Linking CVE's to MITRE ATT&CK Techniques

Aditya Kuppa
University College, Dublin
Ireland
aditya.kuppa@ucdconnect.ie

Lamine Aouad
Tenable Corporation
France
laouad@tenable.com

Nhien-An Le-Khac
University College, Dublin
Ireland
an.lekhac@ucd.ie

ABSTRACT

The MITRE Corporation is a non-profit organization that has made substantial efforts into creating and maintaining knowledge bases relevant to cybersecurity and has been widely adopted by the community. ATT&CK "Adversarial Tactics, Techniques, and Common Knowledge" is a popular taxonomy by MITRE, which describes threat actor behaviors. Techniques are the foundation of the ATT&CK model, they are the actions that adversaries perform to accomplish goals, which translate into the model's tactics. The aim of ATT&CK is to categorize adversary behavior to help improve the post-compromise detection of advanced intrusions.

Software vulnerabilities (CVE) play an important role in cyber-intrusions, mostly classified into 4 ATT&CK techniques, which cover the exploitation phase of the attack chain. Identifying vulnerabilities that are actively exploited by the attackers, and understanding how a vulnerability can enable the attacker at each stage of the attack life cycle is absolutely critical for vulnerability assessments. Given the sparse classification of a CVE into ATT&CK taxonomy, lack of methods to extract labels from threat reports and, the volume of vulnerabilities disclosed defenders lack a concrete approach to prioritize CVE's based on their role in the attack chain and in the context of controls in place.

In this work, we propose a Multi-Head Joint Embedding Neural Network model to automatically map CVE's to ATT&CK techniques. We address the problem of lack of labels for this task, by a novel unsupervised labeling technique. We enrich CVE's with a curated knowledgebase 50 mitigation strategies, which help the model to learn both attacker and defender view of a given CVE. We evaluate our approach with the dataset containing CVE's disclosed from the past 10 years and compare it with standard baseline models and ablation analysis. Using the proposed model, we mapped 62, 000 CVE records to 37 different ATT&CK techniques and show that the proposed multi head design performs well in the absence of labels in the training dataset.

CCS CONCEPTS

• **Security and privacy** → **Vulnerability scanners**; Software security engineering; • **Computing methodologies** → *Information extraction*; *Knowledge representation and reasoning*.

ARES 2021, August 17–20, 2021, Vienna, Austria

ACM ISBN 978-1-4503-9051-4/21/08.
https://doi.org/10.1145/3465481.3465758

KEYWORDS

CVE, Attack Models, Deep Learning, ATT&CK , unsupervised labeling

ACM Reference Format:
Aditya Kuppa, Lamine Aouad, and Nhien-An Le-Khac. 2021. Linking CVE's to MITRE ATT&CK Techniques. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3465481.3465758

1 INTRODUCTION

Improving security and reducing risk are two of the main concerns, and challenges, for most organizations today. The increasing volume of vulnerabilities, the frenetic expansion of the attack surface, the increasing sophistication of attacks and attackers, and multiple other technological and human factors are making cybersecurity the most critical challenge of the digital age. The rise in the number of high-impact vulnerabilities and the complexity of underlying systems opens up an opportunity for an adversary to exploit the vulnerability to achieve his/her goals.

Furthermore, attacks or incidents in cyberspace have a few unique characteristics that make them particularly challenging to defend against. An intrusion can span across multiple stages [2], for instance, attackers can use techniques like drive-by-download [3] or spear-phishing [4] for initial compromise and weaponize a variety of vulnerabilities for reconnaissance, Command and Control (C&C) communications, privilege escalation, lateral movement, and exfiltration stages of the attack.

To efficiently assess and reduce risk, common practice security practitioners follow is to periodically scan assets for known vulnerabilities and improve their security posture by patching the identified vulnerabilities. It may not always be obvious "which" systems have "what" vulnerabilities and "how" attackers can exploit these vulnerable systems [32, 35]. Also, the vast majority of attacks in the wild are carried by a handful of vulnerabilities [1, 5], and the refresh time of attacks in the wild is as slow as around 600 days (i.e. the same exploits are re-used in the wild for almost two years before they are substituted at scale with a new attack) [7]. We surveyed 690 publicly documented advanced cyber intrusion reports from the past 10 years, which describe different attack steps of advanced cyber intrusions and we also infer similar observation that less than 500 CVE's are actively exploited by attackers in their attack campaigns. To address this asymmetry various risk models have been proposed to help defenders prioritize important and relevant vulnerabilities relevant to the environment, risk appetite, and controls in place.

Understanding the attacker's choice of vulnerability for a particular attack stage is a hard problem and often measured by the exploit availability, reusability, ease, opportunity/development cost,

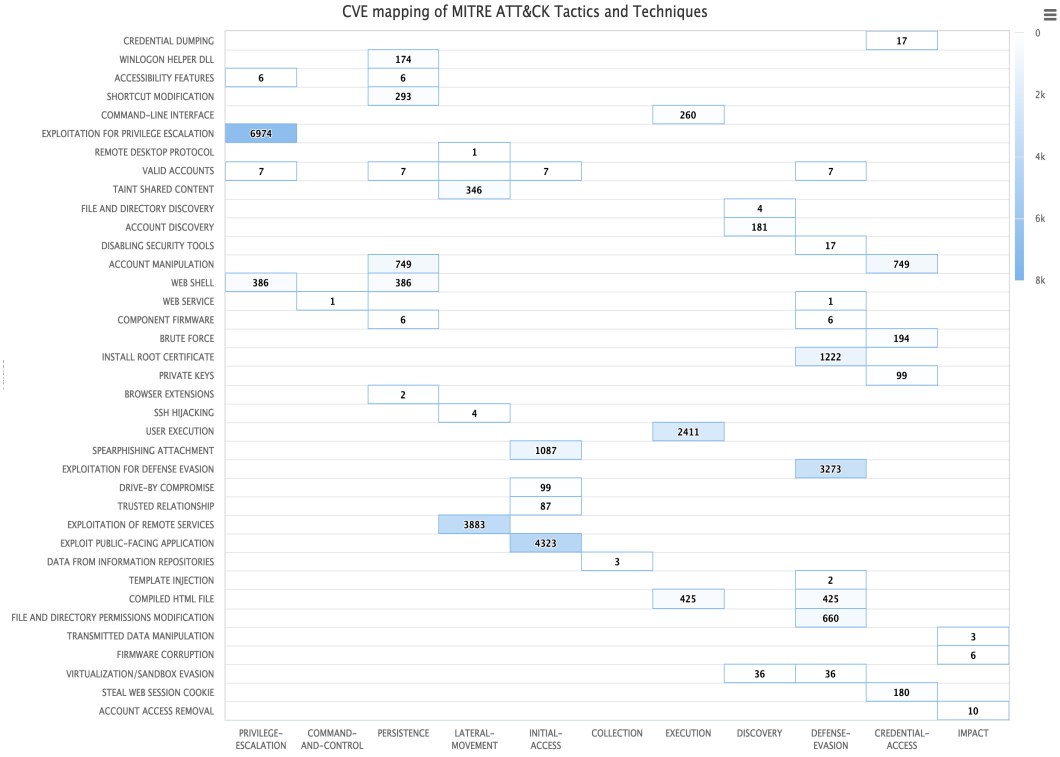

Figure 1: Heat map of MITRE ATT&CK tactics and technique with the all CVE's dataset. The numbers in each cell correspond to CVE count for particular tactic and technique

Attack Group	Common Vulnerabilities and Exposures (CVE)
APT28	CVE-2014-4076, CVE-2015-3043, CVE-2015-2424, CVE-2016-7855, CVE-2015-5119, CVE-2015-1701, CVE-2016-7255, CVE-2015-2590
APT3	CVE-2016-7255, CVE-2014-4113, CVE-2014-1776, CVE-2014-6332, CVE-2015-3113, CVE-2010-3962, CVE-2015-5119, CVE-2017-0199, CVE-2017-11882, CVE-2015-2545, CVE-2015-2387, CVE-2015-3043, CVE-2016-1019, CVE-2015-5122, CVE-2018-0802, CVE-2018-4878, CVE-2015-3105, CVE-2015-2419, CVE-2013-4979, CVE-2016-4117, CVE-2014-8439, CVE-2015-7645, CVE-2016-4119, CVE-2015-8651
APT32	CVE-2016-7255
APT37	CVE-2015-2545, CVE-2015-2387, CVE-2015-3043, CVE-2016-1019, CVE-2015-5122, CVE-2018-0802, CVE-2015-5119, CVE-2018-4878, CVE-2015-3105, CVE-2015-2419, CVE-2013-4979, CVE-2016-4117, CVE-2014-8439, CVE-2017-0199, CVE-2015-7645
APT38	CVE-2016-4119, CVE-2016-1019, CVE-2015-8651
APT41	CVE-2019-3369, CVE-2015-1641, CVE-2012-0158, CVE-2017-11882, CVE-2019-3396, CVE-2017-0199
CARBANAK	CVE-2013-3906, CVE-2013-3660, CVE-2012-0158, CVE-2017-5638, CVE-2016-5165, CVE-2016-5195, CVE-2015-2545, CVE-2015-1770, CVE-2014-6352, CVE-2015-1641, CVE-2015-1701
CLEAVER	CVE-2010-0232
DRAGONFLY	CVE-2012-4792, CVE-2013-1347, CVE-2013-2465, CVE-2012-1723
ELDERWOOD	CVE-2011-0611, CVE-2011-2110, CVE-2012-1875, CVE-2011-0609, CVE-2012-1889, CVE-2010-0249, CVE-2012-1535, CVE-2012-0779
FIN7	CVE-2017-11882
GORGON GROUP	CVE-2017-0199
GROUP5	CVE-2014-4114
KE3CHANG	CVE-2010-3333, CVE-2012-4681, CVE-2010-2883, CVE-2015-2545
LAZARUS GROUP	CVE-2017-0144, CVE-2017-0145
LOTUS BLOSSOM	CVE-2017-11882
MOLERATS	CVE-2017-0199
MUDDYWATER	CVE-2019-2725, CVE-2017-5689, CVE-2017-11882
NAIKON	CVE-2010-3333, CVE-2012-1856, CVE-2012-0158
NEODYMIUM/PROMETHIUM	CVE-2016-0034, CVE-2010-2568, CVE-2012-0507, CVE-2010-3336, CVE-2012-1889, CVE-2013-3896, CVE-2012-0158, CVE-2015-0072, CVE-2011-0097, CVE-2013-0074, CVE-2015-8651, CVE-2012-1723, CVE-2010-1297, CVE-2013-2423, CVE-2014-1761, CVE-2012-0056, CVE-2016-1019, CVE-2016-0189, CVE-2013-2460, CVE-2016-0165, CVE-2015-1671, CVE-2013-0422, CVE-2016-0167, CVE-2010-0840, CVE-2015-5119, CVE-2015-0311, CVE-2010-0188, CVE-2016-4117, CVE-2015-0310, CVE-2013-1493, CVE-2011-0611, CVE-2016-4171, CVE-2014-6332, CVE-2008-2551, CVE-2011-1823, CVE-2013-2551, CVE-2010-3653, CVE-2016-1010, CVE-2015-0313
OILRIG	CVE-2014-0640, CVE-2017-0199, CVE-2015-7547
PATCHWORK	CVE-2014-4114, CVE-2017-8570, CVE-2015-1641, CVE-2012-1856, CVE-2017-0199, CVE-2017-8750, CVE-2017-12824, CVE-2015-2545, CVE-2017-0261
PLATINUM	CVE-2015-2545, CVE-2013-7331, CVE-2015-2546, CVE-2013-1331
RANCOR	CVE-2018-0798
SILENCE	CVE-2018-0802, CVE-2017-11882, CVE-2017-0263, CVE-2008-4250, CVE-2017-0262, CVE-2018-8174, CVE-2017-0199
SUCKFLY	CVE-2014-6332
TROPIC TROOPER	CVE-2018-0802, CVE-2017-5689, CVE-2017-11882
TURLA	CVE-2012-1723, CVE-2012-4681, CVE-2013-2729, CVE-2013-5065, CVE-2009-3129, CVE-2013-3346

Table 1: CVE's used by Attack groups in different attack campaigns

popularity, and spread of vulnerable software. A large part of CVE's exploited by advanced attackers are client-side software, web application, and software belonging to top 100 technologies. The presence of a high impact vulnerability increases the incidence of exploitation in the wild. For example, CVE-2017-0199 alone was used by multiple attack groups namely *patchwork, apt3, apt37, apt41, and oilrig* in different intrusions [1]. Also, low complexity vulnerabilities for which a reliable exploit can be easily engineered lower the production costs[2] and favor the deployment of the exploit [7, 10]. Finally, Allodi and Etalle [10] developed an attacker threat model for SCADA systems and provide evidence that suggests that attackers do not arbitrarily choose vulnerabilities to exploit amongst all available options, and a new vulnerability in a system carries little weight on the overall risk scenario, as most attackers would not be able to exploit it (unless commoditized).

A genuine effort from academic, industry, and government agencies is made to create common knowledge bases to address the need of understanding attacker behaviors. The ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) has gained popularity in recent years for multiple purposes and has been integrated into popular threat information sharing technologies[11]. ATT&CK is a community-curated knowledge base with the taxonomy of adversaries' tactics and techniques, and information about known techniques used by named attackers. The framework defines attacker actions in an extensive manner and is frequently updated.

Table 1 lists documented CVE's used by different attack groups collected from the past 10 years, but there is no complete understanding of which stage of attack chain the CVE is used or what are the other similar CVE's that can be used against a given environment. Although threat behaviors are used to report cyber attacks using the ATT&CK standard, yet there is no end-to-end approach that captures the role of Common Vulnerabilities and Exposures (CVEs) into different attack stages. Grouping CVE's by ATT&CK techniques helps quantitatively re-prioritize vulnerability risks according to attack stage defined in ATT&CK and in the context of controls. This mapping can also help in prioritizing vulnerability assessments, setting risk model parameters, defenders mitigation strategies, and understanding the attacker's actions. The same CVE's when mapped to corresponding ATT&CK techniques and tactics can help defenders to correctly assess the risk and understand which stage of the attack cycle the corresponding CVE's are being used. Figure 1 illustrates a heat map of CVE count of each MITRE ATT&CK tactics, techniques found in the wild.

Let us explain the problem premise with an example, lets take the case of the "Win32k.sys" kernel-mode device driver vulnerability, which allows a local or remote attacker to execute arbitrary code in kernel mode with elevated privileges. Over the years multiple vulnerabilities have been disclosed that effect this kernel-mode driver, and advanced attackers have leveraged these vulnerabilities in their attack campaigns. Let's look closely the disclosed vulnerabilities between 2010-2016. These vulnerabilities can be mapped to MITRE ATT&CK techniques: **"Exploitation for Defense Evasion, Exploitation of Remote Services, Spearphishing Attachment/User Execution"** and CVE-2011-3402,CVE-2014-4148, CVE-2016-7255, CVE-2014-4148, CVE-2016-0174, CVE-2016-0196, CVE-2016-7255, CVE-2016-7246 where CVE's in red are actively exploited by attackers. Now, when we look at disclosed vulnerabilities between 2017-2019, which share same attack techniques for this particular kernel driver: CVE-2018-8120, CVE-2018-8453, CVE-2018-8589,CVE-2018-8453, CVE-2019-0803 are also exploited by attackers in the attack campaigns.

Similarly, when we study remote scripting vulnerabilities disclosed for scripting engine component of Microsoft Edge and Internet explorer over the past years most of them belong to **"Exploitation for Defense Evasion, Exploitation of Remote Services"** ATT&CK techniques. Below is a subset of 27 CVE's exploited in malware campaigns CVE-2017-0015, CVE-2017-0067, CVE-2017-0141, CVE-2017-8601, CVE-2017-8605, CVE-2017-8598, CVE-2018-0953, CVE-2018-8114, CVE-2018-8122, CVE-2018-8133, CVE-2018-0955, CVE-2018-8267, CVE-2018-8275, CVE-2018-8653, CVE-2016-0191, CVE-2016-0193, CVE-2016-3205, CVE-2016-3207, CVE-2016-3206, CVE-2016-3210, CVE-2016-3222, CVE-2016-3199, CVE-2016-3377, CVE-2016-7200, CVE-2016-7201, CVE-2016-7203, CVE-2016-7242 and CVE-2016-0189, which is used in advanced attack campaigns.

Given the potentially positive impact that an effective automated system, which maps CVE's to ATT&CK techniques has on defenders vulnerability assessment processes, in this paper, we design and implement a novel technique for this problem based upon Natural Language Processing (NLP) techniques and Multi-Label Text Classification(MLTC) models. For the MLTC task, we treat the semantic representation of Security Vulnerabilities (SV) as features and ATT&CK techniques as labels. The large corpora of security vulnerabilities descriptions and their properties: (a) attack steps to exploit the vulnerability; (b) mitigation steps and security controls to avoid exploitation; (c) high-level features and ATT&CK descriptions can be mined to learn rich semantic embeddings. We employ different feature extractors based on the structural and information properties of the text.

Traditional techniques link CVE's in threat reports to MITRE techniques using a series of regular expressions which may miss the semantic connection between the nature of technique leveraged within the scope of the threat introduced by CVE. We address this problem by devising a novel labeling method based on attack techniques descriptions and CVE's text in real-life intrusions. Paragraphs containing CVE descriptions are extracted from threat reports by employing a set of regular expressions, then by using element-wise operations we create vector representations of multi-word units leveraging the word representations of both attack techniques descriptions and threat reports we link to one another on the basis of their distance in vector space. The proposed MLTC model consists of a three head joint non-linear input-label embeddings and a joint-space-dependent classification unit, which is trained with the cross-entropy loss to optimize classification performance. The parametrization of the output layer is controlled by the dimensionality of the joint encoding, which makes sure we are not constrained by the dimensionality of the input or label encoding, but is instead flexible.

[1] https://attack.mitre.org/groups/
[2] https://zerodium.com/program.html

In summary, the primary contributions of this paper are as follows:

- We introduce the Multi-Label Text Classification task for mapping of CVE's to ATT&CK techniques from textual descriptions and propose a multi-head joint embedding neural network architecture.
- We address the problem of lacking labels for this task by a novel unsupervised labeling technique, which extracts *phrases* from threat reports and ATT&CK technique descriptions. We could map 17 techniques via our labeling technique. We create a knowledge base of 150 attack scenarios for exploiting vulnerabilities and 50 mitigation strategies to enrich textual descriptions of CVE's. This helps the model to learn both attacker and defender views of the given CVE.
- We evaluate our approach with a dataset with the past 10 years CVE dataset and compare the proposed model with standard baseline models and perform the ablation analysis to highlight the importance of each model component. We additionally find that the model can link 20 additional techniques which are not seen in the training dataset showing the proposed multi-head approach is generalizable in the given settings.

The rest of this paper is organised as follows: Section 2 presents our research motivation through an example. We discuss our approach in detail in Section 3. Our experiments and discussion are illustrated in Section 4. Section 5 belongs to related work on mapping CVE's to ATT&CK techniques. Finally, we provide a conclusion in Sections 6.

2 MOTIVATING EXAMPLE

In this section, we explain the motivation of our research with a concrete example. The CVE-2017-8759 vulnerability was exploited in advanced intrusions with victims[3] [4] spread across Russia, Iraq, Afghanistan, Nigeria, Libya, Jordan, Tunisia, Saudi Arabia, Iran, Netherlands, Bahrain, United Kingdom and Angola. This CVE can be used by attackers either to gain initial access or move around the network via a privilege escalation and can be mapped to MITRE ATT&CK techniques **"User Execution, Exploitation of Remote Services"**.

There are two known attack scenarios for attackers (a) **Document-based** in which the attacker crafts a malicious document, which may include malicious code, replacement memory addresses, and possibly NOP instructions. The attacker uses email or other means to entice an unsuspecting user to view the malicious document executing attacker's code in the context of an application, and (b) **Application-based** in which the attacker constructs a malicious .NET application and uploads it to a network share. This network share can be inside the organization network (lateral movement) or outside of company network (initial access) depending on the attack stage of the intrusion.

[3]https://www.fireeye.com/blog/threat-research/2017/09/zero-day-used-to-distribute-finspy.html

[4]https://securelist.com/blackoasis-apt-and-new-targeted-attacks-leveraging-zero-day-exploit/82732/

Similarly, the defender can follow multiple mitigation steps and configure controls to avoid and get visibility into the attacker actions: (a) Deploy intrusion detection systems to monitor the network traffic for signs of anomalous or suspicious activity to get alerts on the application-based attack scenario; (b) Filtering and monitoring the email traffic for malicious links and attachments to avoid document-based attack scenario; (c) Enabling memory-protection schemes and running non-administrative software as an unprivileged user with minimal access rights, which can reduce the attack surface, and (d) Timely patching and maintaining the good security hygiene. Knowing the ATT&CK technique for a given CVE, defenders can assess the risk of the CVE based on which attack stage attackers may use the CVE, and deploy controls to monitor the intrusions. Furthermore, he/she can group techniques by tactics to prioritize vulnerabilities for patching. Figure 2 lists different features of a CVE with its ATT&CK stages.

3 APPROACH

Before delving deeper into the proposed system, we give a high-level overview and motivation of the approach. The dataset collected has no labels i.e. there is no mapping of ATT&CK techniques for a given CVE. Though there are 4 techniques *Exploit Public-Facing Application, Exploitation for Client Execution, Exploitation for Privilege Escalation, Exploitation of Remote Services*, which cover the exploitation phase of the attack chain but there are no granular categories that can be mapped. To address this constraint we devise an unsupervised labeling process to extract labels from the threat reports and CVE's. As astute readers may note that not all CVE's are covered in threat reports, and we could link only 17 techniques to the CVE's in the dataset. To improve the coverage and make the system more generalizable we propose an MLTC system that takes into consideration not only the labels in the training set i.e. 17 techniques but also covers techniques for which there are threat reports.

3.1 Labeling Process

Here we describe the overall process we followed in our labeling pipeline. In the proposed unsupervised labeling process we have two major steps – (1) A robust feature extractor that can project CVE and ATT&CK data distributions into a common representation; and (2) A distance function that measures the similarity/dissimilarity between the common representations and assign ATT&CK technique to CVE. Figure 3 illustrate the labeling pipeline with different components.

Common Representations: First, we identify CVE's used in particular attacks or malware from the reports scrapped using CVE specific regular expressions. Next, we extract paragraphs from the report in which CVE was mentioned. Now to extract context words surrounding the CVE, we use Spacy[5] to extract all noun and verb phrases from the paragraphs as candidates. For each candidate, we extract all words within the phrase, and the surrounding right and left context words to each side of the phrase, obtaining three separate sequences of words: left context, the phrase, and the right context. For example, below is a sentence from a threat report for

[5]https://spacy.io/

Feature Name	Value
MITRE Techniques	User Execution, Exploitation of Remote Services, Spear phishing Attachment
CVE	CVE-2017-8759
CVE Description	Microsoft .NET Framework 2.0, 3.5, 3.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2 and 4.7 allow an attacker to execute code remotely via a malicious document or application, aka ".NET Framework Remote Code Execution Vulnerability."
Attack Sequences	**Document-based** • An attacker crafts a malicious document to leverage the issue and to carry out some action on their behalf. The document may include malicious code, replacement memory addresses, and possibly NOP instructions. • The attacker uses email or other means to entice an unsuspecting user to view the malicious document. • When the user views the document, the attacker's code will run within the context of the affected application. **Application-based** • The attacker constructs a malicious .NET application and uploads it to a network share. • The attacker entices an unsuspecting victim to execute the application. • When the victim executes the application, the attacker-supplied code will run within the context of the affected system.
Mitigations and Controls for Defender	• Web users should be cautious about following links to sites that are provided by unfamiliar or suspicious sources. • Filtering HTML from emails may help remove a possible vector for transmitting malicious links to users. • Deploy NIDS to monitor network traffic for signs of anomalous or suspicious activity. This includes but is not limited to requests that include NOP sleds and unexplained incoming and outgoing traffic. • Memory-protection schemes (such as nonexecutable stack and heap configurations and randomly mapped memory segments) will complicate exploits of memory-corruption vulnerabilities. • To reduce the impact of latent vulnerabilities, always run non administrative software as an unprivileged user with minimal access rights
High Level Features	• Authentication: Not Required • Availability: User Initiated • Classification: Input Validation Error • Credibility: Vendor Confirmed • Ease: No Exploit Required • Application Type: Web, Application • Vendors: Microsoft • LocalCode: No, RemoteCode: Yes • CWE: CWE-20 • CVSS – CVSS3_Vector: E:H/RL:OF/RC:C – CVSS2_Vector: AV:N/AC:M/Au:N/C:C/I:C/A:C – Severity: 9.4 • CPE: – cpe:2.3:a:microsoft:.net_framework:2.0:sp2:*:*:*:*:*:* – cpe:2.3:a:microsoft:.net_framework:3.5:*:*:*:*:*:*:* – cpe:2.3:a:microsoft:.net_framework:3.5.1:*:*:*:*:*:*:* – cpe:2.3:a:microsoft:.net_framework:4.5.2:*:*:*:*:*:*:* – cpe:2.3:a:microsoft:.net_framework:4.6:*:*:*:*:*:*:* – cpe:2.3:a:microsoft:.net_framework:4.6.1:*:*:*:*:*:*:* – cpe:2.3:a:microsoft:.net_framework:4.6.2:*:*:*:*:*:*:* – cpe:2.3:a:microsoft:.net_framework:4.7:*:*:*:*:*:*:*

Figure 2: Different Features of CVE-2017-8759

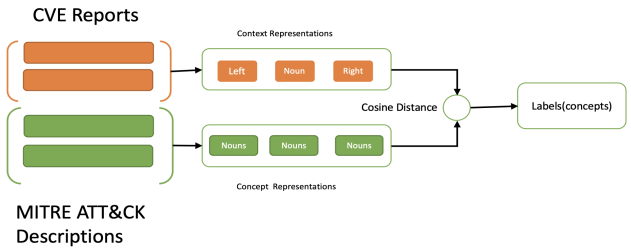

Figure 3: Labeling process of CVEs used in threat reports to MITRE ATT&CK techniques

CVE-2017-8759 highlighting left and right contexts around phrase **CVE-2017-8759 exploit**.

The [left_context]malicious document[left_context] containing CVE-2017-8759 exploit, [right_context]downloads multiple components[right_context], and eventually launches a FINSPY payload. Similar approach is followed to extract candidate phrases and their left-right context words from the technique descriptions available in ATT&CK framework. An element wise mean operation is performed on all three vectors to form a single candidate vector of particular phrase.

Distance Measures: At its core, for the labeling process to be successful we need to measure the similarity/dissimilarity of ATT&CK technique candidate vectors and CVE description representations. For choosing the right distance function, we manually label randomly sampled 200 CVE's found in threat reports with their corresponding ATT&CK techniques and extract the context phrases, and create candidate vectors as described above. Now to choose the similarity measure between ATT&CK technique and CVE description in threat reports, we measure $\mathcal{L}_2$ distance, Cosine distance, Maximum Mean Discrepancy (MMD) [36], and Fisher Linear Discriminant (FLD) [37] and select the best performing distance measurement method for labeling pipeline. This comparison is done to understand the influence of each distance function on underlying data distributions. For example, samples with the same mean and variance might have small $\mathcal{L}_2$ distances even though they are different, whereas MMD can differentiate between them. Table 2 gives the correlation measures of different distance functions on a manually labeled dataset. We choose the cosine distance for the labeling pipeline as it has the highest correlation value that is suitable for this problem.

Given its performance we leverage the cosine distance – which is the measure of similarity between two vectors of an inner product space that measures the cosine of the angle of these vectors given by $D_{\cos} = 1 - \frac{\mu_s \cdot \mu_t}{\|\mu_s\|_2 \|\mu_t\|_2}$. Techniques with the highest cosine similarity with the CVE phrases are assigned the label as a technique. We set the N window size to 5 in our experiments. We could map

Table 2: Correlation measure of ATT&CK techniques and CVE candidate vectors on manually labelled 200 samples.

Distance Function	Correlation Measure
$\mathcal{L}_2$ - $\|\mu_c - \mu_a\|_2$	0.77
Cosine - 1- $\frac{\mu_c \cdot \mu_a}{\|\mu_c\|_2 \|\mu_a\|_2}$	0.89
MMD - $\mathbb{E}_{x^a}[f(x^a)] - \mathbb{E}_{x^c}[f(x^c)]$	0.64
Fisher Linear Discriminant	0.73

17 techniques to CVE's found in threat reports. Table 3 gives a list of phrases mapped to ATT&CK techniques.

Table 3: Subset of Phrases extracted from Threat Reports mapped to ATT&CK Techniques via Labelling process

ATT&CK Technique	Phrases Extracted
Valid Accounts	default accounts, administrative account, default-account, unauthorized creation of user accounts, predictable account credential, perform successive incorrect login attempts, multiple-login protection
Virtualization/Sandbox Evasion	sandbox restrictions, sandboxed process, bypass sandbox protection, sandbox protections
Web Service	malicious web service
Web Shell	upload and execute arbitrary script, shell-upload
Winlogon Helper DLL	creates a malicious dll, malicious dll, unauthorized execution of dll
Spearphishing Attachment	phishing, distributes the page and entices an unsuspecting user
Steal Web Session Cookie	cookie-theft, weak random session, malicious cookie, session-impersonation, cookie guessing, session-hijacking
System Network Connections Discovery	manual scanning, port-scanned, leaks protected network

3.2 MLTC Model

Given a training set with N samples is given as $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i^m), i = 1, ..., N\}$, with $\mathbf{x}_i = \{\mathbf{x}_i^d, \mathbf{x}_i^s, \mathbf{x}_i^c, \mathbf{x}_i^t\}$, where $\mathbf{x}_i^d$ is textual description of CVE, $\mathbf{x}_i^s$ represents the sequence of steps to exploit the CVE, $\mathbf{x}_i^c$ denotes mitigation steps and controls needed to reduce the attack surface for the CVE, $\mathbf{x}_i^t$ represents the high level characteristics of the CVE namely CPE, CVSS base and temporal strings, CWE, classification of the CVE, credibility, local vs remote CVE,

Table 4: Subset of Attack categories

Attack Category	Types
INJECTION	REMOTE/LOCAL CODE, COMMAND, HTML, OS COMMAND, PHP CODE, PHP OBJECT, REMOTE/LOCAL SHELL COMMAND, SQL, XML EXTERNAL ENTITY, SCRIPT
FILE BASED	ACCESS, READ, WRITE, DELETE, UPLOAD, REMOTE/LOCAL INCULDE, TEMPORARY/ARBITRARY CREATION, INSECURE FILE PERMISSIONS
BYPASS	ACCESS, AUTHENTICATION, AUTHORIZATION, BRUTE FORCE AUTHENTICATION, HARD CODED CREDENTIALS/PASSWORD , MAN IN THE MIDDLE, URI PROCESSING
SESSION	FIXATION, HIJACKING, MANIPULATION, WEAK MANAGEMENT
CREDENTIALS	HARD CODED/DEFAULT, MIS-CONFIGURATION, INSECURE/PREDICTABLE RANDOM NUMBER, WEAK PASSWORD ENCRYPTION, CERTIFICATE/SSH SPOOFING, MAN IN THE MIDDLE
ENTRY	DOCUMENT BASED, EMAIL BASED, APPLICATION BASED, CLICK JACKING, MAN IN THE MIDDLE, REQUEST BASED
ESCALATION	NULL POINTER DEREFERENCE, OVERFLOW, HEAP BASED BUFFER OVERFLOW, INTEGER OVERFLOW, STACK BASED BUFFER OVERFLOW, MEMORY CORRUPTION, REMOTE/LOCAL CODE/COMMAND EXECUTION

severity, and y_i^m denotes the corresponding ATT&CK techniques represented as $y_i^m = y_1, y_2..$ for the sample $\mathbf{x}_i$. One of the main goals for the classifier is not only to predict the labels in the training set but also from new/unseen data. This is important to the problem at hand, for the following reasons: (a) Old CVE, which was assigned to a ATT&CK technique can be re-assigned to a new technique based on the evolution of attackers methods; (b) new techniques, CVE's, attack scenarios and mitigations are added to combat new threats and the model has to still work with (new) *concept drift* data; (c) Our proposed labeling process depends on the threat write-ups for which the CVE coverage is limited i.e. we do not have reports covering all ATT&CK techniques. More formally, for a given input $\mathbf{x}_i$ and the labels y_i^m classifier supports both, $\mathcal{Y}_s$, or unseen, $\mathcal{Y}_u$ label sets where $\mathcal{Y}_s \cap \mathcal{Y}_u = \emptyset$ and $\mathcal{Y} = \mathcal{Y}_s \cup \mathcal{Y}_u$.

We design a multi-head deep embedding model to simultaneously embed feature representations of given CVE to predict corresponding ATT&CK techniques of CVE. To achieve this, we project the samples from two domains i.e. the feature domain of CVE and the corresponding ATT&CK domain into a joint latent space, which captures the structure of the labels, the encoded texts and the interactions between the two. Then we use an MLP classifier on the joint latent space that is independent of the label set size. The resulting model has the following properties: (i) Each head of the model learns the label dependency from an attacker, defender, and CVE metadata point of view. (ii) Making joint latent space dimension independent of label size, and input feature dimensions help the model to discover un-seen labels that is critical for our problem, and (iii) The model is trained with the cross-entropy loss with sigmoid function, which is suitable for multi-label classification problem. Figure 4 illustrates the proposed model and underlying components.

3.2.1 Embedding Modules. To embed the features of CVE into a latent space, we use three different embeddings modules. The main purpose of embedding layer is to tranform a sequence of input tokens $x_1, \ldots, x_n$ into vector representation $\boldsymbol{e}_i$ $(i = 1, \ldots, n)$. We use bi-LSTM encoder to extract character-based $\overrightarrow{b_i}$ and word-level $\overleftarrow{b_i}$ embeddings and concatenate both to get the final vector. Features, which have less contextual information but may contain Out of Vocabulary (OOV) tokens pass through token embedding layer that outputs $\boldsymbol{h}_i^M$. Features, which capture mitigation for a given CVE and attack step sequences have more contextual, and sequential in nature so instead of using a Long Short Term

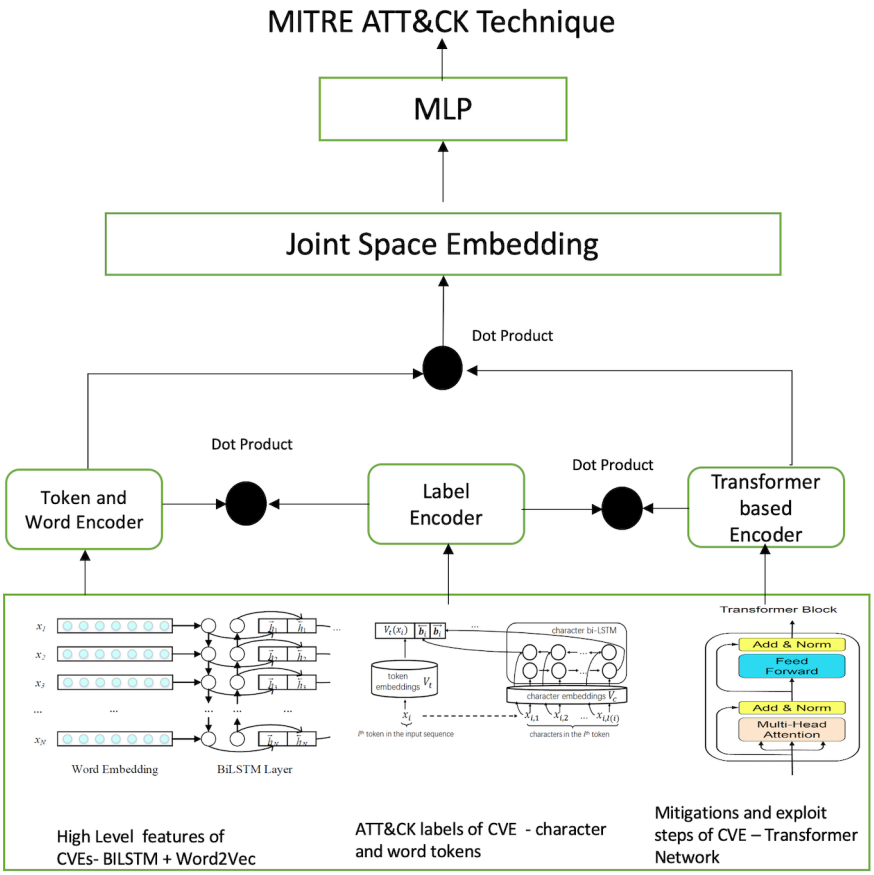

Figure 4: Proposed multi-head deep embedding model. Each textual feature is projected with their label to a join space, the output of which is processed by a MLP with independent label size. Also, each feature branch follows independent encoders based on type of features

Memory (LSTM) encoder, in which sentences interact through recurrent connections, which limits the flow of information between sentences occurring further in the sequence we use the transformer mechanism proposed by [8]. Tranfomer is a multi-head with self-attention and position wise feed-forward sub-layers. Each sub layer has residual connections[6] and normizatiation [9] $LayerNorm(x + Sublayer(x))$ functions for efficient information learning. Here $Sublayer(x)$ denotes the sub-layer function. The attention mechanism is defined on queries, keys and values packed together in matrices $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$, respectively.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\mathbf{V} \tag{1}$$

A multi-head attention for query matrix $\mathbf{Q}$, key matrix $\mathbf{K}$ and value matrix $\mathbf{V}$ is given by

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathbf{H}_1, ..., \mathbf{H}_h)\mathbf{W}^O \tag{2}$$

$$\text{where } \mathbf{H}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{3}$$

Here, $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ are parameter matrices.Every layer in the model outputs a vector of d_{model} dimensions. d_k and d_v are dimensions of key and value, respectively, in a single head and there are h such heads in total. In a self-attention, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ all are from the same layer. In our system, each key, query, and value is a vector corresponding to a sentence. The output of the transformer module is $\boldsymbol{h}_i^A$ embedding vector.

Finally, for label embeddings we train a Word2Vec [12] model with ATT&CK techniques $y_i^m = y_1, y_2...$ and corresponding descriptions and this module outputs $\boldsymbol{h}_i^y$.

For creating joint latent space between two ATT&CK techniques domain and CVE feature domain, we perform a component-wise multiplication of each embedding type with label embdedding $\boldsymbol{h}_i^y$ for their joint representation given by,

$$\boldsymbol{h}_{\boldsymbol{A}joint}^{(i)} = \boldsymbol{h}_i^y \odot \boldsymbol{h}_i^A, and, \boldsymbol{h}_{\boldsymbol{M}joint}^{(i)} = \boldsymbol{h}_i^y \odot \boldsymbol{h}_i^M, \tag{4}$$

$$p_A^{(i)} = \boldsymbol{h}_{\boldsymbol{A}joint}^{(i)} w_A + b_A, and, p_M^{(i)} = \boldsymbol{h}_{\boldsymbol{M}joint}^{(i)} w_M + b_M, \tag{5}$$

where $\odot$ is a component-wise multiplication. The probability for h belong to one of the k known labels is modeled by a linear unit that maps any point in the joint space into a score, which indicates the validity of the combination, where $w \in \mathbb{R}^{d_j}$ and b are a scalar variables. We train the classifier with binary cross-entropy loss and apply sigmoid function :

$$\hat{y_i} = \hat{p}(y_i|x_i) = \frac{1}{1 + e^{-P^{(i)}_{val}}}. \quad (6)$$

4 EXPERIMENTS

In this section, we describe the details of datasets used, seed data generation and hyperparameters used for training the model.

4.1 Datasets

For raw data sets, we crawl 690 cybersecurity articles from a collection of advanced persistent threats (APT) reports that are published from 2008 to 2019[6], zero-day exploits observed by google project zero [7], 63720 vulnerability reports [8], and 37000 threat reports [9]. Figure 5 shows different characteristics of CVE's found in threat reports. The Nessus vulnerability scanner contains a rich set of plugins for performing vulnerability assessments. Each plugin is designed to check for the presence of a set of CVE's via automated scans and contains human-curated rich descriptions about the CVE, steps performed by scan to check for the presence of CVE on a system and metadata about the vulnerability. This dataset was provided by a security company for research purposes.

We manually create a knowledge base of 50 mitigation steps categorized into 7 high-level categories: (a) Restrict and Deny access, (b) Evaluate and Fix Default Configurations, (c) Implement Physical Security, (d) Implement Secure Communication Channels, (e) Inspect and filter network traffic data, (f) Use Strong Authentication and, (g) Use least privilege, and 150 attack scenarios (sequences) from the textual descriptions of Nessus plugins, and recommendations from the CVE vulnerability reports. Table 4 lists a small subset of attack sequence categories from knowledge base. Figure 6 compares the CVSS score distribution of CVE's in entire dataset and CVE's found in threat reports.

We enrich the CVE record with mitigation steps, attack sequences from the knowledge base of the previous step. For initial seed data labeling, we run multiple regular-expressions to extract CVE and neighborhood context words around the use of CVE in the 690 cybersecurity articles and 37000 threat reports. Next, we follow the labeling process described in Section 3 to label CVE's with ATT&CK techniques. We index the all 10 year dataset by the year CVE was publicly disclosed. CVE's disclosed between 2010-2017 were used as training set and data from years 2018-2019 was treated as test set. For the validation, a set of 10% of CVE records were randomly sampled from training set.

For pre-trained token embedding, we apply Word2Vec [12], which are trained with a window size of 8, a minimum vocabulary count of 1, and 15 iterations. The negative sampling number of word2vec is set to 8 and the model type is skip-gram. The dimension of the output token embedding is set to 300. We train the Transformer Network with 2 Transformer blocks, with hidden size 768 and a feed-forward intermediate layer size of 4×768, i.e., 3072. The 768-dimensional representation obtained from Transformer is pooled by the decoder that is a five-layer feed-forward network with ReLU non-linearity in each layer with a hidden size of 200, and a 300-dimensional output layer for the embedding.

We use the adaptive experimentation platform Ax[10] to tune the rest of hyperparameters and the search space for these hyperparameters are: learning rate $\in (10^{-4}, 10^{-3})$ and dropout rate $\in (0.25, 0.75)$. We run model for 5 times. We use the average validation performance as our validation criteria, and report average test performance.

4.2 Evaluation

The performance of model is measured by rank-based evaluation that can accommodate multiple ATT&CK techniques assigned to same CVE. We use Precision at τ ($P@\tau$) defined as

$$P@\tau = \frac{1}{\tau} \sum_{l \in r_\tau(\hat{\mathbf{y}})} \mathbf{y}_l \quad (7)$$

where $\mathbf{y} \in \{0,1\}^k$ is the ground truth label vector of a CVE and $r_\tau(\hat{\boldsymbol{y}})$ is the label indexes of top τ highest scores of the current prediction result. $\| \mathbf{y} \|_0$ counts the number of relevant labels in the ground truth label vector $\mathbf{y}$. Larger $P@\tau$ indicates better performance.

We compare model performance with 3 baseline models namely, Bi-direction Long short-term memory (BI-LSTM), Attention-based BI-LSTM, and Term Frequency–Inverse Document Frequency (TD-IDF) based SVM multi-label classifier. We also run a labelling pipeline on all of the dataset to check - does the model give additional improvements other then just learning the correlation inherent in the noun/verb overlap from the labeling techniques.

TF-IDF approach represents all textual features as vectors with the same length as the vocabulary of the entire text corpus. Each entry in the vector corresponds to a unique word, and its weight gives the frequency of that word in the post (TF) divided by its document frequency (IDF). We set the feature size to 300. These document vectors are then used in the classification task. Also since TF-IDF results in high dimensional representations, we used SVM on TF-IDF features. We train the Word2Vec model using the same pre-training corpus as our proposed model. A BI-LSTM and Attention-based BI-LSTM model is trained for the multi-label task on the Word2Vec extracted embedding. Table 5 summarizes the results of proposed model and different baselines models.

We evaluate the influence of each component of the proposed model by ablation experiments. Note that when we use only the label or only the input embedding in the input-label formulation, the dimensionality of the joint space is constrained to be the dimensionality of the encoded labels and inputs respectively, that is d_j=300 in our experiments. Table 6 summarizes the results.

There are multiple studies [27, 28], which highlight the issues like reproducibility and inconsistencies of textual information in CVE reports. These problems are inherited by our labeling pipeline

[6] https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections
[7] https://googleprojectzero.blogspot.com/p/0day.html
[8] https://www.symantec.com/security-center/vulnerabilities
[9] https://www.symantec.com/security-center/a-z
[10] https://github.com/facebook/Ax

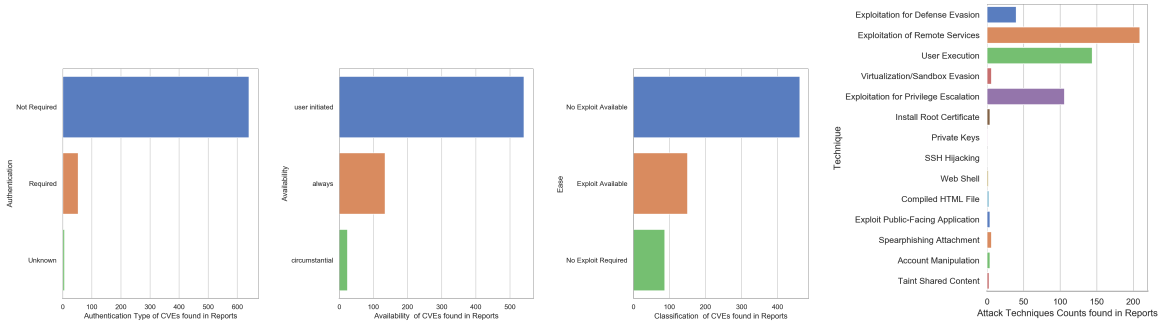

Figure 5: Characteristics of CVE's found in the threat reports

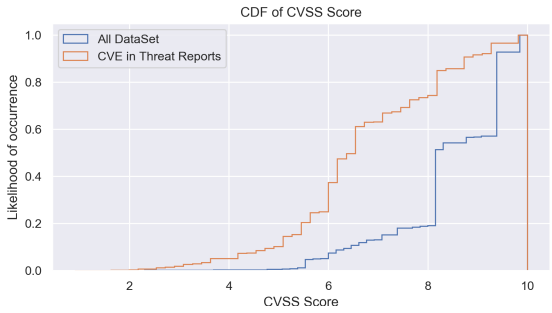

Figure 6: CVSS score cumulative density distribution of CVE's in entire dataset vs CVE's found in report

Model	$P@1$	$P@3$	$P@5$	N_t
Only Labeling Pipeline	0.354	0.392	0.411	17
Bi-LSTM + MLP	0.8557	0.8223	0.838	17
Attention-Based BI-LSTM + MLP	0.8757	0.8234	0.848	17
TD-IDF + SVM	0.7619	0.6246	0.686	17
Proposed Model	0.9316	0.9589	0.945	37

Table 5: Performance comparison of proposed model on the various Baselines. N_t is total number of ATT&CK techniques identified by model across whole dataset.

Layer	Labels		
	$P@1$	$P@3$	$P@5$
$\boldsymbol{h}^M + MLP$	49.84%	32.27%	24.17%
$\boldsymbol{h}^A + MLP$	70.40%	54.98%	44.86%
$(\boldsymbol{h}^A \odot \boldsymbol{h}^M) + MLP$	85.28%	61.12%	52.78%
$(\boldsymbol{h}^A \odot \boldsymbol{h}^M \odot \boldsymbol{h}^y) + MLP$	93.16 %	95.89%	94.50%

Table 6: Ablation test of various modules of proposed model

as well, where it could not map a CVE text to ATT&CK technique because of lack or ambiguous description. Our proposed method could identify 20 techniques that are not seen in the training data (labeling pipeline), which shows the model is resistant to drifts occurring in the underlying data distributions i.e. change of words used to describe the CVE and, highlights the advantages of multi-head design. Results indicate that we can take advantage of label text to explicitly determine the semantic relation between CVE textual features and corresponding ATT&CK techniques. Furthermore, adaptively extracting the proper amount of information from these two textual data benefit the model.

Figure 7 plots the ATT&CK techniques trend of CVE's disclosed over past 10 years. Table 7 summarises different attack techniques identified by the proposed method - A CVE can be part of multiple techniques so depending on stage of attack the attacker can exploit the CVE to achieve his/her goal.

5 RELATED WORK

Our work combines several research areas, and we briefly discuss related work from these areas.Initial works that employed Natural Language Processing (NLP) techniques as feature extractors demonstrated that the CVE text descriptions are rich enough for the classification [30, 31, 38, 39]. With the release of new technologies/products or the discovery of a zero-day attack or vulnerability, new terms appear in descriptions. The appearance of new concepts makes the vulnerability data and patterns change over time [40, 41], which is known as concept drift [42]. It was shown in [43–45] that multiple models have suffered from the concept drift by mixing the new and old vulnerability data in the model validation step, which can lead to biased results as such approach accidentally merges the new information with the existing one. The presence of concept drift makes the models less robust to future prediction due to missing information. There has been some work into exploring the behavior of different attackers, but no work to the best of our knowledge has considered using attacker techniques and defender constraints in concert with machine learning for the purposes of vulnerability or software service management.

In the field of text classification, a large body of work has employed different neural network architectures – Simple feed-forward networks[13, 14], CNN [15–17], and hierarchical recurrent neural networks[18, 19] to address the problem. Recent studies tackled the problem of learning output representations directly from data without any feature extractors [20–23]. In this work, we employ

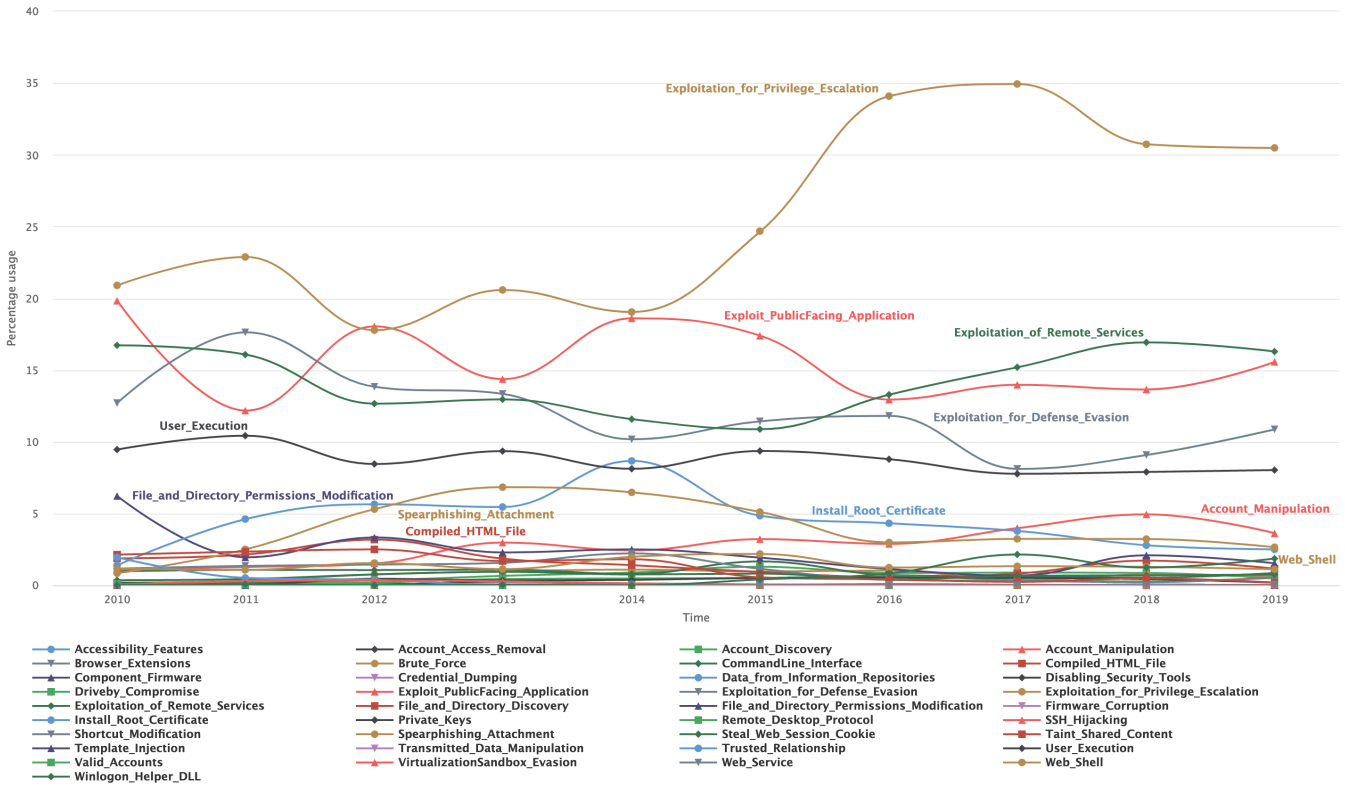

Figure 7: ATT&CK Techniques Trend Plot of CVE's disclosed over past 10 years identified by the proposed model

3 feature extractors as sub-networks including transformers, Bi-LSTM, and word2vec models for the classification tasks.

Recently, the textual information of labels is jointly learned with documents for the MLTC task. EXAM [26] introduces the interaction mechanism to incorporate word-level matching signals into the text classification task. GILE [29] proposes a joint input-label embedding model for neural text classification. LSAN [25] uses label semantic information to determine the semantic connection between labels and documents for constructing the label-specific document representation. Our work follows a similar methodology of learning correlation between the labels and texts but differs in feature transformers and the labeling mechanism.

6 CONCLUSION

In this paper, we define the problem of automatic mapping CVE's to ATT&CK techniques as a multi-label text classification task and propose a multi-head joint embedding neural network model to address it. We devise a novel unsupervised labeling technique that extracts relevant phrases with context from threat reports and ATT&CK technique descriptions to deal with the lack of labels for the task. A knowledge base of 50 mitigation strategies and 150 attack scenarios for exploiting vulnerabilities is curated to enrich CVE's features and help proposed model learn both attacker and defender view of a given CVE's. Our evaluation results are encouraging, with a large set of CVE's disclosed over the past 10 years are mapped to a set of ATT&CK techniques using the proposed approach and also identified CVE's which can be mapped to different stages of attack.

Despite some interesting results of our proposed method, we want to highlight its inherent limitations and subjects for future work. The model learns attacker and defender view of a CVE by feature enrichment from the knowledge base of attack scenarios, and mitigation steps that is incomplete. Also, we found in some cases where CVE records have very little textual information that resulted in no mapping. Our study does not cover the threat modeling in terms of robustness to adversarial attacks and comparison experiments with bench-marking datasets in other domains, which solve MLTC task.

Our future work will focus on the following areas: (a) Extend the model embedding inputs to consume exploit code, patch diffs and Common Weakness Enumeration (CWE) descriptions that will address the incompleteness of knowledge base, and missing textual information; (b) Expand the experiments to compare model performance on benchmarking data sets in other domains, and (c) Examine the security robustness of proposed model under adversarial settings [33, 34] and threat models.

REFERENCES

[1] Luca Allodi, Fabio Massacci Comparing vulnerability severity and exploits using case-control studies ACM Transaction on Information and System Security (TISSEC) volume 17 , 2014.

[2] M Hutchins, Eric & J Cloppert, Michael & M Amin, Rohan Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Leading Issues in Information Warfare & Security Research. 1.

ATT&CK Techniques	CVE Record Count
Account Manipulation, Command-Line Interface, Exploit Public-Facing Application	1
Account Manipulation, Exploitation for Defense Evasion	2
Command-Line Interface, Exploit Public-Facing Application	1
Command-Line Interface,Exploitation for Defense Evasion,User Execution	1
Command-Line Interface, Exploitation of Remote Services	1
Compiled HTML File, Exploitation for Defense Evasion, Exploitation of Remote Services	1
Compiled HTML File, Exploitation for Defense Evasion, Exploitation of Remote Services,Taint Shared Content, User Execution	2
Compiled HTML File, Exploitation for Defense Evasion, Exploitation of Remote Services, User Execution	1
Compiled HTML File, Exploitation for Defense Evasion, Taint Shared Content,User Execution	1
Compiled HTML File, Exploitation of Remote Services, User Execution	1
Exploit Public-Facing Application, Exploitation for Defense Evasion	2
Exploit Public-Facing Application, Exploitation for Defense Evasion, Transmitted Data Manipulation	1
Exploit Public-Facing Application, Transmitted Data Manipulation	1
Exploitation for Defense Evasion, Exploitation for Privilege Escalation	47
Exploitation for Defense Evasion, Exploitation of Remote Services	135
Exploitation for Defense Evasion, Exploitation of Remote Services, Spearphishing Attachment	1
Exploitation for Defense Evasion, Exploitation of Remote Services,Taint Shared Content,User Execution	1
Exploitation for Defense Evasion, Exploitation of Remote Services, User Execution	87
Exploitation for Defense Evasion, Exploitation of Remote Services,User Execution, Virtualization/Sandbox Evasion	1
Exploitation for Defense Evasion, Exploitation of Remote Services, Virtualization/Sandbox Evasion	4
Exploitation for Defense Evasion, Install Root Certificate	3
Exploitation for Defense Evasion, Spearphishing Attachment	4
Exploitation for Defense Evasion, Taint Shared Content	1
Exploitation for Defense Evasion, User Execution	74
Exploitation for Defense Evasion, User Execution, Virtualization/Sandbox Evasion	1
Exploitation for Defense Evasion, Web Shell	2
Install Root Certificate, Private Keys, SSH Hijacking	1

Table 7: Subset of mapped Attack Techniques by proposed Model on full CVE Dataset

[3] Z. Junjie, et al. ARROW: GenerAting SignatuRes to Detect DRive-By DOWnloads Proceedings of the 20th International Conference on World Wide Web WWW '11

[4] D. Rachna, J.D. Tygar, and M. Hearst Why Phishing Works, SIGCHI Conference on Human Factors in Computing Systems, CHI'06

[5] Luca Allodi, Shim Woohyun, and Fabio Massacci Quantitative assessment of risk reduction with cybercrime black market monitoring.. In *In Proc. of IWCC'13.*

[6] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

[7] Luca Allodi, Fabio Massacci and Julian Williams The Work-Averse Cyber Attacker Model. Evidence from two million attack signatures. In *Published in WEIS 2017.* .

[8] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

[9] Lei Ba, J.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450.*

[10] Allodi, L. & Etalle, S. Towards realistic threat modeling: attack commodification, irrelevant vulnerabilities, and unrealistic assumptions. (2017)

[11] Strom, B., Applebaum, A., Miller, D., Nickels, K., Pennington, A. & Thomas, C. MITRE ATT&CK: Design and Philosophy. *Mitre Product Mp.* pp. 18–0944 (2018)

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[13] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

[14] Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188—-1196, Beijing, China.

[15] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, Doha, Qatar.

[16] Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, Denver, Colorado.

[17] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, pages 649–657, Montreal, Canada.

[18] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal. Association for Computational Linguistics.

[19] Nikolaos Pappas and Andrei Popescu-Belis. 2017. Multilingual hierarchical attention networks for document classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1015–1025.

[20] Vivek Srikumar and Christopher D. Manning. 2014. Learning distributed representations for structured output prediction. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3266–3274, Cambridge, MA, USA. MIT Press.

[21] Chih-Kuan Yeh, Wei-Chieh Wu, Wei-Jen Ko, and Yu-Chiang Frank Wang. 2018. Learning deep latent spaces for multi-label classification. In *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, USA.

[22] Isabelle Augenstein, Sebastian Ruder, and Anders Søgaard. 2018. Multi-task learning of pairwise sequence classification tasks over disparate label spaces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1896–1906, New Orleans, Louisiana.

[23] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331. Association for Computational Linguistics.

[24] Majid Yazdani and James Henderson. 2015. A model of zero-shot learning of spoken language understanding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 244–249, Lisbon, Portugal.

[25] Xiao, L., Huang, X., Chen, B. & Jing, L. Label-Specific Document Representation for Multi-Label Text Classification. (2019)

[26] Du, C., Chen, Z., Feng, F., Zhu, L., Gan, T. & Nie, L. Explicit interaction model towards text classification. (2019)

[27] Towards the detection of inconsistencies in public security vulnerability reports, Dong, Ying and Guo, Wenbo and Chen, Yueqi and Xing, Xinyu and Zhang, Yuqing

and Wang, Gang, 28th {USENIX} Security Symposium ({USENIX} Security 19), 869–885, 2019

[28] Understanding the reproducibility of crowd-reported security vulnerabilities Mu, Dongliang and Cuevas, Alejandro and Yang, Limin and Hu, Hang and Xing, Xinyu and Mao, Bing and Wang, Gang, 27th {USENIX} Security Symposium ({USENIX} Security 18), 919–936, 2018

[29] Pappas, N. & Henderson, J. GILE: A Generalized Input-Label Embedding for Text Classification. *Transactions Of The Association For Computational Linguistics*. **7** pp. 139–155 (2019)

[30] Neuhaus, S. & Zimmermann, T. Security trend analysis with cve topic models. (2010)

[31] Urbanska, M., Ray, I., Howe, A. & Roberts, M. Structuring a vulnerability description for comprehensive single system security analysis. *Rocky Mountain Celebration Of Women In Computing, Fort Collins, Co, Usa.* (2012)

[32] Riskwriter: Predicting cyber risk of an enterprise Aditya, K and Grzonkowski, Slawomir and Le-Khac, Nhien-An International Conference on Information Systems Security 88–106, 2018 ,Springer

[33] Black box attacks on deep anomaly detectors Kuppa, Aditya and Grzonkowski, Slawomir and Asghar, Muhammad Rizwan and Le-Khac, Nhien-An Proceedings of the 14th International Conference on Availability, Reliability and Security, 1–10, 2019

[34] Black Box Attacks on Explainable Artificial Intelligence (XAI) methods in Cyber Security Kuppa, Aditya and Le-Khac, Nhien-An, International Joint Conference on Neural Networks (IJCNN) 2020

[35] Effect of Security Controls on Patching Window: A Causal Inference based Approach Kuppa, Aditya and Aouad, Lamine and Le-Khac, Nhien-An Annual Computer Security Applications Conference 556–566 2020

[36] Gretton, A., Borgwardt, K., Rasch, M., Sch"olkopf, B. & Smola, A. A kernel two-sample test. *Jmlr*. (2012)

[37] Friedman, J., Hastie, T. & Tibshirani, R. The elements of statistical learning. (Springer,2001)

[38] Tavabi, N., Goyal, P., Almukaynizi, M., Shakarian, P. & Lerman, K. Darkembed: Exploit prediction with neural language models. (2018)

[39] Khazaei, A., Ghasemzadeh, M. & Derhami, V. An automatic method for CVSS score prediction using vulnerabilities description. *Journal Of Intelligent & Fuzzy Systems*. **30**, 89–96 (2016)

[40] Williams, M., Dey, S., Barranco, R., Naim, S., Hossain, M. & Akbar, M. Analyzing Evolving Trends of Vulnerabilities in National Vulnerability Database. (2018)

[41] Murtaza, S., Khreich, W., Hamou-lhadj, A. & Bener, A. Mining trends and patterns of software vulnerabilities. *Journal Of Systems And Software*. **117** pp. 218–228 (2016)

[42] Bullough, B., Yanchenko, A., Smith, C. & Zipkin, J. Predicting exploitation of disclosed software vulnerabilities using open-source data. (2017)

[43] Almukaynizi, M., Nunes, E., Dharaiya, K., Senguttuvan, M., Shakarian, J. & Shakarian, P. Patch Before Exploited: An Approach to Identify Targeted Software Vulnerabilities. (Springer,2019)

[44] Bozorgi, M., Saul, L., Savage, S. & Voelker, G. Beyond heuristics: learning to classify vulnerabilities and predict exploits. (2010)

[45] Spanos, G. & Angelis, L. A multi-target approach to estimate software vulnerability characteristics and severity scores. *Journal Of Systems And Software*. **146** pp. 152–166 (2018)

[46] Luong, M., Sutskever, I., Le, Q., Vinyals, O. & Zaremba, W. Addressing the rare word problem in neural machine translation. *Arxiv Preprint Arxiv:1410.8206*. (2014)

[47] Huang, C., Yen, H., Yang, P., Huang, S. & Chang, J. Using sublexical translations to handle the OOV problem in machine translation. *Acm Transactions On Asian Language Information Processing (talip)*. **10**, 16 (2011)

[48] Liu, A. & Kirchhoff, K. Context models for oov word translation in low-resource languages. *Arxiv Preprint Arxiv:1801.08660*. (2018)