National College of Ireland

Predicting Attacks on Vulnerabilities using Random Forest

MSc Internship
Cybersecurity

Sarell Lopes
Student ID: x18147241

School of Computing
National College of Ireland

Supervisor: Dr. Muhammad Iqbal

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Sarell Lopes
Student ID:	x18147241
Programme:	MSc Cybersecurity
Year:	2019-20
Module:	MSc Internship
Supervisor:	Dr. Muhammad Iqbal
Submission Due Date:	08/01/2020
Project Title:	Predicting Attacks on Vulnerabilities using Random Forest
Word Count:	5473
Page Count:	17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	29th January 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Predicting Attacks on Vulnerabilities using Random Forest

Sarell Lopes
x18147241

Abstract

Vulnerability assessment is an integral part of information security. CVSS is a globally accepted standard for calculating risk and prioritising the vulnerabilities during the IT system assessment. Automated vulnerability management systems rely on CVSS for their patching processes and ranking weaknesses. CVSS have received some disapproval from the researchers for its limitation to asses the severity factor of vulnerability. The goal of this research is to combine external factors Proof of exploits and attack signatures along with CVSS impact metrics, privilege attribute and user interactions as features for the Random Forest algorithm to evaluate the proposal of predicting an attack on a vulnerability.

1 Introduction

Information System has become a backbone of the organisations in our modern era, it is hard to work without computers and more advance IT infrastructure deployed in today's industry since it eases our job and gives us the control over all the activities that have been taking place in different environments like monitoring the physical security systems, processing data that are comprised of individuals, finances and much more, controlling massive production lines, medical facilities and list goes on. Along with the benefits, Information Technology brings various risk factors which are required to be mitigated, to prevent it from falling into the hands of adversaries who are in constant pursuit of damaging organisations reputation for personal, political or financial gains by proliferating into IT systems and compromising the security of the processes and data. It is achieved by attacking and exploiting the weaknesses present in the organisation's IT infrastructure, to prevent this vulnerability assessment helps to detect the security weaknesses in the IT systems, which can be analysed and fixed before any adversary exploits it [1] [2].

Hence, a vulnerability assessment is an essential part of cybersecurity that helps organisations to address the security of their information systems by discovering the vulnerabilities in the system, analysing risk for them and later securing or patching them according to their criticality. The number of weaknesses in information technology infrastructure may vary for few hundreds to thousands, and hence it is important to prioritise them, Common Vulnerability Scoring System (CVSS) is a standard used widely across the industry to perform risk analysis. CVSS is made up of metrics viz. Base Metric, Temporal Metric and Environmental metric [2] [3] [4]. These metrics are further divided into sub metric, as shown in Figure 1.

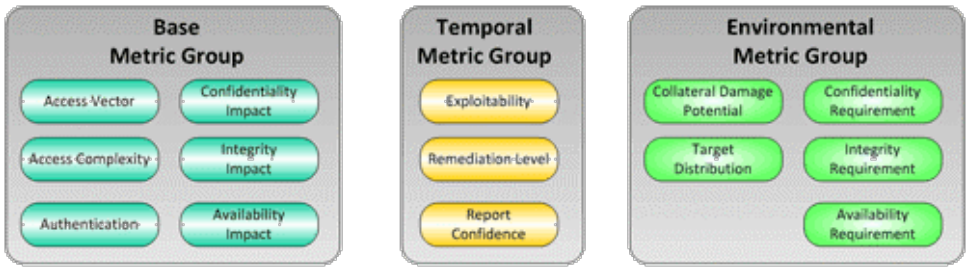

Figure 1: CVSS Metrics & Submetrics

Each metric has a value like 'None', 'Low', 'High', 'Partial' etc. these values represented by a pre-defined score, combing the score through the CVSS formulae they yield result from 0 to 10 where 0 is least critical, 10 is highly critical. Base Score is the metric that's mandatory rest two temporal and environmental are optional as they can only be calculated by vendors and system admin respectively. CVSS helps security professionals to calculate the risk of the vulnerability but alone is not enough to predict whether the vulnerability can be exploited or not. CVSS score for the vulnerabilities is largely maintained by the National Vulnerability Database (NVD), which holds base metric data and information about the software, network device etc. that related to a certain vulnerability. It contains more than 90,000+ listings of weakness and hence has majorly preferred by organisations. Exploit-DB or EDB is another type of database that holds data for Proof of Concepts for the vulnerabilities, is largely maintained by the white hat community that comprises of professionals, industry people and researchers. NVD and EDB being the credible data source have been criticized along with the CVSS for not able to predict the risk for vulnerabilities.

This research focuses on implementing machine learning algorithm random forest to along with attributes from CVSS to address the question "Can we predict the attack on a discovered vulnerability using random forest?"

2 Related Work

CVSS being the de facto standard of vulnerability assessment in the industry has received criticism on its effectiveness on prioritising the vulnerabilities or measuring the risk based on its CVSS score. Researchers L. Allodi and F. Massacci have investigated the relevancy of high CVSS score whether it signifies the vulnerabilities exploited in the wild. The researchers tested the databases obtained from the black market and Symantec's attack and threat resource against CVSS scores from the NVD and Exploit-DB [5]. Different versions of CVSS have emerged since its development, K. Scarfone and P. Mell did theoretical and experimental analysis through which they found that adding more variables to the metric has little effect on the score diversity and increases the complexity of the scoring system, also score shift from version 1 to version 2 has increased number of high priority vulnerabilities i.e. from 59% to 96%. This because the change in version was due to the Payment Card Industry standard in which systems shouldn't have vulnerabilities with CVSS score greater than equal to 4.0. Hence, organizations following the same policies are largely affected by the shift in scores [6]. In one research, the researchers make use of the Bayesian Learning Method to study the vulnerability databases and

CVSS scores; they concluded that NVD along the CVSS score performs well against other databases. But not all metrics are proving to be effective, the Confidentiality, Integrity and Availability (CIA) factors have good accuracy, and metric access complexity has the least accuracy when it comes to analysing the risk for a vulnerability [4].

CVSS is an essential part and contribution towards information security but remains largely unexplored for its actual purpose of vulnerability prioritisation. There is need for listing the vulnerabilities according to the risk they associate to an information system authors Andrej D., Denis T and Borut L. suggest the use of attacker's characteristics to predict whether a vulnerability can be exploited in which they consider various factors to calculate the risk viz. Asset Value, Threat probability and Impact and to predict the threat the factors such as Capability, Opportunity, Motivation, Expected Impact [7]. It is proven in research before, if there are two vulnerabilities with a similar score than one that is exploited most by the adversaries has more chance of getting exploited again [8] [9]. Also, the CVSS metrics that can prove rewarding Temporal and Environmental for assessing risk are rare in use, and they can only be calculated by vendors (Temporal) and System admin or owners (Environment). The authors make use of the Threat Agent Library to define the attacker characteristics and to predict the exploitability and got a better result than the CVSS score [7].

The report published by the security firm Edge Scan focuses on two major categories of the weaknesses are present in IT infrastructure those are, application/software vulnerabilities and network vulnerabilities. It shows that application-related exposures are just 19% and network-related exposures are 81%, but the application related vulnerabilities have a higher percentage of critical susceptibilities, i.e. 19% and network-based susceptibilities were just at 2%. It's named 'Snowflake Effect' as application development is a unique process change according to each organisation and infrastructure does not change, changes and distinct features add more risks [10]. There is research conducted to improve the risk assessment of the software-based vulnerabilities to improve software security. Vulnerability assessment is done based on properties of the software and researchers proposed a metric Structural Severity which they calculate based on entry points in the software and match against the Access complexity metric of CVSS and concluded that considering software attributes such as attack entry points, function calls and vulnerability location can improve the risk assessment of the weaknesses [11]. Researchers from the universities of U.K. and Austria proposed a vulnerability patching system based on CVSS and Game theory in which the analyse the vulnerability by making assumptions based on the interaction between two actors, i.e. attacker and defender as they try to achieve their goals of exploiting and protecting the weakness respectively. The researchers made use of CIA sub-scores from the impact metric of CVSS and Nash equilibrium strategy to predict the criticality of vulnerability and its defence strategies [12].

Studies have shown to improve the classification and ranking of the vulnerabilities in computer networks by metrics that are designed based on CVSS framework. In one paper, authors have developed a dynamic classification and ranking system based on CVSS named as Vulnerability Analysis and Classification countermeasure, which studies and grade Computer Network Threats and Vulnerabilities. CVSS can assess a single vulnerability based on the damage it can cause to IT system, but it fails to measure the damage when there are a group of vulnerabilities that are related to each other which can be targeted by the adversaries, i.e. CVSS is not capable of predicting multi-level attack. The authors proposed two metrics called as Number of Path metric and Shortest Path metric and make use of exploitability [13] [14].

As internet is evolving so are its applications, and new trend of devices have emerged known as Internet of Things (IoT) which designed differently for various purposes viz. security cameras, audio devices, household appliances due to this there is an alarming risk of privacy and security of user, researchers U. Attiq, G. Iqbal, K Joarder and J. Alireza found that CVSS is not reliable for analysing the vulnerability for IoT devices because IoT devices are involved much in interacting with humans, and CVSS doesn't have any metric that takes into consideration about the human safety. Also, the authors argue that due to different manufacturing design and unique characteristics the CVSS metrics won't work for IoT as it's developed for traditional IT systems and they propose the framework based on CVSS, named CVSS IoT in which the modification is done for metrics Attack Vector, Access Complexity and added Human Safety metrics to calculate risk towards human safety [15].

Researchers from the University of Arizona has applied machine learning for predicting exploits in the wild for weaknesses. They use data scraped from the dark web, EDB, NVD and Symantec. The datasets from EDB, Symantec and dark web are check for PoCs and attack signs respectively using the binary features to do so [16].

It is mostly discussed that CVSS score especially the base score is not enough to predict the risk or whether the vulnerability will be exploited in wild because, large number of vulnerabilities with the high score haven't been exploited. Researches have suggested the alternative frameworks and machine learning approach to improve risk assessment. This suggest that there is an opportunity to improve CVSS and its ranking capabilities, hence this research focuses on implementing machine learning algorithm random forest to along with attributes from CVSS to address the research question mentioned in the introduction section.

3 Methodology

The discussion in this section is about the proposed research methodology for predicting whether a vulnerability will be attacked in wild or not, using machine learning approach on datasets acquired from previous research and other sources. The topic covers some important aspects of Data Analytics and Mining.

CRISP-DM is a data mining approach is applied for this research because of its global use in the industry for data mining projects. CRISP-DM full form is CRoss-Industry Standard Process for Data mining that involves 6-Steps viz. 1. Business Understanding 2. Data Understanding 3. Data Preparation 4. Modelling 5. Evaluation 6. Deployment [17].

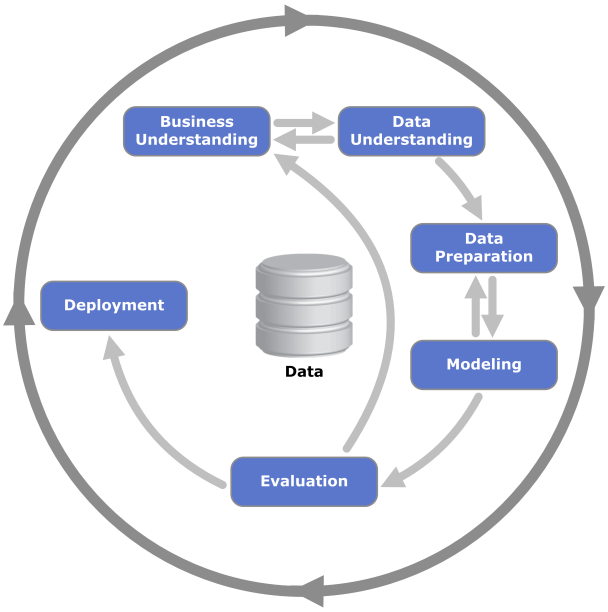

Figure 2: CRISP-DM Methodology

3.1 Business Understanding

This important aspect before commencing the project development is to identify the goal of the project. The leading focus of this research is to predict if the vulnerability will be attacked or not, based on the attack signatures, impact score, number of public exploits and privileges granted. The main purpose of the research is to build an attack prediction model which will assist security administrators in identifying the susceptibilities that will fall prey to the attacks in the wild. The literature on previous research shows the current CVSS framework falls short of predicting whether a vulnerability will be exploited or attacked, on the other hand the challenge is that there are only sheer number of vulnerabilities which have never been exploited or attacked, and in the event of cyber incident companies rarely disclosed about the set of vulnerabilities that were used to target their Information systems. This study will have optimal contribution towards vulnerability assessment and will be beneficial for security professionals and firms globally.

3.2 Data Understanding

This phase involves studying the data before constructing a prediction model. Understanding the data and its attributes are of crucial importance if not done in an orderly manner can lead to difficulties in building a dependable model. Depending on the area of the research, the data source can be easily available or difficult to source example data regarding the population census may be available on government sites but data for the medical research can be limited to organisations premise only and may not be easily found. The obtained data may contain errors, blanks and unwanted values so its necessary to clean and make it relevant. While sourcing the data, it has to be done by obtaining the required permission or license.

The data for this project must contain the discovered vulnerabilities for a certain period along with the attributes explaining if public exploit is available, if they were attacked. The dataset was sourced from multiple repositories those are the following:

1. National Vulnerability Database for CVE details and CVSS score.
2. University of Trento Italy for data from Exploit DB, black markets and Symantec that they've collected.
3. Symantec Attack Signatures.
4. Cyberwatch data for proof of concepts.
5. Zeroday website.

Given the list of sources, the data extracted was not in large numbers except the NVD as the data for attacks happened in the wild is extremely hard and rare to find with respect to specific vulnerability. The past researchers have referred to the NVD, EDB, Symantec for analysis and machine learning project due to their vast coverage over the disclosed vulnerabilities. NVD's JSON feeds consist of data from 2002 to 2019 out of which we analyse data from 2004 to 2014 that consist of 63706 vulnerabilities with their CVSS score and CVSS metric details because this is where the most vulnerabilities are concentrated along with their CVSS v2 scores then CVSS v3 as this is a newly updated standard and hence the number vulnerabilities with CVSS V3 are less. Whereas the datasets from the University of Trento comprised of EDB (proof of concepts), EKITS (CVEs on the black market) and Symantec data collected by researchers during the time of their research. Further, the datasets were sourced from Symantec Attack Signature public website for the attack signatures against the vulnerabilities with their CVE-Ids so as fetch the newer CVEs that aren't available in University of Trento's datasets. CVEs with proof of concept was also scraped from the Cyberwatch's publicly available database along with the Zeroday website for the vulnerabilities that might have faced an attack in the past. The number of public exploits count was extracted from the cvedetails.com, which provides the data for public use. The data from each source are summarised in Table 1.

Datasource	**Number of Vulnerabilities**	**Details**
NVD	63706	CVE-ID and CVSS Scores
EDB	16265	Proof of Concept
EKITS	896	Vulnerabilities on Black market
SYM-Malware-threats	806	Vulnerabilities attacked by malware
SYM-network-attacks	1636	Network Vulnerabilities attacked
Zeroday	408	Exploited Zeroday Vulnerabilities
SYM-Attack Signatures	1014	Attack Signatures recorded by Symantec
Cyberwatch Public Data	42800	Proof of Concept

Table 1: Vulnerabilities Data Sources

The data collected from the above sources are present in two data formats JSON and CSV. The attributes in these datasets are comprised of different data types such as strings, integers and float they are described in Table 2.

Data Source	Variables	Datatype	Data Source	Variables	Datatype
NVD	CVE-ID	string	**EDB**	e-id	integer
	AV (Access Vector)	string		cve-id	string
	AC (Access Complexity)	string		date	string
	Auth (Authentication)	string		osvdb-id	integer
	C (Confidentiality)	string		file	string
	I (Integrity)	string		description	string
	A (Availability)	string		author	string
	CVSSV2(BaseScore)	Float		platform	string
	SEV (SEVERITY)	string		type	string
	EXPL_SCR (Exploitability Score)	Float		port	integer
	IMPACT_SCR (Impact Score)	Float			
	ALL_PRV	boolean			
	USR_PRV	boolean			
EKITS	ek_id	integer	**SYM-Malware-threats**	threat_ID	string
	e_name	string		Type	string
	version	integer		CVE	string
	date	string		String	string
	price	integer	**SYM-network-attacks**	threat_ID	string
	per	string		Type	string
	service1	string		CVE	string
	service2	string		String	string
	service3	string	**SYM-Attack Signatures**	name	string
	cve_id	string		cve	string
	Limited costumers?	string			
	p_source	string			
	s_source	string			
Cyberwatch Public Data	cve-id	string	**Zeroday**	cve-id	string
	cvss score	string			
	vulnerability	string			
	applications	string			
	date	string			

Table 2: Datasets & Attributes

Next, we observe the data through visualisation through the generated graphs: Figure 3 & Figure 4.

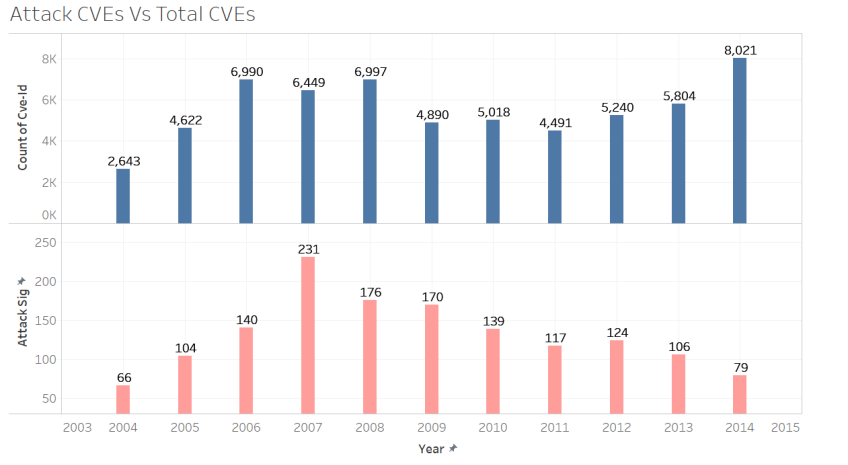

Figure 3: Attack Sig Vs CVE

The Figure 3 explain the heavy imbalance in the list of total vulnerabilities and attacked vulnerabilities this is due to the rare presence of the information about the CVEs exploited in the wild by the threats such as malware or any other attack vectors.

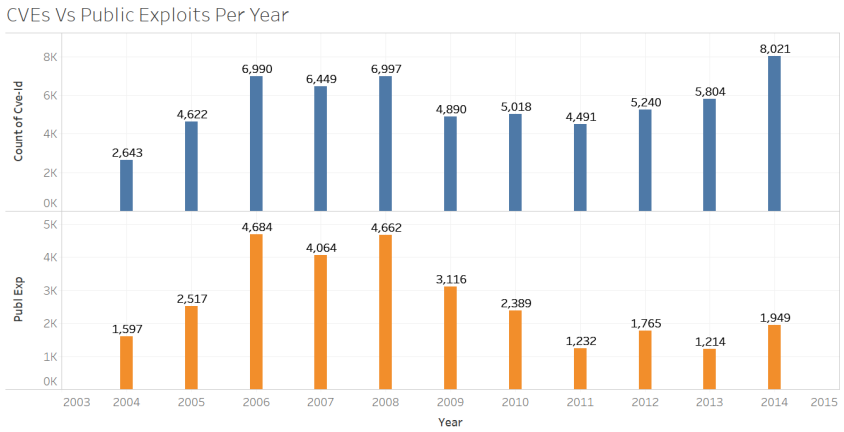

Figure 4: CVEs Vs PoCs

In Figure 4, here we can see that there is quite a balance in data for total vulnerabilities and vulnerabilities with the public exploit as more and more proof of concepts being released by the white hat community.

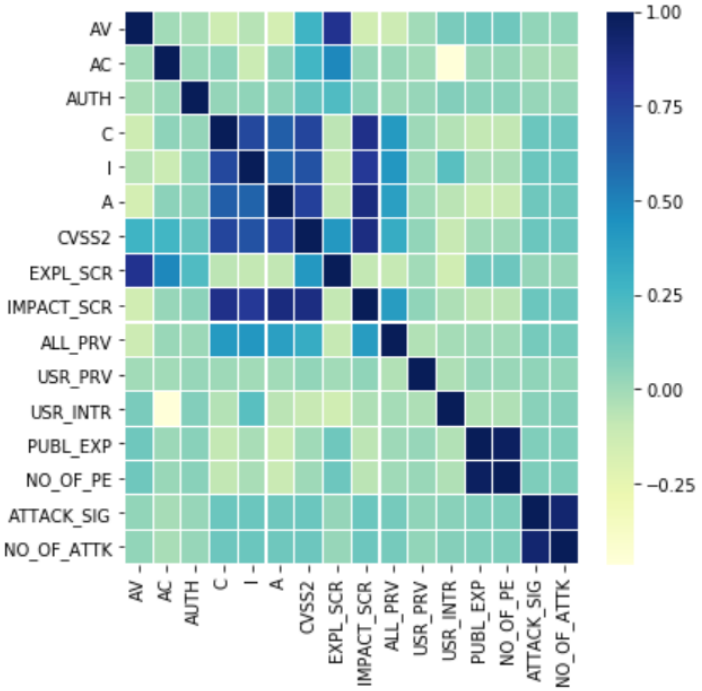

Figure 5: Correlation Test Plot

Correlation between variables can be seen in Figure 5; this will help us for the selection of feature variables during processing and getting the data ready for the model.

3.3 Data Preparation

After studying the data in the previous phase now, we cleanse and extract the features from these datasets for further processing using the models. This phase has to be done diligently as it involves cleaning the data for the empty or garbage values and marking the specific attributes that will be used for making the predictions in other words, not all the attributes present in the dataset will contribute towards the better predication and factors of the machine learning. There is a possibility if loosing or tampering data during this phase and large data size has a higher chance of having irrelevant information. Selection of features is also part of this phase and its vital to choose because not all columns add up for the better prediction and some might be completely neutral in making an impact on the prediction [18]. Further, we divide the phase into steps as, 1) File conversion 2) Stripping empty values 3) Changing values and data types 4) Feature Selection 5) Handling Imbalanced data.

1. File Conversion:
   The dataset obtained from the NVD's website about the vulnerabilities is in JSON format, for feeding this data to our model it has to be in CSV format, hence we used python 3 language to rewrite the data from JSON to CSV format.

2. Stripping the empty values:

   - In NVD's dataset there are vulnerabilities with missing CVSS scores, and the field is empty, these records are eliminated as they'll not be useful and can affect the model performance and accuracy.
   - For the remaining datasets we only obtained records whose CVE-IDs are present, as it is the only distinct attribute that help us to relate to other datasets and gain more insights on activities against or on it viz. Proof of Concept and attacks.

3. Changing values and datatypes:
   For our model we require data in numeric formats and some of the attributes have data in string format, this case is especially for the CVSS metrics related attributes that has values viz. 'Low', 'Medium', 'High', 'Single' and 'Multiple' etc. These string values represents numerical values set by the CVSS standards and we make use of numerical values to replace there corresponding string values.

   Also, we don't make use of all attributes in other datasets we derive boolean values from them, e.g. In attack signature dataset consist attack record on CVE then this respective CVE is marked as '1' for being attacked in custom generated data. Table 3 below will explain the details as follows.

From the table Table 3, the datasets EDB, EKITS, Cyberwatch represents the proof of concept (PoC) present for the weaknesses and number of PoCs hence column PUBL_EXP is derived marking presence of PoC for particular vulnerability represented through boolean value and column NO_OF_PE represents total number of PoCs for a particular vulnerability. Similarly, the datasets SYM-Malware-threats, SYM-network-attacks, SYM-Attack Signatures and Zeroday represents attacks on vulnerability based on which the columns derived are 'ATTACK_SIG' which represent attacked attempt on the vulnerability in form of boolean value and column 'NO_OF_ATTK' the number of times the vulnerability was attacked.

NVD	**CVSS Variables**	**String values**	Numeric Values
	AV (Access Vector)	Local (L)	0.395
		Adjacent Network (A)	0.646
		Network (N)	1
	AC (Access Complexity)	High (H)	0.35
		Medium (M)	0.61
		Low (L)	0.71
	Auth (Authentication)	Multiple (M)	0.45
		Single (S)	0.56
		None (N)	0.704
	C (Confidentiality)	None (N)	0
		Partial (P)	0.275
		Complete (C)	0.66
	I (Integrity)	None (N)	0
		Partial (P)	0.275
		Complete (C)	0.66
	A (Availability)	None (N)	0
		Partial (P)	0.275
		Complete (C)	0.66
EDB	**Derived Columns**	**Assigned Datatype**	**Assigned value**
EKITS	PUBL_EXP (PoC)	Boolean	0/1
Cyberwatch Public Data	NO_OF_PE (Count)	Number of PoCs	integer
SYM-Malware-threats			
SYM-network-attacks	ATTACK_SIG	Boolean	0/1
SYM-Attack Signatures	NO_OF_ATTK	Number of PoCs	integer
Zeroday			

Table 3: Extracted, Derived Features & CVSS values

4. Feature Extraction:
   This step helps to choose the most relevant elements present in our datasets and to discard that are of less importance, it helps to improve the performance and accuracy factors of the model. More the features are included in the dataset the less explanatory it becomes [19]. In this project we incorporate the features from the previous research where authors L. Allodi and F. Massaci suggest the use of data

from Symantec threat database for the attacks on vulnerabilities and for PoCs they recommended to refer the EKITS database maintained by them, hence the features attack signature, number of attacks, public exploits and number of public exploits are derived from their research learnings [7]. The derived columns are binary feature which will be used for identification of whether the vulnerability has PoCs present in the datasets and for the attack signature present in the datasets or not [16]. The next feature, the impact score was selected based on the research outcome of a Bayesian Analysis performed by the authors testing the CVSS metric through which they found out that Confidentiality, Integrity and Availability contributed most for calculating the risk of vulnerability and these attributes are used to calculate the impact score in CVSS metrics. The rest two features 'All Privileges' and 'User Interaction' were chosen based on the correlation test we conducted and also the other features selected had significant correlation with the predictor column attack signature. Table 4 summarizes the selected features for model to process.

Features	Specification
ATTACK_SIG	Attacked Vulnerabilities (Boolean value)
PUBL_EXP	Proof of concepts/ Public Exploits
IMPACT_SCR	CVSS metrics calculated using CIA subfactors
ALL_PRV	All privileges given by vulnerability
USR_INTR	User interaction required to exploit vulnerability

Table 4: Extracted Features for Model

5. Handling imbalance data:
   The data we collected has a high imbalance for the predictor column 'ATTACK_SIG' due to the fact that information related to the attacks are generally not released to public due to the reputation of the company and other reasons. Hence, there is a major difference between data comprised of discovered vulnerabilities in NVD and the vulnerabilities that are actually being exploited or attacked in wild. The imbalanced can cause the data to be biased and prediction can be inaccurate and also model getting trained for the class having maximum values. There are sampling approaches that balances the data through oversampling of minority class or under sampling of majority class or a combination of both, called as SMOTEENN. This technique fits for our research as there is presence of high imbalance of data between attacked and non-attacked weaknesses.

3.4 Modelling

In this section we apply the model to our data that has been processed in the above step. In modeling approach different models tend to give us different results for the data. Further, Random forest model is used to predict whether the vulnerability will be attacked or not.

The research conducted on proactive identification of exploits in wild by Arizona State University tried various machine learning models where Random Forest performed well and yielded better results [16]. Also, the project based on Cyberthreat discovery used multiple algorithms and Random Forest outperformed compared to other models which suggest that it is powerful algorithm that can be used for classification. Random forests is a the grouping of tree predictors where each predictor relies upon the random vector's

value that is being sampled autonomously with the uniform distribution across the trees present in the forest.

4 Design Specification

The section explains about the architecture design and process course for this research. Starting from sourcing the data from the sources and merging the data by taking the important mentions from previous literature that includes binary feature extraction for PoCs and attack signatures and extracting CVSS metric information from the sourced data. Further we cleansed the data from the datasets where records are missing the CVE-IDs and CVSS scores and metric information as it is crucial for the prediction of the attack. Next, the features are extracted based upon the research before through which most relevant columns were identified for the prediction of the attack on vulnerability. The dataset is being divided into train and test data using the stratified K-fold cross validation, SMOTEENN (SMOTE-ENN, SMOTE - Synthetic Minority Oversampling Technique, ENN - Edited Nearest Neighbours) is applied to highly imbalanced data to resample and balance it for better results. The resampled data is being given to the random forest classifier for training and the performance is being tested and analyzed through test data.

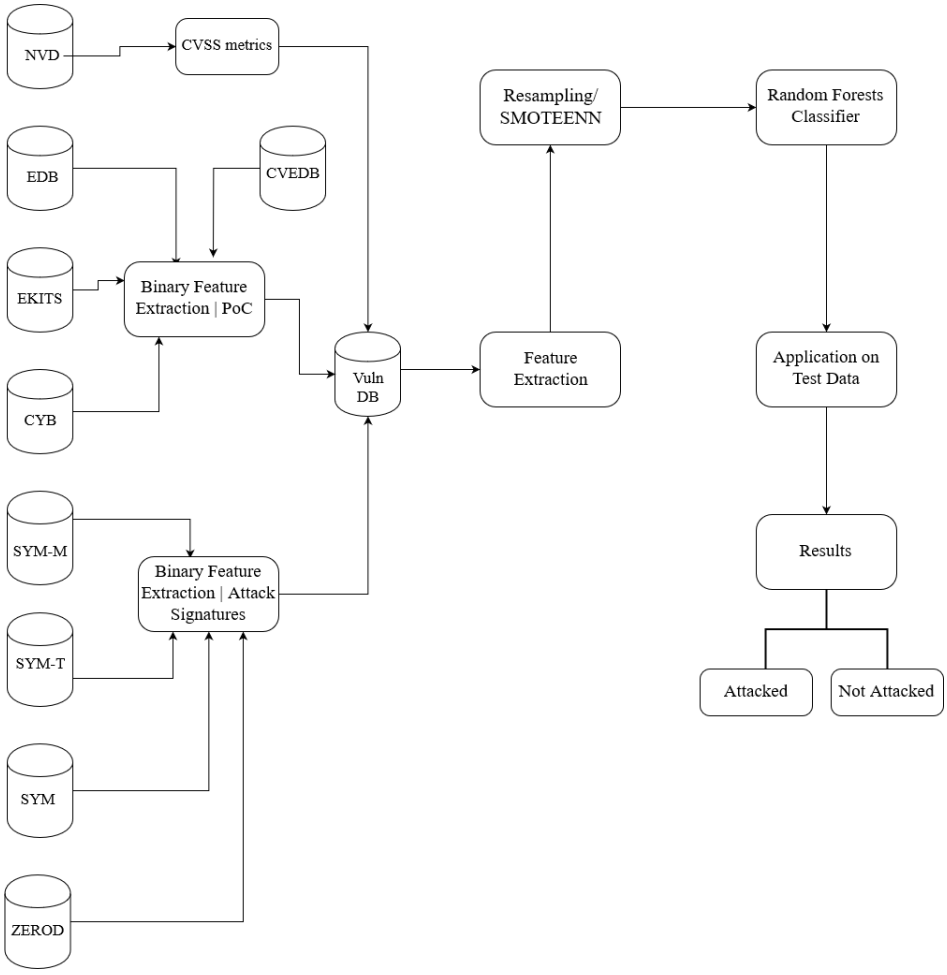

Figure 6: Data Merging & Deriving and Extraction of Features & Random Forest Model

5 Implementation

The proposed design for the prediction of an attack on the vulnerability involve various steps to be executed in a linear manner beginning from sourcing the data to obtaining desired results through machine learning model. Few datasets required to be scraped such as public database of Symantec, Cyberwatch and Cvedetails.com the scraping was performed in two methods manual and automated. For manual method it involved replicating data from datatables present on website into the excel and later saving it in CSV format. For automated process a web browser add-on named 'Web Scraper' was utilized to scrape the data and save it in an CSV format. The 'Web-Scraper' add-on needs to be configured for links and selectors to be extracted. The data sourced was from year 2004 to year 2014 as this period holds large listings of the vulnerabilities discovered.

Once the data was acquired from multiple sources and in various different files and formats, the merging of the datasets into a single dataset for the processing was done with the help of script written in Python programming language. The python version 3.5.9 was used along with KATE editor to code the script for converting the JSON data feed file into CSV and cleaning the unwanted records with empty values. Next, reading other datasets in CSV format and parsing them based on the common and unique CVE-IDs between them the different attributes were consolidated into the single dataset for further process of feature selection. The python language was chosen because it's robust, has clear syntax and has large community contributing towards it which make various libraries easily available especially for machine learning and dependency issues are resolved quickly.

The merged dataset contains the derived binary feature values that checks for the presence of public exploits also known as proof of concepts and for the presence of attack signatures in the datasets. Datasets contained varied information, set of them were parsed for PoCs and set of them for attack signatures. The functionalities for file conversion, extracting data, changing data types and cleaning was written in the same script to avoid reprocessing of files recursively for each attribute. Further the Google Colab setup was done for next level processing, google drive was mounted for file saving and python version 3.6.9 is already integrated in Colab. Such integrated development environment helps in construction of code in snippets and test them individually hence saves times. The data frame is loaded in the python from CSV file using the pandas-package's reading CSV functionality. The correlation test is conducted using the seaborn and matplotlib libraries for feature selection and prior research outcomes are taken into account as well. The selected feature's index is loaded for further processing. The seaborn library is used to plot the graph for the class distribution, through which it is identified that there is high imbalance between the classes and hence we further apply the SMOTEENN algorithm to perform hybrid sampling to synthetically balance the data.

The 'sklearn' package is a machine learning package in python containing classifier algorithms and many more functionalities viz. StratifiedKFold which is used to split the data into training and test datasets. Finally, the data after SMOTEENN iteration is fed to the Random Forest model functions imported from the sklearn learn library for processing the data and predicting the outcome. The model generates the results in form metrics viz. true positives, true negatives, false positives and false negatives in multiple iterations. For each iterations the metrics values are stored in list and at the end the average of each value is calculated and that will be the final outcome of the model. The model's performance was evaluated based on the accuracy, specificity, sensitivity and geometric mean of sensitivity & specificity was analyzed.

6 Evaluation

Assessing the model performance is done by determining the accuracy, sensitivity or recall, specificity and geometric mean of sensitivity/specificity. The metrics, True positives TP, True Negatives TN, FALSE Positives FP and False Negatives FN can be derived from the confusion matrix. The metrics in our research for predicting the attacks on vulnerabilities indicates as:

- TP - Attacked vulnerabilities, and model predicted the attack.
- TN - Not attacked vulnerabilities, and model predicted no attacked vulnerabilities.
- FP - Not attacked vulnerabilities, and model predicted vulnerabilities were attacked.
- FN - Vulnerabilities attacked and model predicted no attacked vulnerabilities

Now, we calculate the model's accuracy, sensitivity or recall, specificity and geometric mean as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Recall/Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$GM = \sqrt{Sensitivity.Specificity} \tag{4}$$

SMOTEENN	**Accuracy %**	**Specificity %**	**Recall/ Sensitivity %**	**Geometric Mean %**
NO	98	0	0	0
YES	83	83	58	69

Table 5: Output with & without use of SMOTEENN

Following two cases were considered:

6.1 Case 1: Model's Performance on unbalanced data:

The model was executed with the original data that was highly imbalanced between the class attacked and not attacked. From the Table 5; it can be stated that the accuracy is too high at 98% and the recall and other attributes are nil which suggests that model has failed to the predict the attack on the vulnerabilities. The high imbalance has caused the model to be biased and generally over fit. In order to tackle this problem and hybrid balancer SMOTEENN was applied to the data before running the model and later evaluated.

6.2 Case 2: Model's Performance on balanced data:

Now, after applying the SMOTEENN and balancing the data the model was executed and have yielded results for accuracy and specificity at 83%, Recall at 58% and the geometric mean for specificity and sensitivity is at 69%. These results are better compared to the imbalanced data and as a recall is crucial for the better prediction result of the model. Based on these results, it can be confirmed that machine learning algorithm random forest can be applied to predict whether a vulnerability can be attacked in the wild.

6.3 Discussion

The balancing of data with SMOTEENN proved to be effective for classifier to improve the performance and its related metrics. The recall metric explains how accurate was the prediction made by our model and this metric is what we depend upon to classify the vulnerabilities that are prey to the attacks in wild, this will help the security professional or administrator to proactively prepare the mitigation techniques. Though in our case the recall value is consider satisfactory but still needs to be good so as to strongly predict the attack on the vulnerabilities. The other metric specificity contributed well towards prediction of not attacked cases and in our project it has produced good results of vulnerability that are not prone to attack but our focus is on the weakness that are susceptible to attacks in the wild which can be predicted well by our developed model if recall percentage is good enough. Also, in this research the feature selected were few, still the result obtained, and the performance of model is satisfactory.

It can be noticed that due the use of hybrid balancer SMOTEENN there was a good improvement in all of the metrics especially the recall metric, given the fact that the data about the attacked vulnerability is sparsely present over the public databases and it is challenging to build a good quality dataset for our models but techniques like SMOTEENN helps to bridge the gap of limited data and can improve the prediction capability of the model.

7 Conclusion and Future Work

In the prior research, the CVSS faced reproval about failing to predict the risk factor of the vulnerability, few types of research did analyse that certain sub metrics are effective especially the Confidentiality, Integrity and Availability which contributed to Impact score. CVSS alone is not enough for prioritising the vulnerabilities, considering this we make use of binary features derive from the PoCs and Attack Signature datasets along with the CVSS impact score and attributes such as 'all privileges', 'user interaction' has promising results when it comes to prioritising the vulnerabilities with the help of machine learning. Feature extraction was done with the help of earlier research and correlation test to identify the best columns for prediction and has a satisfactory outcome and can assist the IT security domain to a certain level of vulnerability assessment.

In future, a better resource for vulnerability's attack-related data and along with better feature extraction techniques can improve the result of the models. The current study was considering vulnerabilities in general, in future classification can be done based on infrastructure, software and IoT related vulnerabilities along with better attributes, e.g. outdated systems, patch available and life of vulnerability.

Acknowledgment

I want to offer my gratitude to my supervisor Dr. Muhammad Iqbal for his valuable guidance and sharing his insights on data analysis and machine learning and consistent support, which kept me motivated throughout the journey. I am grateful to my industry supervisors for providing me with an opportunity to conduct research in their reputed organisation and also for their teachings and knowledge they shared. My special thanks to Department of I.T., University of Trento, Italy for permitting me to use their valuable datasets. I would also like to offer my thanks to our course director Dr. Arghir Moldovan and professor Dr. Irina Tal for helping me throughout the master course. Lastly, I am fortunate to have limitless support from my family and friends who have encouraged me and making the master expedition a great success.

References

[1] A. A. Younis and Y. K. Malaiya, "Comparing and evaluating cvss base metrics and microsoft rating system," in *2015 IEEE International Conference on Software Quality, Reliability and Security*. IEEE, 2015, pp. 252–261.

[2] R. Wang, L. Gao, Q. Sun, and D. Sun, "An improved cvss-based vulnerability scoring mechanism," in *2011 Third International Conference on Multimedia Information Networking and Security*. IEEE, 2011, pp. 352–355.

[3] S. Nanda and U. Ghugar, "Approach to an efficient vulnerability management program," 2017.

[4] P. Johnson, R. Lagerström, M. Ekstedt, and U. Franke, "Can the common vulnerability scoring system be trusted? a bayesian analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 1002–1015, 2016.

[5] L. Allodi and F. Massacci, "A preliminary analysis of vulnerability scores for attacks in wild: the ekits and sym datasets," in *Proceedings of the 2012 ACM Workshop on Building analysis datasets and gathering experience returns for security*. ACM, 2012, pp. 17–24.

[6] K. Scarfone and P. Mell, "An analysis of cvss version 2 vulnerability scoring," in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, 2009, pp. 516–525.

[7] A. Dobrovoljc, D. Trček, and B. Likar, "Predicting exploitations of information systems vulnerabilities through attackers' characteristics," *IEEE Access*, vol. 5, pp. 26 063–26 075, 2017.

[8] C. P. Pfleeger and S. L. Pfleeger, *Security in computing*. Prentice Hall Professional Technical Reference, 2002.

[9] S. Vidalis and A. Jones, "Analyzing threat agents and their attributes." in *ECIW*, 2005, pp. 369–380.

[10] E. KEARY. (2019) 2019, vulnerability statistics report. [Online]. Available: https://www.edgescan.com/wp-content/uploads/2019/02/edgescan-Vulnerability-Stats-Report-2019.pdf

[11] A. A. Younis, Y. K. Malaiya, and I. Ray, "Using attack surface entry points and reachability analysis to assess the risk of software vulnerability exploitability," in *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*. IEEE, 2014, pp. 1–8.

[12] L. Maghrabi, E. Pfluegel, L. Al-Fagih, R. Graf, G. Settanni, and F. Skopik, "Improved software vulnerability patching techniques using cvss and game theory," in *2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*. IEEE, 2017, pp. 1–6.

[13] M. Keramati and M. Keramati, "Novel security metrics for ranking vulnerabilities in computer networks," in *7'th International Symposium on Telecommunications (IST'2014)*. IEEE, 2014, pp. 883–888.

[14] M. Keramati, "A novel system for quantifying the danger degree of computer network attacks," in *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*. IEEE, 2017, pp. 0804–0809.

[15] A. Ur-Rehman, I. Gondal, J. Kamruzzuman, and A. Jolfaei, "Vulnerability modelling for hybrid it systems."

[16] M. Almukaynizi, E. Nunes, K. Dharaiya, M. Senguttuvan, J. Shakarian, and P. Shakarian, "Proactive identification of exploits in the wild through vulnerability mentions online," in *2017 International Conference on Cyber Conflict (CyCon US)*. IEEE, 2017, pp. 82–88.

[17] A. I. R. L. Azevedo and M. F. Santos, "Kdd, semma and crisp-dm: a parallel overview," *IADS-DM*, 2008.

[18] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Data preprocessing for supervised leaning," *International Journal of Computer Science*, vol. 1, no. 2, pp. 111–117, 2006.

[19] H. Liu and H. Motoda, *Feature extraction, construction and selection: A data mining perspective*. Springer Science & Business Media, 1998, vol. 453.