University of California San Diego

Jacobs School of Engineering

Final Report

Using NLP to Predict the Severity of Cybersecurity Vulnerabilities

June 4, 2021

Group 1:
Saba Janamian, Bryan Cook, James Logan
Teck Lim, Ivan Ulloa

Advisors:
Dr. Amarnath Gupta, Dr. Ilkay Altintas

Table of Contents

1. Abstract

Cyber-attacks continue to be one of the world's foremost safety and economic threats, and, in recent years, have become more numerous and severe. MITRE collects and publishes "Common Vulnerabilities and Exposure" (CVE) records, used by cybersecurity engineers to understand and address known threats. CVE records should contain a "Common Vulnerability Scoring System" (CVSS) score, which indicates a human-determined level of severity. These scores are important to cybersecurity engineers in threat prioritization. Unfortunately, nearly half of all CVE records have not yet had CVSS scores ascribed to them. The *VulnerWatch* product is introduced as a machine learning solution for predicting CVSS scores. Bidirectional Encoder Representation (BERT) is used on CVE text descriptions to predict eight criteria that, in aggregate, indicate a CVSS score. *VulnerWatch* provides the user with a prioritized list of previously unclassified CVE issues and associated predicted CVSS scores. It also allows the engineer to manually enter text describing threats and receive a predicted CVSS score in near real-time. The accuracy of predictions for criteria determining CVSS scores is favorable, averaging close to 0.9, with similar levels of precision and recall. Resultant CVSS predictions are also favorably accurate (MSE = 1.27, MAE = 0.5, R2= 0.51). At this level of accuracy, *VulnerWatch* is deemed to be successful in providing a valuable tool in combating cyber-attacks.

2. Introduction and Question Formulation

2.1 The global threat of cyber-attacks

Cyber-attacks continue to be one of the world's foremost safety and economic threats and in recent years have become more numerous and severe. In May 2021 a ransomware attack was made on Colonial Pipeline Co., a transport company for gasoline and other fuels. The attack closed its entire pipeline system leaving many without fuel and causing a state of emergency in the state of Florida. According to *Interesting Engineering [8]*, an F-35 fighter jet is more likely to be brought down by a cyber-attack than a missle. Suffice to say, cyber-attacks represent a serious threat to public safety.

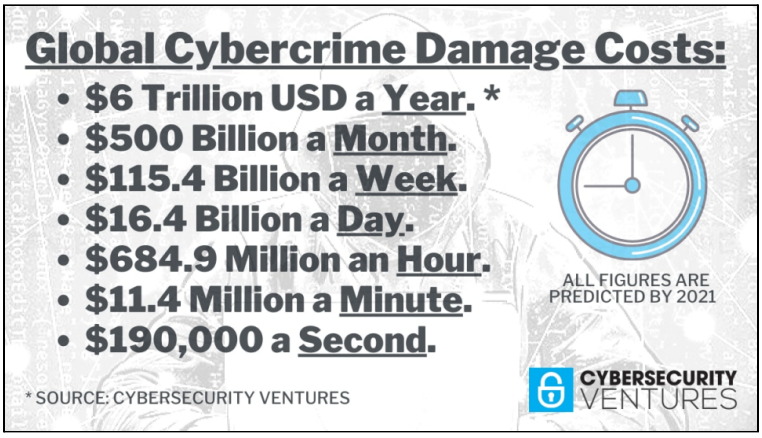

Figure 1: Global Cybercrime Damage Costs

Additionally, there are very serious ramifications to cyber-attacks. As depicted in *Figure 1*, according to Cybersecurity Ventures, cybercrime is a $6T a year problem, a sum larger than the GDP of Japan, the third largest economy in the world.

2.2 Problem: CVE records with no CVSS scores

The primary means to counteract cyber-attacks are through the effort of cyber security engineers. These professionals are tasked with identifying risks and preventing cyber-attacks through the application of tools and patches to network and computer systems. According to a 2020 survey by *(ISC2) [9]*, there are "879,000 cybersecurity professionals in the U.S. workforce and an unfilled need for another 359,000 workers", as well as a potential gap of "nearly 3.12 million unfilled positions" globally. Given the severity of cyber-attacks and the dearth of security professionals professional to combat them, it's critical to arm engineers with as many tools as possible.

One such tool available to engineers are Common Vulnerability and Exposure (CVE) records. A CVE record documents a known cybersecurity issue/risk so that engineers can review them and eliminate issues that may cause systems in their purview to be at risk. CVE records and their management are overseen by the MITRE corporation, and records themselves are available to engineers via the National Vulnerability Database (NVD).

In addition to other technical and non-technical data, CVE records contain a Common Vulnerability Scoring System (CVSS) score, which are human-ascribed values that indicate the level of severity or risk that a CVE represents. CVSS scores are a useful way for cyber security engineers to prioritize the order in which CVE records should be examined and addressed. Unfortunately, nearly 50% of CVE records do not yet contain CVSS scores. Given the importance of quickly addressing cybersecurity issues, the absence of many CVSS scores provides an opportunity to help engineers make better prioritization decisions.

2.3 *VulnerWatch* Solution

To address this issue, the *VulnerWatch* solution was initiated. The primary question put forth is whether machine learning and natural language processing can be used to predict a CVSS score with sufficient accuracy to be an effective surrogate tool for cybersecurity engineers. Assuming such a model is possible, VulerWatch must also provide this functionality to cybersecurity engineers in a convenient interface.

2.4 Related Work

Prior attempts have been made to predict CVSS scores by performing textual analysis of the vulnerability. B*ozorgi et al. [1]* used Bag-of-Words and SVM classification to classify vulnerabilities into different exploitability categories defined by the researchers. The work did not explicitly classify the CVSS vector metrics, but it showed that textual analysis could be beneficial for measuring the severity of vulnerabilities. *Khazaei et al. [2]* further improved the prediction by classifying the numerical severity score using Bag-of-Words, SVM, Random Forest, and fuzzy system. The predicted result was an integer number between 0 to 10 without extracting the CVSS metric vector from the result. The research still showed that there is a potential for predicting CVSS from text mining. *Elbaz et al. [3]* still used Bag-of-Word for processing the text, but they improved the explicability of the predicted score by classifying the metric vectors instead of the numerical score. *Yin et al. [4]* used transfer learning by utilizing a pre-trained ExBERT model. The project's goal was to predict if a vulnerability is exploitable or

not based on the verbiage used in the description of the CVE. Although they obtained a high accuracy of 91%, their model still did not have the potential to explain the predicted result. The proposed design model can explain the predicted score by giving a view of each predicted CVSS metrics. The model also shows which words influenced the classifier's decision with an associated confidence score.

3. Team Roles and Responsibilities

- Teck Lim: Project manager, ETL pipeline engineer
- Saba Janamian: Storyteller, Data engineer
- James Logan: Data engineer
- Ivan Ulloa: Data analyst
- Bryan Cook: Solution architect
- Dr. Amarnath Gupta: Project advisor

4. Data Acquisition

4.1 Data Sources

The majority of the vulnerability intelligence information sources used in this project are publicly available. Examples include CVE, CWE, CAPEC, CPE and CCE listed in Table 1. These are maintained mainly by two organizations, MITRE and the National Institute of Standards of Technology (NIST). For this project the focus lies in the specific use of CVE descriptions to predict the severity of vulnerabilities also known as their respective CVSS score. CVE records may be obtained directly from both of these organizations mentioned above. Although only one type of vulnerability record was used, the rest shouldn't be discarded as potential data sources to enrich or create new machine learning models such as the ones used here.

Title	Link	Description
CVE - Common Vulnerabilities Enumeration	https://cve.mitre.org/ https://nvd.nist.gov/	CVE is a list of records—each containing an identification number, a description, and at least one public reference—for publicly known cybersecurity vulnerabilities.
CWE - Common Weakness Enumeration	https://cwe.mitre.org/	CWE is a community-developed list of software and hardware weakness types. It serves as a common language, a measuring stick for security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.

CAPEC - The Common Attack Pattern Enumeration and Classification	https://capec.mitre.org	The Common Attack Pattern Enumeration and Classification (CAPEC) effort provides a publicly available catalog of common attack patterns that helps users understand how adversaries exploit weaknesses in applications and other cyber-enabled capabilities. "Attack Patterns" are descriptions of the common attributes and approaches employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. Attack patterns define the challenges that an adversary may face and how they go about solving it. They derive from the concept of design patterns applied in a destructive rather than constructive context and are generated from in-depth analysis of specific real-world exploit examples.
CPE - Common Platform Enumeration Dictionary	https://nvd.nist.gov/products/cpe	CPE is a structured naming scheme for information technology systems, software, and packages. Based upon the generic syntax for Uniform Resource Identifiers (URI), CPE includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name.
CCE - Common Configuration Enumeration	https://nvd.nist.gov/config/cce/index	The Common Configuration Enumeration, or CCE, assigns unique entries (also called CCEs) to configuration guidance statements and configuration controls to improve workflow by facilitating fast and accurate correlation of configuration issues present in disparate domains.

Table 1: Examples of cybersecurity datasets available publicly. Here the Common Vulnerabilities Enumeration (CVE) type was utilized to determine the CVSS score.

4.2 Dataset Description

The CVE dataset consists of over 150,000 records each containing specific information for the vulnerability such as: CVE ID, date added, date modified, text description, CVSS score (when available), related CWEs, among others. As of the beginning of 2021, the file size for all CVEs combined can be expected to reach over 180MB in JSON format and expected to continue its growth. From these 150,000 records only about 80,000 have been classified with CVSS scores where the remaining are still pending or simply do not have enough information available to be classified.

4.3 Data Acquisition Pipeline and Storage

As mentioned above, CVE records are downloaded directly from the MITRE or NIST database. The *VulnerWatch* product makes use of separate docker containers to host and prioritize different tasks in each. One of these containers hosts a PostgreSQL database to store the downloaded CVE entries. *VulnerWatch* makes use of the MITRE API to download new CVE raw data where only specific fields of information about the CVEs are stored in the PostgreSQL database.

The process for the acquisition, storage and access of records is shown in *Figure 2*. Data stored for each of the CVEs consists of: CVE ID, publish date, CVSS v3 score, confidence of prediction, method used to classify the CVE, and lastly the full text description of the CVE. Note that the CVSS score, confidence, and method used to classify the CVE may be missing depending on the classification method for the CVSS score.

Figure 2: VulnerWatch data acquisition, storage, and access pipeline for CVE records.

Records stored in the VulnerWatch database may be accessed through the *VulnerWatch* data explorer which provides tools to sort, filter, and search for keywords. The acquisition cycle may be executed by the user manually or set to periodically search for new CVE records on the MITRE database.

5. Data Preparation

All the datasets used were created using human expertise and are therefore less susceptible to noise and contain high quality textual data. Using a data source such as Twitter, in contrast, would require much more processing and effort to extract value. But using human-generated data is not without consequence. machine-generated data tends to accumulate with ease but when humans create a dataset they also become a bottleneck. Data augmentation was therefore required when generating a comprehensive text dataset focused on cybersecurity vulnerabilities.

When constructing a cybersecurity text corpus many curated data sources were utilized. CVEs each have a text description but this field can be short, even only 1 sentence sometimes. Other data sources such as CAPEC, CWE, etc. help add more high quality data but do not contribute relatively much data when compared to CVEs. Thankfully, CVEs have existed long enough to be incorporated into the workflows of major corporations and open source projects. These entities regularly output information about new or updated CVEs relevant to their day-to-day activities. This information is embedded in CVE records as web links and can be therefore scraped for additional data. Each CVE contains many links, most of which point to common websites. These common endpoints can be scraped to augment the cybersecurity corpus.

6. Analysis Methods

6.1 Model Separation

The main objective of the project was to design a heuristic model with sufficient explicability. Prediction of CVSS score without explanation of the factors influencing the prediction was not desired in this project. As described above, most of the prior works on this problem aimed to predict the final numerical score of the CVSS without showing the contribution of underlying metric vectors. To improve the explicability of the output eight separate models were fine-tuned on a BERT uncased model. For this project, we assumed that the result of each CVSS sub metric is mutually exclusive from other sub metrics. Therefore, there was no need to create a multi-class labeling system. *Figure 3* shows a high level overview of model separation structure.

6.2 Word contribution detection

To further improve the explicability of the model, the project adapted two algorithms proposed by *Wu et al. [5]* that could extract words that have the most influence on the sequence classifier of the model. These techniques helped shed light on how BERT made the classification and which words contributed the most to the model's decision-making. The extracted words could also help users better understand the result and decide on the output's validity.

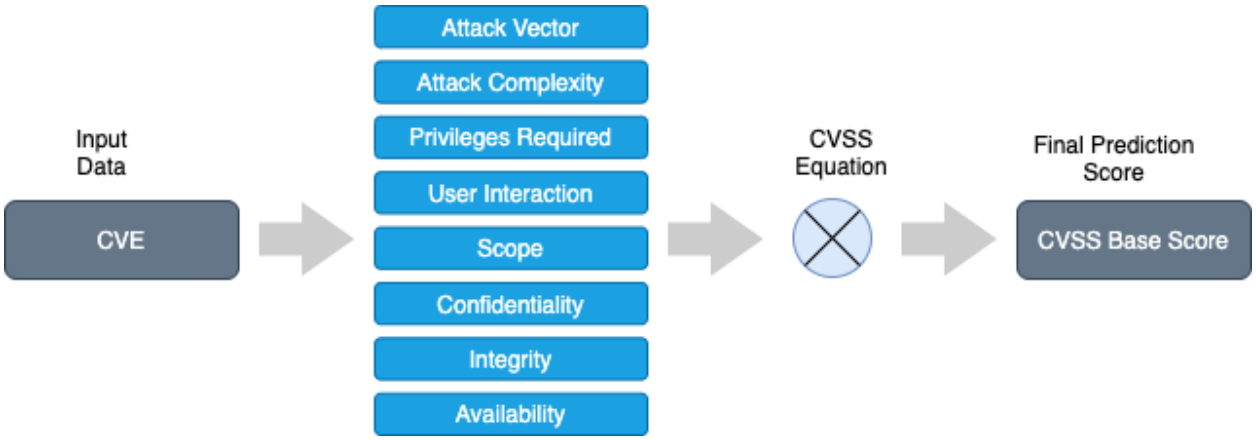

Figure 3: Metric values are predicted independently by different sub-models

6.2.1 Gradient Sensitivity (GS)

The GS method relies on gradients of inputs with respect to each embedding. The algorithm is inspired by the backpropagation algorithm in vision [6]. GS measures how much each input word contributes to the final decision by measuring the first derivative of the input embedding. To formally describe the idea, consider a classification model in which an input with embedding $E$is classified with a golden class c using a scoring function $S_c(E)$. The GS model aims to find which *i-th* dimension of embedding E contributed to the scoring $S_c(E)$which resulted in the decision of class label c.

$$R_i^{GS}(E) = \frac{\partial S_c(E)}{\partial E_i}$$

Where the derivative is with respect to the *i-th* dimension of embedding E.
For clarification the Python implementation of the code is provided in the following:

```
def backward_gradient(sensitivity_grads):
   classifier_out = func_activations['model.classifier']
   embedding_output = func_activations['model.bert.embeddings']
   sensitivity_grads = torch.autograd.grad(classifier_out, embedding_output,
                                           grad_outputs=sensitivity_grads,
                                           retain_graph=True)[0]
   return sensitivity_grads
```

6.2.2 Gradient Sensitivity times Input (GI)

The GI method builds on top of the GS method by multiplying the i-th dimension of input embedding by the GS output [7]. This method will measure how much the output would diverge when the input is changed.

$$R_i^{GI}(E) = E_i \cdot R_i^{GS}(E_i)$$

```
def backward_gradient_input(sensitivity_grads):
   embedding_output = func_activations['model.bert.embeddings']
   return backward_gradient(sensitivity_grads) * embedding_output
```

7. Findings and Reporting

Per Section 1, the *VulnerWatch* initiative set out to answer two key questions: 1) whether CVSS scores could be predicted with sufficient accuracy to be of use to cybersecurity engineers, and 2) whether these predictions could be integrated into a cohesive and useful product.

7.1 Model Accuracy

As was mentioned previously, the CVSS score is determined by the combination of eight independently determined metrics where each of these may consist of 2-4 classes. Below are accuracy results and mean confidence of the model categorizing these sub-models. Accuracy, Matthews Correlation Coefficient (MCC), and F1 metrics are used to measure model performance. Per *Figure 4*, all submodels performed exceptionally well, with only minor fine tuning and minimal hours of training time required. Accuracy and mean confidence are consistently high in all eight trained models, but MCC and F1 suffer when the ground truth of the label is biased towards one class. When the model performance is measured with only high mean confidence predictions, by discarding low confidence predictions, the overall accuracy, MCC and F1 score improve significantly.

Sub-Models	N-Class Labels	Mean confidence	All CVE in test dataset			CVE with 80% confidence in test dataset			CVE with 90% confidence in test dataset		
			Accuracy	MCC	F1	Accuracy	MCC	F1	Accuracy	MCC	F1
Attack Vector (AV)	4	0.9912	0.9257	0.8162	0.8146	0.9287	0.8232	0.8268	0.9525	0.8768	0.8675
Attack Complexity (AC)	2	0.9201	0.9518	0.6421	0.8147	0.9752	0.7753	0.8825	0.9916	0.8266	0.9066
Privilege Required (PR)	3	0.9498	0.8806	0.7441	0.8136	0.9212	0.8236	0.8687	0.9514	0.8775	0.9128
User Interaction (UI)	2	0.9195	0.9374	0.8643	0.9129	0.9672	0.9286	0.9537	0.9851	0.9688	0.9811
Scope (S)	2	0.9327	0.9670	0.8801	0.8989	0.9830	0.9355	0.9447	0.9923	0.9737	0.9783
Confidentiality Impact (C)	3	0.9631	0.8915	0.8062	0.8729	0.9207	0.8567	0.9073	0.9568	0.9218	0.9495
Integrity Impact (I)	3	0.9798	0.9041	0.8413	0.8977	0.9136	0.8569	0.9084	0.9299	0.8839	0.9255
Availability Impact (A)	3	0.9612	0.9108	0.8219	0.7606	0.9357	0.8712	0.7895	0.9588	0.9173	0.8243

Figure 4: Sub-Model Prediction Performance with 61k train data, 15k test data

Of paramount importance and relevance is the notion of whether the model suggests promise in its use as an industry solution. The intention is that cybersecurity engineers will be provided with predicted CVSS scores for uncategorized CVEs, and, in this way, can prioritize addressment of issues with high severity or risk. In combating cybersecurity, time is of the essence, and engineers must focus their time on addressing high profile issues. Industry experts suggest that most successful cybersecurity attacks are the result of unaddressed issues, and not previously unencountered attacks. The engineer is generally faced with more available data than available time, so efficiency is key. As such, the model must yield what should be deemed as a "sufficient" level of accuracy so as to be of value to the engineer. While the accuracy of submodel predictions is vital, in terms of assessing model efficacy, ultimately, the predicted CVSS score is the most important factor.

Scores	All CVE 15404 / 15404			CVE with 80% confidence 9392 / 15404			CVE with 90% confidence 4123 / 15404		
	MSE	MAE	R2	MSE	MAE	R2	MSE	MAE	R2
Exploitability (range 0.1 - 3.9)	0.4280	0.2883	0.4887	0.2468	0.1680	0.7050	0.1176	0.0822	0.8362
Impact (range 0.0 - 6.0)	0.8561	0.3670	0.6049	0.4872	0.2124	0.7683	0.1893	0.0798	0.9114
Base (range 0.0 – 10.0)	1.2760	0.5887	0.5055	0.7333	0.3542	0.7051	0.3127	0.1603	0.8687

Figure 5: CVSS V3 Scores Prediction Performance

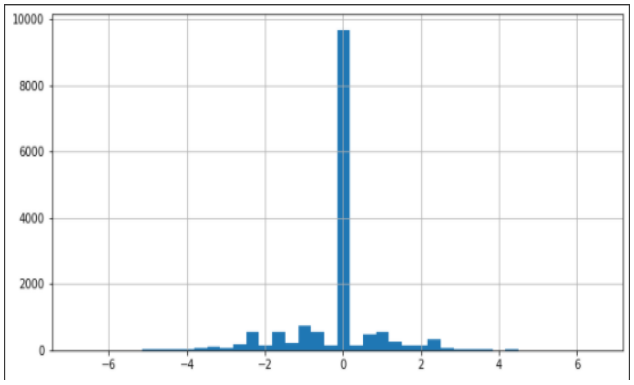

Figure 6: CVSS Base Score Error Distribution

Given that the mean absolute error for all scores is much less than 1.0, this alone suggests value in raising the awareness of CVEs that are labeled as high risk. It also provides reasonable assurance that the cybersecurity engineer won't be inordinately distracted by false positives, e.g. a level 1 issue would, on average", still be marked as an inconsequential CVSS severity. This would be similarly true for issues of high severity; on average high profile CVEs would still raise the awareness of the engineer. And R2 over 0.5 suggests a favorable overall fit.

7.2 Integrated Product

As was discussed previously, the purpose of the data explorer is to provide the cybersecurity engineer with a single portal for reviewing CVE records, their descriptions, and relative severity. At present, the proof of concept for this portal has been implemented in Tableau, as it allows for a fast implementation. Moving forward, it is intended that this will be re-implemented as a web-based application. While the proof of concept does provide all of the requisite features, it does not have acceptable performance, nor does it offer the same portability that a web application would provide.

Text search:

Results Page #: 1

Results per Page: 10

Order By: published_date Descending

Include Manual Scores: ☑
Include Modeled Scores: ☑

CVE	Publish Date	CVSS V3	Confidence	Model	Description
CVE-2019-4722	Jun 01, 2021	4.3	N/A	manual	IBM Cognos Analytics 11.0 and 11.1 could allow a remote attacker to obtain sensitive information via a stack trace due to mishandling of certain error conditions. IBM X-Force ID: 172128.
CVE-2020-4561	Jun 01, 2021	10	N/A	manual	IBM Cognos Analytics 11.0 and 11.1 DQM API allows submitting of all control requests in unauthenticated sessions. This allows a remote attacker who can access a valid CA endpoint to read and write files to the Cognos Analytics system. IBM X-Force ID: 183903.
CVE-2020-4520	Jun 01, 2021	8.8	N/A	manual	IBM Cognos Analytics 11.0 and 11.1 could allow a remote attacker to inject malicious HTML code that when viewed by the authenticated victim would execute the code. IBM X-Force ID: 182395.
CVE-2020-36369	May 28, 2021	5.5	N/A	manual	Stack overflow vulnerability in parse_statement_list Cesanta MJS 1.20.1, allows remote attackers to cause a Denial of Service (DoS) via a crafted file.

Figure 7: VulnerWatch Data Explorer

As depicted in *Figure 7*, a list of CVEs and the data elements described in section 3.1 are present in the portal. In the top right corner, a data range selector is provided, as well as a selector to view CVE records with human-applied CVSS scores, machine predicted scores, or both. Also, a search box is provided to search both the CVE ID list and the CVE descriptions.

The purpose of the "text entry" portal is to provide a means by which the engineer can manually enter text describing a new cybersecurity issue. Once entered, the portal will provide a predicted CVSS score and a confidence level for that score in near real-time. This portal is implemented as a web application with a REST API served by Flask. The functionality and performance requirements of this product were a good fit for this approach, and the implementation was straightforward.

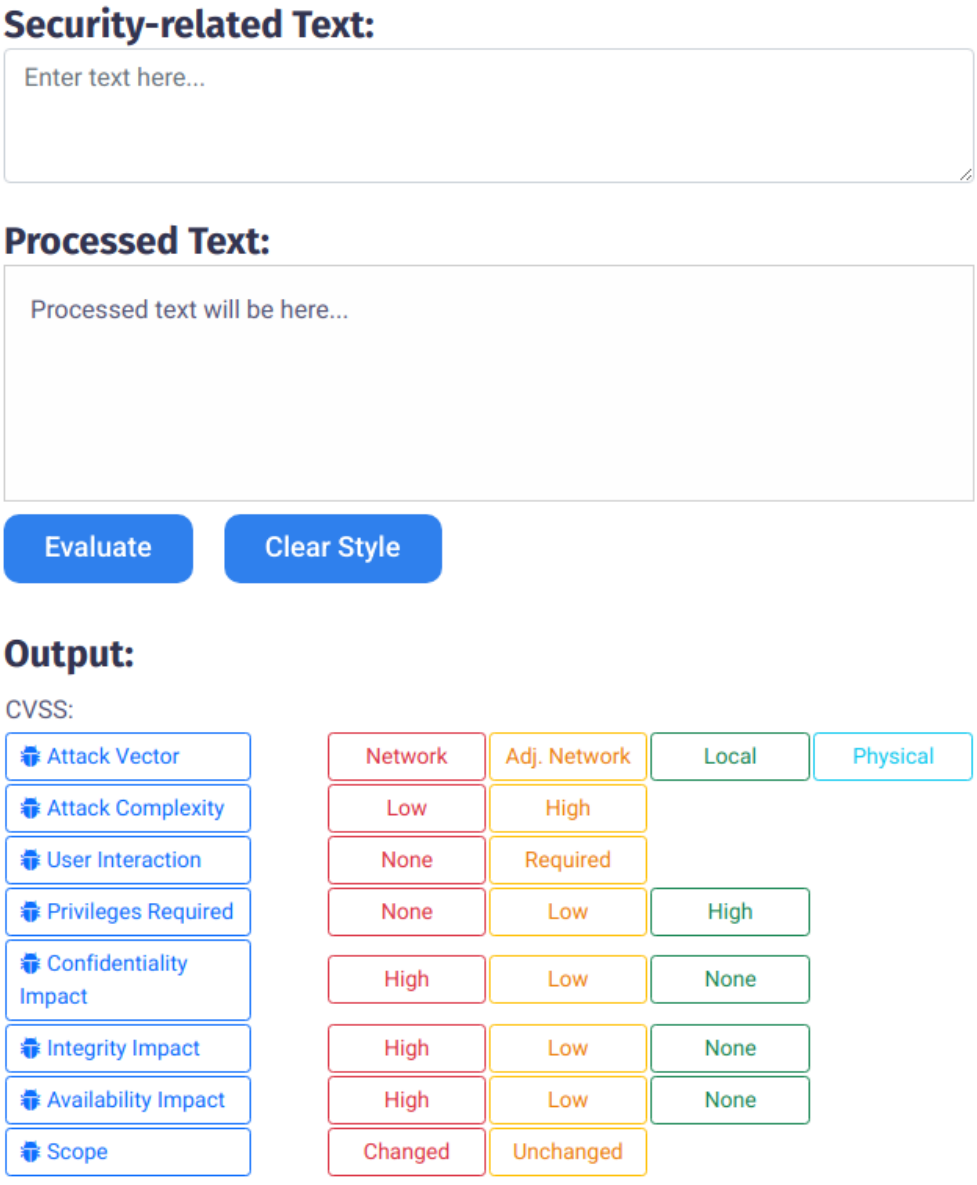

Figure 8: VulnerWatch API Sandbox

Figure 8 depicts the interface in which the engineer can paste text for analysis. This text is evaluated using the models meant for predicting CVSS score, to produce an estimated CVSS score as well as a confidence value. This process runs significantly faster when given access to specialized hardware, namely a GPU. Without access to a GPU the prediction takes 6-7 seconds to complete, but a decent GPU (eg: Tesla K80) speeds up the operation by 10 folds to 0.6-0.7 seconds.

The dashboard providing metadata insights into all of the data is implemented in *Tableau*. This allowed for a speedy implementation and an appealing, feature-rich user experience that would be difficult to implement as a web-based application. Some of the diagrams are affected by selection of human-applied vs. machine predicted CVSS scores, and performance of these configuration changes is generally poor. However, usage of the dashboard should be relatively rare, and modifying its configuration even more so.

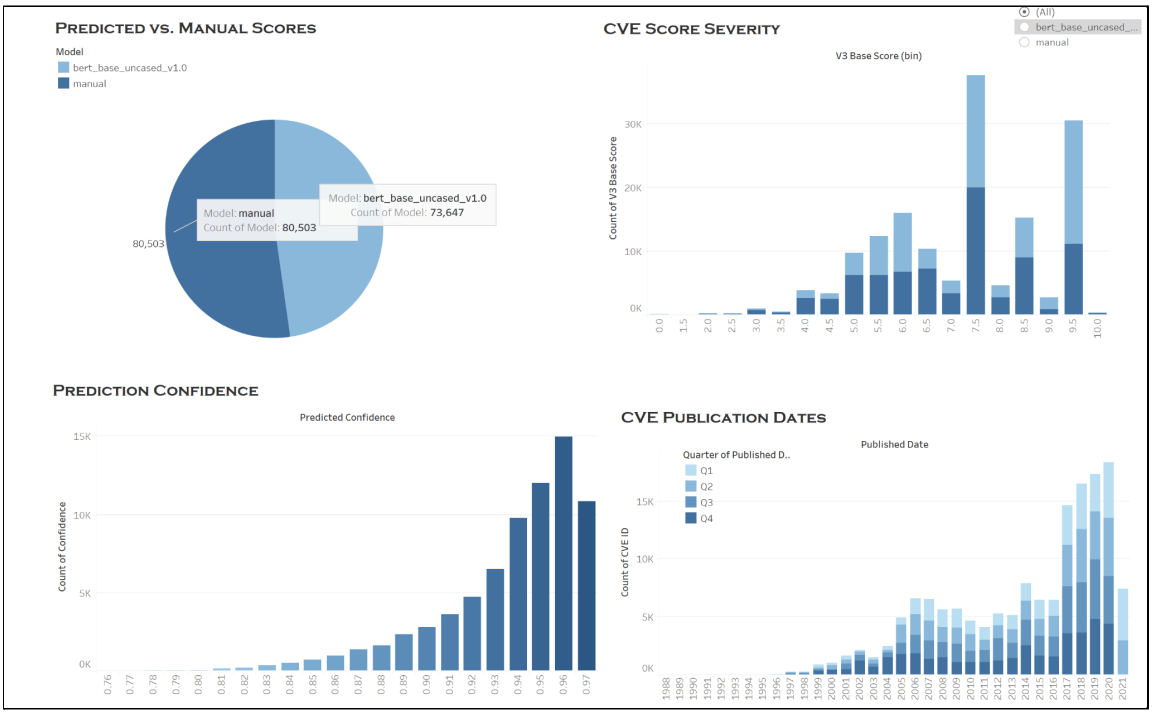

Figure 9: Metadata Dashboard

As depicted in *Figure 9*, the metadata insights described in section 3.2 are present in the portal. Most of the visualizations were best suited for histogram data, while the mix of human vs. machine CVSS scores was best shown as a pie chart. Color was used for a variety of purposes, e.g. to separate new yearly CVE records into their respective quarters. The portal is deemed to convey key points of interest about predictions made by *VulnerWatch*, as well as generally insights about CVE records and cybersecurity.

8. Solution Architecture, Performance, and Evaluation

VulnerWatch solution architecture is designed with scalability and portability in mind. The eight sub-models used in the metric predictions can easily be retrained or replaced individually to continuously improve the overall product performance.

8.1 Scalability & Portability

There are seven major components in the product. They are (1) Postgres relational database, (2) Data downloader module, (3) Predictor module, (4) Trainer module, (5) Flask Web service, (6) Eight fine-tuned BERT models, and (7) Tableau dashboard. These components are encapsulated in four logical docker containers, which are (1) Database container, (2) Database agent container, (3) User Interface container and (4) Trainer container (See *Figure 10)*. They are based on the OpenSUE Leap Distribution image. Containers can be hosted together on a single machine or distributed on multiple machines since containers are designed to communicate with each other through IP addresses. The flexibility of this design allows the performance of each container to be scaled individually. For example, an additional GPU can be added to the Database agent container to speed up prediction time. On the other hand, the Trainer container can be started and stopped at any time to save cost when models are not actively training with new data.

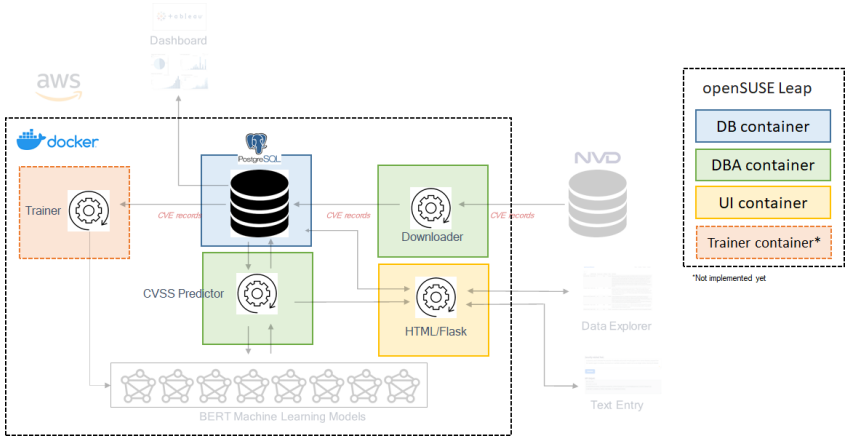

Figure 10:: Solution architecture

8.2 ETL Pipeline

The ETL data pipeline continously ingest CVEs data from NVD. Every hour, the Downloader module is invoked to call standard NVD REST API to download new or modified CVE records. The data is then extracted, transformed and loaded into the Postgres database. The Predictor module, which is preloaded with eight fine-tuned BERT models, continuously checks against the database and predicts unassigned V3 base score CVEs that just came in.

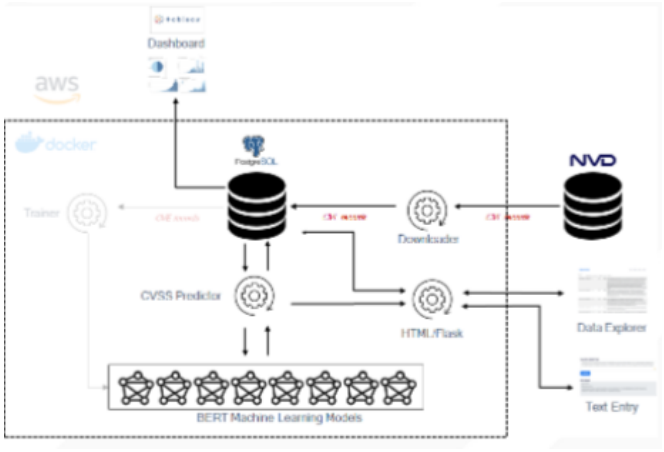

Figure 11:: ETL pipeline

8.3 Training Pipeline

The training pipeline is designed to be inactive most of the time for the purpose of saving cost. The container will only need to be fired up to retrain or fine-tune BERT models as needed. Once the trainer container is started, it partitions CVEs data in Postgres database into train, validation and test datasets. Then use the datasets accordingly to fine-tune and replace existing models.

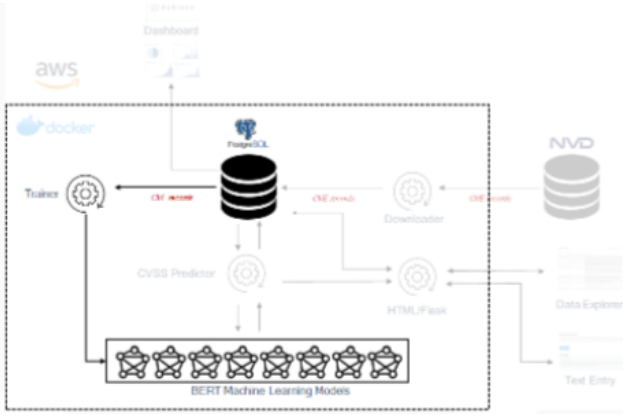

Figure 12: Training pipeline

New cybersecurity vulnerabilities are found and reported to MITRE frequently. The vulnerability descriptions are found to be evolving and changing over time. All models have the highest efficacy in the first year after they are fine-tuned, then they slowly decay each year (See *Figure 12*). To maintain high performance and robustness of all models, all models are recommended to be re-trained and fine-tuned every twelve months at minimum.

Year	# Samples
2015	38
2016	2684
2017	8872
2018	9888
2019	23046
2020	27292
2021	5200

Year	Test Accuracy - CVE Metric: Confidentiality Impact - 3 Label						
# Train Samples	2015	2016	2017	2018	2019	2020	2021
38	0.24	0.30	0.43	0.54	0.53	0.47	
2722	0.78	0.76	0.74	0.70	0.69		
11594	0.85	0.83	0.83	0.80			
21482	0.86	0.85	0.82				
44528	0.87	0.84					
71820	0.85						

Figure 13: Yearly Decay of Model Efficacy

8.4 Website and Dashboard

A Flask web service runs and reads the latest data from the Postgres database to provide a user interactive web interface. A separate dashboard providing metadata insights into all of the data is implemented in Tableau. As discussed in the previous section, the website provides comprehensive access to CVE data as well as user-describe vulnerability prediction. Currently, *VulnerWatch* is hosted on an AWS EC2 instance. The product is accessible and available to the public.

8.5 Cost and Right-sizing Computation

Fine-tuning of BERT requires intense computation. Approximately sixty-one thousand CVE records with minimally two epochs are required to fine-tune each sub-model. To achieve reasonable training time,

GPUs were leveraged. Specifically, the ml.p3.8xlarge AWS instance was used for this training. This product features one Tesla K80 GPU and four vCPUs, and provides total fine-tuning of each model in less than two hours. The cost for this computation is non-trivial, it is estimated to incur approximately $235.01 USD (2 hr/model x 8 models x 14.688 USD/hr instance cost) to fine-tune all 8 models every time. While the cost is in the low hundred of dollars, this is deemed wholly acceptable in the context of high-stakes cybersecurity risks. On the other hand, ETL pipeline cost is relatively affordable to operate on AWS. A t2.xlarge EC2 instance with no GPU provides reasonable horsepower for one to two users accessing the website simultaneously. It costs approximately $82.80 USD (24 hr x 30 days x 0.115 USD/hr instance cost) a month to be operational 24/7.

9. Conclusions

The conclusions drawn from the *VulnerWatch* initiative are as follows:

- As stated in Section 1, the primary question put forth is whether machine learning and natural language processing (NLP) can be used to predict a CVSS score with sufficient accuracy to be an effective surrogate tool for cybersecurity engineers. BERT demonstrated strong accuracy in predicting submodels with precision and recall in the 0.90 range. Moreover, resulting CVSS scores were predicted with a high-degree of accuracy such that they are deemed highly useful by cybersecurity engineers for field work.
- The constituent components included in the *VulnerWatch* product offer a flexible, high-utility interface for engineers to access CVSS predictions and related metadata. The product is highly portable and its economics are well within the range of viability.
- Transfer learning, Language models, Attention weights, and Bidirectional Encoder Representation from Transformers (BERT) are state-of-the-art and cutting-edge technologies used in industry and academia. BERT models have a deeper understanding of language context than traditional single-direction language models. The transfer learning technique also has provided an opportunity to train a general knowledge source and fine-tune it, with significantly less effort, to be used in specific domains. BERT still has many areas for improvement, including reducing the model size, shortening the pre-training time, and making the model less computationally expensive. The use of BERT in this project was a successful approach. The performance of the model outperformed any traditional NLP and classification models. The BERT also provided better explicability and a better view into the decision-making of the neural networks.

10. References

[1] Bozorgi, Mehran, et al. “Beyond Heuristics.” *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '10*, 2010, doi:10.1145/1835804.1835821.

[2] Khazaei, Atefeh, et al. “An Automatic Method for CVSS Score Prediction Using Vulnerabilities Description.” *Journal of Intelligent & Fuzzy Systems*, vol. 30, no. 1, 2015, pp. 89–96., doi:10.3233/ifs-151733.

[3] Elbaz, Clément, et al. “Fighting N-Day Vulnerabilities with Automated CVSS Vector Prediction at Disclosure.” *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, doi:10.1145/3407023.3407038.

[4] Yin, Jiao, et al. “Apply Transfer Learning to Cybersecurity: Predicting Exploitability of Vulnerabilities by Description.” *Knowledge-Based Systems*, vol. 210, 2020, p. 106529., doi:10.1016/j.knosys.2020.106529.

[5] Wu, Zhengxuan, and Desmond C. Ong. “On Explaining Your Explanations of BERT: An Empirical Study with Sequence Classification.” 2021, doi:arXiv:2101.00196.

[6] Li, Jiwei, et al. “Visualizing and Understanding Neural Models in NLP.” *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, doi:10.18653/v1/n16-1082.

[7] Kindermans, Pieter-Jan, et al. . “The (Un)Reliability of Saliency Methods.” 2017, doi:arXiv1711.00867.

[8] *Interesting Engineering*, “Cyber Attacks More Likely to Bring Down F-35 Jets Than Missile”, Fabienne Lang, Feb 25, 2021

[9] *PRNewsWires*, “(ISC)2 Survey Finds Cybersecurity Professionals Have Increasing Level of Concern About SolarWinds Incident”, Mar 29, 2021

11. Appendices

11.1 DSE MAS Knowledge Applied to Project

Every skill learned as part of the DSE program was utilized throughout the capstone project, from python notebook for EDA to user interactive web design. Various python modules and packages, especially numpy, pandas, scikit-learn, were used extensively at the early stage of data exploratory. Matplotlib was used to create simple plots and graphs for visualization and understanding of collected data. Machine learning concepts and algorithms, including NLP, KNN, word2vec, word embedding and more, that were learned as part of the program were invaluable to understand the inner workings of transfer learning and the BERT model. Evaluation metrics such as MSE, MAE, R2, F1, etc that were taught as part of the program were heavily utilized to assess the models efficacy and performance. Besides model building and evaluation, ETL pipeline and database design knowledge helped building an end-to-end product. The visualization concepts and Tableau BI tool completed the final product to commercial-ready standard.

11.2 Library Link and Citation

Cook, Bryan; Janamian, Saba; Lim, Teck; Logan, James; Ulloa, Ivan; Altintas, Ilkay; Gupta, Amarnath (2021). Using NLP to Predict the Severity of Cyber Security Vulnerabilities. In Data Science & Engineering Master of Advanced Study (DSE MAS) Capstone Projects. UC San Diego Library Digital Collections. https://doi.org/10.6075/J0TX3F89

11.3 GitHub Link

https://github.com/twlim1/260_capstone