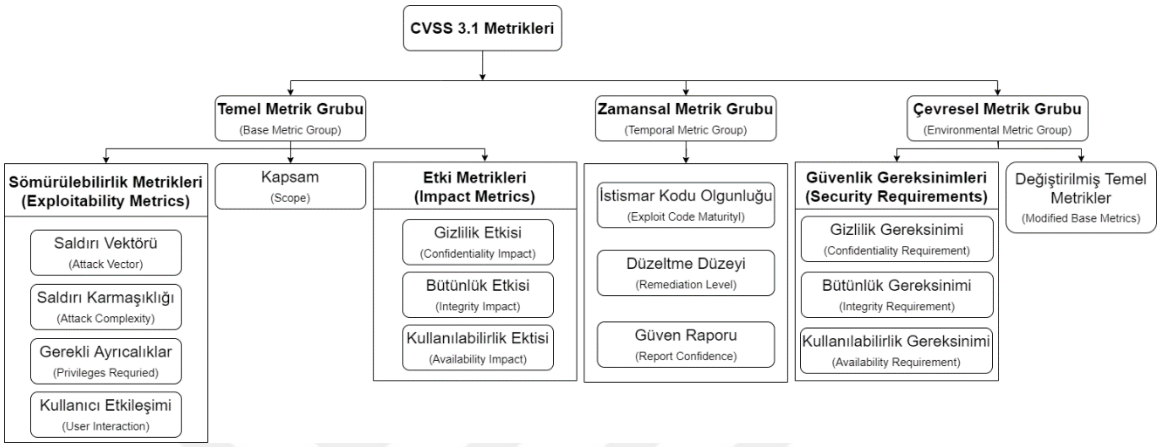
High (H): Güvenlik açığının yıkıcı etkisi olabileceğini ifade eder.[37].

Not Defined (ND): Herhangi bir tanımlama içermez. Hesaplama denklemi için bir işarettir [37].

3.3. CVSS 3.1

Yazılım sistemlerinin güvenlik sorunu günümüzde artarak devam etmektedir. Güvenli sistemler geliştirme yönünde birçok yöntem önerilse de saldırganlarda farklı teknikler geliştirmeye devam etmektedir. Bu durum güvenlik algısının ve yöntemlerinin sürekli değişmesine neden olmaktadır. Sürekli değişen ve kendini yenileyen bu duruma CVSS'de ayak uydurmak zorundadır. Bu nedenle Haziran 2019'da şuanda kullanımdaki en güncel sürümü olan CVSS 3.1 yayınlanmıştır. CVSS 3.1 versiyonu 3.0 ile aynı güvenlik şemasını kullanmaktadır. Ancak karmaşıklaşan hesaplama ve metriklerin daha iyi açıklanması ve anlaşılması için versiyon güncellenmiştir. CVSS 3.0 ve 3.1 arasındaki en büyük değişiklik kavramlara açıklık getirilmesidir. Güvenlik şemasında veya hesaplamalarda bir değişiklik olmamıştır. Bu nedenle NVD veri seti içerisindeki raporlarda

CVSS 3.0 vektör değerlerine sahip kayıtlar değiştirilmemiştir. Yeni kayıtlar ise CVSS 3.1 vektör olarak işlenmektedir. CVSS 3.1 versiyonu da daha önce açıklanan üç farklı metrik grubundan oluşmaktadır. Bunlar Temel, Zamansal ve Çevresel metrik gruplarıdır. Her metrik grubunun alt metrikleri değişiklik göstermektedir. Şekil 3.2'de CVSS 3.1'in metrik grupları ve alt metrikleri görülmektedir [39].



Şekil 0.2. CVSS 3.1 güvenlik metrikleri [39]

3.3.1. CVSS 3.1 Temel Metrikler

Önceki versiyonlarda olduğu gibi temel metrikler zamana ve farklı ortamlara göre değişim göstermeyen bir metrik grubudur. Bu özelliği sayesinde güvenlik açığı için genel bir bilgi sunmaktadır. CVSS 2.0 versiyonunda açıkça belirtilmeyen alt gruplar CVSS 3.1 versiyonun resmi olarak tanımlanmıştır. Temel metrikler Sömürülebilirlik (Exploitability) ve Etki (Impact) metrikleri olarak ikiye ayrılmaktadır. Ayrıca Kapsam (Scope) olarak isimlendirilen bağımsız bir metriğe de sahiptir. Kullanılabilirlik metrikleri Saldırı Vektörü (Attack Vector - AV), Saldırı Karmaşıklığı (Attack Complexity - AC), Gerekli Ayrıcalıklar (Privileges Required - PR) ve Kullanıcı Etkileşimi (User Interaction - UI)'dir. Etki Metrikleri ise Gizlilik (Confidentiality - C), Bütünlük (Integrity - I) ve Kullanılabilirlik(Availability -A) metrikleridir [39]. Tablo 3.2'de temel metrik grubu ve değerleri gösterilmiştir.

Saldırı Vektörü (Attack Vector - AV)

Bir saldırganın, istismar etmek istediği sisteme uzaklığını belirtir. Saldırı ne kadar uzaktan gerçekleştirilebiliyorsa o kadar değer büyük olur. CVSS 2.0 temel metriklerinden Erişim vektörünün karşılığıdır. Ancak alabileceği değerler farklılaşmıştır. Bu metriğin alabileceği değerler; Network (N), Adjacent (A), Local (L) ve Physical (P)'dır [39].

Network (N): Bir güvenlik açığı bir ağ kümesine bağlıdır ve saldırgan birkaç ağ atlaması ile (örneğin birkaç yönlendirici üzerinden) veya internet ortamından açığı istismar edebilmektedir [39].

Adjacent (A): Saldırıya açık sistem veya bileşen bir ağ kümesinin elemanıdır ancak açığın istismarı için aynı ağda paylaşılan (Bluetooth, aynı yerel ağ vb.) bir ortamda bulunulması gerekmektedir [39].

Local (L): Sistem bir ağ kümesine bağlı değildir. Sistem üzerinde saldırganın etkili olabilmesi için yerel veya uzaktan erişmesi gerekmektedir [39].

Physical (P): Saldırganın sistem ile fiziksel olarak aynı ortamda bulunması ve doğrudan sisteme fiziksel olarak dokunarak güvenlik açığını istismar etmesi gerekir [39].

Tablo 0.2. CVSS 3.1 güvenlik metrikleri [39]

Vektör	Açıklama	Değer	Ağırlık	Kategori
Attack Vector	Saldırganın güvenlik açığından yararlanabilmesi için gerekli olan durumu açıklamaktadır. Savunmasız sistemden yararlanabilmek ne kadar uzaktan mümkün olursa o kadar puan artar.	Network (N) Adjacent (A) Local (L) Physical (P)	0.85 0.62 0.55 0.2	Exploitability
Attack Complexity	Saldırganın başarılı bir istismar gerçekleştirebilmesi için gerekli harici koşulları ifade eder. Puan karmaşıklığı az saldırılar için en yüksek değere sahiptir.	Low (L) High (H)	0.77 0.44	Exploitability
Privileges Required	Güvenlik açığından yararlanabilmek için saldırganın sahip olması gereken olan ayrıcalık düzeyini ifade eder. Ayrıcalık gerektirmeyen bir açık için değer en yüksek puana sahiptir.	None (N) Low (L) High (H)	0.85 0.62-0.68* 0.27-0.5*	Exploitability
User Interaction	Güvenlik açığından yararlanabilmek için her hangi bir kullanıcının eyleminin gerekip gerekmediğini ifade eder. Eğer farklı bir kullanıcının eylemi gerekmiyorsa puan yüksektir.	None (N) Required (R)	0.85 0.62	Exploitability
Confidentiality Impact	Açığın istismarının sistemin gizliliği üzerindeki etkisini ifade eder. Sistemde meydana gelecek bir sömürü sonucu gizliliğin ne ölçüde etkileneceğini puanlar. Etki arttıkça puan artar.	High (H) Low (L) None (N)	0.56 0.22 0.0	Impact
Integrity Impact	Başarılı bir saldırının sistem bütünlüğü üzerinde oluşturacağı etkiyi ifade eder. Bütünlük etkisinin artması puanını artırır.	High (H) Low (L) None (N)	0.56 0.22 0.0	Impact
Availability Impact	Sistemin kullandığı kaynakların olası bir saldırı durumunda nasıl etkilendiği belirtir. Etkinin artması güvenlik açığı puanının atmasını sağlar.	High (H) Low (L) None (N)	0.56 0.22 0.0	Impact
Scope (S)	Güvenlik açığının bulunduğu bileşen dışında kalan bileşenlerin kullandığı kaynakları etkiyip etkilemediğini ifade eder. Bir kapsam değişikliği gerçekleşirse puan yükselecektir.	Unchanged(U) Changed (C)		

*Eğer Kapsam Değiştiyse

Saldırı Karmaşıklığı (Attack Complexity - AC)

Bir güvenlik açığından saldırganın yararlanabilmesi için gereken, saldırganın kontrolü dışında kalan koşulları ifade etmektedir. Bir saldırganın saldırıyı gerçekleştirebilmesi için kullanması gereken efor ve teknik bilgiyi belirtir. Bir saldırı için gereksinimler ne kadar karmaşıksa puanı o derece azalır. Alabileceği değerler; Low (L) ve High (H)'dir [39].

Low (L): Başarılı bir saldırı için özel ve karmaşık koşullar mevcut değildir [39].

High (H): Başarılı bir saldırı için özel koşulların sağlanması gerekmektedir. Saldırganı başarılı bir saldırı gerçekleştirebilmesi için yoğun bir hazırlık yapması ve çaba göstermesi gerekmektedir [39].

Gerekli Ayrıcalıklar (Privileges Required – PR)

Bir güvenlik açığından faydalanabilmesi için saldırganın sahip olması gereken ayrıcalık düzeyini tanımlar. Eğer bir güvenlik açığı istismar edilmeden önce bir ayrıcalık gerektirmiyorsa temel puan o derece yüksektir. Alabileceği değerler; None (N), Low (L) ve High (H)'dır [39].

None (N): Saldırıyı gerçekleştirebilmek için herhangi bir ayrıcalığa ihtiyaç yoktur [39].

Low (L): Saldırı gerçekleştirebilmek için bir kullanıcı yetkisinde ayrıcalıklara sahip olunması gerekmektedir. Önemli olmayan kaynaklara erişim yetkisi vardır [39].

High (H): Saldırganın başarılı bir istismar gerçekleştirebilmesi için önemli ayrıcalıklara ihtiyacı vardır [39].

Kullanıcı Etkileşimi (User Interaction – UI)

Bu metrik ile bir güvenlik açığından saldırganın yararlanabilmesi için herhangi bir kullanıcının eylemine ihtiyaç duyup duymadığı ölçülür. Burada bir kullanıcı tarafından başlatılan süreç gereksinimleri de dikkate alınır. Kullanıcı etkileşimi gerektirmeyen güvenlik açıklarının puanı daha yüksektir. Alabileceği değerler; None (N) ve Required (R)'dır [39].

None (N): Herhangi bir kullanıcı etkileşimine ihtiyaç yoktur [39].

Required (R): Güvenlik açığından yararlanılabilmesi için muhakkak bir kullanıcı eylemi gereklidir [39].

Kapsam (Scope – S)

Bu metrik ile bir güvenlik açığına sahip bileşenin istismarı sonucu kapsamı dışındaki farklı bileşenleri ve kaynakları etkileyip etkilemediği ölçülmektedir. Eğer bir güvenlik açığı kapsamı dışındaki bileşenleri ve kaynakları da etkileyebiliyor veya onları da savunmasız ve istismara açık hale getiriyor ise daha tehlikeli bir güvenlik açığına dönüşmektedir. Alabileceği değerler; Unchanged (U), Changed (C)'dır [39].

Unchanged (U): Bir güvenlik açığı sadece bulunduğu bileşen ve kaynakları etkileyebilir [39].

Changed (C): Bir güvenlik açığı kapsamı dışındaki bileşen ve kaynakları da etkileyebilir [39].

Gizlilik (Confidentiality – C)

Gizlilik metriği bir saldırganın sistem üzerinde başarı ile gerçekleştirdiği bir istismar sonucunda meydana gelen bilgi ifşasının etkisini göstermektedir. İstismar sonucu sistemden bilgi sızmasının boyutuna göre değer almaktadır. Bu değerler None (N), Low (L) ve High (H)'dir [37].

None (N): Sistemin gizliliğini etkileyen bir durum söz konusu değildir [39].

Low (L): Sistemin gizliliğini kısmı olarak etkileyen, bazı verilere sınırlı erişimin mümkün olduğu durumlardır [39].

High (H): Saldırganın sistem üzerindeki tüm verilerine erişim yetkisi kazanması durumudur [39].

Bütünlük (Integrity – I)

Bir sistemdeki bilgilerin güvenilirliğinin ve doğruluğunun garanti olup olmaması üzerine saldırının etkisini ölçen metriktir. Saldırı sonucu meydana gelen etkinin bütünlüğüne etkisinin artması skorlama sonucunu doğrudan artıracaktır. Metriğin sınıfları None (N), Low (L) ve High (H)'dir [39].

None (N): Sistemin bütünlüğünü etkileyen bir durum söz konusu değildir [39].

Low (L): Sistemdeki bazı dosyaların veya bilgilerin değiştirilmesi mümkündür [39].

High (H): Sistem saldırı sonucu tamamen korumasız kalmıştır ve sistem bütünlüğü tamamen risk altındadır [39].

Kullanılabilirlik (Availability - A)

Bir sistemin kullandığı kaynakların (bilgi kaynaklarına erişim, ağ bant genişliği, işlemci döngüleri, disk kullanımı vb.) başarılı bir saldırı sonrası etkilenmesini ölçen metriktir. Bu metriğin alabileceği değerler, None (N), Low (L) ve High (H)'dir [39].

None (N): Sistemin kullanılabilirliğini etkileyen bir durum söz konusu değildir [39].

Low (L): Sistemdeki bazı kaynakların kısmi olarak kesintileri veya performans düşüklükleri mümkündür [39].

High (H): Sistemde saldırı sonucu etkilenen kaynaklar tamamen işlevsiz kalmıştır [39].

3.3.2. CVSS 3.1 Zamansal Metrikler

Zamansal metrikler bir güvenlik açığından yararlanılabilme durumunu, mevcut çözümlerini ve bunların onaylanıp onaylanmadığını ölçen metriklerdir. Güvenlik açığına ait metrik grubunun değerleri zamanla değişebilmektedir. Bu metrik grubunun alt metrikleri Sömürü Kodu Olgunluğu

(Exploit Code Maturity - E), Düzeltme Düzeyi (Remediation Level - RL) ve Güven Raporu (Report Confidence - RC) şeklinde tanımlanmıştır [39].

Sömürü Kodu Olgunluğu (Exploit Code Maturity - E)

Bildirilen bir güvenlik açığının saldırganlar tarafından istismar edilebilirliğinin düzeyini gösteren metriktir. Bir güvenlik açığı teorik olarak gerçek dünyada her zaman sömürülebilir. Ancak bunun için ihtiyaç duyulan teknik ihtiyaçların ayrıntıları veya herhangi bir gereksinime ihtiyaç duyulmadan istismar edilmesi tehlike düzeyini belirlemektedir. Ayrıca başlarda bir kavramsal kanıt alarak belirtilen bildirim zamanla tutarlı bir istismar koduna dönüşebilir. Bu metriğin alabileceği değerler; Unproven (U), Proof-of-Concept (POC), Functional (F), High (H) ve Not Defined (ND)'dır [39].

Unproven (U): Bilinen herhangi bir istismar kodu henüz mevcut değildir [39].

Proof-of-Concept (POC): Kavramsal olarak istismar kodu olabilir ancak pratik olarak kullanılamaz [39].

Functional (F): Kullanılabilir bir istismar kodu vardır. Güvenlik açığının bazı durumlarında çalıştırılabilir.

High (H): Güvenlik açığı için istismar kodu otonom veya manuel olarak güvenlik açığının her durumunda çalışabilir [39].

Not Defined (ND): Diğer seçeneklerden herhangi birini seçmek için yeterli bilgi olmadığını bildirmektedir [39].

Düzeltme Düzeyi (Remediation Level - RL)

Bir güvenlik açığı ilk bildirildiği anda hemen bir güvenlik yaması ile çoğu zaman düzeltilmez. Çoğu zaman ilk aşamada belirli geçici çözümler, düzeltmeler ve sonrasında bir resmi yama yayınlanmaktadır. Bazen de geçici çözümlerden sonra yama yerine resmi bir yeni sürüm yükseltmesinde sorun çözülmektedir. Bu durum zamanla düzeltilen bir süreci işaret etmektedir. Bu metriğin alabileceği değerler, Official Fix (OF), Temporary Fix (TF), Workaround (W), Unavailable (U) ve Not Defined (ND)'dır [39].

Official Fix (OF): Satıcı tarafından resmi olarak yayınlanmış bir yama veya yükseltme ile güvenlik açığına dair tüm riskler ortadan kaldırılmıştır [39].

Temporary Fix (TF): Satıcı tarafından güvenlik açığının risklerini azaltmak için resmi bir geçici çözüm yayınlanmıştır [39].

Workaround (W): Güvenlik açığı için resmi olmayan bir çözüm mevcuttur [39].

Unavailable (U): Güvenlik açığı için herhangi bir çözüm mevcut değildir [39].

Not Defined (ND): Diğer seçeneklerden herhangi birini seçmek için yeterli bilgi olmadığını bildirmektedir [39].

Güven Raporu (Report Confidence - RC)

Tespit edilen güvenlik açığı bildirilirken, açık hakkında rapor edilen teknik ayrıntı düzeyine göre açığın güven derecesini belirleyen metriktir. Bir güvenlik açığı tüm teknik ayrıntıları ile duyurulabileceği gibi hiçbir ek ayrıntı vermeden, sadece varlığı duyurulabilir. Eksik veriler ile bildirilen açıklar sonradan doğrulanabilir ve onaylanabilir. Bir güvenlik açığının resmi olarak onaylanması ciddiyetini artıracaktır. Ayrıca bu metrik saldırıda kullanılabilecek teknik bilgi düzeyini de göstermektedir. Bir güvenlik açığının resmi satıcı veya diğer kurumlar tarafından da doğrulanıp ve onaylanması puanını artıracaktır. Bu metriğin alabileceği değerler; Unknown (U), Reasonable (R), Confirmed (C) ve Not Defined (ND)'dır [39].

Unknown (U): Güvenlik açığının varlığı bildirilmiştir ancak açığın nedeni ve potansiyel riskleri bilinmemektedir. Güvenlik açığının etkisi üzerindeki raporlar farklılık göstermektedir ve tam olarak güvenli bir temel puan hesaplanması konusunda güven yoktur [39].

Reasonable (R): Güvenlik açığının doğruluğunu gösteren kavram kanıt gibi makul kanıtlar vardır. Güvenlik açığı ile ilgili önemli ayrıntılar yayınlanmıştır. Ancak güvenilir değerlendiricilerin kaynak koda erişimleri olmadığı için güvenlik açığı doğrulanamamaktadır [39].

Confirmed (C): Resmi satıcı tarafından güvenlik açığı onaylanmıştır [39].

Not Defined (ND): Diğer seçeneklerden herhangi birini seçmek için yeterli bilgi olmadığını bildirmektedir. [39].

3.3.3. CVSS 3.1 Çevresel Metrikler

Bu metrik grubu, güvenlik açığının bir kullanıcı ortamına göre özelleştirilebilmesine imkan sağlar. Bu durum farklı ortamlara sahip kuruluşların durumlarına göre özelleştirilmiş CVSS puanları hesaplamalarını sağlamaktadır. Bu metrik grubunun alt metrikleri Değiştirilmiş Temel Metrikler (Modified Base Metrics) ve Güvenlik Gereksinimleri (Security Requirements)'dir. Güvenlik gereksinimleri metriği kendi içerisinde üç farklı metrik grubuna ayrılmaktadır. Bunlar Gizlilik Gereksinimi (Confidentiality Requirement – CR), Bütünlük Gereksinimi (Integrity Requirement – IR) ve Kullanılabilirlik Gereksinimi (Availability Requirement – AR)'dir [39].

Güvenlik Gereksinimleri (Security Requirements)

Bir güvenlik açığına sahip sistemin kullanıcısı olan kurumlar açısından önemini ölçen metriktir. Bu metrik değeri temel metriklerdeki gizlilik, bütünlük ve kullanılabilirlik metriklerinin kullanıcı kurumlarının ortamları açısından özelleştirilmesini sağlar. Bu açıdan üç alt metrik değerine sahiptir. Bunlar; Gizlilik Gereksinimi (Confidentiality Requirement – CR), Bütünlük Gereksinimi (Integrity Requirement – IR) ve Kullanılabilirlik Gereksinimi (Availability Requirement – AR). Alt metriklerin her birinin alabileceği değerler; High (H), Medium (M) ve Low (L) ve Not Defined (ND)'dır [39].

Low (L): Güvenlik açığının sınırlı olumsuz etkisini ifade eder [39].

Medium (M): Güvenlik açığının potansiyel ciddi etkileri olabileceğini ifade eder [39].

High (H): Güvenlik açığının yıkıcı etkisi olabileceğini ifade eder [39].

Not Defined (ND): Herhangi bir tanımlama içermez. Hesaplama denklemi için bir işarettir [39].

Değiştirilmiş Temel Metrikler (Modified Base Metrics)

Bu metrik değeri ile bir güvenlik açığı kullanıldığı ortamın özelliklerine dayalı olarak temel metrik değerlerinin bireysel olarak kullanıcı ortamına göre özelleştirilmesine imkân sağlar. Bu metrik değerleri tanımlanırken analistler temel metriklerin alabileceği değerinin artırıcı veya azaltıcı etkenlerini tanımlamaktadırlar. Tüm temel metrik grubunun alt metriklerine karşılık gelen alt sınıfları değiştirebilirler [39].

3.4. CVSS Hesaplaması

Formülleri ve metrik değerleri aşağıda verilen güvenlik açığı puanlama sistemleri ile tespit edilen güvenlik açıklarının temel puanları hesaplanmaktadır. Hesaplanan puanlar 0.0 – 10.0 arasında bir değerdir. Bulunan bu değerler, nitel önem derecesi ölçeği kullanılarak sınıflandırılmaktadır. Tablo 3.3 ve 3.4'de sırayla CVSS 2.0. ve 3.1'in ölçekleri gösterilmektedir. Her iki sürümün skor hesaplama formülleri aşağıdaki verilmiştir. Bu formüllerde geçen ifadeler bir güvenlik açığının güvenlik vektöründe bulunan metrik değerlerine karşılık gelen Tablo 3.1 ve 3.2'deki sayısal değerlerin yerlerine konulması ile hesaplanmaktadır.

CVSS 2.0 temel skor hesaplama formülleri;

$$BaseScore = RoundTo1Decimal((0.6 * Impact + 0.4 * Exploitability - 1.5) * f(Impact)) \quad (3.1)$$

$$Impact = 10.41 * (1 - (1 - ConfImpact) * (1 - IntegImpact) * (1 - AvailImpact)) \quad (3.2)$$

$$Exploitability = 20 * AccessComplexity * Authentication * AccessVector \quad (3.3)$$

$$f(Impact) = 0 \; if \; Impact = 0; \; 1.176 \; otherwise \quad (3.4)$$

CVSS 2.0 denklemleri aşağıda 3.1 – 3.4 numaralı formüller ile tanımlanmıştır. Formüllerden 3.1 incelendiğinde temel skor değerinin hesaplanabilmesi için öncelikle Impact, Exploitability değerleri 3.2 ve 3.3 formülleri kullanılarak hesaplanmalıdır. Ayrıca temel skor Impact değerinin sonucuna göre hesaplanan 3.4 formülündeki fonksiyon değeri ile çarpılmaktadır. Eğer Impact değeri 0 ise 3.4 formülünün değeri de 0 çıkacaktır. Bu durumda CVSS 2.0 temel puanı da 0 olarak hesaplanacaktır. Aksi halde 3.4 formülünün değeri 1.176 değeri alır ve temel puan bu değer ile

çarpılarak hesaplanmaktadır. Formül 3.2 ve 3.3 sonuçlarına göre hesaplanan Impact değeri 0.6, Exploitability değeri ise 0.4 ile çarpılıp toplamaktadır. Bulunan değerden 1.5 değeri çıkarılarak formül 3.4 sonucu ile çarpılmaktadır. Bulunan sonuç en yakın bir ondalık basamağa yuvarlanarak temel skor hesaplanmaktadır.

CVSS 3.1 temel skor hesaplama formülleri;

$$ISS = 1 - [(1 - Confidentiality) \times (1 - Integrity) \times (1 - Availability)] \quad (3.5)$$

$$Eğer\ Scope\ Değişmemişse,\ Impact = 6.42 \times ISS \quad (3.6)$$

$$Eğer\ Scope\ Değişmişse,\ Impact = 7.52 \times [ISS - 0.029] - 3.25 \times [ISS - 0.02]^{15} \quad (3.7)$$

$$Exploitability = 8.22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction \quad (3.8)$$

Temel Skor (BaseScore) değeri;

$$Eğer\ Impact\ Değeri\ 0\ ise,\ BaseScore = 0 \quad (3.9)$$

$$Scope\ Değişmemişse, Base\ Score = Roundup(Minimum[(Impact + Exploitability), 10]) \quad (3.10)$$

$$Scope\ Değişmişse,\ Base\ Score = Roundup(Minimum[1.08 \times (Impact + Exploitability), 10]) \quad (3.11)$$

CVSS 3.1 temel skor değerinin hesabında 3.5 ve 3.11 arasındaki formüller kullanılmaktadır. Hesaplama işleminin ilk adımı formül 3.5 kullanılarak etki alt skorunun (Impact Sub Score - ISS) bulunmasıdır. Impact değerinin hesaplanabilmesi için ISS değeri gereklidir. Bu işlemde etki metrik değerleri 1'den çıkarılarak çarpılmakta ve sonucun tekrar 1 değerinden çıkarılması ile ISS değeri hesaplanmaktadır. ISS değeri hesaplandıktan sonra Scope metriğinin değerine göre Impact değeri hesaplanmaktadır. Eğer Scope metriğinin değeri değişmemiş (Unchanged) olarak işaretlenmişse formül 3.6 kullanılmaktadır. Eğer metriğin değeri değişmiş (Changed) olarak işaretlenmişse Impact değeri formül 3.7 kullanılarak hesaplanmaktadır. Temel skor değerini oluşturan diğer bir değişken ise Exploitability değeridir. Bu değerin hesaplanması için Exploitability metrik grubunda bulunan metrikler çarpılmakta ve sonucun 8.22 katı alınmaktadır. Bu işlem formül 3.8'de gösterilmiştir. Impact ve Exploitability değerleri hesaplandıktan sonra temel skor (BaseScore) hesaplanabilir. Bu aşama ilk olarak Impact değerine bakılmaktadır. Eğer Impact değeri sıfıra eşit veya küçükse temel skor formül 3.9'da gösterildiği gibi sıfıra eşitlenir. Eğer Impact sıfır değerinden büyük hesaplanmış ise Scope değerine göre temel Skor hesaplanmaktadır. Formül 3.10'da görüldüğü gibi eğer Scope değişmemiş ise Impact ve Exploitability değerleri toplanır. Bulunan sonuç 10 değerinden küçükse Roundup fonksiyonuna gönderilir. Eğer 10 değerinden büyük ise Roundup fonksiyonuna 10 değeri gönderilmektedir. Roundup fonksiyonunun sonucu temel skor değerini vermektedir.

```
1. function Roundup (input):
2.     int_input = round_to_nearest_integer (input * 100000)
3.     if (int_input % 10000) == 0:
4.         return int_input / 100000.0
5.     else:
6.         return (floor(int_input / 10000) + 1) / 10.0
```

Şekil 0.3. Roundup fonksiyonu [39]

Scope değerinin değişmiş olarak işaretlendiği güvenlik açıklarının BaseScore değerleri formül 3.11'e göre hesaplanmaktadır. Bu formüle göre Impact ve Exploitability değerlerinin toplamı 1.08 ile çarpılır ve bulunan sonuç 10 değerinden küçük ise Roundup fonksiyonuna gönderilir. Eğer bulunan sonuç 10 değerinden büyük ise Roundup fonksiyonuna 10 değeri iletilmektedir. Buradaki Roundup fonksiyonu programlama dillerinden veya donanımdan kaynaklanan kayan nokta hesaplama hatalarını önlemek için kullanılmaktadır [39]. Roundup fonksiyonunun sözde kodu Şekil 3.3'de gösterilmiştir.

Tablo 0.3. CVSS 2.0 nitel önem derecesi ölçeği [12]

Önem Derecesi	Temel Puan Aralığı
Low	0.0 - 3.9
Medium	4.0 - 6.9
High	7.0 – 10.0

Temel skor her zaman 0.0 -10.0 aralığında bir değer olarak hesaplanmaktadır. Bir güvenlik açığı sahip olduğu temel skor aralığına göre bir önem derecesi ile sınıflandırılmaktadır. Bu sınıflandırma işlemine nitel önem derecesi denilmektedir. CVSS 2.0 ve 3.1 için farklı bir önem derecesi sınıflandırması kullanılmaktadır. Tablo 3.3 ve 3.4'de CVSS'nin her iki sistemi için kullanılan önem dereceleri gösterilmektedir. CVSS 2.0 için sadece 3 farklı sınıf kullanılırken, CVSS 3.1 için 5 farklı sınıflandırma kullanılmaktadır.

Tablo 0.4. CVSS 3.1 nitel önem derecesi ölçeği [12]

Önem Derecesi	Temel Puan Aralığı
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

4. MATERYAL VE METOT

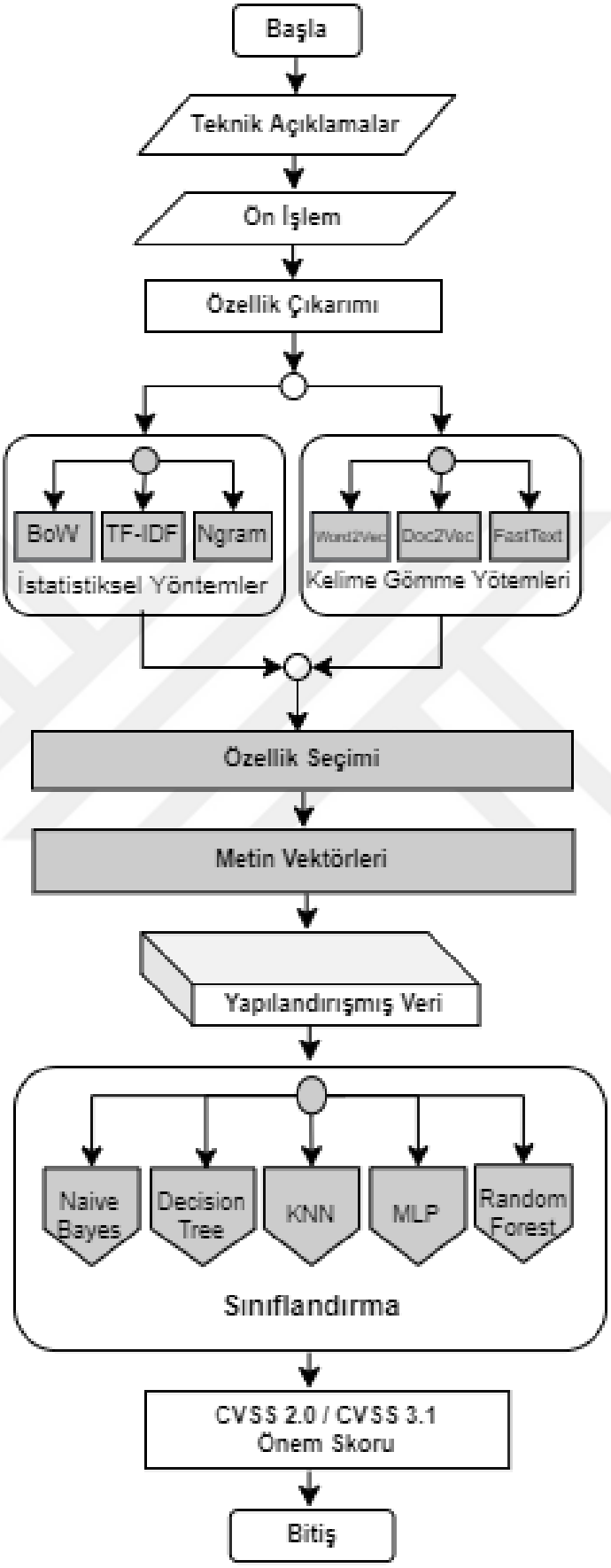
Tez çalışmasının amacı güvenlik açığı raporlarını kullanarak güvenlik açıklarının sahip olduğu metrik değerlerini ve önem derecelerini hesaplamaktır. Önerilen metodolojide bu amacı gerçekleştirebilmek için kullanılacak temel özellik, raporların sahip olduğu teknik açıklama metinleridir. Tablo 4.1'de bulunan rapor örneği incelendiğinde açıklama metinlerinin uzmanlar tarafından doğal dille yazılmış olduğu görülmektedir. Yapılandırılmamış haldeki bu büyük veri kümesinin deterministik bir sistemde kullanılabilmesi için yapılandırılmış bir veri setine dönüştürülmesi gerekmektedir. Metinleri kullanarak oluşturulacak bir modelin genel çerçevesi, ön işleme, öznitelik çıkarımı, öznitelik seçimi ve sınıflandırma aşamalarından oluşmaktadır [40]. Bu tez çalışması da prensip olarak bu aşamaları barındırmaktadır.

Öz nitelik çıkarımı ve seçimi aşamaları metin madenciliği tekniklerinden geleneksel istatistiki yöntemlerin yanında modern derin öğrenme tabanlı kelime gömme algoritmaları kullanılarak geçekleştirilmiştir. Geleneksel istatistiki yaklaşımlardan Bag of Words (BoW) [41], Term Frequency Inverse Document Frequency (TF-IDF) [42], Ngram [43] ve kelime gömme algoritmalarından Word2Vec [44,45], Doc2Vec [46] ve FastText [47] kullanılmıştır. Sınıflandırma aşamasında temel yaklaşım bütün güvenlik açığı vektörlerinin ve önem derecesinin aynı anda tahmin edilmesidir. Bunun için çok sınıflı sınıflandırma yapan Naive Bayes (NB) [48], Desicion Tree (DT) [49], K-Nearest Neighbors (KNN) [50], Multi-layer Perceptron (MLP) [51] ve Random Forest (RF) [52] algoritmaları seçilmiştir. Sınıflandırma işlemi Bölüm 3'te ayrıntıları verilen CVSS güvenlik metriklerinin değerlerinin tahmin edilmesi işlemidir. Tahmin edilen değerlerden güvenlik açığı vektörleri oluşturulmaktadır.

Metodolojinin son aşaması yazılım güvenlik açığı skorunun hesaplanmasıdır. Bu aşamada önceki bölümde açıklanan nedenlerle CVSS skorlama sisteminin kullanımda olan iki versiyonu seçilmiştir. Şekil 4.1'de önerilen yöntemin tüm aşamaları, teknik açıklamaların işlenmesi, özniteliklerin çıkarılması, doküman vektörlerinin oluşturulması, sınıflandırma ve hesaplama adımları ayrıntılarıyla gösterilmektedir.

Önerilen metodolojideki sınıflandırma algoritmaları, geleneksel istatistiki yöntemler BoW, TF-IDF, 2-3 Ngram ve derin öğrenme temelli kelime gömme yöntemleri Word2Vec, Doc2Vec, FastText vektörleri ile oluşturulan eğitim setleri ile ayrı ayrı eğitilmektedir. Bunun sonucu olarak CVSS 2.0 için altı ve CVSS 3.1 için ise sekiz tane güvenlik metrik değeri hesaplanmaktadır.

Yukarıda bahsedilen metodoloji açık kaynaklı bir programlama dili olan Python [53] kullanılarak gerçekleştirilmiştir. Bunun nedeni geniş bir bilimsel topluluk tarafından kabul görmüş olması ve ek kütüphaneleri aracılığı ile hızla gelişen yapısıdır. Mevcut metodolojiyi uygulamak için Python paketlerinden, metin analizi için Pandas [54,55], Numpy [56], NLTK [57], Gensim [58] ve sınıflandırma için ise Scikit Learn [59] kullanılmıştır.



Şekil 0.4. Önerilen metodolojinin genel yapısı

4.1. Veri Tabanı

Bu tez çalışmasında, ayrıntıları Bölüm 2'de açıklanan veri tabanlarından NVD kullanılmıştır. NVD güvenlik açığı verilerini en güvenilir şekilde araştırmacıların kullanımına sunan veri tabanıdır [60]. NVD veri tabanında yayınlanan raporlarda iki farklı skorlama sistemi bulunmaktadır. Bu sistemler temelde bir birinin devamı gibi görünse de kullandıkları şemalar ve güvenlik vektörleri farklıdır. Çalışmamızda iki şemada kullanılmıştır. Kullanılan şemalar 3. Bölümde ayrıntılarıyla anlatılmıştır.

Tablo 4.1'de NVD veri tabanı güvenlik raporlarının bir örneği ve özellikleri görülmektedir. Rapor incelendiğinde iki farklı versiyon CVE-ID ve Description değerlerinin aynı olduğu görülmektedir. Ancak diğer değerler farklılık göstermektedir. Bunun nedeni skorlama sistemin versiyonlarının farklı güvenlik şemaları kullanmalarıdır. CVSS 2.0 için güvenlik vektörü 6 alt metrikten oluşurken bu değer CVSS 3.1 için 8 alt metrik olarak karşımıza çıkmaktadır. Bu metriklerin bazıları birbirine benzese de alabildikleri değerler farklıdır.

Tablo 0.5. Örnek güvenlik açığı raporu

	CVSS 3.1	**CVSS2.0**	
CVE-ID	CVE-2020-11998		
Description	A regression has been introduced in the commit preventing JMX re-bind. By passing an empty environment map to RMIConnectorServer, instead of the map that contains the authentication credentials, it leaves ActiveMQ open to the following attack: https://docs.oracle.com/javase/8/docs/technotes/guides/management/agent.html "A remote client could create a javax.management.loading.MLet MBean and use it to create new MBeans from arbitrary URLs, at least if there is no security manager. In other words, a rogue remote client could make your Java application execute arbitrary code." Mitigation: Upgrade to Apache ActiveMQ 5.15.13		
Attack Complexity	LOW	**Access Vector**	NETWORK
Attack Vector	NETWORK	**Access Complexity**	LOW
Privileges Required	NONE	**Authentication**	NONE
User Interaction	NONE	NaN	
Scope	UNCHANGED	Nan	
Confidentiality Impact	HIGH	PARTIAL	
Integrity Impact	HIGH	PARTIAL	
Availability Impact	HIGH	PARTIAL	
Exploitability Score	3.9	10	
Impact Score	5.9	6.4	
Base Score	9.8	7.5	
Severity	CRITICAL	HIGH	
Vector String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	AV:N/AC:L/Au:N/C:P/I:P/A:P	

Tez çalışması kapsamında yapılan analizler için 3 farklı zamanda elde edilen ve literatürde kullanılan veri setlerinin kullanımı tercih edilmiştir. İlk veri kümesi veri tabanının ilk üretildiği tarihten 31.12.2020 tarihine kadar yayınlanmış yaklaşık 158.059 güvenlik açığını içermektedir. Tüm kayıtlarda güvenlik açığını ifade eden teknik açıklama yer alsa da tamamında resmi güvenlik vektör değerleri ve önem skorları yer almamaktadır. Mevcut kayıtlardan CVSS 2.0 vektörlerine sahip toplam kayıt sayısı 147.275'tir. CVSS 3.1 vektör değerlerine sahip kayıt sayısı ise 73.907'dir.

Bundan dolayı metodolojimizi uygularken resmi değerlere sahip olmayan veriler çıkarılmıştır. İlk veri setinde modelimizin eğitimi için CVSS 2.0 verilerinden 103.092 adeti eğitim ve doğrulama için, 44.183 adeti test için kullanılmıştır. CVSS 3.1 modelinin eğitimi için ise 51.735 adet veri kullanılırken, test için 22.172 adet veri kullanılmıştır.

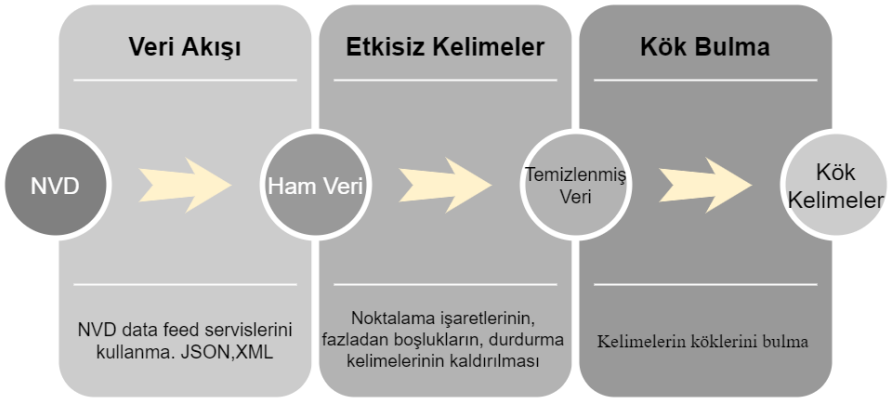
Bu tez çalışması kapsamında hazırlanan ikinci veri seti için 31.10.2021 tarihine kadar yayınlanmış 173,0241 kayıt seçilmiştir. Yapılan analizlerde bazı raporların önem skorlarına sahip olmadığı tespit edilmiştir. Doğrulama yapılamayacağı için resmi değerlere sahip olmayan raporlar veri setinin dışında bırakılmıştır. Bunun sonucunda CVSS 2.0 değerlerine sahip 163,229 kayıt, CVSS 3.1 değerlerine sahip 89,868 kayıt veri seti içerisine alınmıştır. İki skorlama sistemi için modeller ayrı ayrı uygulanmıştır. Bunun için veri seti, eğitim-doğrulama için %70 ve test için %30 oranında ayrılmıştır. Eğitim-doğrulama ve test setleri CVSS 2.0 için 114,260 – 48,967 kayıttan oluşmuştur. CVSS 3.1 için ise 53,907 eğitim-doğrulama için, 26,961 kayıt test için ayrılmıştır. Çalışmamızda ayrıca Spanos vd.[9] oluşturdukları veri seti de analizleri karşılaştırmak için kullanılmıştır. Tablo 4.2'de kullanılan veri setlerinin karşılaştırması görülmektedir.

Tablo 0.6. Kullanılan Veri Tabanlarının Karşılaştırılması

Veri Seti	**CVSS 2.0**			**CVSS 3.1**		
	Eğitim	**Test**	**Toplam**	**Eğitim**	**Test**	**Toplam**
VS 2	114,260	48,967	163,229	62,907	26,961	89,868
VS 1 [61]	103,092	44,183	147,275	51,735	22,172	73,907
Spanos vd. [9]	75,156	23,935	99,091	-	-	-

4.2. Ön İşlemler

Genel olarak yapılan ilk işlem NVD veri tabanının veri beslememelerinden elde edilen verilerin düzenlenmesi ile başlar. NVD verileri JSON dosya formatında sunulmaktadır. Ancak raporlar yıllık bazda dosyalar halindedir. Bunlardan tek bir veri seti oluşturacak şekilde dosyalar bütünleştirilerek kullanılacak veri seti oluşturulmuştur. Bu sürecin sonunda verilerin ön işleme aşamalarına geçilmiştir. Şekil 4.2'de ön işleme adımları gösterilmiştir. Ön işleme aşaması da diğer aşamalar kadar önemlidir. Uysal ve ark. [62] çalışmalarında ön işleme aşamasının sınıflandırmaya olan etkisini incelemişlerdir. Sonuç olarak ön işleme, özellik seçimi ve sınıflandırma kadar önemlidir. Ön işleme adımları noktalama işaretlerinin, fazladan boşlukların, durdurma kelimelerinin kaldırılması işlemi ile başlamaktadır. Sonraki işlem kelimelerin köklerini bulma işlemidir. Bu işlemler sonucunda metinler üzerinden özellik çıkarımı gerçekleştirilebilecek yapılar elde edilmiştir.



Şekil 0.5. Ön işleme adımları

4.2.1. NVD Veri Beslemeleri

NVD veri tabanı sunduğu tüm verileri www.w3.org standartlarına uygun JSON ve XML formatında veri beslemeleri sunmaktadır. Ancak güvenlik açıkları raporları için artık sadece JSON veri formatını kullanmaktadır. Ayrıca bazı verilerin anlık sunumu için RSS yayınları da mevcuttur. Güvenlik açığı raporları yıllık dosyalar halinde verilmektedir. Bu çalışma kapsamında NVD'nin sağladığı JSON verilerini indirecek bir python scripti oluşturulmuştur. İndirilen yıllık rapor dosyaları formatlı bir şekilde birleştirilerek CSV dosyası olarak yeniden oluşturulmaktadır. Tüm verileri içeren CSV dosyalarında bulunan teknik açıklamalar üzerinde ön işleme aşamasının ikinci adımı olan stop words ve diğer anlam içermeyen işaretlerin kaldırılmasına geçilmektedir.

4.2.2. Stop Words

Doğal dil işlemede metin verisinin temel işlemlerinden biri olan ve stop words olarak isimlendirilen bu aşamada metin içerisinde bulunan noktalama işaretleri, fazladan eklenmiş boşluklar, versiyon numaralandırmaları ve kelime olarak her hangi bir tanımlama içermeyen bağlaç, zamir ve belirteç gibi Stop Words ifadelerin metinden temizlenmesi gerekmektedir. Bu işlemi yerine getirebilmek için ilk olarak düzenli ifadeler (regular expression) kullanılmıştır. Stop Words ifadelerin kaldırılması için ise hazır kütüphanelerden yararlanılmıştır [57].

4.2.3. Kök Bulma

Bu işlem lemmatization olarak adlandırılmaktadır. Bir kelime farklı formlara sahip olabilir. Lemmatization işlemi ile farklı gibi görünen aynı köke sahip kelimeler tek bir öğe olarak analiz edilebilir [63]. Bu işlem düzgün gerçekleştirilemez ise aynı kök kelimeden türetilmiş kelimeler farklı kelimeler olarak algılanacaktır. Böyle bir durum veri setinde aynı şeyi ifade eden birden çok kelimenin bulunması durumuna neden olacaktır. Çalışmamızın bu aşamasının doğru bir şekilde yapılması çok önemlidir.

4.3. Metin Vektörleştirme

İkinci aşama metinleri oluşturan temizlenmiş verilerden sınıflandırma algoritmalarında kullanılabilecek özellik vektörlerinin hazırlanmasıdır. Metin analizi üzerine uzun zamandır araştırmacılar çalışmaktadır. Bunun sonucunda pek çok farklı öznitelik çıkarım tekniği önerilmiştir. Önerilen yöntemler iki kategoriye ayrılmaktadır. Bunlar istatistiki ve derin öğrenme tabanlı kelime gömme (Word Embedding) yaklaşımlardır [64]. Bu çalışma kapsamında kullanılmak üzere geleneksel istatistiki yaklaşımlardan Bag of Words (BoW) [41], Term Frequency Inverse Document Frequency (TF-IDF) [42] ve Ngram [43] seçilmiştir. Bu yöntemler pek çok problemde başarı ile kullanılmıştır. İstatistiki yöntemler, yalnızca kelime frekansına odaklanmaktadır. Bu durum kelimeler arasındaki anlamsal ilişkilerin kaybolmasına neden olmaktadır. Diğer bir eksikleri ise yüksek başarı elde etmek için gerekli işlemlerin yüksek mühendislik becerileri gerektirmesidir. Ancak bu eksikliklerine karşın istatiksel modeller makine öğrenimi temelli sınıflandırma modellerinde karşılaştırılabilir performans elde etmek için çok kullanışlıdır [40].

Metin tabanlı verilerden öznitelik çıkarımında, kelime frekansına odaklanan geleneksel istatistiki yöntemler kullanılmaya devam etmektedir. Ancak bu yöntemler ile birlikte son yıllarda derin öğrenme tabanlı kelimeler arasındaki anlamsal ilişkilere odaklanan metin sınıflandırma yöntemlerinden kelime gömme algoritmalarının kullanımı giderek artmaktadır [64][65]. Bu nedenle önerilen metodolojide metin vektörlerinin oluşturulması için öznitelik çıkarılması aşamasında kelime gömme algoritmalarından Word2Vec [44,45], Doc2Vec [46] ve FastText [47] kullanılmıştır. Modelimiz çok sınıflı bir sınıflandırma işlemi gerçekleştirecektir. Bununla birlikte farklı sınıflandırma algoritmaları kullanarak sınıflandırma için en başarılı özellik şeçimi ve algoritmanın hibrit bir modelini kullanacağımız için karşılaştırılabilir yöntemleri kullanmak önemlidir.

4.3.1. Bag of Word

Bu modelde doküman kelimelerin frekansından oluşan bir vektör olarak temsil edilir. Metinlerin özniteliklerini çıkarırken kullanılan yaygın bir yöntemdir [41]. En temel istatistiki metin vektörleştirme yöntemidir. Bir corpus içeresindeki her bir kelimeyi içeren bir kelime tablosu oluşturur. Daha sonra corpus içerisindeki dokümanların kelime tablosunda bulunan kelimelerinin sayar ve kaç adet olduklarını o tabloda dokümanın satırına yazar. Corpusta yer alan ancak doküman içerisinde yer almayan kelimelere 0 değerini verilir. Bir corpus içerisinde bulunan kelime sayısının oldukça fazla olabileceği düşünüldüğünde oluşturulan metin vektörlerinin büyük bir seyrek matris (sparse matrix) oluşturacağı açıktır. Bu durumun önüne geçebilmek için en sık kullanılan kelimelerden oluşan vektörler tercih edilmektedir [66]. Ayrıca stop words ve aynı kök değerine sahip kelimelerin farklı kelimeler olarak listelenmemesi için veri ön işleme adımlarının özenle

yapılması önemlidir. BoW modelinin diğer bir dezavantajı corpus içerisinde çok fazla geçen kelimelerin ayırt ediciliğinin az olmasına rağmen vektörde yüksek değerlere sahip olmasıdır.

4.3.2. Term Frequency Inverse Document Frequency

BoW modelinde her metin için kelimelerin frekansı ile vektör oluşturulurken, tf-idf'de bir kelimenin doküman ve tüm dokümanların birleşiminden oluşan, çoğu zaman belirli bir alana özgü olan külliyat (corpus) [67] içerisindeki önem derecesinin hesaplaması ile vektör oluşturmaktadır. Bunun için Terimin Geçtiği Doküman Sayısı / Doküman Sayısının logaritmasının mutlak değeri alınmaktadır [42]. Bu sayede bir kelimenin külliyat içerisinde çok fazla bulunması yani dokümanların çoğunda bulunan kelimelerin ayırt ediciliği az olanlarının düşük değerler alması ve ayırt ediciliği çok olan kelimelerin değerinin yüksek olması sağlanmaktadır [42].

Tf-idf vektörleri oluşturulurken öncelikle terim frekansları bulunmaktadır [68]. Bu işlem gerçekleştirilirken farklı yöntemler kullanılabilmektedir. Genel olarak temel yöntem dokümanda bulunan bir kelimenin o dokümanın toplam kelime sayına bölünmesi ile hesaplanır. Yani, bir kelimenin belgedeki görünme sıklığını ölçer. Ancak binary frekans, bir terimin, dokümandaki geçme sayısına bölünmesi ile hesaplanan raw frekansı, log normalizasyonu, double 0.5 ve double K normalizasyon yöntemleri ile de terim frekansları hesaplanabilmektedir [69]. Tf terim frekansı hesaplandıktan sonra idf yani ters belge frekansı (Inverse Document Frequency) hesaplanmalıdır [70]. Idf hesaplamak için Dokuman Sayısı / Terimin Geçtiği Dokuman Sayısının logaritmasının mutlak değeri alınmaktadır. Bu işlemin sonucunda eğer bir kelime her dokümanda geçiyorsa idf sonucu 0 olarak hesaplanacaktır. Eğer bir kelime sadece bir dokümanda geçiyorsa idf değeri 1 olarak hesaplanacaktır.

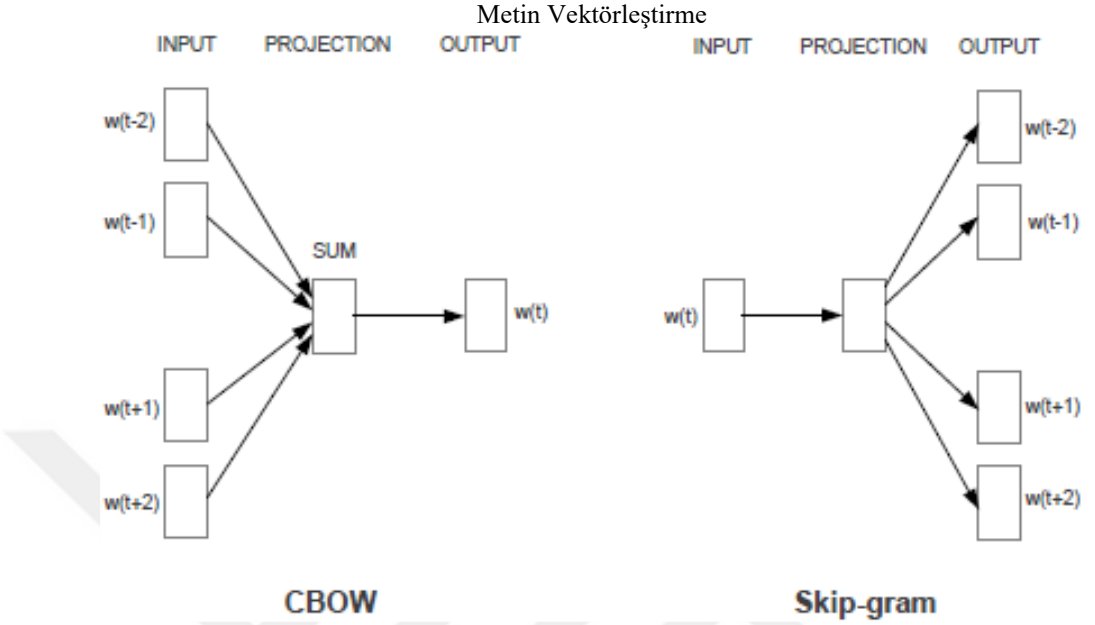
4.3.3. Ngram

BoW modeline benzer ancak kelimeleri ikili, üçlü veya daha fazla gruplar halinde seçerek saymaktadır. N değeri ile farklı kelime gruplarına sahip metin vektörleri oluşturulabilir. N değeri bir seçilirse oluşturulan model BoW modeli ile aynı olacaktır [43]. N değeri 1 seçilirse unigram, N değeri 2 ise bigram (digram olarak ta isimlendirilir),N değeri 3 ise trigram ve N değeri daha büyük ise (genellikle) ngram olarak isimlendirilmektedir [71].

4.3.4. Word2Vec

Mikolov vd [44,45] tarafından önerilen, bir külliyat içerisindeki kelimeleri vektör uzayında temsil etmeye yarayan denetimsiz (unsupervised) ve tahmin temelli bir mimarı ve optimizasyon modelidir. Yazarlar yöntemlerini gerçekleştirmek için iki farklı alt yöntem önermektedir. Temelde benzer olan bu yöntemler; CBOW(Continous Bag of Words) ve Skip-Gram olarak

isimlendirilmektir. CBOW ve Skip-Gram modellerinin temel farklı input ve output katmanlarının farklı olmasıdır. Şekil 4.3'te Word2Vec mimarisi gösterilmektedir.



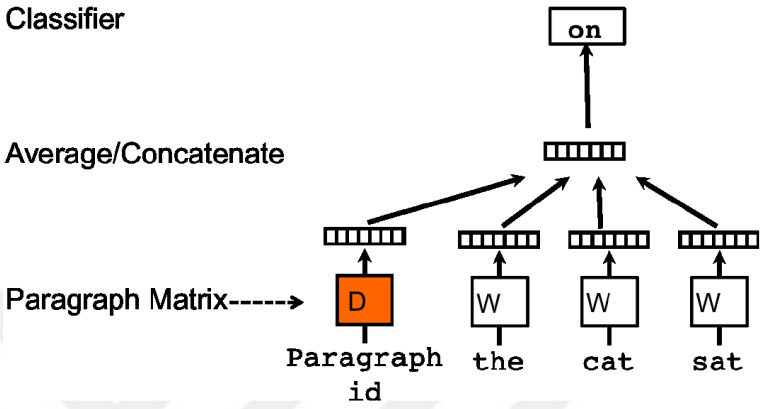
Şekil 0.6. Word2Vec mimarisi [45]

Algoritmanın en temel parametresi Window size'dir. Window Size parametresi hesaplanacak kelimenin sağından ve solundan kaç kelimenin eğitim için kullanılacağını belirtmektedir. CBOW yönteminde Window size parametresinin merkezi dışındaki kelimeler input olarak alınır. Skip-Gram yönteminde ise window size parametresinin ortasında yer alan kelime input olarak alınmaktadır. Output olarak ise CBOW'da merkezdeki kelime tahmin edilmeye çalışılırken, Skip-Gram'da merkezde olmayan kelimeler tahmin edilmeye çalışılmaktadır. Bu işlem tüm dokümandaki kelimeler bitene kadar devam etmektedir. Arka planda derin sinir ağlarını kullanarak sözcükler arasındaki ilişkiyi öğrenmeye odaklanmıştır. Her kelime için farklı vektörler oluşturmaktadır. Oluşturulan vektörler diğer kelimelere olan benzerliklere göre belirlenmiş bir vektör uzayıdır. Model eğitildikten sonra eş anlamlı kelimeler ve kelimeler arasındaki anlamsal ilişkiler belirlenebilmektedir [44,45].

4.3.5. Doc2Vec

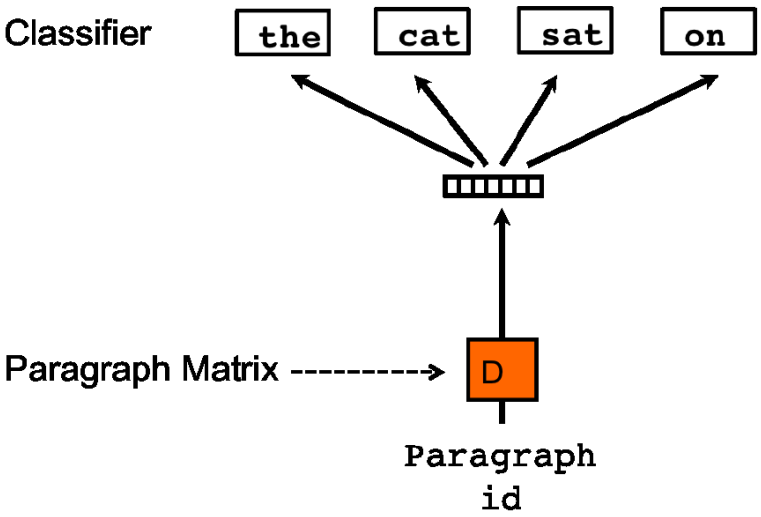
Word2Vec'in doğası gereği belge vektörleri değil sadece kelime vektörleri oluşturulabilmektedir. Belgeler içerisindeki cümleler, kelimelerin aksine belirli bir anlamsal bağlantı içermez. Ancak Word2Vec modeline yeni bir katman eklenerek basit ve akıllıca bir çözüm üretilmiştir. CBOW ve Skip-Gram yapılarını kullanan ve kelimeler dışında belgelere özgü özellik vektörlerinin de eklendiği Doc2Vec yapısı, bir belgenin uzunluğundan bağımsız olarak vektörel bir

temsilinin oluşturulması için önerilmiştir. Word2Vec'in iki farklı yöntemini de kullanabilir. CBOW temelli yaklaşım, Distributed Memory Model of Paragraph Vectors (PV-DM) olarak ve Skip-Gram tabanlı yaklaşım ise Distributed Bag of Words version of Paragraph Vector (PV-DBOW) olarak adlandırılır [46]. Şekil 4.4 ve Şekil 4.5'te PV-DM ve PV-DBOW modeller gösterilmektedir.



Şekil 0.7. PV-DM modeli [46].

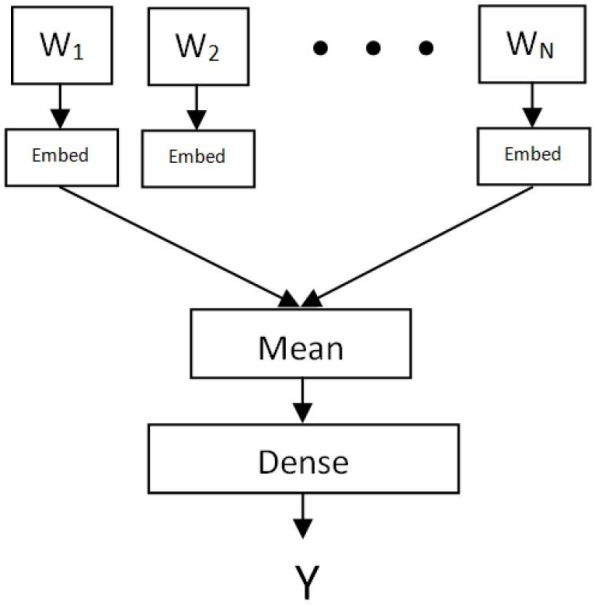
PV-DM ve PV-DBOW modelini gösteren Şekil 4.4 ve Şekil 4.5'ten de görülebileceği gibi bu modeller CBOW ve Skip-Gram yöntemlerine oldukça benzer algoritmalardır. Tek fark eğitim sonucunda bir doküman id değerinin elde edilmesidir. Ayrıca Doc2Vec yönteminde kelime vektörlerini hafızada tutmak gerekmediğinden daha az bellek ihtiyacı duyulmaktadır. Mikolov vd [46], çalışmalarında PV-DM modelinin sınıflandırma başarısının daha iyi olduğunu belirtmektedirler. Ancak sonuçların daha doğru olabilmesi için her iki yöntemi birlikte kullanan hibrit bir yöntemin kullanılmasını tavsiye etmektedirler.



Şekil 0.8. PV-DBOW model [46].

4.3.6. FastText

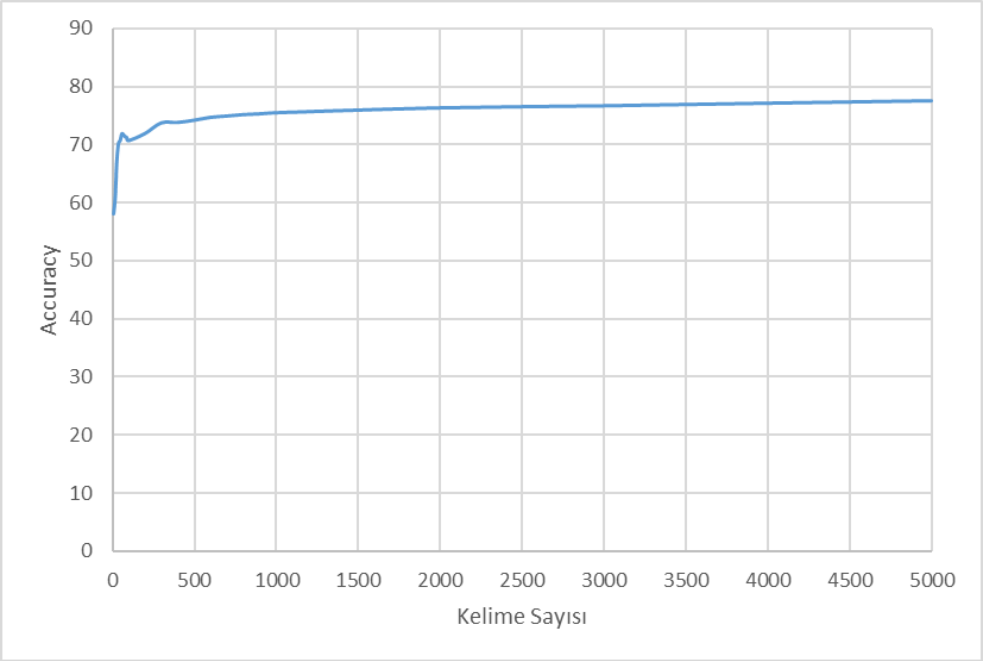
Word2Vec gibi sürekli eğitilmiş sözcük vektörlerini kullanan ve büyük boyutlu etiketlenmiş temsiller üreten modeller doğal dil işlemede oldukça başarılı olmuşlardır. Bu modellerin en büyük eksikleri kelimelerin morfolojisini ihmal etmeleridir. Bu durum özellikle nadir kelimelerin ve eş anlamlı ifadelerin çok olduğu diller için dezavantajlı bir durumdur. FastText yöntemi Skip-Gram yapısını kullanmaktadır. FastText'in kelimelerin morfolojine odaklanması diğer yöntemlerden ayrılmasını sağlamaktadır. Yöntem hızlıdır ve büyük verilerin eğitimine izin vermektedir. Morfolojik yapısı, yüksek performans elde etmesini sağlamaktadır [47]. Şekil 4.6'da FastText modeli gösterilmiştir.



Şekil 0.9. FastText modeli [72].

4.4. Öznitelik Şeçimi

Modelimizin vektörleştirme aşamasında bağımsız kelime sayısı 86.554 olarak bulunmuştur. Bu büyüklükte bir değerle işlem yapmak çok fazla kaynak gerektirmektedir. Yin ve ark. [73] çalışmalarında metin sınıflandırma için optimal kelime sayısını 300 olarak belirtmişlerdir. Bununla birlikte farklı kelime sayıları ile yaptığımız denemelerde 300 üzeri kelime sayılarının model performansı üzerinde yüksek bir etki oluşturmadığı gözlemlenmiştir. Şekil 4.7'de yapılan deneysel çalışmaların grafiği verilmiştir. Grafikten de anlaşılacağı üzere 300 kelime uzunluğuna sahip vektörler kullanmak, modelimizin performansını değerlendirmek için yeterlidir.



Şekil 0.10. Optimum kelime sayısı

Vektörleştirme işleminde, en çok kullanılan 300 kelime ile eğitim setinden değerler hesaplanmıştır. Hesaplanan bu değerler ile test verilerinin vektör değerleri oluşturulmuştur. Öznitelik çıkarımı ve seçimi süreçlerinden sonra yapılandırılmış bir veri seti hazırlanmıştır.

4.5. Çok Sınıflı Sınıflandırma Algoritmaları

Metodolojinin sınıflandırma aşaması için çoklu öğrenme desteğine sahip, yerleşik, stratejiye göre gruplandırılmış tahmin ediciler olan Naive Bayes (NB) [48], Desicion Tree (DT) [49], K-Nearest Neighbors (KNN) [50], Multi-layer Perceptron (MLP) [51] ve Random Forest (RF) [52] algoritmaları seçilmiştir. Sınıflandırma modelleri oluşturulurken Cross Validation yöntemi kullanılmıştır. Cross Validation parametresi olarak 10 değeri seçilmiştir. Cross Validation bağımsız veri setleri üzerinde modelin doğruluğunu denetleyen bir istatiksel analizdir [74]. Temel amacı bir sistemin pratikte doğruluk oranını kestirmektir. Bu işlemin amacı, modelin yeni verilere karşı hassasiyetini belirleyerek aşırı uyum (overfitting) veya yanlı seçim (selection bias) problemlerini tespit etmektir [75].

4.5.1. Naive Bayes

Günümüzdeki önemli sınıflandırma problemlerinden biri olan metin verileri giderek artmaktadır. Bu alanda en yaygın kullanılan sınıflandırma algoritmalarından biri Naive Bayes'dir. Farklı veri setleri ile birçok problemin çözümünde üstün performans göstermiştir [48]. Naive Bayes algoritması Bayes teoremine dayanan ve öz nitelikler arasındaki güçlü bağımsızlık varsayımlarına

dayanarak sınıf etiketlerini tahmin eden modeller oluşturmak için geliştirilmiştir [76]. Farklı geleneksel yaklaşımlarla uzun süredir rekabet ederek birçok alanda kullanılmıştır [77].

4.5.2. Desicion Tree

Desicion tree algoritması biyoinformatikten, metin sınıflandırmaya kadar pek çok farklı alanda etkili bir şekilde uygulanan istatiksel bir makine öğrenmesi algoritmasıdır [78]. Metin sınıflandırmada yaygın olarak kullanılan algoritmalardan biridir. Kurallar kök ve yapraklar arasında, özelliklerin otomatik olarak işlenmesi ile oluşturulur [79]. Karar ağaçları decision, chance ve end (uç) olmak üzere üç farklı tür düğümlerden oluşmaktadır. En optimum sonuca ulaşmak için yenilemeli bir işlem uygular. Sonuçlara göre farklı dallar oluşturan bir ağaç yapısına dönüşür. Ancak düşük boyutlu verilerde aşırı uyum problemi olabilmektedir. [80].

4.5.3. K-Nearest Neighbors

KNN algoritması yaygın olarak veri madenciliği ve istatistiki problemlerin çözümlerinde kullanılan, uygulaması kolay ve performansı yüksek bir yöntemdir [81]. Algoritma eğitim aşamasında verilen vektör uzayındaki k kadar en yakın komşu içerisinde hangi sınıfa ait olduğunu hesaplar. Aynı sınıfa ait elemanlar arasından yüksek benzerlik değerlerine sahip iken farklı sınıflar düşük benzerlik değerlerine sahip olurlar [82]. Algoritma basitlik ve doğruluk gibi avantajlara sahip olmasının yanında yüksek hesaplama maliyetlerine sahiptir [83]. Metin sınıflandırmada sıklıkla kullanılan algoritmalardan biridir [84].

4.5.4. Multi-layer Perceptron

Metin sınıflandırma algoritmaların da yaygın olarak kullanılan ileri beslemeli yapay sinir ağının bir alt sınıfı olan MLP, bir den fazla perceptron katmanından oluşan ağları ifade etmek için kullanılmaktadır. Bir MLP giriş, gizli ve çıktı olmak üzere en az üç katmandan oluşmaktadır [85]. Geri yayılım algoritmasını kullanan denetimli bir makine öğrenme algoritmasıdır [86]. Doğrusal olmayan aktivasyon fonksiyonlarını kullanan nöronlardan oluşmaktadır [87]. Doğrusal olmayan problemlerin çözümünde kullanılabilmektedirler [88].

4.5.5. Random Forest

Random forest çok sayıda karar ağacının bir araya gelmesi ile oluşur. Hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Karar ağaçlarında bulunan aşırı uyum problemini gidermektedir [89]. Birçok avantajından dolayı pek çok gelişmiş uygulamada random forest kullanılmıştır [90]. Denetimli bir algoritmadır. Problemin her olasılığı için rasgele karar

ağaçlarından oluşan bir orman oluşturur. Daha sonra oluşturulan karar ağaçları birleştirilerek doğru ve istikrarlı bir tahmin hesaplanır [91].

4.6. Doğrulama

Çoklu-sınıf sınıflandırma problemlerinde kullanılmış ve çok iyi bilinen aşağıdaki değerlendirme ölçütleri önerilen metodolojiyi değerlendirmek için kullanılmıştır [92].

Accuracy: Bir örnek kümesi için doğru tahmin edilen etiketlerin sayısının toplam veri kümesine oranını ifade eder.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{4.1}$$

Recall: Pozitif olarak tahmin etmemiz gereken işlemlerin ne kadarını pozitif olarak tahmin ettiğimizi gösteren bir metriktir.

$$Recall = \frac{TP}{TP+FN} \tag{4.2}$$

Precision: Pozitif olarak tahmin ettiğimiz değerlerin gerçekten kaç adetinin pozitif olduğunu göstermektedir.

$$Precision = \frac{TP}{TP+FP} \tag{4.3}$$

F1 Score: Precision ve Recall değerlerinin harmonik ortalamasını göstermektedir.

$$F_1 = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{4.4}$$

Accuracy ile karşılaştırıldığında precision, recall ve f1 score daha güvenilir ölçümlerdir. Ayrıca bu ölçüm kümesi çok sınıflı sınıflandırma modellerinin değerlendirilmesi üzerine çalışan diğer araştırmacıların önerileri ile de uygun düşmektedir [93].

Bununla birlikte modelin tahmin ettiği güvenlik açığı vektörlerinden hesaplanan önem puanlarını değerlendirme ve modelimizi doğrulamak için kullanılan ölçütler şunlardır [9].

- Mean Absolute Error (MAE): $\frac{1}{n}\sum_{i=1}^{n}|y_i - y'_i|$ (4.5)
- Median Absolute Error (MDAE): $median(|y_i - y'_i|)_{i=1}^{n}$ (4.6)

- Root Mean Square Error (RMSE): $\sqrt{\frac{\sum_{i=1}^{n}(y_i - y'_i)^2}{n}}$ (4.7)

MAE tahmin edilen değerler ile gerçek değerler arasındaki farkın mutlak değerlerinin ortalamasıdır. Düşük bir değer doğruluğu gösterir. MDAE değeri medyan değerlere dayandığı için aşırı hatalı değerlerden etkilenmez. RMSE hata değerlerinin karesi alınarak hesaplandığı için büyük hataların küçük hatalardan daha fazla etkisi vardır. Bu nedenle MAE aykırı değerlere karşı daha doğru doğru sonuçlar sağlar [94].

5. BULGULAR VE TARTIŞMA

Bu tez çalışmasında önerilen yöntemin sonuçlarının daha doğru analizi için modellerin sonuçlarını ayrıntılı bir şekilde sunuyoruz. Önerilen modellerin uygulanması sırasında veri setimizin %70'i eğitim ve %30'da test amaçlı kullanılmıştır. Dahası eğitim aşamasında Cross Validation tekniği kullanılarak modelin doğruluğu denetlenmiştir. Cross Validation değeri 10 olarak seçilmiştir. Sınıflandırma problemlerinin değerlendirmesinde ilk bakılan metriklerden biri accuracy'dir. Accuracy, doğru sınıflandırmaların toplam numune sayısı üzerindeki oranına karşılık gelmektedir. Multi-class classification problemlerinde sınıfların düzgün dağılmama durumları modelleri değerlendirirken baskın sınıfın performansı yönünde sonuçları etkileyebilir. Bu durum bizim problemimiz için de geçerlidir. Accuracy önemli bir gösterge olsa da çok sınıflı problemlerde tek başına yeterli olmayabilir [94]. Bu nedenle her alt sınıfın performanslarının ortalamaları ile oluşturulan macro Precision, Recall ve F1 Score ölçekleri ile de modelimiz değerlendirilmiştir. Değerler yüzdelik olarak sunulmuştur. Bulgular iki ayrı ana başlıkta sunulmuştur. İlk olarak CVSS güvenlik vektör metriklerinin tahmin sonuçları ayrıntılı bir şekilde verilmektedir. Sonrasında CVSS Severity Score değerlerinin tahmin ve hesaplama sonuçları sunulmuştur.

5.1. CVSS Güvenlik Vektörü Metriklerinin Tahmin Sonuçları

Tez çalışması kapsamında önerilen metodolojinin sonuçlarını doğru ve sağlıklı bir şekilde değerlendirebilmek için veri kümesi hakkında bazı istatistiki bilgiler faydalı olacaktır. Tezin amacı yaklaşık 165.000 NVD raporunun güvenlik açığı vektörlerinin tahmin edilmesi ve önem puanlarının hesaplanmasıdır. Bu amaçla oluşturulan veri seti Bölüm 4.1'de ayrıntılarıyla açıklanmıştır. CVSS skorlama sisteminin iki farklı versiyonu için bu işlem gerçekleştirilmiştir. CVSS 2.0 versiyonu için veri seti 147.275, CVSS 3.1 versiyonu için ise 73.907 kayıttan oluşmaktadır.

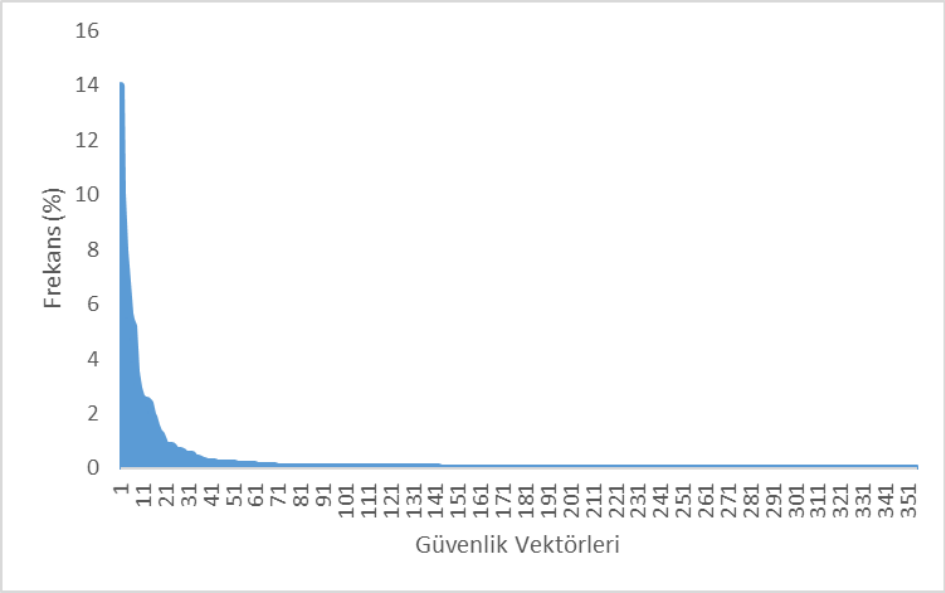
Spanos ve ark.[9] 2018 yılındaki çalışmalarında yaklaşık 100.000 veri ile CVSS 2.0 vektörlerini çok hedefli bir yaklaşımla tahmin etmeye çalışmıştır. İlgili çalışmalarında güvenlik vektörlerinin %0.01 ile % 16.27 arasında değişen frekansla yayıldığını ve en sık kullanılan 10 vektörün kümülatif toplamının neredeyse %70 olduğunu belirtmektedirler. Farklı güvenlik açığı vektör sayısının 729 olasılıktan 337 adet olduğu bilgisini paylaşmaktadırlar. Bu durumunda güvenlik vektörlerinin çoğunluğunun oldukça düşük frekans yüzdelerine sahip olduğunu açıklamışlarıdır. Bu sebeple bu vektörlerin tahmininin çok zor bir iş olduğunu vurgulamaktadırlar. İlgili çalışmanın yapıldığı zamandan bugüne güvenlik açığı verisi %70 artmıştır [61].

Önerilen metodoloji uygulanırken CVSS 2.0 için farklı vektör sayısı 355 adet olarak tespit edilmiştir. Ayrıca %1 ve üzeri yüzdelik dilime sadece ilk 20 vektörün girdiği tespit edilmiştir.

Ayrıca bu vektörlerin kümülatif toplamı yaklaşık %83 olmuştur. Şekil 5.1 ve Tablo 5.1'den anlaşılacağı üzere CVSS 2.0 vektörlerinin frekans dağılımları %0.01 ile %14.00 arasında değişen frekanslardadır ve tahmin için veri setinin daha da zor hale geldiğini göstermektedir [61].

Tablo 0.1. En sık kullanılan 10 CVSS 2.0 vektör dizisi.

CVSS 2.0 Vektörleri	Sayısı	Frekans
AV:N / AC:L / Au:N / C:P / I:P / A:P	20605	%14.00
AV:N / AC:M / Au:N / C:N / I:P / A:N	14757	%10.03
AV:N / AC:M / Au:N / C:P / I:P / A:P	11685	%7.94
AV:N / AC:L / Au:N / C:P / I:N / A:N	10042	%6.82
AV:N / AC:L / Au:N / C:C / I:C / A:C	8312	%5.65
AV:N / AC:L / Au:N / C:N / I:N / A:P	7964	%5.41
AV:N / AC:M / Au:N / C:C / I:C / A:C	7663	%5.21
AV:L / AC:L / Au:N / C:C / I:C / A:C	5194	%3.53
AV:N / AC:M / Au:S / C:N / I:P / A:N	4293	%2.92
AV:L / AC:L / Au:N / C:P / I:P / A:P	3820	%2.60



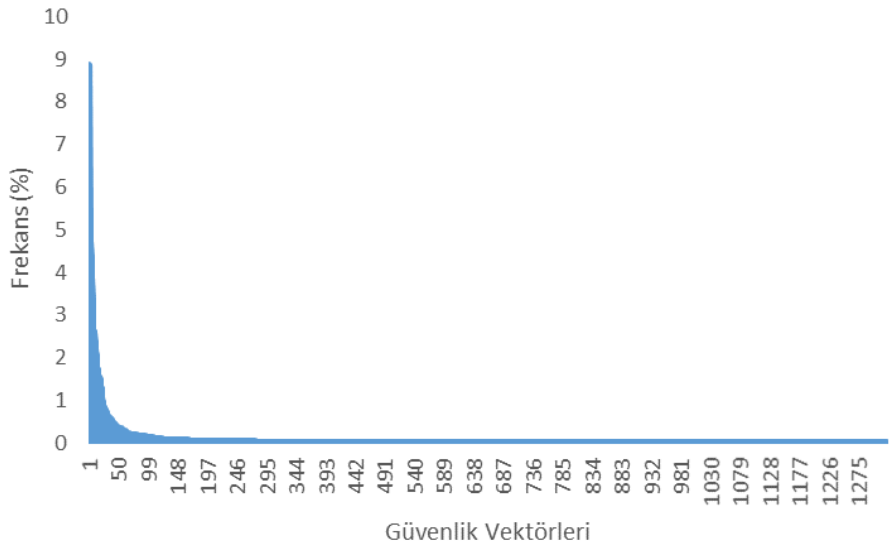
Şekil 0.1. CVSS 2.0 güvenlik açığı vektörlerinin dağılımı.

CVSS 3.1 vektörlerinin frekans dağılımları Şekil 5.2'de verilmiştir. Vektörlerin frekans dağılımı %0.01 ile %8.85 arasındadır. Tablo 5.2'de ilk 10 vektörün toplam sayısı ve frekansları görülmektedir. Buna göre farklı vektör sayısı 2.592 olasılıktan 1.322 tanedir. Bu vektörlerden sadece %1'lık dilime ilk 22 vektör girmektedir. Bu vektörlerin kümülatif toplamı ise yaklaşık %60'dır. Spanos ve ark. [9] çalışmalarını yaptıkları skorlama sisteminin sadece CVSS 2.0

versiyonu olduğu göz önüne alındığında 3.1 versiyonun değerlerinin tahminin çok daha zor bir iş olduğu yukardaki istatistiki bilgilerden anlaşılmaktadır [61].

Tablo 0.2. En sık kullanılan 10 CVSS 3.1 vektör dizisi.

CVSS 3.1 Vektörleri	Sayısı	Frekans
AV:N / AC:L / PR:N / UI:N / S:U / C:H / I:H / A:H	6539	%8.85
AV:N / AC:L / PR:N / UI:R / S:C / C:L / I:L / A:N	3714	%5.03
AV:N / AC:L / PR:N / UI:N / S:U / C:H / I:H / A:H	3474	%4.71
AV:N / AC:L / PR:N / UI:R / S:U / C:H / I:H / A:H	3190	%4.32
AV:L / AC:L / PR:N / UI:R / S:U / C:H / I:H / A:H	2850	%3.86
AV:L / AC:L / PR:L / UI:N / S:U / C:H / I:H / A:H	2462	%3.34
AV:N / AC:L / PR:N / UI:N / S:U / C:N / I:N / A:H	2330	%3.16
AV:N / AC:L / PR:N / UI:N / S:U / C:H / I:N / A:N	1957	%2.65
AV:L / AC:L / PR:L / UI:N / S:U / C:H / I:H / A:H	1930	%2.62
AV:N / AC:L / PR:N / UI:R / S:C / C:L / I:L / A:N	1897	%2.57



Şekil 0.2. CVSS 2.0 güvenlik açığı vektörlerinin dağılımı

5.1.1. CVSS 2.0 Yazılım Güvenlik Açığı Metriklerinin Sonuçları

Bu bölümde önerilen metodolojinin değerlendirilmesi için CVSS 2.0 skorlama sisteminin güvenlik vektörlerine ait metrikler için sınıflandırma sonuçları sunulmaktadır. Sınıflandırma, veri öğelerinin ait olduğu sınıfı belirtir. Bizim problemimizde güvenlik açığı vektörleri CVSS 2.0 için 6 metrikten oluşmaktadır. Her metrik ise Bölüm 3.1'de ayrıntıları ile anlatıldığı gibi 3 alt sınıfa ayrılmaktadır. Bu çok olasılıklı yapının kısıtlı bir teknik açıklamadan tahmin edilmesindeki zorluk ve nedenleri açıklanmıştır. Ancak kurulan farklı modeller ile elde edilen sonuçlar umut vericidir.

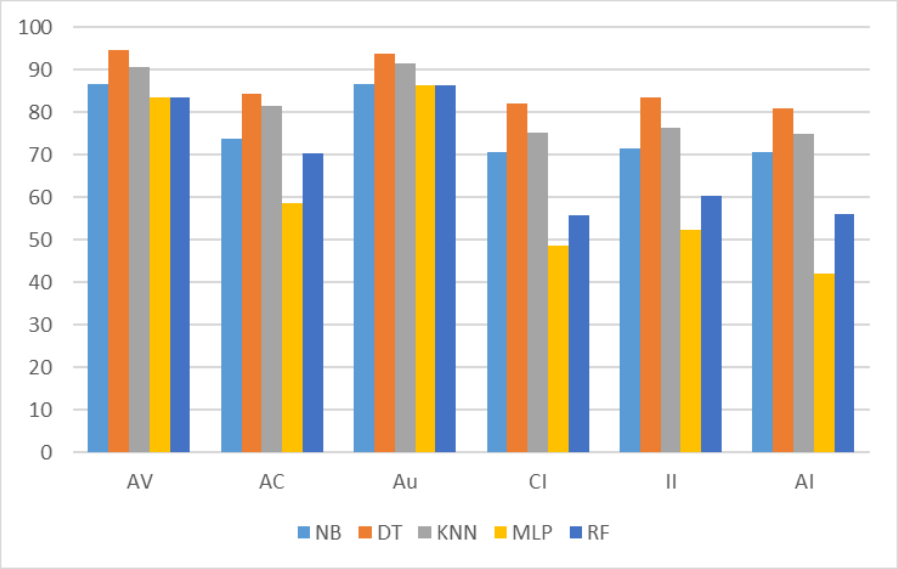
Tablo 0.3. En yüksek tahmin değerine sahip CVSS 2.0 modellerinin sonuçları

Metrik	Model	Alg.	Acc.	Prec.	Recall	F1
AV	BoW	DT	0.95	0.91	0.78	0.83
	TF-IDF	KNN	0.95	0.92	0.80	0.85
	2Ngram	MLP	0.94	0.92	0.74	0.81
	3Ngram	DT	0.92	0.92	0.67	0.75
	Word2Vec	**KNN**	**0.96**	**0.92**	**0.84**	**0.87**
	Doc2Vec	MLP	0.89	0.74	0.66	0.68
	FastText	MLP	0.95	0.89	0.80	0.81
AC	BoW	DT	0.84	0.84	0.66	0.71
	TF-IDF	KNN	0.88	0.86	0.71	0.76
	2Ngram	KNN	0.82	0.80	0.64	0.68
	3Ngram	MLP	0.76	0.78	0.56	0.60
	Word2Vec	**KNN**	**0.89**	**0.87**	**0.75**	**0.78**
	Doc2Vec	KNN	0.73	0.67	0.53	0.55
	FastText	KNN	0.85	0.82	0.63	0.67
Au	BoW	DT	0.94	0.66	0.56	0.59
	TF-IDF	KNN	0.94	0.83	0.61	0.67
	2Ngram	KNN	0.94	0.77	0.60	0.65
	3Ngram	KNN	0.92	0.82	0.53	0.58
	Word2Vec	**KNN**	**0.95**	**0.80**	**0.68**	**0.73**
	Doc2Vec	MLP	0.89	0.53	0.49	0.50
	FastText	MLP	0.92	0.64	0.74	0.68
CI	BoW	DT	0.82	0.82	0.79	0.80
	TF-IDF	KNN	0.85	0.85	0.84	0.84
	2Ngram	KNN	0.81	0.81	0.78	0.79
	3Ngram	MLP	0.74	0.77	0.69	0.71
	Word2Vec	**KNN**	**0.88**	**0.87**	**0.87**	**0.87**
	Doc2Vec	KNN	0.66	0.63	0.62	0.62
	FastText	KNN	0.82	0.80	0.78	0.79
II	BoW	DT	0.83	0.83	0.80	0.81
	TF-IDF	KNN	0.87	0.86	0.85	0.86
	2Ngram	KNN	0.82	0.81	0.79	0.80
	3Ngram	MLP	0.75	0.77	0.69	0.71
	Word2Vec	**KNN**	**0.89**	**0.88**	**0.88**	**0.88**
	Doc2Vec	MLP	0.68	0.65	0.64	0.64
	FastText	KNN	0.82	0.80	0.79	0.80
AI	BoW	DT	0.81	0.81	0.79	0.80
	TF-IDF	KNN	0.86	0.85	0.84	0.85
	2Ngram	KNN	0.78	0.78	0.77	0.77
	3Ngram	KNN	0.71	0.72	0.69	0.69
	Word2Vec	**KNN**	**0.87**	**0.87**	**0.87**	**0.87**
	Doc2Vec	KNN	0.64	0.63	0.61	0.62
	FastText	KNN	0.81	0.79	0.79	0.79

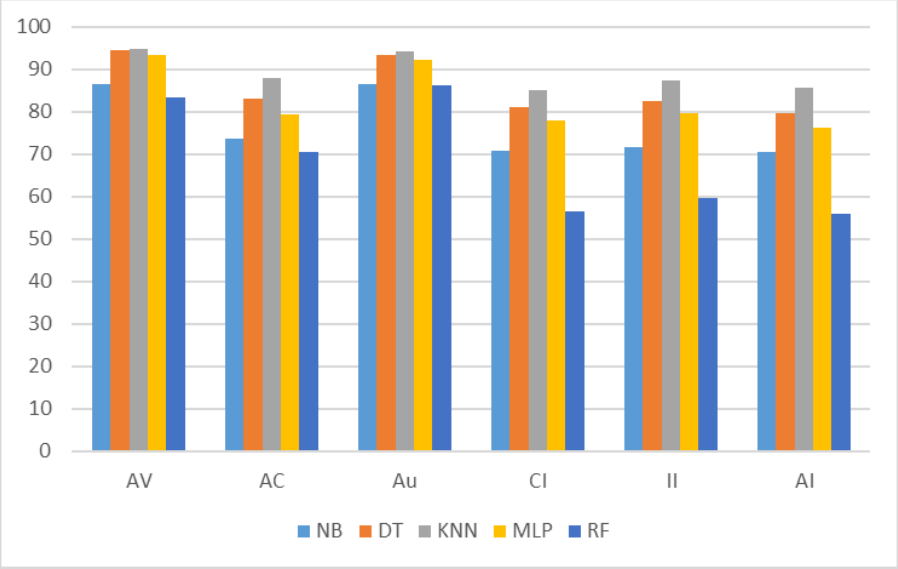
Tablo 5.3, tüm modellerin en iyi performanslarını göstermektedir. Sonuçlar Accuracy, Precisison, Recall ve F1 Score ölçeklerine göre değerlendirilmiştir. Tabloda belirtilen precision ve recall değerleri her sınıfın performansının ortalaması alınarak bulunan macro değerlerdir. Ayrıca F1-score değeri precision ve recall değerlerinin harmonik ortalaması ile hesaplanmaktadır. Tez çalışması içerisinde önerilen yöntem farklı özellik çıkarımı ve sınıflandırma algoritmalarından oluşmaktadır. Metin vektörleştirme aşamasında istatistiki ve derin öğrenme tabanlı Word Embedding olmak üzere altı farklı yöntem kullanılmıştır. Ayrıca sınıflandırma aşamasında beş

farklı çok sınıflı sınıflandırma yapan algoritma tercih edilmiştir. Tüm hesaplamalar sonucu elde edilen bulgularda model performansları karşılaştırılmıştır. Önerilen modelin genel bir değerlendirmesini yapmadan önce tüm modellerin performanslarını kendi içerisinde değerlendirmek daha doğru olacaktır.

Şekil 5.3'te BoW özellik çıkarım tekniğini kullandığımız modele ait sınıflandırma performansı görülmektedir. Görüldüğü üzere Decision-Tree sınıflandırma algoritması diğerlerinden daha iyi sonuç üretmiştir. Bu modelde KNN algoritması da Decision-Tree'ye benzer performans göstererek diğer algoritmalardan ayrılmaktadır. MLP algoritması bu modelde diğer algoritmaların performansından çok geride kalmıştır.



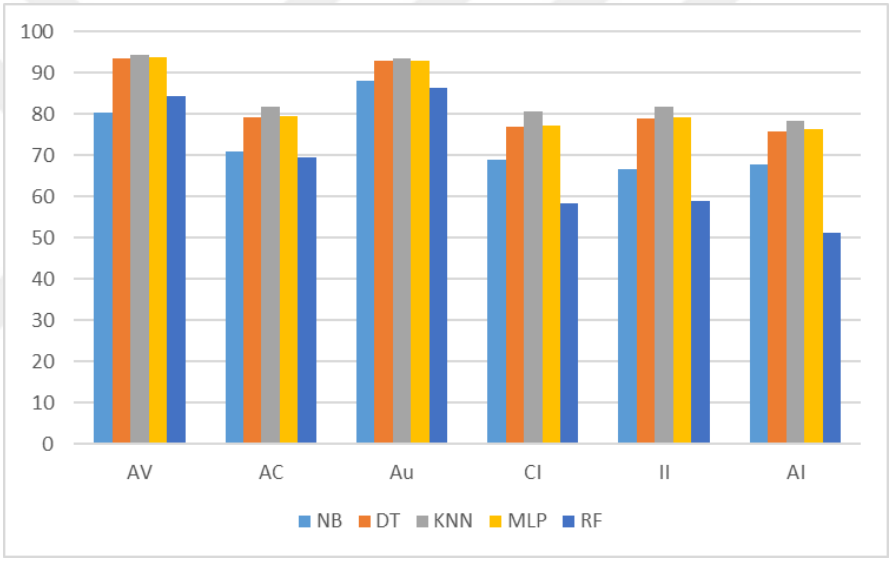
Şekil 0.3. BoW modelinin CVSS 2.0 tahmin sonuçları



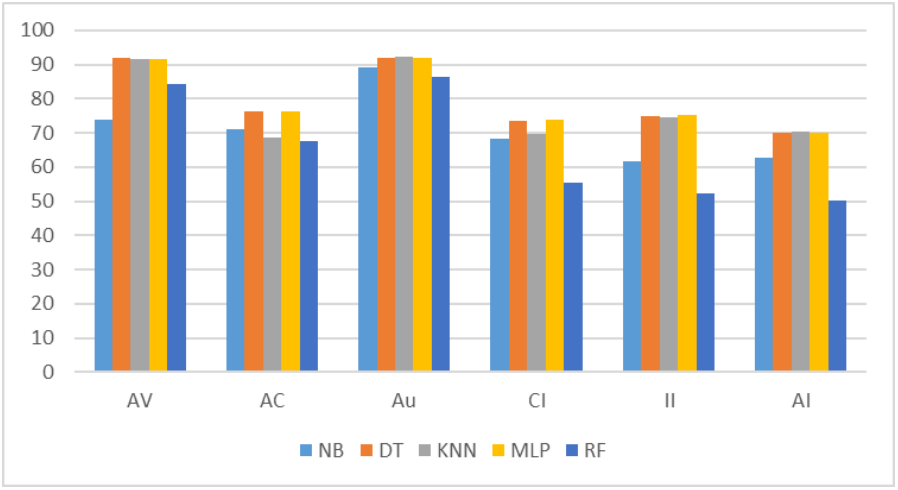
Şekil 0.4. TF-IDF modelinin CVSS 2.0 tahmin sonuçları

TF-IDF özellik çıkarım tekniğini kullandığımız modelin performans değerleri Şekil 5.4'te sunulmuştur. Görüleceği üzere KNN algoritmasının başarısı bu model için oldukça yüksektir. Diğer başarılı algoritmalar sırayla Decision-Tree ve MLP olarak görülmektedir. Ancak 6 vektörün tamamında KNN algoritmasının bariz bir şekilde diğer seçeneklerden üstün olduğu anlaşılmaktadır. Random Forest algoritması bu modelin en başarısız sınıflandırıcısı olmuştur.

2-Ngram özellik çıkarım tekniğini kullandığımız modelin ürettiği tahmin değerlerinin başarı oranları Şekil 5.5'de sunulmuştur. Beş farklı sınıflandırma algoritmasına bakıldığında KNN, MLP ve Decision-Tree algoritmalarının diğerlerinden başarı olarak ayrıldığı görülmektedir. Bu algoritmalar içerisinde KNN'nin başarısının daha iyi olduğu da gözlemlenmektedir. Bu model içinde random forest algoritması Access Vector metriği haricinde başarı oranı en düşük algoritma olmuştur.

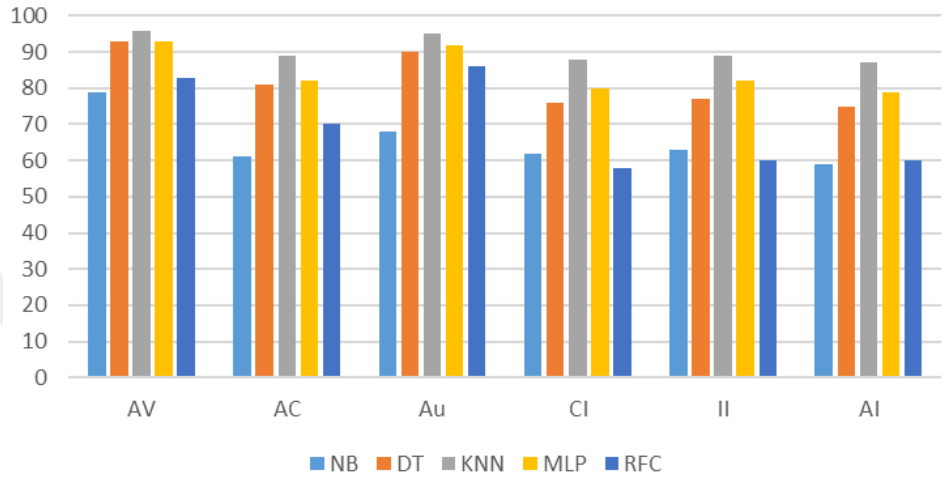


Şekil 0.5. 2-Ngram modelinin CVSS 2.0 tahmin sonuçları



Şekil 0.6. 3-Ngram modelinin CVSS 2.0 tahmin sonuçları

3-Ngram özellik çıkarım tekniğini kullandığımız modelin sonuçlarının sunulduğu Şekil 5.6 incelendiğinde KNN, Decision-Tree ve MLP'nin benzer başarılar ile öne çıktığı görülmektedir. Ancak Access Complexity ve Confidentiality Impact değerlerinde KNN algoritmasının sonuçları diğer iki algoritmadan geride kalmıştır. Ayrıca Decision-Tree algoritmasının Access Vector tahmin değeri diğer algoritmaların önündedir. 2-Ngram modelinde de olduğu gibi Random Forest tek metrik dışında en düşük sınıflandırma performansına sahip algoritma olarak görülmektedir.

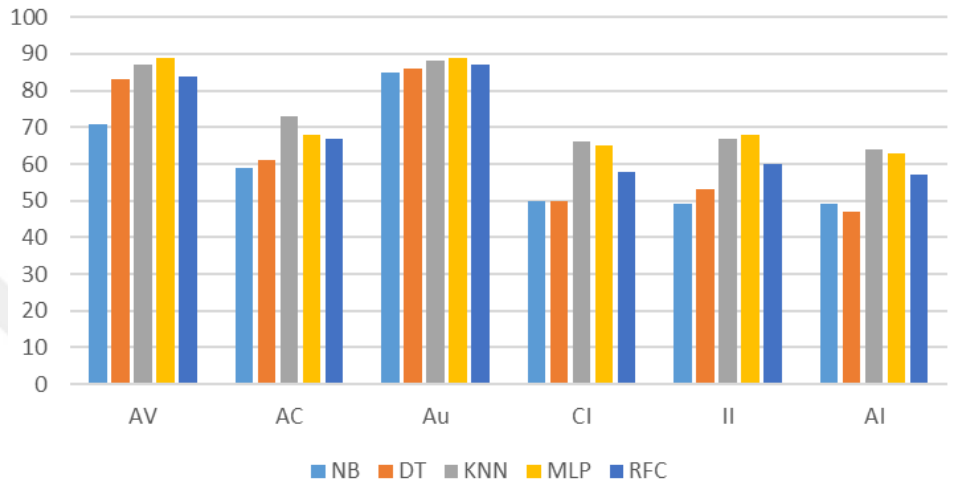


Şekil 0.7. Word2Vec modelinin CVSS 2.0 tahmin sonuçları

Şekil 5.7'de Word2Vec kelime gömme algoritması ile elde edilen sınıflandırma sonuçları görülmektedir. Bilindiği üzere Word2Vec kelimeler arasıdaki benzerlik oranlarını hesaplamaktadır. Ancak bizim problemimiz kelimerlerle değil tam bir metin ile çalışmaktadır. Bu nedenle her bir dökümanın vektör değerini hesaplamak için Word2Vec kelime benzerliklerinin ortalamaları alınarak doküman vektörleri oluşturulmuştur. Sonuçlar incelediğinde Word2Vec yöntemi ile elde edilen sonuçların oldukça başarı değerler olduğu görülmektedir. özellikle KNN algoritması ile elde edilen değerler tüm metriklerde en yükse başarı değerini işaret etmektedir. MLP ve DT algoritmaları başarılı sayılabilecek diğer algoritmalar olarak görülmektedir. NB ve RFC en düşük sınıflandırma performansı gösteren algorimalar olmuştur.

Doc2Vec doğrudan bir dokümanın sayısal vektörünü çıkarmak için geliştirilmiş bir yöntemdir. Veri setimizdeki her bir güvenlik açığına ait teknik açıklama bir doküman kabul edilerek vektörleştirilmiştir. Elde edilen sonuçlar ile sınıflandırma algoritmaları ile tahminler gerçekleştirilmiştir. Şekil 5.8 incelendiğinde Doc2Vec yöntemi ile en başarılı sonuçlar üreten algoritmaların MLP ve KNN algoritmaları olduğu görülmektedir. Bu yöntemde RFC algoritmasının da başarısının artığı görülmektedir. Ancak NB ve DT algoritmaları bu modelde en düşük performansı göstermiştir.

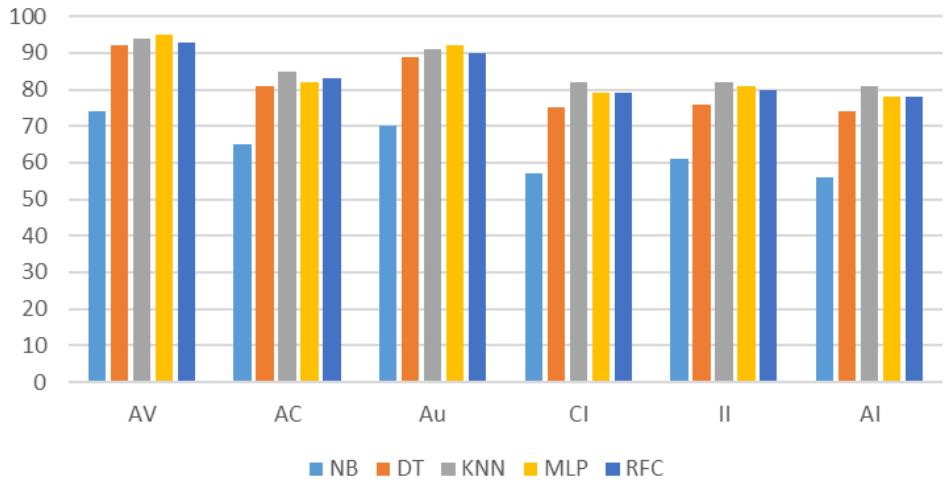
Şekil 5.9 FastText ile eğitilen modellerin performanslarını göstermektedir. Sonuçlar incelendiğinde tüm metriklerin dördünde MLP, ikisinde ise KNN algoritmasının en yüksek başarı değeri gösterdiği açıktır. Ayrıca RFC algoritmasının başarısı diğer yöntemlere göre FastText algoritması ile gözle görünür bir artış göstermiştir. DT algoritması bu algoritmaların gerisinde kalsa da tüm metrik değerleri için oldukça başarılı sonuçlar üretmiştir. NB algoritması bu modelimizde en başarısız algoritma olarak görünmektedir.



Şekil 0.8. Doc2Vec modelinin CVSS 2.0 tahmin sonuçları

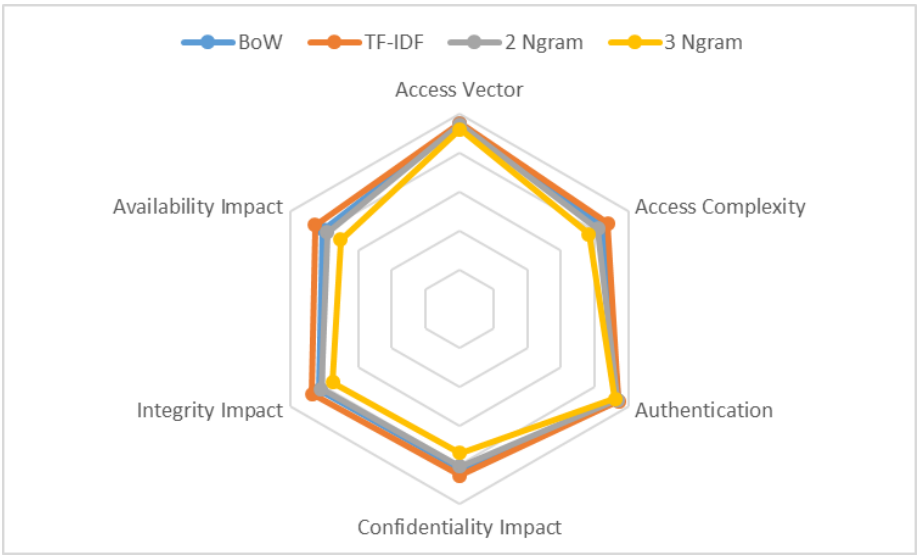
Tablo 5.3 tüm modellerin en yüksek tahmin değerlerini göstermektedir. Önerilen yöntem temelde yazılım güvenlik açıklarının sunulduğu güvenlik raporlarından, güvenlik açığı vektörlerine ait metrikleri tahmin etmektedir. Bu işlemi yerine getirirken raporlarda bulunan teknik açıklamaları kullanmaktadır. Bu aşama teknik açıklamalardan öznitelik çıkarımı için iki ana kategoriye ayırabileceğimiz altı farklı algoritma kullanılmaktadır. Ana kategoriler istatistiki ve kelime gömme olarak isimlendirilebilir. İstatistiki yöntemler Bow, TF-IDF ve Ngram'dır. Kelime gömme yaklaşımlarından ise Word2Vec, Doc2Vec ve FastText algoritmalarıdır. Sınıflandırma aşamasında ise tüm yöntemler beş farklı çok sınıflı sınıflandırma algoritması kullanılmıştır. Kullanılan tüm yöntemlerin bulguları ayrıntılı bir şekilde yukarıda sunulmuştur. Yöntemlerden en başarılı olanları sunmadan istatistiki ve kelime gömme yöntemlerinin sonuçlarını kendi içerisinde karşılaştırmak sonuçları değerlendirmek için daha doğru bir yaklaşım olacaktır.

İstatistiki yöntemler incelediğinde tüm modeller için 3 algoritmanın öne çıktığı görülmektedir. Bunlar KNN, Decision-Tree ve MLP'dir. Naive Bayes ve Random Forest bu modeller için en başarısız sınıflandırıcılar olmuştur. Tüm modeller için metriklerin altısında da en yüksek başarı değerine sahip olan KNN en başarılı algoritma olmuştur. Decisison-Tree 6 metrikten 2'sinde KNN algoritmasının başarısını yakalamış olsa da 4 metrikte gerisinde kalmıştır. KNN algoritmasının sonuçlarının önerilen 4 model için de umut verici olduğu söylenebilir.



Şekil 0.9. FastText modelinin CVSS 2.0 tahmin sonuçları

TF-IDF özellik çıkarım tekniğini kullandığımız modelin diğer modellerden daha başarılı sınıflandırma sonuçlarına sahip olduğu Şekil 5.10'da görülmektedir. Ayrıca KNN algoritmasının, incelenen diğer sınıflandırıcılardan çok daha iyi sonuçlar ürettiği de görülmektedir. Özellikle %95 ile Access Vector değerinin en iyi tahmin edilen metrik olduğu görülmektedir. Ayrıca Authentication vektörünün de %94'lük tahmin değeri oldukça iyi bir sonuçtur. Bununla birlikte Access Complexity değeri %88, Confidentiality Impact %85, Integrity Impact %87 ve Availability Impact %86 başarı oranı ile tahmin edilmiştir. Tüm metrikler için ortalama başarı oranı %89,2 olmuştur.



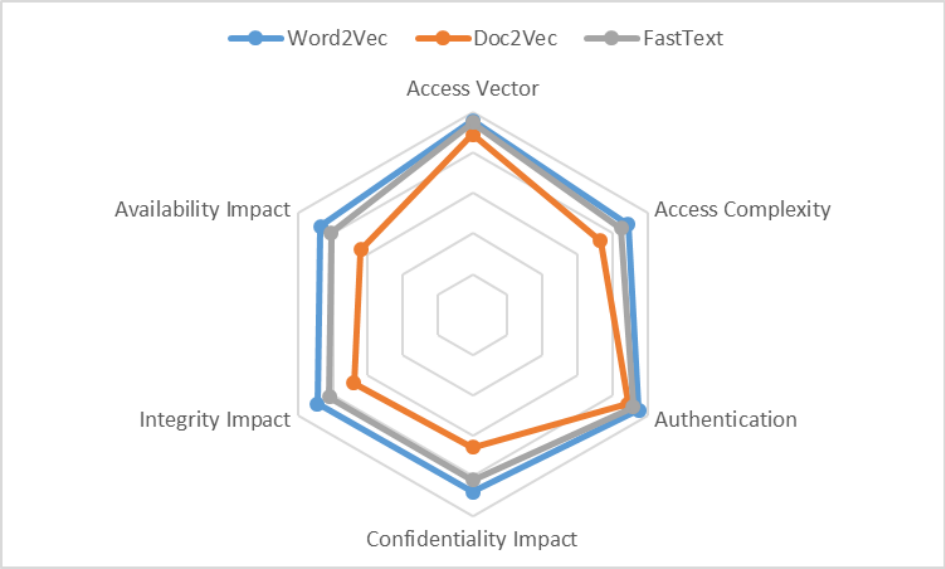
Şekil 0.10. İstatistiki modellerin CVSS 2.0 için karşılaştırmaları

Şekil 5.11'de Kelime gömme moddellerinin tahmin değerlerinin karşılaştırmalı bir sunumu görülmektedir. Grafikte gösterilen sonuçlar doğruluk değerlerinin yüzdelik karşılıklarıdır. Access

Vektör, elde edilen %96'lik başarı oranı ile tahmin değeri en yüksek metrik olmuştur. Authentication metrik değeri %95 ile ikinci olurken, %89 tahmin başarısı ile Access Complexity üçüncü sırayı almışlardır. Confidentiality Impact ve Integrity Impact %88 tahmin başarısı elde etmiştir. Availability Impact %87 ile bu metrikleri takip etmektedir.

Tüm metriklerde Word2Vec+KNN en iyi tahmin sonuçlarını göstermiştir. Ancak diğer modellerin içerisinde de çok başarılı tahmin performansı gösteren modeller olduğu Şekil 5.11'den görülmektedir. Özellikle FastText+MLP algoritması oldukça başarılı sonuçlar ortaya koymuştur. Model performansları karşılaştırıldığında Word2Vec sonuçları ile en başarılı kelime gömme yöntemi olmuştur. FastText elde ettiği sonuçlar ile ikinci sırada yer alırken, Doc2Vec elde edilen sonuçlara göre en düşük tahmin değerlerine sahip yöntem olarak görülmektedir. Sınıflandırma algoritmalarının başarımları elde edilen sonuçlara göere KNN, MLP, DT, RFC ve NB olarak sıralanmıştır.

Bulgular incelendiğinde, CVSS 2.0 için güvenlik metriklerinin tahmininde elde edilen sonuçlardan istatistiki modellerden en başarılı olanın TF-IDF ve kelime gömme modellerinden ise Word2Vec olduğu görülmektedir. Sınıflandırma algoritmalarının sonuçlarına kendi içersinde değerlendirildiğinde ise KNN, her iki vektörleşrime yöntemi için de en başarılı algoritma olduğu görülmektedir. Ancak Word2Vec+KNN modeli tüm modeller içirsinde, güvenlik vektörleri için en başarılı metrik değerlerini tahmin eden model olmuştur. Diğer modellerden bazıları Tablo 5.3'ten görülebileceği için metriklerin tamamında olmasada oldukça başarılı sonuçlar elde etmişlerdir. Özellikle BoW ve FastText yöntemlerinin sonuçları kayda değerdir. Ayrıca sınıflandırmada özellikle MLP ve DT algoritmalarının sonuçları umut vericidir.



Şekil 0.11. Kelime gömme modellerinin CVSS 2.0 için karşılaştırılması.

Modelimizin başarı değerlerinin daha doğru analiz edilebilmesi için önerilen yöntem literatürdeki benzer çalışmalardan Spanos vd.[9] sonuçları ile karşılaştırılmıştır. Ayrıca bu çalışmalarda kullanılan veri setleri ile önerilen yöntem yeniden eğitilerek elde edilen sonuçlar karşılaştırılarak sunulmuştur. Çalışmamızda kullanılan veri seti diğer çalışmalarda kullanılan tüm veri setlerinden daha büyük oranda veri içermektedir. Bu durum özellikle tahmin edilecek vektör olasılıklarının da sürekli arttığı anlamına gelmektedir. Ancak literatürdeki çalışmalar giderek zorlaşan bu problem için farklı öznitelik çıkarımı ve sınıflandırma modellerinin denenmesini tavsiye etmektedir. Elde edilen sonuçlar araştırmacıların öngülerini doğrular niteliktedir.

Tablo 5.4'te önerilen metodolojinin farklı veri setleri ve modeller ile karşılaştırılması görülmektedir. Görülebileceği gibi bu çalışmada elde edilen sonuçlar tüm CVSS 2.0 metrik değerlerine göre tahmin başarısında artış göstermektedir. Ayrıca karşılaştırılan modellerde kullanılan veri setleri ile yapılan eğitimler sonucu elde edilen tahmin değerleri de diğer modellerden daha başarılı sonuçlar vermiştir.

Tablo 0.4. CVSS 2.0 için modelimizin diğer modeller ve veri tabanları ile karşılaştırılması

Model	AV	AC	Au	CI	II	AI
Kelime Gömme Model	96	89	95	88	89	87
İstatistiki Model	95	88	94	85	87	86
Spanos et al	85	65	85	68	69	69
SpanosDS+Our Model	94	80	94	78	81	79

5.1.2. CVSS 3.1 Yazılım Güvenlik Açığı Metriklerinin Tahmin Sonuçları

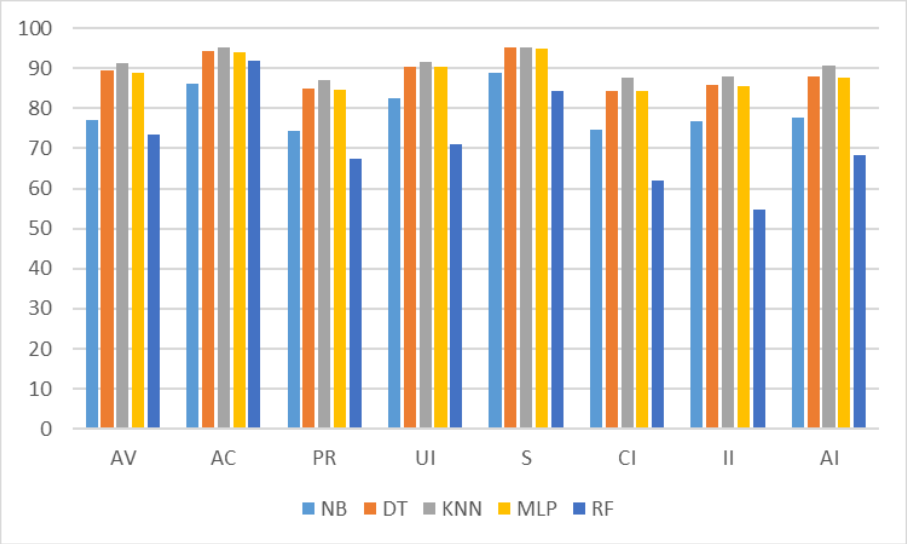
Güvenlik açığı skorlama sistemleri içerisinde en gelişmiş ve kapsayıcı olanı CVSS 3.1 sürümüdür. Bu skorlama sistemi 2019 yılında tanıtılmış ve kullanılmaya başlanmıştır. Şuanda NVD tarafından resmi olarak kullanılan iki versiyondan biridir ve en güncelidir. Bölüm 1.1'de incelediğimiz literatürden de anlaşıldığı üzere, mevcut çalışmanın bu versiyon ile bu çapta yapılan ilk çalışma olduğunu düşünüyoruz.

Sonuçları değerlendirmeden önce şunu hatırlatmak istiyoruz. Tahmin edilmeye çalışılan vektör boyutu sistemin bu versiyonu için çok daha karmaşık bir hal almaktadır. Vektörler, 1322 olasılık içeresinden ve her biri 2, 3 ve 4 ayrı sınıfa ayrılmış metriklerden tahmin edilmektedir. Ayrıca sonuçları değerlendirmeden bu zor problemin CVSS 2.0 versiyonuna göre neredeyse %50 daha az bir veri seti üzerinden tahmin edilmeye çalışıldığını hatırlamak istiyoruz. Problemin zorluğunu tekrar vurgulamakla birlikte farklı özellik seçimi ve sınıflandırma algoritmaları ile oluşturduğumuz modellimizin sonuçlarının beklentilerin üzerinde olduğunu vurgulamak isteriz.

Tablo 0.5. En yüksek tahmin değerine sahip CVSS 3.1 modellerinin sonuçları

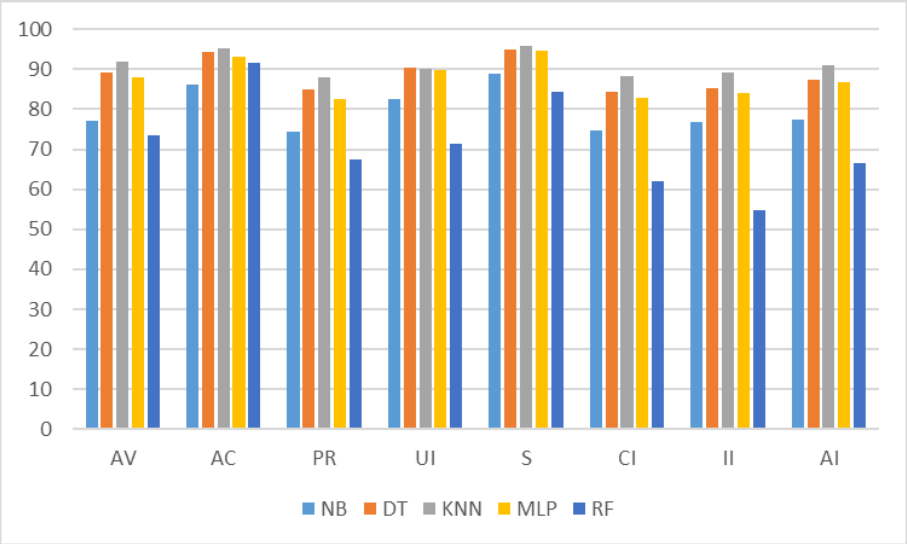
Metrik	Model	Alg.	Acc.	Prec.	Recall	F1
AV	BoW	KNN	0.91	0.79	0.69	0.72
	TF-IDF	**KNN**	**0.92**	**0.80**	**0.71**	**0.74**
	2Ngram	KNN	0.89	0.77	0.60	0.65
	3Ngram	DT	0.83	0.68	0.45	0.50
	Word2Vec	KNN	0.88	0.85	0.56	0.63
	Doc2Vec	MLP	0.82	0.59	0.56	0.57
	FastText	MLP	0.90	0.81	0.68	0.68
AC	**BoW**	**KNN**	**0.95**	**0.92**	**0.73**	**0.80**
	TF-IDF	**KNN**	**0.95**	**0.91**	**0.74**	**0.80**
	2Ngram	KNN	0.94	0.85	0.71	0.76
	3Ngram	KNN	0.93	0.83	0.64	0.69
	Word2Vec	KNN	0.94	0.89	0.63	0.68
	Doc2Vec	KNN	0.93	0.90	0.57	0.60
	FastText	**KNN**	**0.95**	**0.90**	**0.65**	**0.71**
PR	BoW	KNN	0.87	0.86	0.74	0.79
	TF-IDF	**KNN**	**0.88**	**0.87**	**0.75**	**0.80**
	2Ngram	KNN	0.82	0.81	0.67	0.71
	3Ngram	KNN	0.78	0.78	0.56	0.61
	Word2Vec	KNN	0.81	0.76	0.63	0.67
	Doc2Vec	KNN	0.75	0.71	0.52	0.56
	FastText	KNN	0.82	0.77	0.64	0.68
UI	**BoW**	**KNN**	**0.91**	**0.91**	**0.91**	**0.91**
	TF-IDF	MLP	0.90	0.89	0.89	0.89
	2Ngram	KNN	0.85	0.86	0.82	0.83
	3Ngram	DT	0.81	0.83	0.75	0.77
	Word2Vec	MLP	0.89	0.87	0.86	0.86
	Doc2Vec	MLP	0.79	0.78	0.77	0.77
	FastText	**MLP**	**0.91**	**0.91**	**0.89**	**0.89**
Scope	BoW	MLP	0.95	0.93	0.89	0.91
	TF-IDF	**KNN**	**0.96**	**0.93**	**0.92**	**0.93**
	2Ngram	KNN	0.92	0.90	0.79	0.83
	3Ngram	KNN	0.91	0.92	0.74	0.79
	Word2Vec	KNN	0.93	0.89	0.83	0.86
	Doc2Vec	MLP	0.88	0.78	0.79	0.79
	FastText	KNN	0.95	0.94	0.89	0.90
CI	**BoW**	**KNN**	**0.88**	**0.85**	**0.85**	**0.85**
	TF-IDF	**KNN**	**0.88**	**0.86**	**0.86**	**0.86**
	2Ngram	KNN	0.79	0.78	0.72	0.74
	3Ngram	DT	0.74	0.82	0.60	0.65
	Word2Vec	KNN	0.81	0.80	0.75	0.76
	Doc2Vec	MLP	0.70	0.66	0.59	0.61
	FastText	KNN	0.84	0.80	0.79	0.80
II	BoW	KNN	0.88	0.85	0.85	0.85
	TF-IDF	**KNN**	**0.89**	**0.89**	**0.88**	**0.88**
	2Ngram	KNN	0.79	0.81	0.74	0.76
	3Ngram	DT	0.71	0.82	0.63	0.67
	Word2Vec	MLP	0.81	0.80	0.78	0.79
	Doc2Vec	MLP	0.70	0.67	0.67	0.67
	FastText	KNN	0.84	0.84	0.82	0.82
AI	**BoW**	**KNN**	**0.91**	**0.89**	**0.73**	**0.78**
	TF-IDF	**KNN**	**0.91**	**0.89**	**0.74**	**0.79**
	2Ngram	KNN	0.84	0.85	0.68	0.72
	3Ngram	DT	0.73	0.78	0.55	0.60
	Word2Vec	KNN	0.86	0.80	0.67	0.70
	Doc2Vec	MLP	0.76	0.70	0.54	0.57
	FastText	KNN	0.87	0.82	0.66	0.70

İstatistiki yöntemlerden BoW özellik çıkarım tekniğini kullanılan modelin tüm sınıflandırma algoritmaları ile elde ettiği accuracy skorları Şekil 5.12'de görülmektedir. En başarılı sonuçları sırayla KNN, DT ve MLP'nin verdiği görülmektedir. Random forest algoritması tüm metriklerin tahmininde en geride kalan algoritma olmuştur.



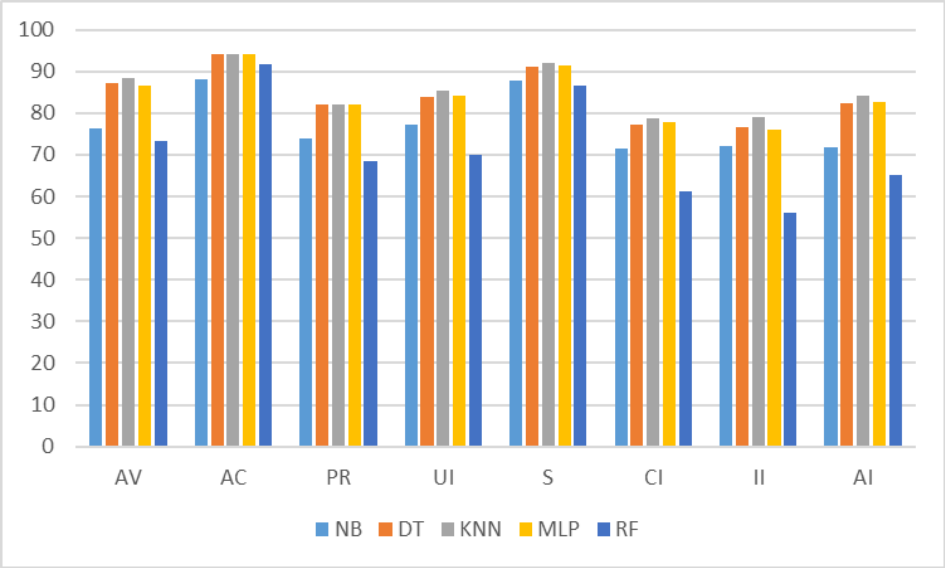
Şekil 0.12. BoW modelinin CVSS 2.0 tahmin sonuçları

Genel olarak istatistiki en başarılı model olarak tespit ettiğimiz TF-IDF özellik çıkarım tekniğini kullandığımız modele ait accuracy sonuçları Şekil 5.13'te görülmektedir. Ortalama başarı değerinin yaklaşık %90 olduğu KNN en başarılı sınıflandırma algoritması olarak karşımıza çıkmaktadır. KNN algoritmasını DT ve MLP takip etmektedir. En başarısız algoritmanın Random Forest olduğu görülmektedir.

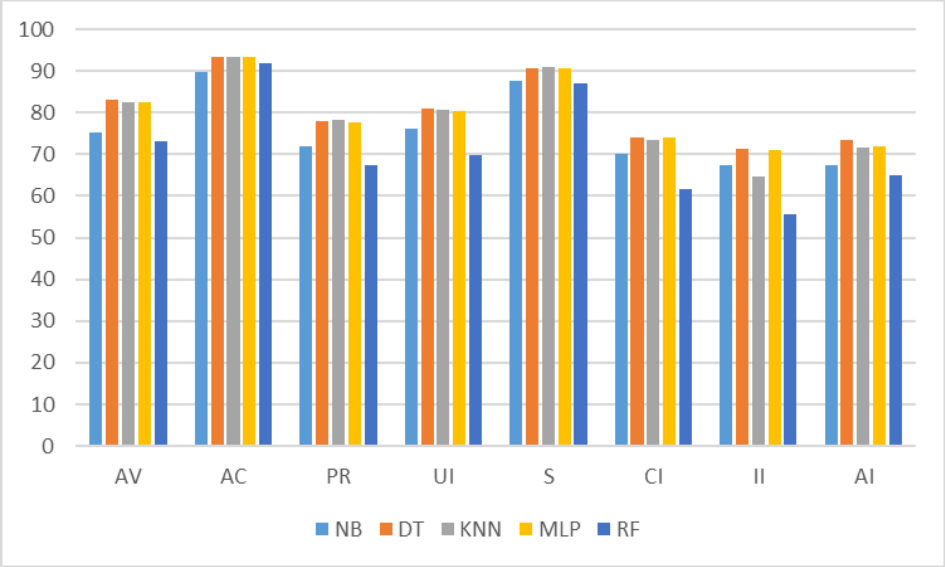


Şekil 0.13. TF-IDF modelinin CVSS 2.0 tahmin sonuçları

Ngram özellik çıkarım tekniğini kullandığımız modele ait performanslar incelendiğinde genel olarak başarılı tahminler yapmış olsalar da diğer modellerin gerisinde kaldıkları görülmektedir. Ngram modelleri kendi içerisinde değerlendirildiğinde n değeri artıkça performansta düşüş olduğu gözlemlenmiştir. Şekil 5.14'te 2-Ngram modelinin, Şekil 5.15'te ise 3-Ngram modelinin accuracy değerleri görülmektedir. KNN algoritması 2-Ngram modelinde diğer sınıflandırma algoritmalardan daha iyi sonuçlar üretmiştir. 3 Ngram modelinde en başarılı sonuçlar DT algoritması ile elde edilmiştir.

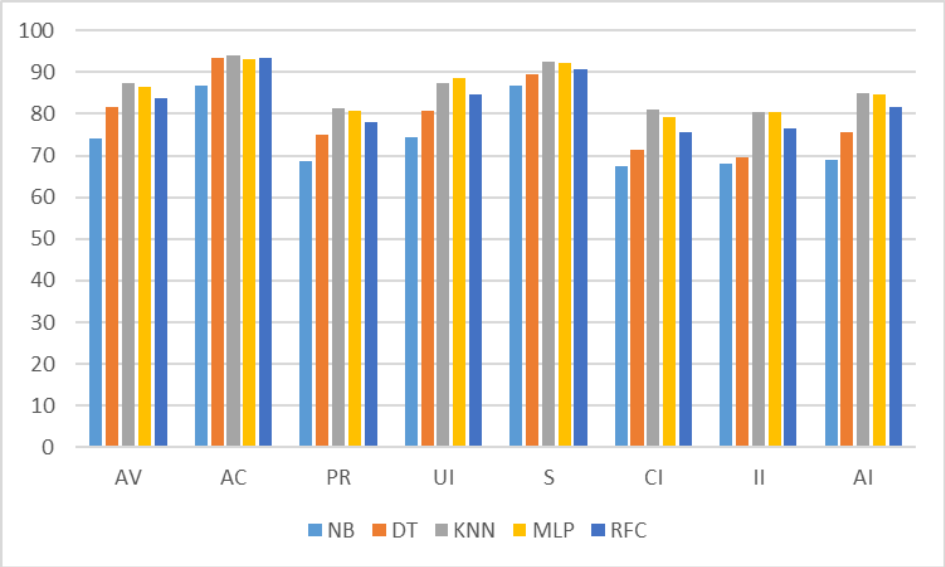


Şekil 0.14. 2-Ngram modelinin CVSS 2.0 tahmin sonuçları



Şekil 0.15. 3-Ngram modelinin CVSS 2.0 tahmin sonuçları

CVSS 2.0 güvenlik vektörü metriklerinin tahmininde en başarılı model olan Word2Vec yöntemi elde ettiği sonuçlar itibari ile kendi içerisinde değerlendirildiğinde oldukça başarılı sınıflandırma sonuçları elde etmiştir. Tüm metrik değerlerinde en az %80 ve üzeri tahmin başarısı göstererek tutarlı bir tahmin modeli olabileceğini göstermiştir. Özellikle KNN ve MLP algoritmaları ile elde ettiği değerler oldukça yüksektir. Ayrıca RFC algoritmasının sonuçları ile de bazı metrik değerlerinde oldukça başarılıdır. Şekil 5.16'da gösterilen Word2Vec modelimizin sonuçları değerlendirildiğinde en başarılı algoritma altı metrik değerinde KNN olarak görülmektedir. İki metriğin sınıflandırmasında ise MLP en başarılı algoritma olmuştur. Sonuçlar itibari ile RFC ve DT onları takip etmektedir. Bu modelimizin en düşük başarı gösteren algoritması ise NB olmuştur.

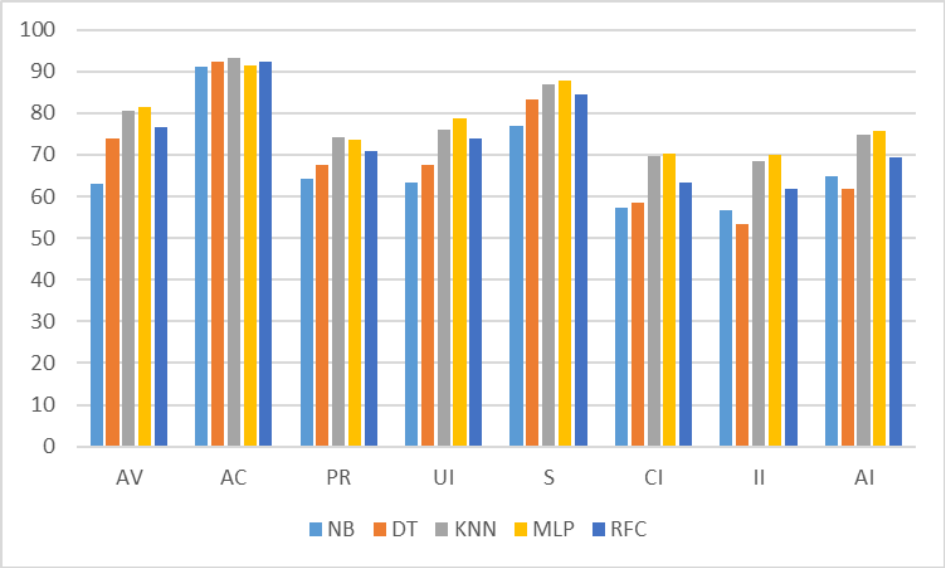


Şekil 0.16. Word2Vec modelinin CVSS 3.1 tahmin sonuçları

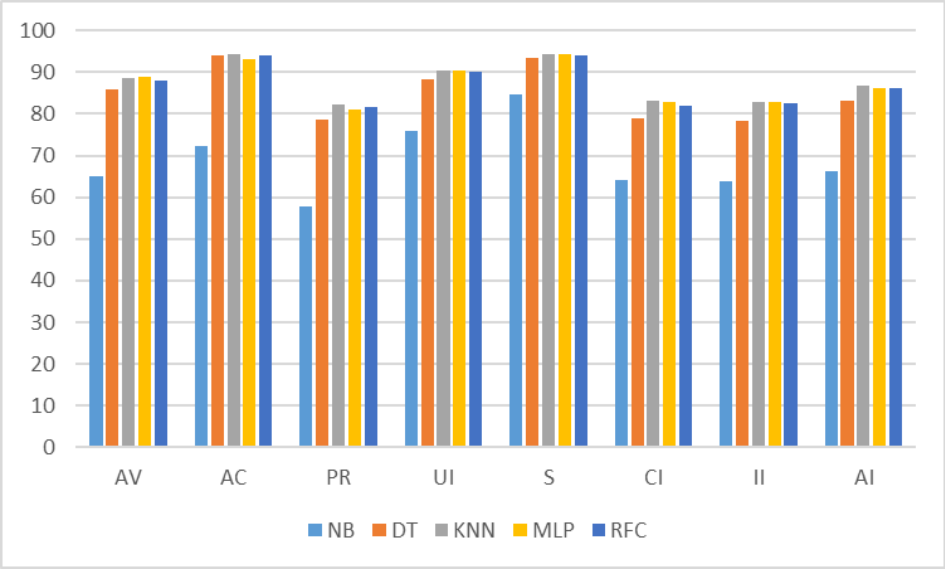
Doc2Vec algoritması CVSS 2.0 versiyonunda olduğu gibi CVSS 3.1 versiyonunun güvenlik metriklerini tahmin etme sonuçları itibari ile en düşük başarı gösteren modelimiz olmuştur. Ancak sonuçlar kendi içerisinde değerlendirildiğinde kabul edilebilir sonuçlar elde edildiği görülmektedir. Özellikle MLP algoritması ile oldukça başarı sonuçlar elde ettiği Şekil 5.17'de görülmektedir. Görüldüğü gibi altı metrik değerinde en başarılı olan MLP algoritmasını iki tahmin sonucu ile KNN algoritması takip etmiştir. RFC algoritması elde ettiği tüm sonuçlarda bu iki algoritmayı takip etmektedir. Bu algoritmaları sırayla DT ve NB izlemiştir.

FastText yöntemi ile elde edilen güvenlik açığı metrik değerleri Şekil 5.18'de gösterilmiştir. CVSS 3.1 skorlama sisteminin değerlerinin tamamının tahmin edilmesinde FastText en yüksek değerleri elde eden yöntem olmuştur. Metriklerin biri dışında yedi metrik değerinin sınıflandırmasında en yüksek başarı oranına KNN algoritması ile ulaşılmıştır. MLP ise sadece bir

metrik değerinde en yüksek tahmin değerini elde etse de sonuçlar KNN'e çok yakındır. Ancak özellikle RFC ve DT algoritmalarının sonuçları da bu iki algoritmaya çok yakındır. NB algoritması ile elde edilen sonuçlar ise diğer algoritmaların oldukça gerisinde kalmıştır.



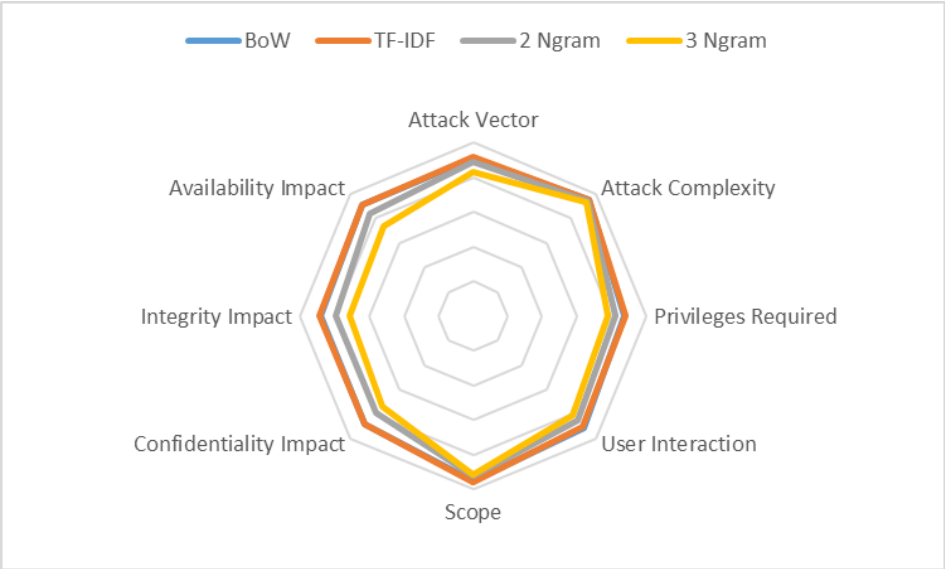
Şekil 0.17. Doc2Vec modelinin CVSS 3.1 tahmin sonuçları



Şekil 0.18. FastText modelinin CVSS 3.1 tahmin sonuçları

Önerilen metodolojinin ilk bölümünde kullanılan BoW, TF-IDF ve Ngram istatistiki yöntemlerinin CVSS 3.1 güvenlik vektörlerinin metriklerini tahmin etmede göstermiş oldukları performans tek tek açıklanmıştır. Bu yöntemler içersinde en başarılı özellik çıkarım tekniğinin TF-IDF ve en başarılı sınıflandırma algoritmasının KNN olduğu görülmektedir. Modellerin başarılarının karılaştırıldığı radar grafik Şekil 5.19'da sunulmuştur. Şekilden de anlaşılabileceği

gibi tüm metrik değerlerinde TF-IDF en başarılı yöntem olsa da BoW modelinin sonuçları bu modele oldukça yakındır. Ngram modellerinin sonuçları başarılı sonuçlar üretmiş olsa da BoW ve TF-IDF'in gerisinde kalmıştır. Ayrıca N değeri arttıkça tahmin başarısının düştüğü görülmektedir.

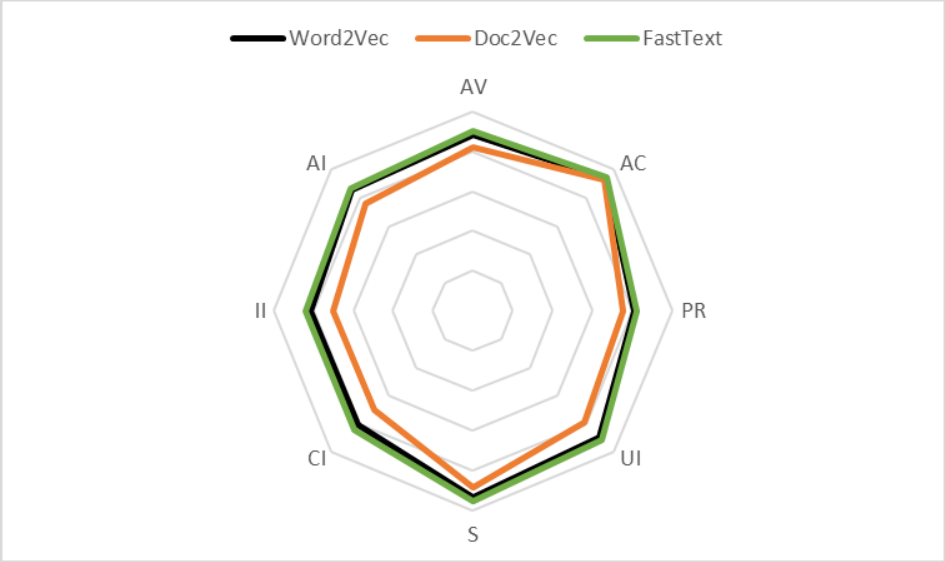


Şekil 0.19. İstatistiki modellerin CVSS 3.1 için karşılaştırmaları

Önerilen metodolojinin ikinci kısmını oluşturan kelime gömme yöntemlerinşin tahmin başarılarının birbiri ile karşılaştırılması sonucu elde edilen radar grafik Şekil 5.20'te görülmektedir. Grafikten de anlaşılacağı gibi AC metriğinin tahmini dışında tüm metriklerin sınıflandırılmasında Doc2Vec'in diğer yöntemlerin oldukça gerisinde kaldığı açıkça görülmektedir. Ayrıca FastText ve Word2Vec yöntemlerinin sınıflandırma sonuçlarının bir birine oldukça yakın oldukları da açıkça görülmekle birlikte FastText yönteminin tüm metrik değerlerinin tahmininde en başarı yöntem olduğu görülmektedir.

Tablo 5.5 incelendiğinde TF-IDF+KNN algoritmaları ile %96 değeri ile en başarılı tahmin edilen güvenlik açığı vektör metriğinin Scope olduğu görülmektedir. İkinci en başarılı tahmin BoW, TF-IDF ve FastText modelleri ve KNN sınıflandırma algoritması ile %95 tahmin başarısı ile Attack Complexity olmuştur. Bu değeri TF-IDF+KNN ile %92 tahmin başarısı elde edilen Attack Vector takip etmiştir. User Interaction ve Availability Impact %91 değeri ile tahmin edilmiştir. Availability Impact metriğini en başarılı tahmin eden algoritmalar BoW ve TF-IDF kullanan yöntemlerdir. User Interaction ise BoW+KNN ve FastText+MLP yöntemleri ile tahmin edilmiştir. TF-IDF+KNN yöntemi %89 tahmin başarısı ile Integrity Impact metriğini sınıflandırmıştır. Güvenlik vektörlerin diğer metriklerinden Privileges Required ve Confidentiality Impact tahmin değerleri %88 olmuştur. Privileges Required TF-IDF+KNN ile Confidentiality Impact ise hem TF-IDF hem de BoW+KNN ile bu sonucu elde etmiştir. Bu değerlerin yedisi TF-IDF+KNN, dördü

BoW+KNN ile ve sadece bir tanesi FastText+MLP modeli ile elde edilmiştir. Bu durum TF-IDF+KNN modelini en başarılı model yapmaktadır.



Şekil 0.20. Kelime gömme modellerin CVSS 3.1 için karşılaştırmaları

Ayrıca önceki bölümlerde bahsedildiği gibi CVSS'nin 3.1 versiyonu oldukça yeni bir skorlama yöntemidir ve üzerinde henüz yeterli araştırma yapılmamıştır. Literatürde çalışmamızla karşılaştırabileceğimiz bir çalışmaya rastlanmamıştır. Bu nedenle hem istatistiki yöntemlerle elde ettiğimiz sonuçları hem de kelime gömme yöntemi ile elde edilen sonuçlar Tablo 5.6'da karşılaştırılmıştır. Görüleceği üzeri önerilen modeller bazı metrik değerlerinde aynı sonuçları tekrar etse de istatistiki modellerin sonuçları genel olarak kelime gömme modellerinin sonuçlarından daha iyi performans gösterdiği ifade edilebilir.

Tablo 0.6. CVSS 3.1 için modellerimizn karşılaştırılması

Model	AV	AC	PR	UI	S	CI	II	AI
Kelime Gömme Model	90	95	82	91	95	84	84	87
İstatistiki Model	92	95	88	91	96	88	89	91

5.2. Önem Skoru ve Derecelerinin Sonuçları

Yazılım güvenlik açıklarının analizi ve derecelendirilmesi önemli bir süreçtir. Yazılım güvenlik açığı raporlarının nihai sonucu bir önem skorunun hesaplanması ve buna bağlı olarak önem derecesinin tayinidir. Önem derecesi yazılım güvenlik açığı hakkında ilk ve en önemli değerdir. İstismarı mümkün olan güvenlik açıkları toplam açıkların sadece %20'sidir. İstismarların büyük çoğunluğu ilk iki hafta içerisinde gerçekleşmektedir [10]. Önem derecelerinin zaman gecikmeleri yaşanmadan belirlenmesi bu nedenle elzemdir. Kurumsal ağların ve sistemlerin

tehditlere ve zafiyetlere kaşı izlenmesinden ve güvenliğinden sorumlu birimlerin en önemli görevlerinden biri oluşan güvenlik ihlallerinin analizini gerçekleştirmektir. Bu durumda analistler öncelikle oluşan durumla ilgili açık kaynak sistemleri üzerinden istihbarat araştırılması yapmaktadırlar [95]. Özellikle zafiyet veri tabanları ve güvenlik skorları bu aşamada en çok tercih edilen referanslar olarak dikkati çekmektedir (SCAP, OVAL vb.) [96]. NVD veri tabanında yayınlanan tüm raporlar bir önem skoruna sahiptir. Tez çalışmasının amaçlarından biri bu skorların hesaplanmasında kullanılan manuel işlemlerin makine öğrenmesi ve doğal dil işleme teknikleri ile yapılabileceğini göstermektir. Bu noktada iki farklı yöntem ile sonuçlar elde edilmiştir. İlk olarak bir önceki bölümde olduğu gibi önerilen metodoloji ile güvenlik raporlarının önem dereceleri tahmin edilmiştir. İkinci olarak ise bir önceki bölümde tahmin edilen güvenlik açığı vektörlerinin en iyi tahmin sonuçlarına sahip metriklerden oluşturulmuş olan güvenlik vektörleri üzerinden Bölüm 3.1'de ayrıntıları verilen 3.1 – 3.11 arasındaki formüller kullanılarak önem skorlarının hesaplanmıştır. Tablo 3.3 ve 3.4'te nitel önem derecesi ve skorlarına ait ölçekler sunulmuştur.

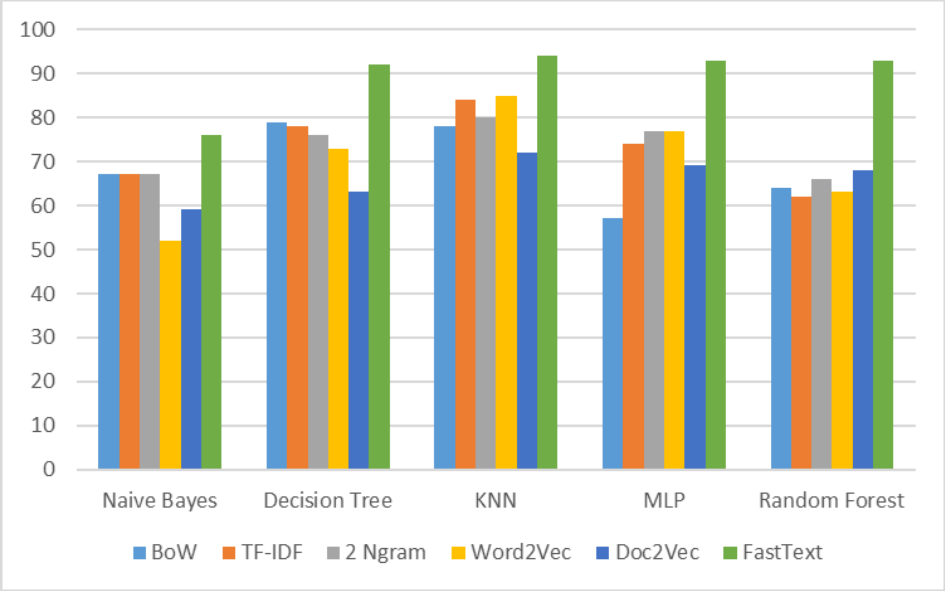
5.2.1. CVSS 2.0 Önem Derecelerinin Tahmin Sonuçları

Bu bölümde tez çalışmasında önerilen metodolojinin değerlendirilmesi için CVSS 2.0 için önem derecelerinin sınıflandırma sonuçları sunulmuştur. CVSS 2.0 önem dereceleri üç farklı sınıf içerisinden doğru tahmini yapmaya çalışmıştır. Bu sınıflar Tablo 3.3'te gösterilmiştir. Tablo 5.7'de tüm modellerin en iyi performans değerleri sunulmuştur. Sonuçlar Accuracy, Precisison, Recall ve F1 skor ölçeklerine göre değerlendirilmiştir.

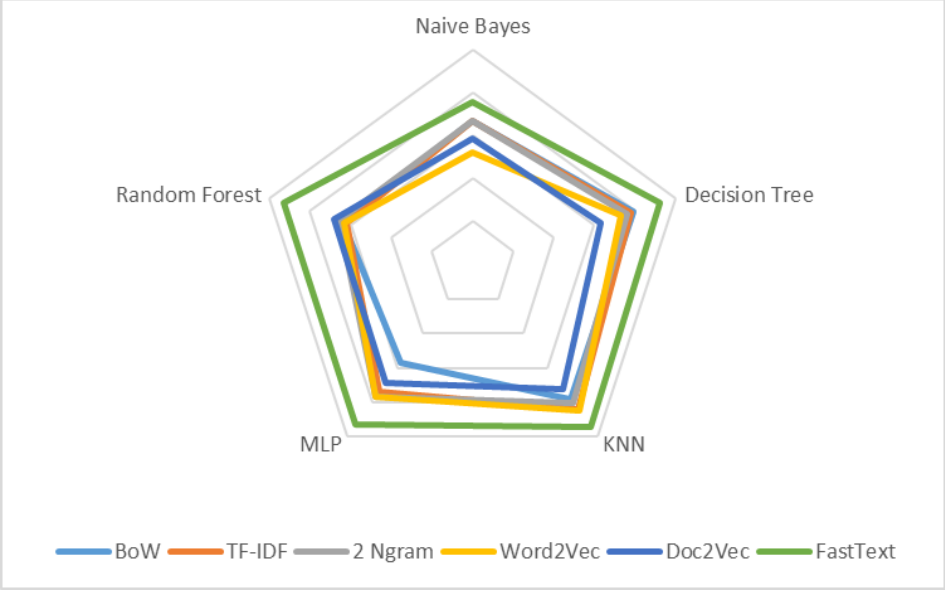
Doğru değerlendirme yapabilmek için öncelikle her metin vektörleştirme modeli, sınıflandırma algoritmalarına göre ayrı ayrı değerlendirilmiştir. Şekil 5.21 tüm modellerin sınıflandırma performanslarını göstermektedir. Görüldüğü üzere en başarılı model FastText kullanılan modellerdir. Tüm sınıflandırma algoritmaları içerisinde en başarılı değerleri elde etmiştir. KNN kullanan TF-IDF en yüksek başarıyı gösteren istatistiki model olmuştur. Şekil 5.22 modellerin sınıflandırma performanslarını karşılaştırmalı bir şekilde gösteren bir radar grafiktir. Buradan da FastText ile birlikte kullanılan KNN diğer algoritmalardan daha başarılı sınıflandırma yaptığı görülmektedir.

Ayrıca Tablo 5.7 incelendiğinde tahmin modelimizin doğrulama değerlerinin tutarlığı görülecektir. FastText modeli tüm algoritmalar ile en başarılı yöntem olmuştur. KNN algoritması ile %94 ile en yüksek başarı oranını göstermiştir. Ayrıca DT, MLP ve RF algoritmalarının başarı oranları da oldukça yüksektir. FastText yönteminin en düşük değer elde ettiği NB algoritmasının başarı oranı %76'dır. Diğer Word Embedding yöntemlerinin sonuçları incelendiğinde Word2Vec ile birlikte kullanılan MLP algoritmasının %85 ve Doc2Vec ile kullanılan KNN algoritmalarının %77 ile en yüksek başarı oranlarını yakaladıkları görülmektedir. Bu iki yöntem başarılı

sayılabilecek sınıflandırma performansları göstermiş olsalar da FastText'in oldukça gerisinde kalmışlardır.



Şekil 0.21. CVSS 2.0 için önem derecesi tahmininin grafiği



Şekil 0.22. CVSS 2.0 modellerinin karşılaştırmalı doğruluk radar grafikleri

İstatiksel yöntemlerden, KNN ve TF-IDF %84'lük sınıflandırma başarısı göstermiştir. Diğer modellerden Ngram KNN ile %80 sınıflandırma başarı elde ederek ikinci model olmuştur. BoW modeli, DT algoritmasının kullanıldığı modelde %79 ile en iyi başarı oranını yakalamıştır. İstatistiksel ve Word Embedding modeller karşılıklı olarak karşılaştırıldıklarında FastText'in diğer

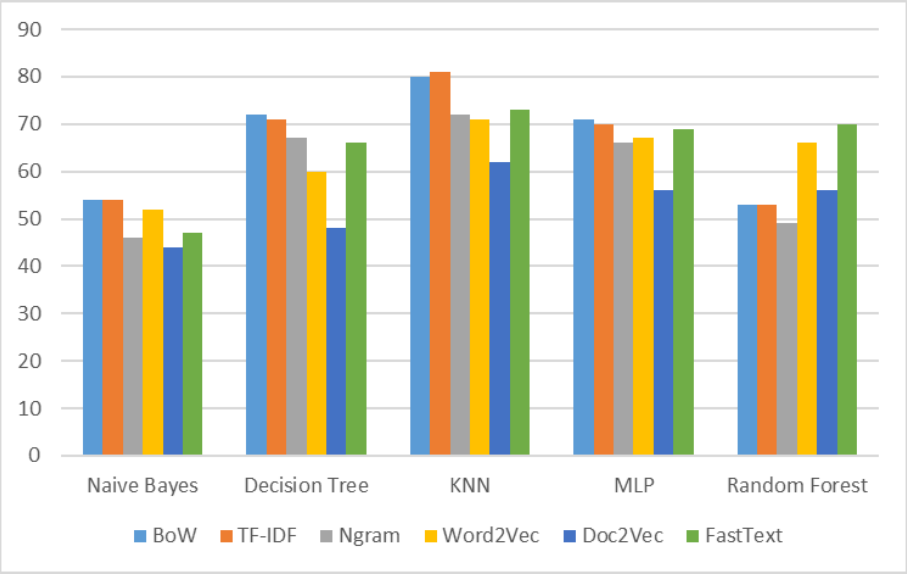
modellerden bariz şekilde başarılı sonuçlar elde ettiği açıktır. Ancak TF-IDF ve Word2Vec modellerinin KNN algoritması ile elde ettiği sonuçlar da umut vericidir.

Tablo 0.7. CVSS 2.0 Tahmin değerlerini sonuçları

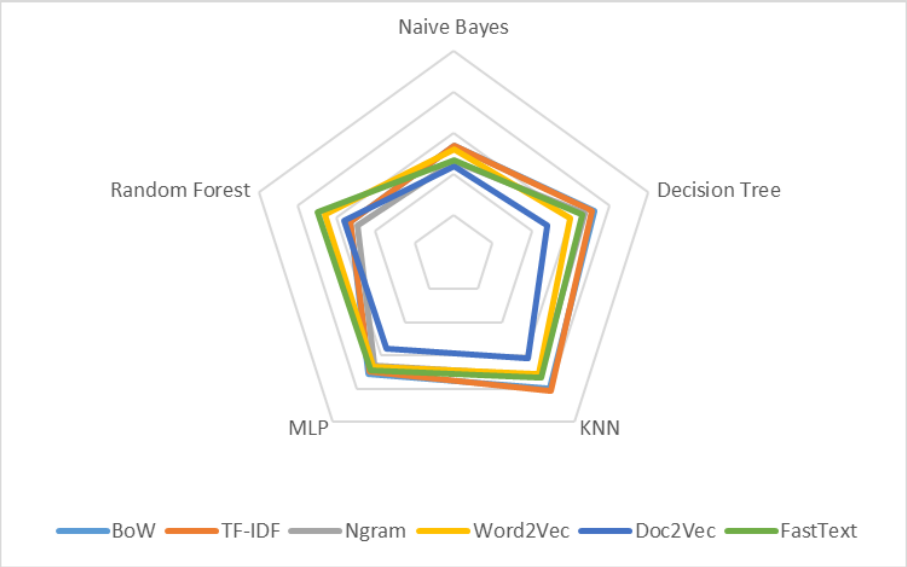
Metric		Alg.	Acc.	Prec.	Recall	F1
İstatistiksel Modeller	BoW	NB	0.67	0.60	0.64	0.61
		DT	**0.79**	**0.77**	**0.70**	**0.73**
		KNN	0.78	0.71	0.71	0.71
		MLP	0.57	0.19	0.33	0.24
		RF	0.64	0.48	0.42	0.39
	TF-IDF	NB	0.67	0.60	0.64	0.61
		DT	0.78	0.77	0.69	0.72
		KNN	**0.84**	**0.80**	**0.80**	**0.80**
		MLP	0.74	0.68	0.68	0.68
		RF	0.62	0.48	0.39	0.36
	Ngram	NB	0.67	0.60	0.64	0.62
		DT	0.76	0.73	0.67	0.69
		KNN	**0.80**	**0.76**	**0.73**	**0.74**
		MLP	0.77	0.75	0.67	0.70
		RF	0.66	0.48	0.44	0.42
Word Embedding Models	Word2Vec	NB	0.52	0.49	0.54	0.48
		DT	0.73	0.72	0.60	0.64
		KNN	**0.85**	**0.81**	**0.81**	**0.81**
		MLP	0.77	0.73	0.70	0.72
		RF	0.63	0.48	0.41	0.38
	D oc2Vec	NB	0.59	0.53	0.58	0.54
		DT	0.63	0.58	0.44	0.44
		KNN	**0.72**	**0.66**	**0.59**	**0.61**
		MLP	0.69	0.62	0.62	0.62
		RF	0.68	0.80	0.47	0.48
	FastText	NB	0.76	0.53	0.72	0.57
		DT	0.92	0.91	0.70	0.77
		KNN	**0.94**	**0.90**	**0.77**	**0.82**
		MLP	0.93	0.80	0.81	0.80
		RF	0.93	0.93	0.71	0.79

5.2.2. CVSS 3.1 Önem Derecelerinin Tahmin Sonuçları

Güvenlik açığı skorlama sistemlerinin en günceli CVSS 3.1 versiyonudur. CVSS 3.1 versiyonu ile önem derecelerinin ait olduğu sınıf sayısı beşe yükselmiştir. Bunlar Tablo 3.4'te sunulmuştur. Veri sayısının %50 az olmasına rağmen CVSS 3.1 modelimizin sonuçları oldukça tatmin edicidir. Tablo 5.8 tüm modellerin ve sınıflandırma algoritmaların sonuçlarını göstermektedir. Tablodan da görüldüğü gibi test sonuçlarının doğru değerlendirilebilmesi için tüm doğrulama yöntemlerinin değerleri açıkça sunulmuştur. Modellerin ve algoritmaların sınıflandırma performansları Şekil 5.23'te görülmektedir. CVSS 3.1 önem derecelerinin tahmin edilmesinde en yüksek başarıyı elde eden TF-IDF ve KNN modelidir. En düşük performans değeri ise Doc2Vec ve NB modeli olarak görülmektedir. Şekil 5.24'te modellerin performansını karşılaştırmalı olarak veren radar grafik sunulmuştur.



Şekil 0.23. CVSS 3.1 için önem derecesi tahmininin grafiği



Şekil 0.24. CVSS 3.1 modellerinin karşılaştırmalı doğruluk radar grafikleri

Tablo 5.8'de modellerin sınıflandırma performanslarının ayrıntıları incelendiğinde TF-IDF ve BoW modellerinin KNN algoritmaları ile çok yakın performans değerlerine sahip olduğu görülmektedir. TF-IDF+KNN modeli %81'lik sınıflandırma başarı ile, %80 sınıflandırma başarısı elde eden BoW+KNN modelini geride bırakarak en başarılı model olmuştur. İkinci olan BoW+KNN modelini, %73 ile FastText+KNN modeli takip etmiştir. Ngram+KNN ve BoW+DT modelleri %72'lik doğruluk oranı ile başarılı görünen diğer modellerdir. Diğer modellerden %70 ve üzeri başarı değerine sahip modeller sırayla BoW+MLP, TFIDF+DT/MLP, Word2Vec+KNN ve FastText+RF modelleri olmuştur. En düşük başarı oranına sahip olan model NB ile %44

sınıflandırma başarısı elde eden Doc2Vec'tir. Sonuçlar incelediğinde Doc2Vec dışında tüm modellerin en az bir algoritma ile %70 üzeri tahmin performansı gösterdiği açıkça görülmektedir. En başarılı vektörleştirme yöntemlerinin BoW ve TF-IDF, en başarılı algoritmanın ise KNN olduğu görülmektedir.

Tablo 0.8. CVSS 3.1 Tahmin değerlerini sonuçları

Metric		Alg.	Acc.	Prec.	Recall	F1
İstatiksel Yöntemler	BoW	NB	0.54	0.43	0.47	0.43
		DT	0.72	0.68	0.57	0.61
		KNN	**0.80**	**0.73**	**0.73**	**0.73**
		MLP	0.71	0.69	0.55	0.57
		RF	0.53	0.30	0.31	0.27
	TF-IDF	NB	0.54	0.43	0.47	0.43
		DT	0.71	0.67	0.56	0.59
		KNN	**0.81**	**0.74**	**0.74**	**0.74**
		MLP	0.70	0.59	0.58	0.58
		RF	0.53	0.31	0.31	0.27
	Ngram	NB	0.46	0.41	0.44	0.37
		DT	0.67	0.61	0.53	0.55
		KNN	**0.72**	**0.64**	**0.62**	**0.62**
		MLP	0.66	0.70	0.50	0.52
		RF	0.49	0.33	0.29	0.23
Word Embedding Yöntemler	Word2Vec	NB	0.52	0.41	0.45	0.41
		DT	**0.60**	**0.65**	**0.41**	**0.43**
		KNN	**0.71**	**0.69**	**0.51**	**0.54**
		MLP	0.67	0.55	0.47	0.48
		RF	0.66	0.73	0.45	0.47
	Doc2Vec	NB	0.44	0.33	0.35	0.34
		DT	0.48	0.31	0.30	0.28
		KNN	0.62	0.60	0.43	0.45
		MLP	0.56	0.42	0.42	0.42
		RF	0.56	0.74	0.34	0.32
	FastText	NB	0.47	0.41	0.46	0.38
		DT	0.66	0.66	0.47	0.50
		KNN	0.73	0.69	0.55	0.58
		MLP	0.69	0.59	0.55	0.57
		RF	0.70	0.74	0.50	0.53

Tablo 5.9 tüm modellerin en yüksek sınıflandırma başarılarını göstermektedir. Bu tablodan en başarılı sınıflandırıcının KNN algoritması olduğu açıkça görülmektedir. Sadece DT algoritması BoW ile birlikte CVSS 2.0 için bir hesaplamada en yüksek başarıyı elde etmiştir. Diğer tüm modeller için KNN en yüksek başarı oranına sahip algoritma olmuştur. Vektörleştirme yöntemlerinden CVSS 2.0 için FastText en yüksek başarı oranına sahip olurken, CVSS 3.1 için en yüksek başarı oranı TF-IDF ile yakalanmıştır. İstatistiki yöntemler ve Word embedding yöntemler kendi içerisinde karşılaştırıldıklarında TF-IDF ve FastText yöntemlerinin öne çıktıları görülmektedir. İstatistiki yöntemleri başarı oranları birbirlerine yakındır. Ancak Word embedding yöntemlerde Doc2Vec diğer yöntemlerin oldukça gerisinde kalmıştır.

Tablo 0.9. Tahmin modellerinin en yüksek değere sahip sonuçları

CVSS	Metric	Alg.	Acc.	Prec.	Recall	F1
2.0	BoW	DT	0.79	0.77	0.70	0.73
	TF-IDF	KNN	0.84	0.80	0.80	0.80
	Ngram	KNN	0.80	0.76	0.73	0.74
	Word2Vec	KNN	0.85	0.81	0.81	0.81
	Doc2Vec	KNN	0.72	0.66	0.59	0.61
	FastText	**KNN**	**0.94**	**0.90**	**0.77**	**0.82**
3.1	BoW	KNN	0.80	0.73	0.73	0.73
	TF-IDF	**KNN**	**0.81**	**0.74**	**0.74**	**0.74**
	Ngram	KNN	0.72	0.64	0.62	0.62
	Word2Vec	KNN	0.71	0.69	0.51	0.54
	Doc2Vec	KNN	0.62	0.60	0.43	0.45
	FastText	KNN	0.73	0.69	0.55	0.58

5.2.3. Önem Skorlarının Tahmin Edilen Güvenlik Vektörlerinden Hesaplanması

Yazılım güvenlik açıklarının sahip olduğu güvenlik metrikleri ve bunlardan oluşturulan güvenlik açığı vektörlerinin önemi ve hesaplamaları önceki bölümlerde açıklanmıştır. Güvenlik açığı vektörlerinin önemi bu vektörler kullanılarak ilgili açık için temel bir önem skorunun hesaplanmasından kaynaklanmaktadır. Mevcut çalışmanın temel amaçlarından birisi de tahmin edilen vektörlerden önem skorlarının hesaplanarak gerçek değerleri ile kıyaslanmasıdır. Bu skorlar güvenlik açığı hakkında en önemli bilgilerden birini yansıtmaktadır. Önerilen modellerin performans değerleri Tablo 5.3 ve 5.5'te sunulmuştur.

Tablo 0.10. Güvenlik açığı önem skorlarının hesap değerlerinin ölçüm sonuçları

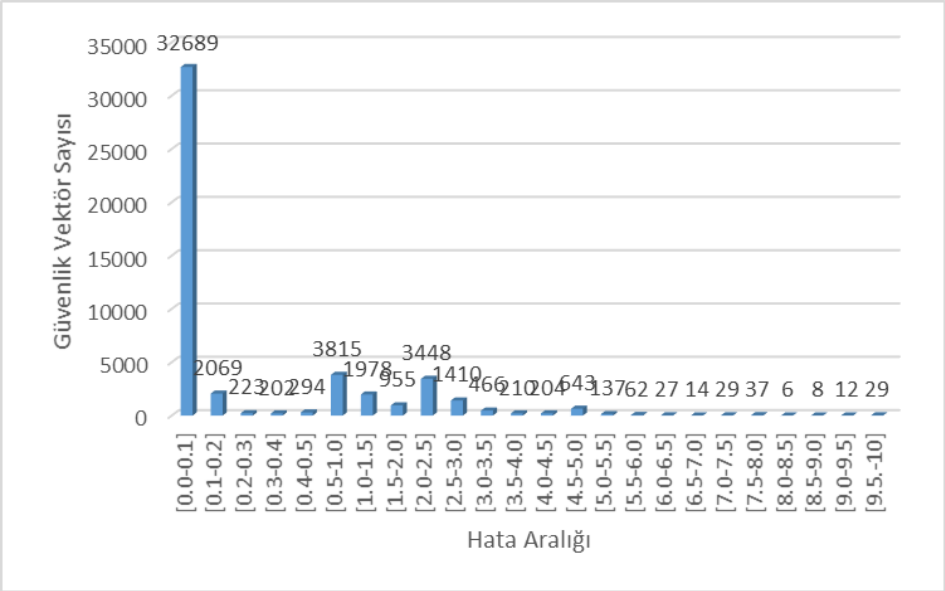
	MAE	MDAE	RMSE
CVSS 2.0 Base Score	0.75	0.0	1.56
CVSS 3.1 Base Score	0.12	0.09	2.13

Hesaplamalarda, bu tablolarda belirtilen performans değeri en yüksek model ve algoritma çiftlerinin kullanımıyla elde edilen vektörler kullanılmıştır. CVSS 2.0 ve 3.1 için ayrı ayrı hesaplamalar gerçekleştirilmiştir. Sonuçlar, Bölüm 4.6'da belirtilen doğrulama ölçütlerinden MAE, MDAE, ve RMSE kullanılarak değerlendirilmiştir. Tablo 5.10'da hesaplanan değerlerin ölçüm sonuçları verilmiştir.

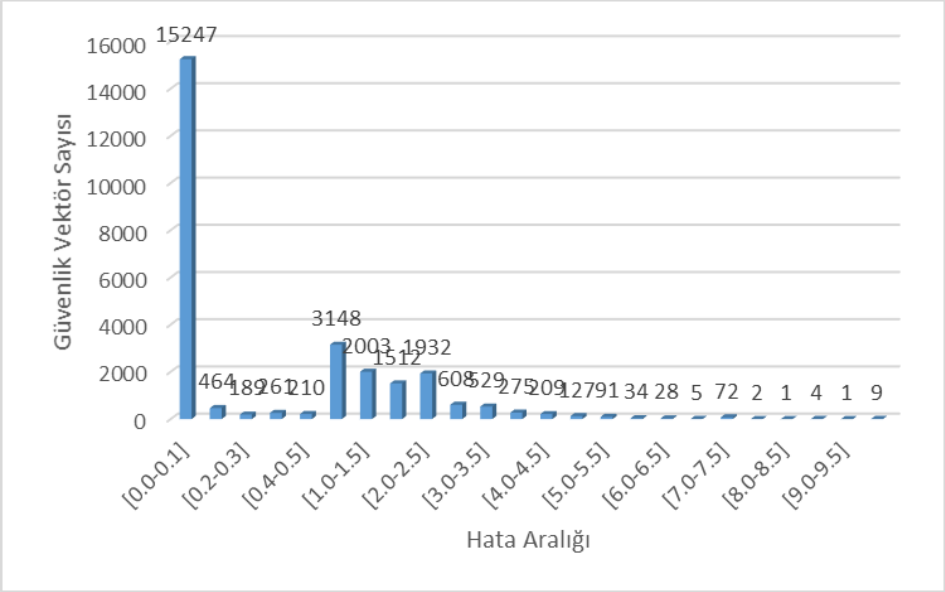
Tablo 0.11. Önem skorlarının gerçek ve hesaplanan değerleri

CVSS	Data	Mean	Std	Mdn	Min	Max
2.0	Gerçek Değer	5.98	2.01	5.5	0.0	10.0
	Hesaplanan	5.85	2.11	5.3	0.0	10.0
3.1	Gerçek Değer	7.24	1.65	7.5	1.9	10.0
	Hesaplanan	7.29	1.94	7.6	0.0	10.0

Gerçek skorlar ile tahmin edilen skorların değerleri Tablo 5.11'de görülmektedir. Gerçek değerlerin ortalama değerleri versiyonlar için sırayla 5.98 ve 7.24'tür. Modelimizin tahmin değerleri ise sırayla 5.85 ve 7.29 olmuştur. Görülebileceği gibi çok yakın değerlerdir. Ayrıca Şekil 5.25 ve 5.26'da hata oranlarının histogramları gösterilmiştir. Hatalar, her iki versiyon içinde çoğunlukla 0.0-0.1 aralığındadır.



Şekil 0.25. CVSS 2.0 hesaplama hatalarının histogramı



Şekil 0.26. CVSS 3.1 hesaplama hatalarının histogramı

Önem skorları, güvenlik açığının değerlendirilmesi için ilk yol gösterici bilgiyi vermektedir. Yapılan deneysel çalışmalarda yuvarlama hatalarının gerçek değerler üzerinde de oluşabildiği görülmüştür. Bu nedenle tahmin değerlerindeki küçük hataların göz ardı edilebileceğini düşünüyoruz. Elde edilen sonuçlar, bu açıdan değerlendirilirse, modelin başarımının daha doğru analiz edilebileceği söylenebilir.

5.3. Tartışma

Modern dünyanın ayrılmaz bir parçası haline gelen bilgi iletişim teknolojilerinin güvenliği çok önemli bir konudur. Tüm bilgi iletişim teknolojilerinin temel parçası ise yazılımlardır. Bu durumda yazılımların güvenliği bunlarla oluşturulan sistemlerin güvenliğini doğrudan etkilemektedir. Bu durumun farkında olan uzmanlar tarafından daha güvenli sistemler geliştirilmesine katkı sağlamak ve tarafların güvenlik açıklarından doğacak etkilenmelerini minimize etmek için güvenlik açıkları listelenmekte, analiz edilerek raporlanmakta ve sınıflandırılmaktadır. Bu işlem günümüzde insanlar tarafından manuel olarak yapılmaktadır. Yazılım güvenlik açıklarının son yıllarda durdurulamaz artışı ve doğurduğu sonuçlar ortadadır. Tespit edilen güvenlik açıklarının sahip olduğu potansiyel risklerin uzman olmayan kişiler tarafından da algılanabilmesi için derecelendirilmeleri önemlidir. Yazılım güvenlik açığı raporlarının teknik açıklamaları doğal dille insanlar tarafından yazılan oldukça kısa metinlerdir. Bu çalışmada bu kısıtlı metin bilgileri kullanılarak doğal dil işleme teknikleri ve sınıflandırma algoritmaları ile sınıflandırma yapan bir model önerilmiştir. Bu bölümde elde edilen bulguların ayrıntılı bir tartışması sunulmuştur. Bu noktada önerdiğimiz model ile uzmanlar tarafından oluşturulan teknik açıklamaların kullanılabileceğini öngördük. Doğal dille yazılmış bu metinlerin işlenebilmesi için istatistiki ve kelime gömme özellik çıkarım yöntemi ve beş çok sınıflı sınıflandırma algoritmasından oluşan hibrit bir yapı oluşturulmuştur.

Önerilen yöntem, doğal dil işleme tekniklerinden istatistiki ve kelime gömme olarak adlandırılan yöntemleri kullanmaktadır. Bu yöntemlerden literatürde en çok kullanılan istatistiki yöntemlerden BoW, Tf-Idf, Ngram ve kelime gömme yöntemlerinden Word2Vec, Doc2Vec, FastText yöntemleri tercih edilmiştir. Ayrıca sınıflandırma aşamasında beş farklı sınıflandırma algoritması kullanılarak olabildiğince modelimizin sonuçlarının geniş ve kapsayıcı olması amaçlanmıştır. Çok sınıflı ve literatürde oldukça zor bir sınıflandırma işlemi olarak tanımlanan araştırma problemine karşı önerdiğimiz modellerin sonuçları oldukça umut vericidir. Ancak güvenlik açığı raporlarının sınıflandırılmasında kullanılan sistemler resmi olarak iki farklı versiyon olarak sunulmaktadır. Bütüncül bir bakış açısıyla multi-class sınıflandırma açısından tüm modellerin belli ölçüde başarılı sonuçlar verdiğini değerlendirebiliriz. Ancak oldukça farklı güvenlik şemasına ve sınıflandırmasına sahip olan bu iki versiyonun sonuçlarını ayrı ayrı tartışmak gerekmektedir.

Yazılım güvenlik açığı raporlarının sınıflandırılmasında kullanılan skorlama sistemlerinden CVSS 2.0 veriyonunun bulguları incelendiğinde tüm metin vektörü oluşturma yöntemlerinden Word2Vec'in diğer yöntemlere göre daha başarılı sonuçlar ürettiği görülmektedir. Özellikle KNN algoritması ile elde ettiği sonuçlar ile CVSS 2.0 güvenlik metriklerinin tahmininde önerdiğimiz ilk modeldir. Dahası önerilen yöntem litetatürdeki diğer çalışmalar ile karşılaştırıldığında da en başarı model olarak görülmektedir. Bu aşamada modelimizin başarısı önceki çalışmaların veri setleri ile de eğitilerek ortaya konmuştur. Farklı veri setleri ile yapılan denemeler sonucu elde edilen karşılaştırmalar önerilen modelin sonuçlarının değerlendirilmesinde daha doğru bir yaklaşımdır.

Yazılım güvenlik açığı skorlama sisteminin CVSS 3.1 versiyonun sonuçlarını değerlendirdiğimizde istatistiki yöntemlerinden Tf-Idf yönteminin ön plana çıktığını görmekteyiz. BoW yönteminden az bir farkla da olsa daha iyi tahmin değerlerine sahip olmuştur. KNN algoritması bu modelimizin tahmin başarısında da oldukça başarılı sonuçlar üretmiştir. Ancak MLP algoritmasının sonuçları da oldukça yüksektir. Literatürde kullanılan istatistiki ve kelime gömme yöntemleri ile yapılan karşılaştırmalarda kelime gömme yöntemlerinin sınıflandırma başarılarının özellikle istatistiki yöntemlerin başarısının gerisinde kaldığı görülmektedir. Bu durum veri seti boyutu azaldıkça istatiksel yöntemlerin, kelime gömme yöntemlerinden daha başarılı olduğunu göstermektedir.

Mevcut metodolojinin farklı özellik vektörü oluşturma eğilimi sebebi ile kullanılan sınıflandırma algoritmalarında farklı değerler elde edilmiştir. İstatistiki yöntemlerin sonuçları değerlendirildiğinde KNN algoritmasının başarısı açıkça ortaya çıkmaktadır. Neredeyse tüm özellik vektörleri ile en iyi sonuçları elde etmiştir. Neredeyse dememizin nedeni bazı metrik değerlerinde diğer algoritmalarla aynı başarı değerine sahip olmasındandır. Başarı açısından Decision Tree algoritmasının ikinci en iyi sınıflandırma performansına sahip olduğunu söyleyebiliriz. Özellikle BoW modelinde CVSS 2.0 vektörlerinin tamamında en iyi sonuçları elde etmiştir. Ayrıca CVSS 3.1 vektörlerinin tahminin 8 metrikten 5 tanesinde 3Ngram yönteminde en başarılı algoritmadır. Modeller içerisinde MLP algoritması CVSS 2.0 metriklerinden 3 Ngram yönteminde 6 metrikten 3 tanesinde ve CVSS 3.1 metriklerinden TF-IDF ile sadece birinde en yüksek sonucu elde ederek üçüncü en başarılı algoritma olmuştur. Sınıflandırma algoritmalarından Naive Bayes ve Random Forest modellerin hiçbirinde en yüksek sonuçları elde etmeyi başaramamışlardır.

Kelime gömme yöntemleri ile kullanılan sınıflandırma algoritmaları açısından model performansları karşılattırıldığında tüm algoritmalar içerisinde KNN algoritmasının en yüksek başarı değerlerine sahip olduğu görülmektedir. Ancak diğer algoritmalardan özellikle MLP algoritmasının sonuçları KNN algoritmasına oldukça yakındır. KNN algoritması CVSS 2.0 güvenlik metriklerinin altısının tamamında en yüksek başarı oranına sahip algoritma olmuştur. Ancak CVSS 3.1 versiyonun ait sekiz güvenlik metriğinden altı tanesinde KNN en yüksek başarı

oranına sahipken, iki tanesinde MLP algoritması daha yüksek başarıyı elde etmiştir. Diğer algoritmalar açısından RFC ve DT algoritmaları birbirine yakın sonuçlar ile diğer başarılı algoritmalardır. Ancak NB algoritması sonuçları itibari ile diğer algoritmalara göre bazı metriklerde başarılı sonuçlar üretse de genel olarak tahmin başarısı en düşük sınıflandırma algoritması olmuştur.

Yazılım güvenlik açıklarının önem derecelerinin doğrudan tahmin edilmesi hipotezi bu tez kapsamında araştırılmıştır. CVSS 2.0 sonuçları incelendiğinde FastText yönteminin NB dışındaki tüm sınıflandırma algoritmaları ile üstün bir başarı ortaya koyduğu açıktır. Dahası Word2Vec yöntemi KNN algoritması ile en başarılı ikinci değere sahip yöntem olarak karşımıza çıkmaktadır. Word embedding yöntemlerin bu problem özelinde başarısını ortaya koymaktadır. İstatistiki yöntemler incelendiğinde TF-IDF yönteminin KNN algoritması ile en başarılı istatistiki model olmuştur. Ancak diğer yöntemlerin sonuçları da oldukça yakındır ve başarılı sayılabilecek düzeydedir. Nispeten yeni bir versiyon olan ve üzerinde henüz olgunlaşmış akademik çalışmalar yapılmamış olan CVSS 3.1 versiyonunun tahmin değerleri CVSS 2.0'a göre oldukça düşmektedir. Unutulmamalıdır ki CVSS 3.1 versiyounu hem tahmin edilmesi gereken sınıf sayısı, hem de veri seti boyutu bakımından daha zor bir problemi ifade etmektedir. Bu tez çalışmamız CVSS 3.1 versiyonunu kullanan ilk çalışmalardan olmasından dolayı bir referans noktası oluşturacaktır. Özellikle CVSS 2.0 versiyonunda Word embedding modellerin başarıları ön plana çıkarken CVSS 3.1 versiyonunda istatistiki yöntemlerin tahmin performansları dikkat çekmektedir. Özellikle TF-IDF ve BoW yöntemlerinin KNN algoritması ile gösterdiği performans ön plana çıkmaktadır. Dahası KNN algoritması Ngram yönteminde de başarılı sayılabilecek bir sonuç göstermektedir. CVSS 2.0 versiyonunun tahmininde oldukça başarılı sonuçlar veren FastText ve Word2Vec yöntemleri göreceli olarak CVSS 3.1 versiyonunda da KNN algoritması ile başarılı sayılabilecek sonuçlar üretse de istatistiki modellerin oldukça gerisinde kalmıştır. Doc2Vec elde edilen sonuçlar yönünden en başarısız sonuçlar üreten yöntem olarak görülmektedir. Doc2Vec modelleri iki sistem için de en düşük başarı değerlerine sahip olmuştur. Ngram yöntemleri kendi içerisinde değerlendirildiğinde N değerindeki artışın tahmin başarısını aşağı çektiği açıkça görülmektedir.

Önerilen metodolojinin bir diğer amacı önem skorunun hesaplanmasıdır. Önem puanları güvenlik açığı vektörlerinin sahip oldukları metriklerin bir fonksiyonudur. Bu nedenle her güvenlik açığı vektörü farklı metrik değerlerinin birleşmesi ile oluşturulur. Bu sebeple öne sürülen hipotez, başarı oranı en yüksek olan özellik seçim yöntemi ile sınıflandırmada en iyi sonucu elde eden algoritmalardan hibrit bir model oluşturmaktır. Bu sayede yazılım güvenlik açığı vektörlerinin temsil derecelerinin en doğru şekilde elde edileceği varsayılmıştır. Açıklandığı şekilde test kümesi için tüm metrikler oluşturulmuştur. Sonrasında CVSS 2.0 ve CVSS 3.1 için önem skorları hesaplanmıştır. Elde edilen sonuçlar üç farklı değerlendirme ölçütü ve istatistiki değerler ile karşılaştırılmıştır. Ölçüm sonuçlarındaki hataların farklı yöntemler ile toplamı oldukça düşüktür.

Sonuç olarak önerilen yöntemin, önem skorlarını yeterince doğru tahmin edebildiği ortaya konulmuştur. İki skorlama sistemi için hata oranlarının histogramları verilmiştir. İncelendiğinde hata oranlarının çoğunun oldukça düşük olduğu görülmektedir. Özellikle hesaplamalarda kullanılan yuvarlama işlemlerinin sonuçlarda küçük hatalara neden olması kaçınılmazdır. Ayrıca gerçek değerler ile hesaplanan değerlerin istatistiki karşılaştırmasında veri kümelerinin ortalama değerleri çok yakın çıkmıştır. Ayrıca iki küme arasındaki korelasyon önerilen yöntemin hesaplama sonuçlarının temel bir öngörü için değerlendirilebilir olduğunu ortaya koymaktadır.

6. SONUÇLAR

Yazılım güvenlik açıkları, insanların günlük işlemlerinin büyük çoğunluğunu bilgi sistemlerinin desteğiyle gerçekleştirildiği çağdaş toplumda büyük bir tehdit oluşturmaktadır. NVD veri tabanı istatistiklerine göre yazılım güvenlik açıkları son dört yılda %100'ün üzerinde artış göstermiştir. Bu nedenle bir yazılım sistemi, geliştirme sürecinden itibaren, temel güvenlik ilkelerini destekleyebilmeli ve sahip olacağı akıllı modeller yardımı ile güvenliği kontrol altında tutabilmelidir. Bu sorunun çözümüne katkı sağlamak amacıyla doğal dil işleme tekniklerini ve çok sınıflı sınıflandırma algoritmalarını kullanan bir model önerilmiştir. Literatürde kullanılan pek çok yöntem istenilen başarı oranını yakalayamamaktadır. Elde edilen sonuçlar umut vericidir. Ancak literatürdeki önerilen diğer metodolojiler gibi mevcut çalışmanın önerdiği model de henüz insanların yerini alabilecek durumda değildir. Ancak zaman alan ve uzmanlık gerektiren güvenlik açıklarının analizi ve keşfi alanında uzmanlara yardımcı olacak ve alacakları kararları hızlandıracaktır.

Önerilen metodolojinin sonuçları, bu modelin yazılım güvenlik açıklarının skorlanmasını başarı ile yapabileceğini göstermiştir. Elde edilen sonuçlardan hata oranlarının oldukça düşük olduğu görülmektedir. Deneysel çalışmalar, kamuya açık en büyük ve güvenilir veri seti kullanılarak yapıldığı için analizi ve doğrulamasının yapılması kolaydır. Literatürde daha önce bildirilen sonuçlarla karşılaştırdığımız sonuçlarımız sayesinde önerilen metodolojinin başarısı ortadadır. Ayrıca çalışmada kullanılan veri seti boyutu ile alanında en geniş kapsamlı çalışmadır.

Önerilen yöntemde, doğal dil işleme tekniklerinden son dönemlerde yaygın olarak kullanılmaya başlanan kelime gömme ve geleneksel istatistiki teknikler tercih edilmiştir. Yapılan analizlerin sonuçları incelediğinden kelime gömme yöntemlerinin sonuçlarının özellikle veri seti boyutu artıkça daha başarılı sonuçlar verdiği görülmektedir. Bu da bizlere word embedding yöntemlerinin yüksek boyutlu veriler ile daha başarılı sonuçlar ürettiğini açıkça göstermektedir. istatistiki doğal dil işleme yöntemleri ile sonuçlar karşılaştırıldığında bu durum daha açık olarak görülmektedir. Özellikle veri seti boyutunun azalmasından kelime gömme yöntemlerinin başarı oranları oldukça etkilenmektedir. Ancak istatistiki yöntemlerin veri seti boyutu az veriler ile daha başarı olduğu görülmektedir. Ayrıca önerilen modelde temel olarak doküman frekansına göre sınıflandırma yapılmasına rağmen Doc2Vec her iki versiyon için en düşük sonuçları üreten yöntem olmuştur. Bu durumun kullanılan veri setindeki teknik açıklamaların oldukça kısa metinlerden oluşmasından kaynaklandığını düşünmekteyiz. Dahası veri seti azaldıkça FastText yönteminin başarısının diğer kelime gömme yöntemlerinin önünde olduğu görülmektedir. Bu durumun FastText algoritmasının kelimelerin morfolojik özelliklerine odaklanmasının bir sonucu olduğu kanaatindeyiz. Bu sayede düşük boyutlu veri setlerindeki kelimeler arasındaki anlamsal ilişkileri

daha doğru tahmin ettiği açıktır. Yapılan tüm karşılaştırmalar ve analizler göstermektedir ki öz nitelik çıkarım yöntemleri sınıflandırma algoritmalarının başarısında doğrudan etkilidir.

ÖNERİLER

Çalışmamızda makine öğrenmesi ve veri madenciliğinin bu problemler özelinde avantajları ve sınırlamaları sunulmuştur. Ayrıca literatürde daha önce önerilen yöntem ve farklı veri setleri ile karşılıklı analizler yapılmıştır. Çalışma bu nedenle yazılım güvenlik açıklarının veri madenciliği ve makine öğrenmesi algoritmaları ile analizi ve keşfi üzerine bir hipotez niteliğindedir. Bu bağlamda çalışmanın bir sonucu olarak alandaki zorlukların ve bu alanında gelecekte yapılabilecek çalışmalara ilham verebilecek bazı açık alanlar tespit edilmiştir.

Özellikle son dönemde tüm araştırmacılara derin öğrenme üzerine çalışmalar yapılması tavsiye edilmektedir. Gelecekteki çalışmalarda derin öğrenme yöntemleri ile çalışmaların genişletilmesini öneriyoruz. Derin öğrenme algoritmalarını öznitelik çıkarımı ve sınıflandırma aşamalarında kullanmanın sınıflandırma performansını artırabileceğini öngörüyoruz. Ayrıca farklı öznitelik ve doğal dil işleme tekniklerinin kullanımının sınıflandırma performansını etkilediği açıktır. Bu nedenle farklı yöntemler ile sınıflandırma başarılarının artırılması yönünde çalışmaların yapılması gerektiğini düşünmekteyiz.

Ayrıca farklı veri tabanlarındaki verileri inceleyerek farklı güvenlik açığı veri tabanlarıyla çalışmaların genişletilmesini tavsiye ediyoruz. Bu sayede farklı veri tabanları tarafından sunulan verilerin tutarlığını ve genel kabul görmüş veri tabanı sağlayıcıları dışında kalan veri tabanlarının önemlerinin ortaya çıkarılabileceğini düşünüyoruz. Bu aşamada özellikle birikmiş ve sürekli gelmeye devam eden verilerin yeni bir yapıyla raporlanması ve arşivlenmesi ihtiyacının olduğunu düşünmekteyiz.

KAYNAKLAR

[1] L.P. Kobek, The State of Cybersecurity in Mexico: An Overview, Wilson Centre's Mex. Institute, Jan. (2017).

[2] T.W. Moore, C.W. Probst, K. Rannenberg, M. van Eeten, Assessing ICT Security Risks in Socio-Technical Systems (Dagstuhl Seminar 16461), Dagstuhl Reports. 6 (2017) 63–89. https://doi.org/10.4230/DagRep.6.11.63.

[3] J. Ruohonen, A look at the time delays in CVSS vulnerability scoring, Appl. Comput. Informatics. 15 (2019) 129–135. https://doi.org/10.1016/j.aci.2017.12.002.

[4] C. Theisen, L. Williams, Better together: Comparing vulnerability prediction models, Inf. Softw. Technol. 119 (2020). https://doi.org/10.1016/j.infsof.2019.106204.

[5] S.M. Ghaffarian, H.R. Shahriari, Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey, ACM Comput. Surv. 50 (2017). https://doi.org/10.1145/3092566.

[6] X. Wu, W. Zheng, X. Chen, F. Wang, D. Mu, CVE-assisted large-scale security bug report dataset construction method, J. Syst. Softw. 160 (2020) 110456. https://doi.org/10.1016/j.jss.2019.110456.

[7] R. Raducu, G. Esteban, F.J.R. Lera, C. Fernández, Collecting vulnerable source code from open-source repositories for dataset generation, Appl. Sci. 10 (2020). https://doi.org/10.3390/app10041270.

[8] D. Miyamoto, Y. Yamamoto, M. Nakayama, Text-mining approach for estimating vulnerability score, Proc. - 2015 4th Int. Work. Build. Anal. Datasets Gather. Exp. Returns Secur. BADGERS 2015. (2017) 67–73. https://doi.org/10.1109/BADGERS.2015.12.

[9] G. Spanos, L. Angelis, A multi-target approach to estimate software vulnerability characteristics and severity scores, J. Syst. Softw. 146 (2018) 152–166. https://doi.org/10.1016/j.jss.2018.09.039.

[10] H. Yang, S. Park, K. Yim, M. Lee, Better not to use vulnerability's reference for exploitability prediction, Appl. Sci. 10 (2020). https://doi.org/10.3390/app10072555.

[11] V.-V. Patriciu, I. Priescu, S. Nicolaescu, Security metrics for enterprise information systems, J. Appl. Quant. Methods. 1 (2006) 151–159.

[12] NVD, NVD, Natl. Vulnerability Database. (2020). https://nvd.nist.gov (accessed July 25, 2020).

[13] E.R. Russo, A. Di Sorbo, C.A. Visaggio, G. Canfora, Summarizing vulnerabilities' descriptions to support experts during vulnerability assessment activities, J. Syst. Softw. 156 (2019) 84–99. https://doi.org/10.1016/j.jss.2019.06.001.

[14] E. Yasasin, J. Prester, G. Wagner, G. Schryen, Forecasting IT security vulnerabilities – An empirical analysis, Comput. Secur. 88 (2020) 101610. https://doi.org/10.1016/j.cose.2019.101610.

[15] R. Sharma, R. Sibal, S. Sabharwal, Software vulnerability prioritization using vulnerability description, Int. J. Syst. Assur. Eng. Manag. 12 (2021) 58–64. https://doi.org/10.1007/s13198-020-01021-7.

[16] R. Malhotra, Vidushi, Severity Prediction of Software Vulnerabilities Using Textual Data, in: V.K. Gunjan, J.M. Zurada (Eds.), Proc. Int. Conf. Recent Trends Mach. Learn. IoT, Smart Cities Appl., Springer Singapore, Singapore, 2021: pp. 453–464.

[17] M. Aota, H. Kanehara, M. Kubo, N. Murata, B. Sun, T. Takahashi, Automation of Vulnerability Classification from its Description using Machine Learning, in: 2020 IEEE Symp. Comput. Commun., 2020: pp. 1–7. https://doi.org/10.1109/ISCC50000.2020.9219568.

[18] Y. Fang, Y. Liu, C. Huang, L. Liu, Fastembed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm, PLoS One. 15 (2020) 1–28. https://doi.org/10.1371/journal.pone.0228439.

[19] C.B. Şahin, Ö.B. Dinler, L. Abualigah, Prediction of software vulnerability based deep symbiotic genetic algorithms: Phenotyping of dominant-features, Appl. Intell. 51 (2021) 8271–8287. https://doi.org/10.1007/s10489-021-02324-3.

[20] I.V. Krsul, Software vulnerability analysis, Purdue University, 1998.

[21] A. Ozment, Improving vulnerability discovery models: Problems with definitions and assumptions, Proc. ACM Conf. Comput. Commun. Secur. (2007) 6–11. https://doi.org/10.1145/1314257.1314261.

[22] I.S.C. Committee, others, IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos, CA IEEE Comput. Soc. 169 (1990).

[23] A. Kuehn, M. Mueller, Shifts in the cybersecurity paradigm: Zero-day exploits, discourse, and emerging institutions, in: Proc. 2014 New Secur. Paradig. Work., 2014: pp. 63–68.

[24] O. Bozoklu, C.Z. Çil, Yazılım Güvenlik Açığı Ekosistemi Ve Türkiye'deki Durum Değerlendirmesi, Uluslararası Bilgi Güvenliği Mühendisliği Derg. 3 (2017) 6–26.

[25] Mitre Corporation, (2020). https://www.mitre.org (accessed July 25, 2020).

[26] CVE, CVE, Common Vulnerabilities Expo. (2020). https://cve.mitre.org (accessed July 25, 2020).

[27] G. Schryen, Security of open source and closed source software: An empirical comparison of published vulnerabilities, AMCIS 2009 Proc. (2009) 387.

[28] G. Schryen, Is Open Source Security a Myth?, Commun. ACM. 54 (2011) 130–140. https://doi.org/10.1145/1941487.1941516.

[29] ExploitDB, Exploit Database, (2020). https://www.exploit-db.com (accessed July 25, 2020).

[30] Rapid7, Rapid7, (2020). https://www.rapid7.com/db/ (accessed July 25, 2020).

[31] SecurityFocus, SecurityFocus, (2020). https://www.securityfocus.com (accessed July 25, 2020).

[32] Snyk, Snyk, (2020). https://snyk.io (accessed July 25, 2020).

[33] SARD, SARD-Software Assurance Reference Dataset Project, (2020). https://samate.nist.gov (accessed July 25, 2020).

[34] G. Spanos, A. Sioziou, L. Angelis, WIVSS: A New Methodology for Scoring Information Systems Vulnerabilities, in: Proc. 17th Panhellenic Conf. Informatics, Association for Computing Machinery, New York, NY, USA, 2013: pp. 83–90. https://doi.org/10.1145/2491845.2491871.

[35] G. Spanos, L. Angelis, Impact metrics of security vulnerabilities: Analysis and weighing, Inf. Secur. J. A Glob. Perspect. 24 (2015) 57–71.

[36] M. Schiffman, C.I.A.G. Cisco, A Complete Guide to the Common Vulnerability Scoring System (CVSS) v1 Archive, (2005). https://www.first.org/cvss/v1/guide (accessed January 1, 2021).

[37] P. Mell, K. Scarfone, S. Romanosky, A Complete Guide to the Common Vulnerability Scoring System Version 2.0, FIRSTForum Incid. Response Secur. Teams. (2007) 1–23. https://www.first.org/cvss/cvss-v2-guide.pdf (accessed January 1, 2021).

[38] Common Vulnerability Scoring System v3.0: User Guide, (n.d.). https://www.first.org/cvss/v3.0/user-guide (accessed January 1, 2021).

[39] Common Vulnerability Scoring System v3.1: User Guide, (n.d.). https://www.first.org/cvss/v3.1/user-guide (accessed January 1, 2021).

[40] A. Fesseha, S. Xiong, E.D. Emiru, M. Diallo, A. Dahou, Text Classification Based on Convolutional Neural Networks and Word Embedding for Low-Resource Languages: Tigrinya, Information. 12 (2021). https://doi.org/10.3390/info12020052.

[41] Y. Zhang, R. Jin, Z.-H. Zhou, Understanding bag-of-words model: a statistical framework, Int. J. Mach. Learn. Cybern. 1 (2010) 43–52.

[42] A. Aizawa, An information-theoretic perspective of tf--idf measures, Inf. Process. Manag. 39 (2003) 45–65.

[43] S. Banerjee, T. Pedersen, The design, implementation, and use of the ngram statistics package, in: Int. Conf. Intell. Text Process. Comput. Linguist., 2003: pp. 370–381.

[44] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Adv. Neural Inf. Process. Syst., 2013: pp. 3111–3119.

[45] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, ArXiv Prepr. ArXiv1301.3781. (2013).

[46] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Int. Conf. Mach. Learn., 2014: pp. 1188–1196.

[47] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2017) 135–146.

[48] S. Aggarwal, D. Kaur, Naïve Bayes Classifier with Various Smoothing Techniques for Text Documents, in: 2013.

[49] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and regression trees, CRC press, 1984.

[50] E. Fix, Discriminatory analysis: nonparametric discrimination, consistency properties, USAF school of Aviation Medicine, 1951.

[51] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133.

[52] L. Breiman, Random Forests, Mach. Learn. 45 (2001) 5–32. https://doi.org/10.1023/A:1010933404324.

[53] G. Van Rossum, F.L. Drake, Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, 2009.

[54] T. pandas development team, pandas-dev/pandas: Pandas, (2020). https://doi.org/10.5281/zenodo.3509134.

[55] W. McKinney, Data Structures for Statistical Computing in Python, in: S. van der Walt, J. Millman (Eds.), Proc. 9th Python Sci. Conf., 2010: pp. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a.

[56] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with {NumPy}, Nature. 585 (2020) 357–362. https://doi.org/10.1038/s41586-020-2649-2.

[57] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, " O'Reilly Media, Inc.," 2009.

[58] Ř. Radim, P. Sojka, Software Framework for Topic Modelling with Large Corpora, in: Proc. Lr. 2010 Work. New Challenges NLP Fram., ELRA, Valletta, Malta, 2010: pp. 45–50.

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine Learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html.

[60] H. Booth, D. Rike, G.A. Witte, others, The national vulnerability database (nvd): Overview, (2013).

[61] H. Kekül, B. Ergen, H. Arslan, A multiclass hybrid approach to estimating software vulnerability vectors and severity score, J. Inf. Secur. Appl. 63 (2021) 103028. https://doi.org/https://doi.org/10.1016/j.jisa.2021.103028.

[62] A.K. Uysal, S. Gunal, The impact of preprocessing on text classification, Inf. Process. Manag. 50 (2014) 104–112. https://doi.org/https://doi.org/10.1016/j.ipm.2013.08.006.

[63] A.A. Jalal, B.H. Ali, Text documents clustering using data mining techniques., Int. J. Electr. Comput. Eng. 11 (2021).

[64] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, Text classification algorithms: A survey, Information. 10 (2019) 150.

[65] M. Aydoğan, A. Karci, Turkish Text Classification with Machine Learning and Transfer Learning, in: 2019 Int. Artif. Intell. Data Process. Symp., 2019: pp. 1–6. https://doi.org/10.1109/IDAP.2019.8875919.

[66] N. Kejriwal, S. Kumar, T. Shibata, High performance loop closure detection using bag of word pairs,

Rob. Auton. Syst. 77 (2016) 55–65.

[67] S. Kim, Y. Choi, J.-H. Won, J. Mi Oh, H. Lee, An annotated corpus from biomedical articles to construct a drug-food interaction database, J. Biomed. Inform. 126 (2022) 103985. https://doi.org/https://doi.org/10.1016/j.jbi.2022.103985.

[68] H.P. Luhn, A statistical approach to mechanized encoding and searching of literary information, IBM J. Res. Dev. 1 (1957) 309–317.

[69] I. Arroyo-Fernández, C.-F. Méndez-Cruz, G. Sierra, J.-M. Torres-Moreno, G. Sidorov, Unsupervised sentence representations as word information series: Revisiting TF–IDF, Comput. Speech Lang. 56 (2019) 107–129. https://doi.org/https://doi.org/10.1016/j.csl.2019.01.005.

[70] K.S. Jones, A statistical interpretation of term specificity and its application in retrieval, J. Doc. (1972).

[71] A. Tripathy, A. Agrawal, S.K. Rath, Classification of sentiment reviews using n-gram machine learning approach, Expert Syst. Appl. 57 (2016) 117–126.

[72] Y. su, R. Lin, C. Kuo, On Tree-structured Multi-stage Principal Component Analysis (TMPCA) for Text Classification, (2018).

[73] Z. Yin, Y. Shen, On the dimensionality of word embedding, ArXiv Prepr. ArXiv1812.04224. (2018).

[74] R. Kohavi, others, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai, 1995: pp. 1137–1145.

[75] G.C. Cawley, N.L.C. Talbot, On over-fitting in model selection and subsequent selection bias in performance evaluation, J. Mach. Learn. Res. 11 (2010) 2079–2107.

[76] S. Russell, P. Norvig, Artificial intelligence: a modern approach, (2002).

[77] J.D. Rennie, L. Shih, J. Teevan, D.R. Karger, Tackling the poor assumptions of naive bayes text classifiers, in: Proc. 20th Int. Conf. Mach. Learn., 2003: pp. 616–623.

[78] E.K. Mallory, A. Acharya, S.E. Rensi, P.J. Turnbaugh, R.A. Bright, R.B. Altman, Chemical reaction vector embeddings: towards predicting drug metabolism in the human gut microbiome., in: PSB, 2018: pp. 56–67.

[79] B. Kamiński, M. Jakubczyk, P. Szufel, A framework for sensitivity analysis of decision trees, Cent. Eur. J. Oper. Res. 26 (2018) 135–159. https://doi.org/10.1007/s10100-017-0479-6.

[80] J.R. Quinlan, Simplifying decision trees, Int. J. Man. Mach. Stud. 27 (1987) 221–234. https://doi.org/https://doi.org/10.1016/S0020-7373(87)80053-6.

[81] Y. Yang, An evaluation of statistical approaches to text categorization, Inf. Retr. Boston. 1 (1999) 69–90.

[82] X. Deng, Y. Li, J. Weng, J. Zhang, Feature selection for text classification: A review, Multimed. Tools Appl. 78 (2019) 3797–3816.

[83] Z. Chen, L.J. Zhou, X. Da Li, J.N. Zhang, W.J. Huo, The Lao Text Classification Method Based on KNN, Procedia Comput. Sci. 166 (2020) 523–528. https://doi.org/https://doi.org/10.1016/j.procs.2020.02.053.

[84] Y. Tan, An Improved KNN Text Classification Algorithm Based on K-Medoids and Rough Set, in: 2018 10th Int. Conf. Intell. Human-Machine Syst. Cybern., 2018: pp. 109–113. https://doi.org/10.1109/IHMSC.2018.00032.

[85] D.A. Simanjuntak, H.P. Ipung, A.S. Nugroho, others, Text classification techniques used to faciliate cyber terrorism investigation, in: 2010 Second Int. Conf. Adv. Comput. Control. Telecommun. Technol., 2010: pp. 198–200.

[86] F. Rosenblatt, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, 1961.

[87] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, 1985.

[88] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control. Signals Syst. 5 (1992) 455.

[89] K. Shah, H. Patel, D. Sanghvi, M. Shah, A comparative analysis of logistic regression, random forest and KNN models for the text classification, Augment. Hum. Res. 5 (2020) 1–16.

[90] Y. Sun, Y. Li, Q. Zeng, Y. Bian, Application Research of Text Classification Based on Random Forest Algorithm, in: 2020 3rd Int. Conf. Adv. Electron. Mater. Comput. Softw. Eng., 2020: pp. 370–374. https://doi.org/10.1109/AEMCSE50948.2020.00086.

[91] S. Sawangarreerak, P. Thanathamathee, Random Forest with Sampling Techniques for Handling Imbalanced Prediction of University Student Depression, Information. 11 (2020). https://doi.org/10.3390/info11110519.

[92] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (2009) 427–437. https://doi.org/https://doi.org/10.1016/j.ipm.2009.03.002.

[93] C. Bielza, G. Li, P. Larrañaga, Multi-dimensional classification with Bayesian networks, Int. J. Approx. Reason. 52 (2011) 705–727. https://doi.org/https://doi.org/10.1016/j.ijar.2011.01.007.

[94] D. Ballabio, F. Grisoni, R. Todeschini, Multivariate comparison of classification performance measures, Chemom. Intell. Lab. Syst. 174 (2018) 33–44. https://doi.org/https://doi.org/10.1016/j.chemolab.2017.12.004.

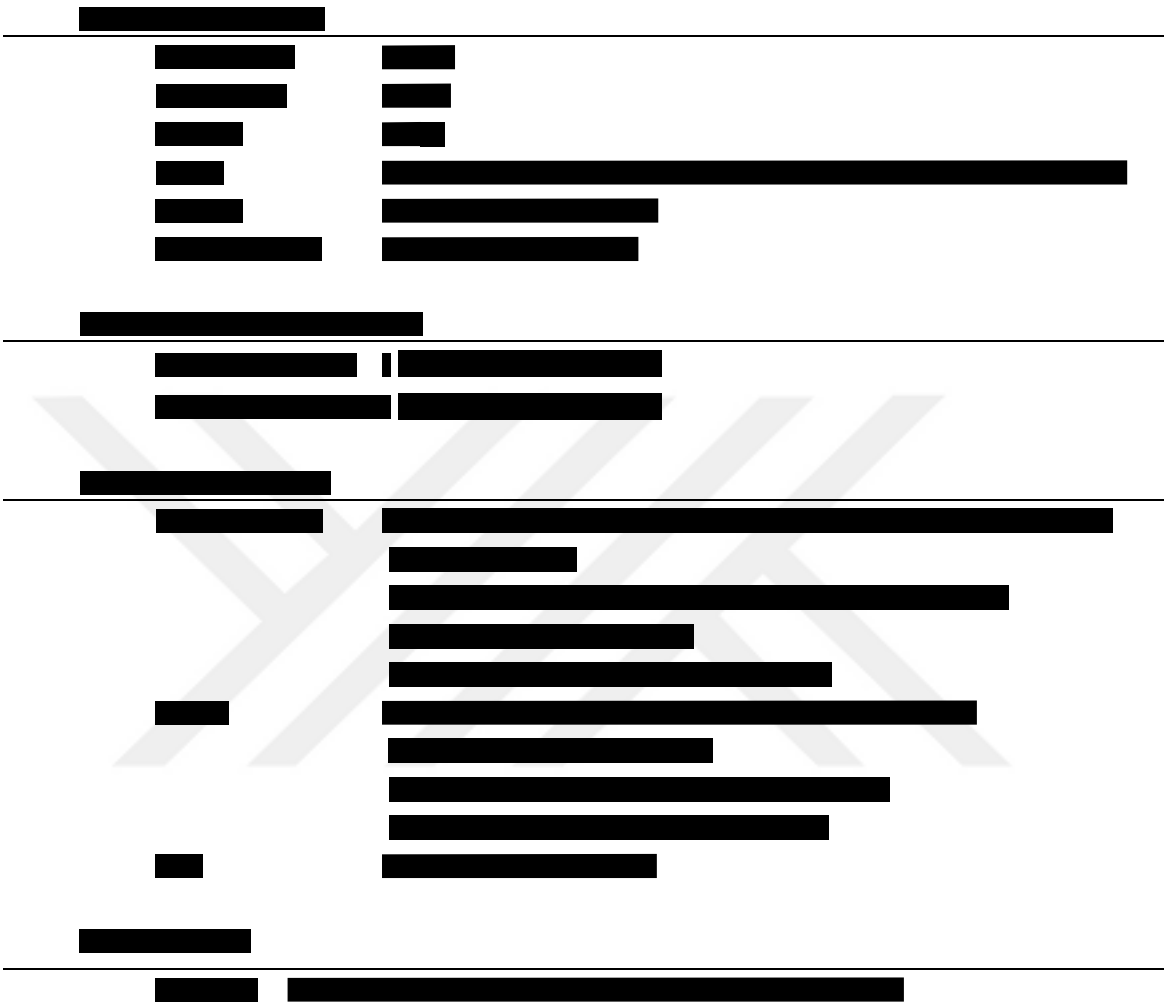
[95] F.D. János, N. Huu Phuoc Dai, Security Concerns Towards Security Operations Centers, in: 2018 IEEE 12th Int. Symp. Appl. Comput. Intell. Informatics, 2018: pp. 273–278. https://doi.org/10.1109/SACI.2018.8440963.

[96] K. Kritikos, K. Magoutis, M. Papoutsakis, S. Ioannidis, A survey on vulnerability assessment tools and databases for cloud-based web applications, Array. 3–4 (2019) 100011. https://doi.org/10.1016/j.array.2019.100011.

ÖZGEÇMİŞ

Hakan KEKÜL

[redacted]



[redacted]

AKADEMİK FAALİYETLER

Makaleler:

1. Kekül, H., Ergen, B., & Arslan, H. (2021). A multiclass hybrid approach to estimating software vulnerability vectors and severity score. Journal of Information Security and Applications, 63, 103028. DOI: 10.1016/j.jisa.2021.103028
2. Kekül, H., Ergen, B., & Arslan, H. (2022). Estimating vulnerability metrics with word embedding and multiclass classification methods. Information and Software Technology
3. Kekül, H., Ergen, B., & Arslan, H. (2021). A new vulnerability reporting framework for software vulnerability databases. International Journal of Education and Management Engineering (IJEME), Vol.11, No.3, pp. 11-19, 2021. DOI: 10.5815/ijeme.2021.03.02
4. Kekül, H., Ergen, B., & Arslan, H. (2021). Yazılım Güvenlik Açığı Veri Tabanları. Avrupa Bilim ve Teknoloji Dergisi, (28), 1008-1012. DOI: 10.31590/ejosat.1012410
5. Kekül, H., Ergen, B., & Arslan, H. (2022). A Multiclass Approach to Estimating Software Vulnerability Severity Rating with Statistical and Word Embedding Methods. International Journal of Computer Network and Information Security(IJCNIS)

6. Kekül, H., Ergen, B., & Arslan, H. (2022). Estimating Missing Security Vectors in NVD Database Security Reports. International Journal of Engineering and Manufacturing(IJEM)
7. Kekül, H., Ergen, B., & Arslan, H. (2022). Systematic Review and Comparison of Software Vulnerability Databases. International Journal of Engineering and Manufacturing(IJEM)

Bildiriler:

8. Kekül, H., Ergen, B., & Arslan, H. (2021). Systems Software Vulnerability Databases. 1st International Conference on Applied Engineering and Natural Sciences. s.971. ISBN: 978-625-00-0389-3

Projeler:

1. Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK-1002) Proje No: 121E298, "Yazılım Güvenlik Açıklarının Skorlanması ve Kategorilerinin Belirlenmesinde Yeni Bir Yöntem", Yürütücü: Hakan KEKÜL, Danışman: Prof. Dr. Burhan ERGEN, Proje Bütçesi: 14,200.00 TL