Machine learning algorithms for conversion of CVSS base score from 2.0 to 3.x*

Maciej Nowak[1 [0000−0002−7747−4867]], Michał Walkowski[1 [0000−0003−4887−6098]], and Sławomir Sujecki[1 [0000−0003−4588−6741]]

Department of Telecommunications and Teleinformatics, Wroclaw University of Science and Technology, 50-370 Wroclaw, Poland
{maciej.nowak, michal.walkowski, slawomir.sujecki}@pwr.edu.pl

Abstract. The Common Vulnerability Scoring System (CVSS) is the industry standard for describing the characteristics of software vulnerabilities and measuring their severity. However, not all publicly known vulnerabilities have criticality rating in CVSS 3.x, which is the latest and most advanced version of the standard. This is due to the large time gap between the publication of the CVSS 2.0 and CVSS 3.x standards, the large number of the detected and published vulnerabilities at the time, and significant differences in the method of determining vulnerability criticality and assigning vector properties to evaluation components. Consequently, organizations using CVSS to prioritize vulnerabilities use both CVSS versions and abandoned the full transition to CVSS 3.x standard. In this paper authors introduce machine learning algorithms for performing conversions from CVSS 2.0 to CVSS 3.x, scores, which should significantly facilitate the upgrade to CVSS 3.x standard for all stakeholders. The considered case corresponds to a real world application with a large potential impact of the research.

Keywords: Security · CVSS Score · Well-known Vulnerabilities · Machine Learning.

1 Introduction

The Common Vulnerability Scoring System (CVSS) is the industry standard for describing the characteristics of a software vulnerability and measuring its severity [6]. CVSS was first introduced as a research project by the National Infrastructure Advisory Council (NIAC) in 2005 [18], and afterwards accepted by other organizations [26]. It consists of three categories: Base (B_S), Temporal (T_S) and Environmental (E_S). The B_S category is marked by vulnerability properties that are time invariable. The B_S category properties include access complexity, access vector and the assessment of the extent to which the vulnerability may threaten confidentiality, integrity and system availability. The T_S category describes properties that may change over time. In particular, the T_S category contains information on the existence of a public exploit and patches

* Supported by organization Wroclaw University of Science and Technology.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

availability, whilst the E_S category includes information on the systems environment affected by the detected vulnerability.

The standard update to version 3.0 was published in 2015 [10] and significantly differs from the previous version in the method of determining the vulnerability criticality and assigning vector properties to the evaluation component. The following changes were introduced in the B_S category in the latest CVSS version (Table 1):

- the measurement parameters of confidentiality, availability and integrity, have been changed to: none, low or high,
- a physical value (P) has been added to the attack vector, which indicates the security vulnerability in which the attacker must have a physical access to the system in order to be able to exploit it,
- the required permissions (PR) parameter has been added to indicate whether administrative permissions or other need to be achieved on the target system, in order to successfully exploit the vulnerability.

Table 1. Difference between CVSS 2.0 and CVSS 3.1. properties of B_S category

CVSS 2.0	CVSS 3.1
Access Vector (AV)	Attack Vector (AV)
Access Complexity (AC)	Attac Complexity (AC)
Authentication (Au)	Privileges Required (PR)
	User Interaction (UI)
Confidentiality Impact (C)	Confidentiality Impact (C)
Integrity Impact (I)	Integrity Impact (I)
Availability Impact (I)	Availability Impact (A)
	Scope (S)

In the T_S category, the name of the vector *Exploitability* was changed to *Exploit Code Maturity*, whereas the rest of the vectors remained unchanged. In the E_S category, the implemented change involves the replacement of two indicators with so-called modified base scores. Essentially, each of the base metrics can be modified by the organization in order to reflect differences in the tested environment (Table 2).

After the publication of the CVSS 3.0 standard, a large number of companies criticized it for inaccurate parameter descriptions and thereby allowing for their different interpretation [14]. In 2019, the CVSS 3.1 update was released, which specified the description of the metrics, making them more understandable for the recipient. However, the method of calculating the vulnerability assessment, as well as the parameters form and vector representation, have not changed [11].

Since there is a 10-year difference between the publications of the CVSS 2.0 and CVSS 3.0 standard, it was not possible to convert all scores from CVSS 2.0 to the newer version automatically. Consequently, only new vulnerabilities

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

Table 2. Difference between environmental category for CVSS 2.0 and CVSS 3.1.

CVSS 2.0	CVSS 3.1
Collateral Damage Potential (CDP)	
Target Distribution (TD)	
	Attack Vector (MAV)
	Attack Complexity (MAC)
	Privileges Required (MPR)
	User Interaction (MUI)
	Scope (MS)
	Confidentiality Impact (MC)
	Integrity Impact (MI)
	Availability Impact (MA)
	Confidentiality Impact (MC)
	Integrity Impact (MI)
	Availability Impact (MA)
Confidentiality Requirement (CR)	Confidentiality Requirement (CR)
Integrity Requirement (IR)	Integrity Requirement (IR)
Availability Requirement (AR)	Availability Requirement (AR)

contain both criticality assessments (currently 73179), the remaining 73856 vulnerabilities, which accounts for 51% of the known vulnerabilities, so far have not been assigned the CVSS score according to the latest version of the standard [20]. Consequently, organizations using CVSS in order to prioritize vulnerability fixes, use both versions [8]. The described situation also causes delays in publishing the vulnerability with its assessment [6], which additionally increases the time gap between the vulnerability publication and its repair [26]. Not every factor responsible for the criticality of the given vulnerability will not be considered, if the CVSS 2.0 is used. Moreover, in case the person responsible for maintaining the IT service obtains the same vulnerability with two different assessments (CVSS 2.0 and CVSS 3.x), the use of these both standards causes a dissonance in vulnerabilities prioritization.

The main objective of this work is to calculate all components of the CVSS vector for B_S category for all the remaining 73856 vulnerabilities, which have not so far been assigned the base score. The novel contribution of this work consists in performing automatic conversions from CVSS 2.0 to CVSS 3.x base score, i.e. B_S category, using machine learning algorithms. The value of this contribution is further enhanced by the fact that the machine learning algorithms are applied to difficult data derived from CVSS database and that the considered case corresponds to a real world application with a large potential impact of the research. This paper is divided into the following sections:

- Related Work - section presents other work related to the present topic. It is devoted to a brief description of work related to machine learning algorithms and CVSS predictions.
- Methods – section presents the description of methods used for the preparation of the database and the machine learning algorithms.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

– Results - section contains the discussion of the results describing the advantages of the proposed machine algorithms.
– Conclusions - section gives the summary of the presented results and introduces fields for further research.

2 Related Work

A large number of authors criticized the currently used CVSS standard explicitly and attempted to improve the criticality assessment of the detected vulnerabilities, [21, 22, 7, 15]. Despite the aforementioned criticism, the CVSS standard is widely used by corporations to prioritize vulnerability fixes, supporting the vulnerability management process [8]. However, as noticed in [26] each organization approaches the problem differently. Therefore, in order to help organizations to streamline the process by more effective adaptation of the vulnerability assessment to the vulnerability assessment more efficiently to the consequences of its use, new methods were proposed, for instance, by introduction of solutions enabling the vulnerability predictions [27, 12] or facilitating vulnerability assessment using patterns and machine learning [24]. Additionally, the developed algorithms for automatic estimation of the CVSS B_S score, which is obtained from the vulnerability description in order to accelerate the publication process in public National Vulnerability Databases (NVD) [6, 13] and to include other categories such as T_S [23] and E_S [17, 26]. Despite the indicated works, corporations are still forced to use both standards in order to prioritize the vulnerability correctly [8].

However, to the best knowledge of the authors nobody tried so far to use machine learning to convert scores from CVSS 2.0 to CVSS 3.x standard by estimating all the component values of the result vector included in the final assessment, in order to facilitate the stakeholder vulnerability management process only by the latest version of the standard.

3 Methods

In this section first the description of methods used for the preparation of the database is given and then is followed by the description of the machine learning algorithms.

Open-source software suite: Vulnerability Management Center (VMC) [26, 25, 5], was used to prepare a database for known vulnerabilities and weaknesses considering information collected from the public sources. The VMC software also provides an Application Programming Interface (API) for Python that was used to write a script for selection of training data of the vulnerability criticality assessment for both CVSS 2.0 and CVSS 3.x standards. This generated a database with 73179 items. For each item, the script transformed an alphanumeric value of the vector to a numerical form using the CVSS 2.0 specification, cf. an example for Access Vector field given in Table 3 [10]. Similarly, the data has been converted from an alphanumeric form of the CVSS 3.x vector to the

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

numbers of the corresponding classes, cf. an example given in Table 4 for Attack Vector field. Thus one obtains a 7 element vector for B_S CVSS 2.0 standard and an 8 element vector for CVSS 3.x standard. The allowed values of the vectors' elements are defined by the respective CVSS 2.0 and 3.x standards. The B_S scores that are known for both CVSS 2.0 and 3.x standards establish a mapping between both sets of vectors. An initial analysis of this mapping revealed however, that there are many instances of CVSS 2.0 vectors that map spuriously on different CVSS 3.x vectors. As a consequence the machine learning algorithms applied to this mapping performed moderately. Therefore, in order to reduce the number of vectors that map spuriously and to improve the performance of the machine learning algorithms the CVSS 2.0 vector was augmented by 50 elements. The additional elements of the CVSS 2.0 vector were derived from the description fields of each vulnerability, which are available from the CVSS 2.0 vulnerability database. Performing the statistical analysis of the available descriptions, the 50 most frequently occurring keywords were selected, ordered and converted into numbers. The details of this process can be found at [16]. The number corresponding to a keyword was obtained by counting the number of occurrences and dividing by 100. Thus obtained vectors were concatenated with the existing 7 element CVSS 2.0 vector forming a 57 element extended CVSS 2.0 vector. The necessary text processing was carried out using library Natural Language Toolkit (NLTK) [2].

After extending the CVSS 2.0 vector a new mapping was established between the B_S scores that are known for both CVSS 2.0 and 3.x standards, which is the subject of the subsequent machine learning analysis. Table 5 gives an insight into the nature of this mapping. As can be seen there is a large imbalance in the learning data. The greatest imbalance occurs for the AV parameter, where 841 CVSS 2.0 vectors were obtained for the class with index 0, while class 3 contains 53732 vectors. The difficulty of the problem is further increased by a large number of the considered classes.

Table 3. Enumerated members for CVSS 2.0 field Access Vector (AV) and their associated numerical values as described by the CVSS 2.0 specification.

Metric Value	Numerical Value
Local (AV:L)	0.395
Adjacent Network (AV:A)	0.646
Network (AV:N)	1.0

For the defined mapping the training set formation procedure was performed in two stages. In stage one undersampling [9] was performed. This stage involves the following 2 steps:

– for each CVSS 3.x vector class the median of CVSS 2.0 extended vectors is obtained by selecting the most frequently occurring CVSS 2.0 extended vector.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

Table 4. Enumerated members for CVSS 3.x field Attack Vector (AV) and their proposed associated numerical values.

Metric Value	Numerical Value
Network (N)	3
Adjacent (A)	2
Local (L)	1
Physical (P)	0

Table 5. The number of CVSS 2.0 vectors that map onto a specific classs of the CVSS 3.x vector.

CVSS 3.x	Class index	No of CVSS 2.0 vectors	CVSS 3.x	Class index	No of CVSS 2.0 vectors
AV	1	841	UI	1	26449
	2	1573		2	46730
	3	17033	C	1	14074
	4	53732		2	15932
AC	1	6021		3	43173
	2	67158	I	1	12818
PR	1	4452		2	22831
	2	19380		3	37530
	3	49347	A	1	1883
S	1	12156		2	28440
	2	61023		3	42856

– Once the median is known the 80 most correlated vectors with the median were selected,

In stage two all the vectors selected for in stage one were included in a common training set. Thus $22 \cdot 80 = 1760$ vectors were obtained, where 23 turned out to be common and hence were removed. After completing stage two, even though the obtained data is imbalanced it is related to a constant selection rule.

Vectors that are not used for training constitute a testing set. The testing set must contain vectors from each class of all the CVSS 3.x vector components. For the initial optimization the testing sets containing 100 vectors were used. The used testing sets were selected randomly and the selection procedure was repeated 5 times. The statistics obtained from the results were used to select the most appropriate classification models for conversion from B_S CVSS 2.0 to B_S CVSS 3.x scores.

In the final stage the estimation of the classification effectiveness for the selected algorithms was conducted carrying out 30 trials with testing sets, containing randomly selected 1000 vectors.

Preprocessing was carried out using the Principal Component Analysis (PCA). Then, the obtained data was processed using six machine learning algorithms:

– Naive Bayes classifier (NB),

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

– k-nearest neighbors algorithm (kNN) with euclidean metric (E) and cosine metric (C), 3 closest neighbors were attained
– Kernel Support Vector Machine (KSVM) in three configurations - with a linear kernel function and the two others, using Gaussian Radial Basis Functions (GRBF). KSVM (GRBF) has been split into a non-trained kernel (UGRBF) version and a trained kernel (TRBF) version, which adopts interrelated 2-fold Cross-validation (2-fold CV), Misclassifcation Ratio (MCR) and NM algorithm with 30 starting points set.

The detailed description of these algorithms can be found in [3, 1, 4]. More detailed information regarding TRBF is provided in [19]. The results obtained using the described algorithms are presented in the next section.

4 Results

In this section the accuracy of CVSS 3.x score prediction using algorithms which were described in section 3 when applied to the CVSS classification problem (section 3) is studied. The original data has been preprocessed by PCA. The algorithm's usefulness for the classification of the CVSS 3.x vector was analyzed using algorithms described in section 3. All classification models were implemented in MATLAB 2015b and were run on a computing server with two CPUs [Intel Xeon®X5650, 2.66 GHz].

Figure 1 shows the average accuracy for the selected machine learning algorithms calculated by performing five test trials on 100 randomly selected vectors. The accuracy was calculated by counting the number of correct predictions and dividing by the number of all testing vectors, i.e. 100 in this instance.

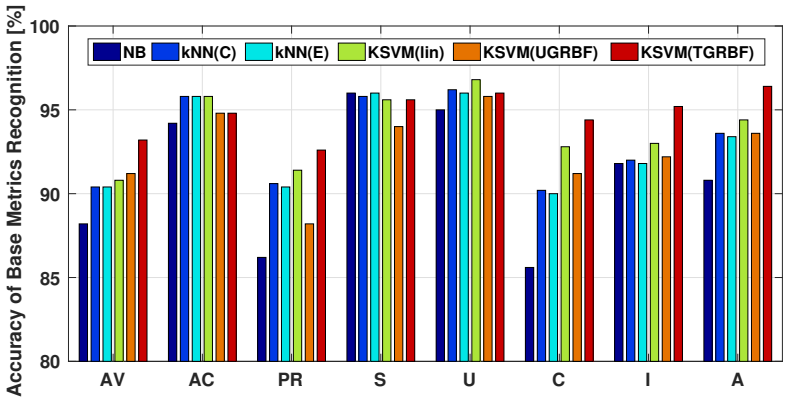

Fig. 1. The comparison of the average accuracy of the CVSS 3.x vector parameters recognition with extended CVSS 2.0 vector, using six algorithms: NS, kNN (C), kNN (E), KSVM (lin), KSVM (UGRBF) and KSVM (TGRBF)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

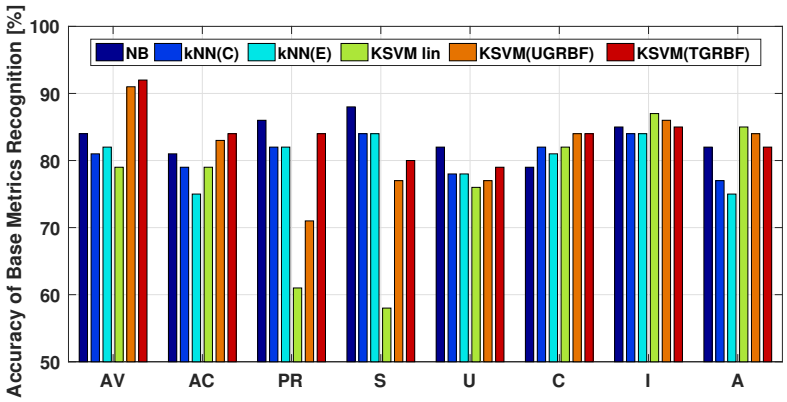

Fig. 2. The comparison of the average accuracy of the CVSS 3.x vector parameters recognition with original CVSS 2.0 vector, using six algorithms: NS, kNN (C), kNN (E), KSVM (lin), KSVM (UGRBF) and KSVM (TGRBF)

Results from Figure 1 show that for the AV parameter the highest average classification efficiency was obtained for KSVM (TRGBF) and is equal to 93.2%. Thus KSVM (TRGBF) can be qualified for the next stage of the research.

Considering the AC parameter high average efficiency - 95.8% was recorded for 3 algorithms: kNN (C), kNN (E) and KSVM (lin). The linear class separability algorithms results indicate significant advantage over more complex algorithms - KSVM (UGRBF) and (TRGBF), whose efficiency is equal to 94.8%. NB obtains the lowest efficiency level. However, it still exceeds 94%. Consequently, also kNN (C), kNN (E) and KSVM (lin) can be selected for the next stage of the research.

Considering the PR parameter a significant decrease in the average efficiency with respect to the classification of other parameters of the CVSS 3.x vector, is visible. Thus, the classification constitutes a concern for NB and KSVM (UGRFB). Due to kernel's parameters tuning, KSVM (TRGBF) obtains a 4.4% higher efficiency than the version with no adaptation - 92.6%. Consequently, KSVM (UGRFB) algorithm proved to be best suited for this task.

With respect to S parameter all five algorithms obtained an average efficiency of over 95.5%. The KSVM (UTRGBF) attained the worst performance. Due to the kernel's adaptation, the SVM (TGRBF) obtained the same efficiency as the KSVM with the linear function - 95.6%. The highest efficiency corresponds to simple algorithms - NB and kNN (E) 96% each. Thus, they can be selected for further analysis.

All the algorithms classify the U parameter with high accuracy, obtaining an efficiency result in the range of 95%. For kNN (E), the highest average accuracy, of all the algorithms marked, was 96.8%. The above-mentioned average constitutes the highest average efficiency classification attained among all parameters. Therefore, kNN (E) proves to be the only suitable candidate for further analysis.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

Similarly to PR, C seems to be difficult to classify, therefore two algorithms were selected for further tests - KSVM (TRBF) with an average efficiency equal to 94.4% and kSVM (lin) - 92.8%. The remaining algorithms, especially the kNN and NB families, are marked by a noticeably lower efficiency – in the range of (85.6-90.2)%.

For I parameter the classification with KSVM (TRBF) provides an average efficiency in the range of 95.2% and is 2.2% higher than the second KSVM (lin) in order. The remaining algorithms constitute a similar efficiency level – in the range of (91.8-92.2)%. Thus the KSVM (TGRBF) becomes the only candidate.

In the case of the A parameter the obtained results of the average classification efficiency indicate a 2% advantage of KSVM (TGRBF) over KSVM (lin) with an exceptional result of 94.4%. Therefore, both algorithms can be accepted for further analysis. The remaining algorithms also provide a satisfying average classification efficiency, with the worst result indicating 90.8% for NB.

For the sake of comparison the calculations presented in Fig. 1 were repeated for the short (7 element) CVSS 2.0 vector. The results of these calculations are summarised in Fig. 2. Comparing the results from Fig. 2 and Fig. 1 clearly shows the advantages of using the extended CVSS 2.0 vector for for the considered machine learning problem.

The selection of the best value for the number of PCs is performed in order to maximize the classification accuracy and to obtain repeatability of the results. If several algorithms were initially selected whilst considering the results from Fig. 1, then after PCs analysis only the best performing algorithms were selected. These algorithms are listed in Table 6 together with the corresponding number of PCs and the average classification accuracy with standard deviation.

Table 6. The comparison of the best candidates (algorithms) for classification, including the number of PCs and the indicated average accuracy obtained from the 5-time testing trial of the algorithms with a randomly selected 100-element testing set.

Base Metrics	Algorithm	Number of PCs / average accuracy	Alternative algorithm	Number of PCs. average accuracy
AV	KSVM(TGRBF)	31 / $93.2 \pm 2.64\%$	-	-
AC	kNN(E)	26 / $95.8 \pm 1.87\%$	-	-
PR	KSVM(TGRBF)	11 / $92.6 \pm 2.83\%$	KSVM(lin)	52 / $91.4 \pm 4.44\%$
S	NB	33 / $96.0 \pm 1.60\%$	-	-
U	KSVM(lin)	46 / $96.8 \pm 1.76\%$	-	-
C	KSVM(TRGBF)	8 / $94.4 \pm 1.43\%$	KSVM(lin)	51 / $92.8 \pm 1.72\%$
I	KSVM(TGRBF)	36 / $95.2 \pm 0.74\%$	-	-
A	KSVM(TGRBF)	22 / $96.4 \pm 1.72\%$	-	-

Quality of classification was estimated by calculating MCR, Precision, Recall and F1-score. The classification results obtained with algorithms tested are compared using 10 repetitions of 5-fold CV. The results obtained are presented in Table 7.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

Table 7. Precision, Recall, F1-score and MCR calculated for algorithms listed in Table 6

CVSS 3.x	Class index	Precision	Recall	F1 score	MCR [%]
AV	1	0.1707	0.4487	0.2443	14.16 ± 0.78
	2	0.7926	0.6921	0.7387	
	3	0.7144	0.9497	0.8145	
	4	0.9550	0.9340	0.9444	
AC	1	0.9529	0.9433	0.9480	9.34 ± 0.20
	2	0.5809	0.6279	0.6034	
PR	1	0.9361	0.9588	0.9472	18.15 ± 1.12
	2	0.7143	0.6791	0.6931	
	3	0.3930	0.3979	0.3847	
S	1	0.8698	0.9271	0.8976	15.35 ± 0.83
	2	0.7670	0.6340	0.6941	
U	1	0.9281	0.9592	0.9434	7.09 ± 0.33
	2	0.9307	0.8809	0.9051	
C	1	0.9863	0.9885	0.9874	10.13 ± 0.50
	2	0.7434	0.8475	0.7904	
	3	0.9316	0.8784	0.9039	
I	1	0.9931	0.9641	0.9784	7.40 ± 0.29
	2	0.8056	0.9183	0.8579	
	3	0.9345	0.8953	0.9144	
A	1	0.9867	0.9585	0.9724	7.76 ± 0.21
	2	0.2724	0.8021	0.4049	
	3	0.9629	0.9031	0.9320	

Next, the classification statistics for the selected algorithms is calculated. Fig. 3 shows the calculated statistics of classifying the CVSS 3.x vector parameters for the selected in Table 6 algorithms. For this calculations the testing sets consisted of 1000 elements while the tests were repeated 30 times. The greatest dispersion of the classification accuracy occurs for the parameters PR and C. For the KSVM(lin) algorithm, used for PR classification, the efficiency range equals (79.9-93.5%, considering the outlier. The median is equal to 88.35%. The median value is higher - 88.6%, when using KSVM (TGRBF). However, the values range dispersion is larger and corresponds to (72.9-92.1)%. Despite the higher median value when compared with KSVM(lin), KSVM (TGRBF) performed worse. In the case of parameter C, the situation is similar. The alternative KSVM (lin) algorithm, except for the lower adjacents, indicates a higher rate for minimum, maximum and median values - 87.1%, 94.1% and 90.95%, respectively, in comparison to 82.3%, 91.8% and 88.8% obtained for KSVM(TGRBF). It is noted that 5 outliers were obtained for KSVM (TGRBF). The classification accuracy calculated using 1000 vectors for the remaining CVSS 3.x parameters is also lower than the one shown in Table 6. The median values for these parameters are as follows: AV - 89.75%, AC - 93.3%, S - 95.0%, U - 94.25%, I - 92.6%, A - 94.80%. The dispersion of values for these parameters is significantly smaller than for PR and C parameters.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

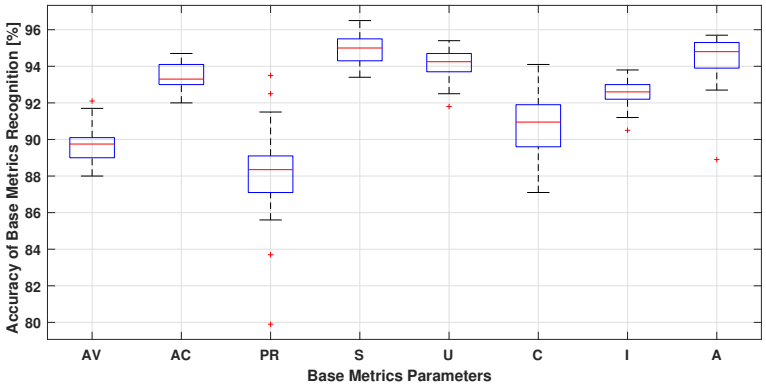

Fig. 3. The classification statistics of CVSS 3.x vector parameters using the program developed considering Table 6

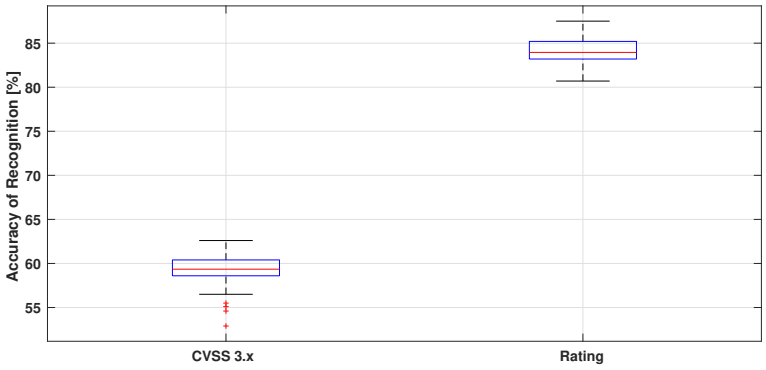

Fig. 4. The efficiency statistics of the B_S CVSS 2.0 to B_S CVSS 3.x conversion and Qualitative Severity Rating Scale levels.

Full statistical analysis for algorithms used is shown in Table 8.

Comparing results from Tables 7 and 8 one can observe that AV, PR and S from validated using real tests perform better than what cross-validation would suggest (100% - MCR[%] vs. accuracy and F1-score). Additionally, with 10-fold CV when compared with 5-fold CV MCR is reduced by 0.6%, 0.35% and 2.1%, respectively for AV, PR and S, whilst other parameters stay unchanged. This implies that AV, PR and S are very sensitive to the number of samples of the learning set. These observations particularly apply to the first class in the case of AV and third class for PR.

Finally, the statistics of B_S CVSS 2.0 to CVSS 3.x conversion accuracy for the and the Qualitative Severity Rating Scale levels and the final CVSS 3.x score

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

Table 8. Precision, Recall, F1-score and MCR calculated for algorithms listed in Table 6 obtained from the 30-time testing trial of the algorithms with a randomly selected 1000-element testing set.

CVSS 3.x	Class index	Precision	Recall	F1 score	Accuracy
AV	1	0.2167	0.2273	0.2198	89.75 ± 1.0
	2	0.7296	0.8274	0.7674	
	3	0.8201	0.9429	0.8672	
	4	0.9626	0.9334	0.9474	
AC	1	0.9592	0.9693	0.9642	93.4 ± 0.73
	2	0.6541	0.5854	0.6163	
PR	1	0.9790	0.9582	0.9683	88.12 ± 2.49
	2	0.7565	0.8086	0.7744	
	3	0.3219	0.3100	0.3137	
S	1	0.9851	0.9558	0.9702	94.94 ± 0.71
	2	0.7689	0.9097	0.8330	
U	1	0.9454	0.9629	0.9539	94.15 ± 0.81
	2	0.9345	0.9064	0.9197	
C	1	0.9813	0.9834	0.9823	90.72 ± 1.75
	2	0.7198	0.8059	0.7537	
	3	0.9407	0.9149	0.9269	
I	1	0.9828	0.9508	0.9665	92.44 ± 0.77
	2	0.8489	0.8377	0.8420	
	3	0.9141	0.9387	0.9260	
A	1	0.9601	0.9589	0.9595	94.47 ± 1.30
	2	0.1961	0.4974	0.2686	
	3	0.9646	0.9481	0.9562	

as described in the CVSS 3.x documentation [10] was conducted taking into consideration 30 repetitions of a randomly selected set of 1000 testing vectors, are shown in Fig. 4. Due to the conversion of B_S CVSS 2.0 to 3.x considering 8 parameters related to the formula described in the CVSS 3.1 documentation, the expected accuracy value can be approximated by the product of all accuracy values obtained for each component of CVSS 3.x vector. Thus obtained value using the numbers shown in Fig. 3 agrees with the results shown in Fig. 4 to within + 5%. Further, it is noted that the median accuracy of converting B_S CVSS 2.0 to 3.x is equal to 59.35%.

For conversion accuracy to Qualitative Severity Rating Scale levels the median accuracy equals 83.95% and all values are included in the range (80.7-87.5)%.

The results described above were confirmed by performing an extreme test - testing with a set consisting of 50,000 vectors. This trial yielded a B_S 2.0 to 3.x conversion efficiency, equal to 59.82% and conversion to QSRS levels indicating 84.12%, which do not differ significantly from the ones shown in Fig. 4.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

5 Conclusions

The paper presents machine learning algorithms for converting B_S CVSS 2.0 scores to B_S CVSS 3.x ones. In order to improve the learning accuracy the CVSS 2.0 vector was augmented by 50 keywords selected from the description field of each vulnerability. The results obtained confirmed the improved performance of the machine learning algorithms when applied to the extended CVSS 2.0 vectors. Further it was found that different machine learning algorithms are best suited for different CVSS 3.x vector components. For the CVSS 3.x components AV, C, I, A the best performing algorithm is KSVM(TGRBS) while for parameters PR, U and C best results are obtained with KSVM(lin). Finally, for AC parameter kNN algorithm performed best while NB algorithm did best for S parameter. The future research will focus on improvement of classification for parameters AV, PR and C.

6 Acknowledgments

The authors wish to thank Wroclaw University of Science and Technology (statutory activity) for financial support and Agata Szewczyk for proofreading and translation.This publication was created as a part of the Regional Security Operations Center (RegSOC) project (Regionalne Centrum Bezpieczeństwa Cybernetycznego), cofinanced by the National Centre for Research and Development as part of the CyberSecIdent-Cybersecurity and e-Identity program.

References

1. Barber, D.: Bayesian reasoning and machine learning. Cambridge University Press (2012)
2. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
3. Bishop, C.M.: Pattern recognition and machine learning. springer (2006)
4. Bonaccorso, G.: Machine learning algorithms. Packt Publishing Ltd (2017)
5. DSecure.me: VMC: Vulnerability Management Center (2021 (accessed January 2, 2021)), https://github.com/DSecureMe/vmc
6. Elbaz, C., Rilling, L., Morin, C.: Fighting n-day vulnerabilities with automated cvss vector prediction at disclosure. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. pp. 1–10 (2020)
7. F-Secure: Vulnerability Management Tool (2021 (accessed January 2, 2021)), https://www.f-secure.com/us-en/business/ solutions/vulnerability-management/radar
8. Fall, D., Kadobayashi, Y.: The common vulnerability scoring system vs. rock star vulnerabilities: Why the discrepancy? In: ICISSP. pp. 405–411 (2019)
9. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from imbalanced data sets. Springer (2018)
10. FIRST: Common Vulnerability Scoring System v3.0: Specification Document (2017 (accessed January 2, 2021)), https://www.first.org/cvss/v3.0/specification-document

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21

11. FIRST: Common Vulnerability Scoring System v3.1: Specification Document (2019 (accessed January 2, 2021)), https://www.first.org/cvss/v3.1/specification-document
12. Hovsepyan, A., Scandariato, R., Joosen, W., Walden, J.: Software vulnerability prediction using text analysis techniques. In: Proceedings of the 4th international workshop on Security measurements and metrics. pp. 7–10 (2012)
13. Jacobs, J., Romanosky, S., Adjerid, I., Baker, W.: Improving vulnerability remediation through better exploit prediction. Journal of Cybersecurity **6**(1), tyaa015 (2020)
14. Klinedinst, D.J.: CVSS and the Internet of Things (2015 (accessed January 2, 2021)), https://insights.sei.cmu.edu/cert/2015/09/cvss-and-the-internet-of-things.html
15. Luers, A.L., Lobell, D.B., Sklar, L.S., Addams, C.L., Matson, P.A.: A method for quantifying vulnerability, applied to the agricultural system of the yaqui valley, mexico. Global Environmental Change **13**(4), 255–267 (2003)
16. Maciej, N., Walkowski, M., Sujecki, S.: CVSS 2.0 extended vector database (2021 (accessed January 21, 2021)), https://github.com/mwalkowski/cvss-2-extended-vector-database
17. Mell, P., Hu, V., Lippmann, R., Haines, J., Zissman, M.: An overview of issues in testing intrusion detection systems (2003)
18. Mell, P., Scarfone, K., Romanosky, S.: Common vulnerability scoring system. IEEE Security & Privacy **4**(6), 85–89 (2006)
19. Nowak, M.R., Zdunek, R., Plinski, E., Swiatek, P., Strzelecka, M., Malinka, W., Plinska, S.: Recognition of pharmacological bi-heterocyclic compounds by using terahertz time domain spectroscopy and chemometrics. Sensors **19**(15), 3349 (2019)
20. NVD: National Vulnerability Database (2021 (accessed January 2, 2021)), https://nvd.nist.gov/
21. Qualys: Vulnerability Management Tool (2021 (accessed January 2, 2021)), https://www.qualys.com /apps/vulnerability-management/
22. Rapid7: Vulnerability Management Tool (2021 (accessed January 2, 2021)), https://www.rapid7.com/products/nexpose/
23. Ruohonen, J.: A look at the time delays in cvss vulnerability scoring. Applied Computing and Informatics **15**(2), 129–135 (2019)
24. Tavabi, N., Goyal, P., Almukaynizi, M., Shakarian, P., Lerman, K.: Darkembed: Exploit prediction with neural language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
25. Walkowski, M., Krakowiak, M., Oko, J., Sujecki, S.: Distributed analysis tool for vulnerability prioritization in corporate networks. In: 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). pp. 1–6. IEEE (2020)
26. Walkowski, M., Krakowiak, M., Oko, J., Sujecki, S.: Efficient algorithm for providing live vulnerability assessment in corporate network environment. Applied Sciences **10**(21), 7926 (2020)
27. Younis, A.A., Malaiya, Y.K.: Using software structure to predict vulnerability exploitation potential. In: 2014 IEEE Eighth International Conference on Software Security and Reliability-Companion. pp. 13–18. IEEE (2014)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_21