IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 13, 2022, accepted May 27, 2022, date of publication June 2, 2022, date of current version June 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3179692

Predicting CVSS Metric via Description Interpretation

JOANA CABRAL COSTA, TIAGO ROXO, (Member, IEEE), JOÃO B. F. SEQUEIROS, HUGO PROENÇA, (Senior Member, IEEE), AND PEDRO R. M. INÁCIO, (Senior Member, IEEE)

Department of Computer Science, Instituto de Telecomunicações, University of Beira Interior, 6201-001 Covilhã, Portugal

Corresponding author: Joana Cabral Costa (joana.cabral.costa@ubi.pt)

This work was supported in part by the Fundação para a Ciência e a Tecnologia (FCT)/Programa Operacional Temático Competitividade e Internacionalização (COMPETE)/Fundo Europeu de Desenvolvimento Regional (FEDER) under the scope of Project SECURIoTESIGN under Grant POCI-01-0145-FEDER-030657; in part by the Portuguese FCT/Ministério da Ciência, Tecnologia e Ensino Superior (MCTES) through National Funds and, when applicable, co-funded by EU funds under Project UIDB/50008/2020; in part by the FCT Doctoral Grant SFRH/BD/133838/2017, Grant 2020.09847.BD, and Grant 2021.04905.BD; in part by the C4—Competence Center in Cloud Computing co-financed by the European Regional Development Fund (ERDF) through the Programa Operacional Regional do Centro (Centro 2020), in the scope of the Sistema de Apoio à Investigação Científica e Tecnológica, Programas Integrados de Investigação Científica e esenvolvimento Tecnológico (IC&DT) under Project CENTRO-01-0145-FEDER-000019.

ABSTRACT Cybercrime affects companies worldwide, costing millions of dollars annually. The constant increase of threats and vulnerabilities raises the need to handle vulnerabilities in a prioritized manner. This prioritization can be achieved through Common Vulnerability Scoring System (CVSS), typically used to assign a score to a vulnerability. However, there is a temporal mismatch between the vulnerability finding and score assignment, which motivates the development of approaches to aid in this aspect. We explore the use of Natural Language Processing (NLP) models in CVSS score prediction given vulnerability descriptions. We start by creating a vulnerability dataset from the National Vulnerability Database (NVD). Then, we combine text pre-processing and vocabulary addition to improve the model accuracy and interpret its prediction reasoning by assessing word importance, via Shapley values. Experiments show that the combination of *Lemmatization* and *5,000-word addition* is optimal for DistilBERT, the outperforming model in our experiments of the NLP methods, achieving state-of-the-art results. Furthermore, specific events (such as an attack on a known software) tend to influence model prediction, which may hinder CVSS prediction. Combining *Lemmatization* with vocabulary addition mitigates this effect, contributing to increased accuracy. Finally, binary classes benefit the most from pre-processing techniques, particularly when one class is much more prominent than the other. Our work demonstrates that DistilBERT is a state-of-the-art model for CVSS prediction, demonstrating the applicability of deep learning approaches to aid in vulnerability handling. The code and data are available at https://github.com/Joana-Cabral/CVSS_Prediction.

INDEX TERMS Common vulnerability scoring system, deep learning, interpretability, natural language processing, security.

I. INTRODUCTION

Cyber threats force companies to increase their investments in security, which resulted in a $170 billion security aspects related market in 2015 [1]. These threats impact 556 million people annually, costing $3 trillion worldwide, with an expected increase to $10.5 trillion by 2025 [2]. Additionally, there was an increase of vulnerability entries in VulDB, with 61 new daily entries, in 2021, relative to the 41 reported in 2016 [3]. This tendency provides a clear picture of the increased risk of threats and cybercrime, raising concern among Information Technology (IT) administrators, which often lack the resources to handle all incoming threats [4]. Given this context, there is an inherent need to define which vulnerabilities should be tackled first.

To aid in the prioritization of vulnerability handling, experts typically use the Common Vulnerability Scoring System (CVSS) [5], a *de facto* standard, to accurately assign a score to a vulnerability. New vulnerability entries are enumerated via Common Vulnerability Enumeration

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

(CVE) [6], with a unique identifier, description, and CVSS Base score metrics, the latter specified by the National Vulnerability Database (NVD).

The score metric assignment is performed manually from vulnerability description analysis, for which vendors do not always provide enough detail [7] for experts to accurately create these scores. Furthermore, some CVSS metrics are subjective [8], heavily relying on the previous experience at assigning CVSS metrics. The inherent problems of this process are exacerbated by the temporal mismatch of CVSS metric assignment and vulnerability finding: 19 days to populate a vulnerability with the respective CVSS and six days to find a new one [9]. Therefore, to reduce the time/cost spent while also mitigating the subjective aspect of score assignment, we explore the use of a deep learning approach to predict the CVSS metrics based on the vulnerability description.

We start by obtaining the vulnerabilities descriptions and respective CVSS metrics using the NVD Application Programming Interface (API). The collected data is processed for the most recent version of CVSS (version 3) and serves as input for the deep learning approach. We select the DistilBERT for sequence classification given its out-performance, in the created dataset, over other state-of-the-art Natural Language Processing (NLP) models. Since the vulnerability descriptions contain technical expressions and have reduced length size, we assess the effect of text pre-processing techniques and vocabulary addition. Our results show that text pre-processing improves the baseline model accuracy, exhibiting incremental performance with vocabulary addition.

One drawback of using a deep learning approach is that the reasoning behind their outputs is not easily disclosed. To overcome this limitation, we use the Shapley value [10], a game-theoretic approach to explain machine learning outputs, to perceive the correlation between description words and the predicted CVSS metric. This process allows us to understand the importance of each word towards the CVSS metric prediction, assessing their importance variance with text pre-processing and vocabulary addition.

The main contributions of our work are summarized as follows:

- We present a vulnerability dataset, derived from NVD data, with vulnerability descriptions and CVSS (version 3) metrics;
- We demonstrate the applicability of deep learning approaches to predict CVSS metrics, in combination with text pre-processing and vocabulary addition, achieving state-of-the-art results;
- We confer interpretability to model prediction by analyzing the importance of word descriptions, via Shapley value.

The remainder of this paper is organized as follows: Section II summarizes the most relevant CVSS-based works; Section III describes the methodologies used; Section IV describes the vulnerability dataset; and Section V discusses the results obtained. Finally, the main conclusions and future work are presented in Section VI.

II. RELATED WORK

A. CVSS APPLICABILITY

CVSS has been extensively analyzed and applied to multiple domains to prioritize or estimate security risks. Younis and Malaiya [11] compared the CVSS base metrics and the Microsoft rating system, declaring that both measures have a very high false-positive rate, with CVSS significantly affected by the software type. Joh [12] concluded that most vulnerabilities are compromised due to no authentication required systems, by analyzing the CVSS base scores for vulnerabilities of currently supported Windows operating systems, suggesting the addition of an authentication process in every system. CVSS base metrics have been used to assess cybersecurity risks in IT systems [4], using the risk formula, and calculating risk probability and impact. The same study reported that an identification of security properties in the early stages of development positively impacts the security of the systems. In the same context, Wirtz and Heisel [13] proposed a semi-automatic method to estimate security risks in the early stages of software development, using CVSS formulas to assess the threat severity. Since CVSS has already demonstrated its validity in typical IT systems, it was also adapted to calculate vulnerabilities regarding hybrid IT and IoT systems [14], [15] accurately. Following this idea, Mishra and Singh [16] proposed a taxonomy for Cloud-specific vulnerabilities, using the CVSS score to represent each major Cloud vulnerability severity. Finally, a guide for applying CVSS to medical devices was also proposed [17], consisting of questions that identify a value for a specific CVSS metric.

B. CVSS AND ARTIFICIAL INTELLIGENCE

The combination of Artificial Intelligence techniques and CVSS scores of individual vulnerabilities has also been reported. Sheehan *et al.* [18] proposed using Bayesian Networks to identify connected and autonomous vehicle cyber risks, using CVSS scores to predict knowledge gaps or potential new cyber vulnerabilities. Furthermore, Frigault *et al.* [19] employed Bayesian Networks and Attack Graphs to measure network security, using the CVSS scores as probabilities and considering metric values of each vulnerability to be independent. However, applying Bayesian Networks to assess CVSS scores has limitations [20], leading to the proposal of an approach that considers the dependency relationships between the CVSS base metrics, combining scores into three aspects: probability, effort, and skill. Allouzi and Khan [21] proposed using the Markov Chain to compute the probability distribution of Internet of Medical Things security threats, using CVSS scores to assign severity to the acknowledged vulnerabilities. One first attempt to predict CVSS final scores was made through the employment of fuzzy systems [22], outperforming Support Vector Machine (SVM) and Random-Forest. In this context,

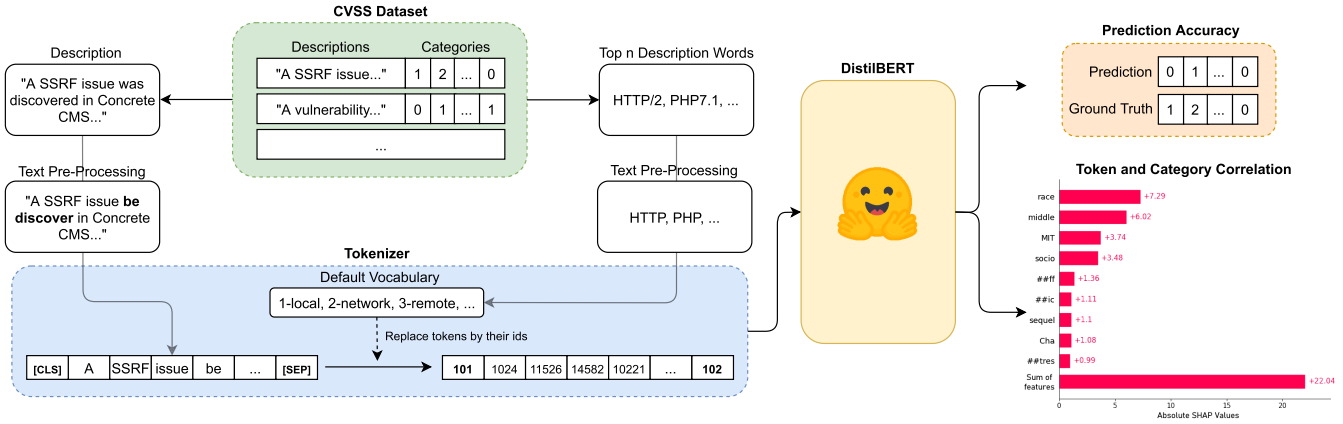


FIGURE 1. **Overview of the methodology used to assess DistilBERT performance in vulnerability detection, using CVSS data descriptions and categories. We evaluate the model performance by varying two key aspects: 1) text pre-processing approaches; and 2) vocabulary addition. Furthermore, we evaluate the correlation of tokens and category, via Shapley value, to assess the tokens more influential towards each category prediction.**

fuzzy CVSS [23] was used to calculate the final severity score for vulnerabilities, employing fuzzy theory to reduce the error rate. To predict CVSS values for base metrics, Elbaz *et al.* [24] propose a linear regression model, using a bag of words approach, with the removal of irrelevant words.

C. CVSS AND DEEP LEARNING

Deep learning is also known for its effectiveness in solving complex problems, with the drawback of time-costly training. Therefore, to resemble security experts decision-making [25], the usage of Neural Networks was proposed, automatically providing a vulnerability report through CVSS metrics. Deep reinforcement learning was also used to assess the cyber-physical security of electric power systems [26], which adapted CVSS to estimate the complexity of attack path. As a result, CVSS base metrics have been adopted as the guide for identifying and prioritizing threats among multiple systems. This indicates that correctly and swiftly predicting the metrics for CVSS is a valuable effort.

Sahin and Tosun [27] concluded that Long Short Term Memory (LSTM) was the most accurate model to predict CVSS final scores, when compared with Convolutional Neural Networks (CNN) and XGBoost. The two previously presented approaches gathered data from Open Source Vulnerability Database (OSVDB) and NVD, respectively, to train their models. Alternatively, Twitter discussions [28], with NVD as ground truth for CVSS scores, were fed to a Graph Convolutional Network with Attention-based input Embedding to predict the CVE severity scores. However, predicting CVSS final scores does not provide any insight to the experts about the values for the CVSS metrics.

D. VULNERABILITY INTERPRETABILITY

The analysis and interpretation of vulnerability descriptions is also reported in the literature. An empirical study based on the NVD vulnerability descriptions [29] concluded that information about the *asset*, *attack*, and *vulnerability* type is relevant to increase vulnerability scoring accuracy. Another work used the Local Interpretable Model-Agnostic (LIME) framework to explain the vulnerability descriptions [30], providing relevant words for a small number of vulnerabilities.

To the best of our knowledge, the work presented herein is the first to combine Deep Learning and NLP approaches to extract information from vulnerability descriptions and output CVSS metrics, while using interpretability to assess model predictions.

III. METHODOLOGY

The methodology used in our experiments is displayed in Fig. 1. We start by creating a CVSS dataset, using information from the NVD. Then, we vary two major performance-related aspects: 1) text pre-processing; and 2) vocabulary addition. Finally, we evaluate model accuracy and assess token correlation with category prediction, using Shapley value.

A. MODEL DETAILS AND EVALUATION METRICS

We used the following models in our experiments: BERT [31], DistilBERT [32], RoBERTa [33], ALBERT [34], and DeBERTa [35]. Our reasoning for model choice is linked to the importance of BERT for the NLP area. It is one of the most used models in NLP, in a variety of tasks, with proven quality. Then, we opted to choose other variations of BERT to assess what is the better model for CVSS metric prediction. Specifically, we choose ALBERT and DistilBERT for having fewer parameters than BERT and RoBERTa and DeBERTa for having more parameters than BERT. The chosen models belong to the BERT family while having specific characteristics, such as the number of parameters. As such, our work focused on finding the best performing state-of-the-art NLP models for CVSS metrics prediction.

We finetune each model following the authors' methodology: regarding the *learning rate*, RoBERTa was set to 1.5×10^{-5}, DistilBERT was set to 5×10^{-5}, while BERT,

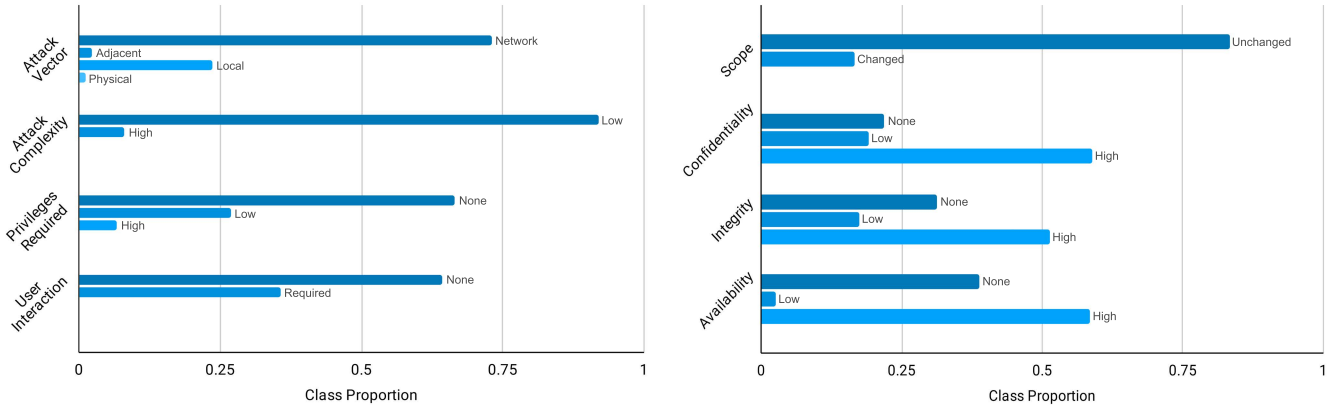


FIGURE 2. Bar plots of the eight categories analyzed of CVSS version 3, linked to vulnerability assessment, of the vulnerability dataset used. Each category displays the associated classes and respective class prior.

ALBERT, and DeBERTa have all been set to 3×10^{-5}; for the *number of training epochs*, RoBERTa was trained for 2 epochs, DeBERTa for 10, and BERT, ALBERT, and DistilBERT for 3; regarding *batch size*, we used 8 for ALBERT and DistilBERT, and 4 for BERT, RoBERTa, and DeBERTa; finally, RoBERTa has a *weight decay* of 0.01, while the remaining models have the default value (0). We use the default losses and architectures of each model, from Hugging Face [36]. To obtain category classification, we use a *PyTorch Softmax* layer [37] on the model output.

To compare the performance of each model, we use the accuracy, F1 score, and balanced accuracy from the *scikit-learn* library [38]. To compare our results with state-of-the-art for CVSS metric inference, we use the accuracy metric.

B. TEXT PROCESSING AND VOCABULARY SELECTION

To assess the contribution of each word to the classification of the considered categories (discussed in section IV), we start by processing vulnerability descriptions. We use two pre-processing methods, namely, *Lemmatization* and *Stemming*. Finally, we tokenize the text to input to the model, evaluating its accuracy based on the pre-processing approach. Both text pre-processing approaches use Natural Language Toolkit (NLTK) methods [39], while tokenization is achieved using Transformers library, from Hugging Face [36]. We choose *Lemmatization* and *Stemming*, given their wide use as text pre-processing approaches in the NLP area. By using *Lemmatization* and *Stemming*, we intend to process text to maintain as much relevant data as possible while ignoring noisy data. This is achieved by ignoring variants of words that have the same "base". In the case of *Stemming* is the same stem, while in *Lemmatization* is the same lemma.

In our experiments, we also evaluate the effect of vocabulary addition. Moreover, we also assess this effect in conjunction with the best performing text pre-processing approach. We evaluate the accuracy of the used model when adding 5,000, 10,000, and 25,000 words to the default vocabulary of the tokenizer. To select the added words, we order them by frequency of appearance in the descriptions, choosing the top n words. To avoid redundancy, we only consider words that appear exclusively in the description and not in the default vocabulary.

Given the existence of software versions and code snippets in some data descriptions, we use regular expressions to filter digits and special characters. This approach reduces the "noise" of vocabulary addition, since this filtered data is not relevant to category classification and could potentially dissipate the importance of relevant added words.

C. SHAPLEY VALUE

Deep learning models have shown high performance in multiple tasks while providing little to no explanation for the reasoning for model prediction. To tackle this issue, we use Shapley value, an interpretability technique that allows us to interpret the reasoning of the model when providing predictions. The Shapley value, coined by Shapley in 1953 [40], is a cooperative game theory-based method used for assigning payouts to players, depending on their contribution towards the total payout. In the machine-learning context, the Shapley value is used to evaluate how each feature (player) of a given instance contributed (assigning payout) towards the model prediction of the instance (total payout).

The use of Shapley value in our experiments is linked to our interest in analyzing how each word contributed to category classification. For categories with more than n classes, and n higher than 2, we perform n Shapley value analysis, each considering a class versus the remaining classes of the category. The considered class is given the value 1, with the remaining receiving the value 0. If a word contributes positively, it means that it influences the considered class. The higher the absolute Shapley value is, the higher the feature influence. We use the SHapley Additive exPlanations (SHAP) framework [41] and the Explainer model, from a publicly available implementation in [42].

TABLE 1. Percentage of class prior of the eight vulnerability related categories for all dataset, train, and test set. Class prior displays the likelihood of an outcome in the dataset and each subset.

Category	Class Prior (%)		
	Train	Test	Dataset
Attack Vector			
Network	73.04	73.57	73.14
Adjacent	2.21	2.14	2.20
Local	23.62	23.18	23.53
Physical	1.13	1.11	1.13
Attack Complexity			
High	8.03	7.74	7.97
Low	91.97	92.26	92.03
Privileges Required			
High	6.70	6.32	6.62
Low	26.79	26.90	26.81
None	66.51	66.78	66.57
User Interaction			
Required	35.80	35.37	35.71
None	64.20	64.63	64.29
Scope			
Changed	16.72	16.36	16.65
Unchanged	83.28	83.64	83.35
Confidentiality			
High	58.94	59.15	58.98
Low	19.37	18.65	19.23
None	21.69	22.20	21.79
Integrity			
High	51.27	51.36	51.29
Low	17.52	17.32	17.48
None	31.21	31.32	31.23
Availability			
High	58.49	58.85	58.56
Low	2.60	2.71	2.63
None	38.91	38.44	38.81

IV. VULNERABILITY DATASET

The vulnerability dataset is based on NVD information, a United States government repository of standards-based vulnerability management data. We obtain the information through their API, starting from index 0 to 152,000, representing data collected until April 2021. Finally, we process the collected data to retrieve vulnerability descriptions and the classes for each of the eight categories analyzed: *Attack Vector*, *Attack Complexity*, *Privileges Required*, *User Interaction*, *Scope*, *Confidentiality*, *Integrity*, and *Availability*. Based on the CVSS documentation, these classes are grouped into Exploitability metrics (Attack Vector, Attack Complexity, Privileges Required, and User Interaction), Scope, and Impact metrics (Confidentiality, Integrity, and Availability). Tables and Figures throughout this paper consider this grouping. A visual representation of class proportions, for each category, of our dataset is displayed in Fig. 2.

Though the collected data corresponds to 152,000 vulnerability descriptions and categories, we only consider descriptions related to version 3 of CVSS in this work. For this reason, the total number of instances in our dataset is 79,810. We divide them into train and test sets, composed of 63,848 and 15,962 instances, respectively, corresponding to a 0.2 test ratio. The average description length is 43.85 and 44.55 words for train and test split, respectively. Each set follows a similar proportion of classes, exhibited in Fig 2, whose analytical values are shown in Table 1. The dataset is publicly available for repeatability purposes, and it serves as a basis for other models to evaluate their performance and compare with the proposed methodology.

V. EXPERIMENTS

A. MODEL COMPARISON

We start by comparing the performance of five different NLP methods in the proposed dataset. The accuracy, F1 score, and balanced accuracy for each of the eight categories are presented in Table 2. The results suggest that DistilBERT is the outperforming model for all the categories, in all the considered metrics. The method with the worst performance is ALBERT, which has the least number of parameters (11M), while DeBERTa, BERT, and RoBERTa, with over 100M parameters, also have worse performance than DistilBERT (65M). Since we intend to assess the class inference, given a vulnerability description, the number of parameters may be linked to the performance variance. In this case, too few parameters (ALBERT) are insufficient for the model to learn, and too many leads to poorer fine-tuning. The similarity of various accuracy values between BERT, ALBERT, and DeBERTa, for different categories, can be explained by dataset imbalance. In these cases, the values displayed represent a scenario where the models opted to achieve higher accuracy by outputting the same value in every instance. Thus, in cases of dataset imbalance, the use of accuracy can be deceptive, justifying the use of other metrics such as balanced accuracy.

In this experiment, we use the default pretraining weights (provided by HuggingFace [36]) and training parameters of every model. The models used are typically applied/evaluated in tasks where the association of two sentences is analyzed (*e.g.*, GLUE [43]) or the aim is finding answers in a text, given a question (*e.g.*, SQuAD [44]). These types of tasks differ from predicting a category given a vulnerability description (the aim of this work), which may justify the underperformance of state-of-the-art methods in our experiments. Based on the obtained results, we selected DistilBERT for continuing the experiments involving the usage of Deep Learning.

B. TEXT PRE-PROCESSING

We assess the performance of DistilBERT, for all eight considered categories, regarding different text pre-processing approaches. We present our results, using balanced accuracy, in Table 3, with *Baseline* referring to the condition where no pre-processing approach is used.

When comparing category-related performance variance, we observe that all categories benefit from pre-processing. Regarding processing-related performance variance, *Lemmatization* promotes better results than *Stemming*, for all categories. *Stemming* truncates words by chopping off letters from the end until the stem is reached. This is a more

TABLE 2. Model accuracy (Acc), F1 score, and Balanced Accuracy (BA) for each of the eight categories analyzed. The outperforming model for each metric and category is shown in bold.

Category	BERT			RoBERTa			ALBERT			DeBERTa			DistilBERT		
	Acc	F1	BA	Acc	F1	BA	Acc	F1	BA	Acc	F1	BA	Acc	F1	BA
Attack Vector	73.57	62.37	25.00	90.55	90.18	62.03	90.87	90.60	67.76	80.86	78.99	37.17	**91.04**	**90.85**	**70.60**
Attack Complexity	92.26	88.55	50.00	92.26	88.55	50.00	92.26	88.55	50.00	92.26	88.55	50.00	**95.16**	**94.32**	**70.74**
Privileges Required	66.78	53.48	33.33	66.86	83.49	67.61	66.78	53.48	33.33	66.86	63.65	41.15	**86.21**	**85.91**	**75.09**
User Interaction	64.63	50.75	50.00	82.73	92.57	91.26	64.63	50.75	50.00	82.73	82.85	81.90	**93.01**	**92.98**	**91.95**
Scope	83.64	76.19	50.00	96.02	95.91	90.20	83.64	76.19	50.00	83.64	76.19	50.00	**96.28**	**96.19**	**90.98**
Confidentiality	59.15	43.97	33.33	86.20	85.90	81.08	86.18	85.98	81.74	86.20	86.05	82.37	**86.29**	**86.16**	**82.67**
Integrity	61.52	50.10	55.84	86.93	86.85	84.41	51.36	34.85	33.33	51.37	34.87	33.34	**87.46**	**87.42**	**85.52**
Availability	58.85	43.61	33.33	83.50	87.89	67.49	58.85	43.61	33.33	83.50	82.47	57.54	**88.55**	**88.00**	**67.51**

TABLE 3. Category balanced accuracy of DistilBERT for baseline conditions (*Tokenization*), and using text pre-processing approaches (*Lemmatization* and *Stemming*). The outperforming approach for each category is shown in bold.

Category	Balanced Accuracy (%)		
	Baseline	Lemmatization	Stemming
Attack Vector	70.60	**71.03**	68.40
Attack Complexity	70.74	**71.45**	64.50
Privileges Required	75.09	**75.14**	73.83
User Interaction	91.95	**92.05**	91.75
Scope	90.98	**91.12**	90.40
Confidentiality	82.67	**83.12**	82.44
Integrity	85.52	**85.77**	84.98
Availability	67.51	**69.66**	68.69

crude approach than *Lemmatization*, which justifies the underperformance using this approach. Given the superiority displayed by *Lemmatization* over *Stemming*, this is the chosen pre-processing approach to use in the remaining experiments.

C. VOCABULARY ADDITION

We also evaluate the effect of vocabulary addition on prediction accuracy. Furthermore, we compare the vocabulary addition with its combination with a pre-processing approach. We display our results in Table 4.

Relative to the baseline, most variations of vocabulary addition translate into performance increase, for all categories. Regarding the vocabulary variations, 5,000-word addition was the condition with better results overall. This suggests that adding more words is beneficial to model accuracy improvement. However, subsequent vocabulary addition (10,000 and 25,000-word addition) does not promote incremental performance increase. Given that vocabulary addition is linked to word frequency in the description, adding more words may disperse the model attention towards less relevant words, hindering its performance. This aspect is more noticeable when 25,000-word addition has worse performance than baseline (*e.g.*, Attack Vector, Attack Complexity). For all categories, 25,000-word addition does not generally translate into performance improvement relative to 5,000-word addition, suggesting the existence of word importance redundancy with this approach.

Regarding the combination of vocabulary addition with *Lemmatization*, we observe that this approach generally improves the balanced accuracy, relative to vocabulary addition alone, for most vocabulary variations. This suggests that word importance may vary with processing approaches, which corroborates the importance of text pre-processing, even in the context of vocabulary addition. The results suggest that *5,000-word addition* with *Lemmatization* is the best approach for overall category prediction, exhibiting the importance of text processing and pertinent word addition in description-based classification.

D. STATE-OF-THE-ART COMPARISON

We compare DistilBERT, and its combination with pre-processing and vocabulary addition, with the state-of-the-art. To the best of our knowledge, only Ebalz *et al.* [24] evaluates class prediction accuracy in version 3 of CVSS. To compare our results with them, we also display the accuracy of *Baseline* and *5,000-word addition with Lemmatization*, whose balanced accuracy is presented in Table 4. Since the authors presented their results in a bar plot, not displaying the analytical values, we register the rounded values observed in said plot. We display the state-of-the-art comparison in Table 5.

Ebalz *et al.* use a bag of words approach, with the removal of irrelevant words, to input a regression model. Using DistilBERT, a deep learning approach, in conjunction with text pre-processing and vocabulary addition, we obtain substantial accuracy improvements in the majority of categories. The categories where Ebalz's approach was closer to ours were *Attack Complexity*, *User Interaction*, and *Scope*, which could be linked to these categories being two-classed. In these cases, the regression model used by Ebalz *et al.* can compete with deep learning approaches. However, for the remaining categories, with over two classes, the performance disparity is substantially larger, with up to a 28% accuracy increase. Furthermore, using the text pre-processing approach and adding vocabulary promotes an accuracy increase of DistilBERT, further enhancing its performance. The results suggest that DistilBERT is a state-of-the-art approach for vulnerability category prediction, particularly for multi-class categories.

TABLE 4. Category balanced accuracy of DistilBERT for baseline conditions (*Tokenization*), and with different vocabulary addition, assessing the effect of *Lemmatization*. *Base*, in each vocabulary column, refers to the vocabulary addition with *Tokenization*, without text pre-processing. The expression *w/ Lemm* refers to *Lemmatization* combination with vocabulary addition. The outperforming approach for each category is shown in bold.

Category	Balanced Accuracy (%)						
	Baseline	5,000 Words		10,000 Words		25,000 Words	
		Base	w/ Lemm	Base	w/ Lemm	Base	w/ Lemm
Attack Vector	70.60	70.79	**71.26**	71.05	71.19	68.66	69.83
Attack Complexity	70.74	71.42	**72.92**	68.73	68.78	65.59	68.06
Privileges Required	75.09	75.44	**75.81**	75.14	74.95	74.76	74.87
User Interaction	91.95	92.11	92.40	**92.48**	92.09	91.81	91.96
Scope	90.98	**91.60**	91.49	90.99	91.01	90.70	91.03
Confidentiality	82.67	82.89	**83.18**	82.82	82.66	82.73	82.38
Integrity	85.52	85.81	**85.97**	85.53	85.56	85.13	85.36
Availability	67.51	67.73	**69.35**	69.18	68.56	67.80	68.32

TABLE 5. Category accuracy of DistilBERT, DistilBERT-Enhanced (DistilBERT-E) and Ebalz's work [24]. DistilBERT-Enhanced refers to DistilBERT using *Lemmatization* and *5,000-word addition*. The outperforming approach for each category is shown in bold.

Category	Accuracy (%)		
	DistilBERT	DistilBERT-E	Ebalz [24]
Attack Vector	91.04	**91.41**	78.00
Attack Complexity	95.16	**95.20**	95.00
Privileges Required	86.21	**86.42**	79.00
User Interaction	93.01	**93.33**	89.00
Scope	96.28	**96.40**	96.00
Confidentiality	86.29	**86.71**	69.00
Integrity	87.46	**87.61**	63.00
Availability	88.55	**88.81**	60.00

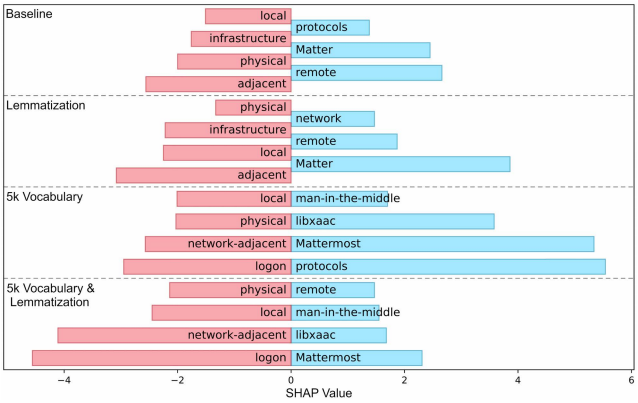
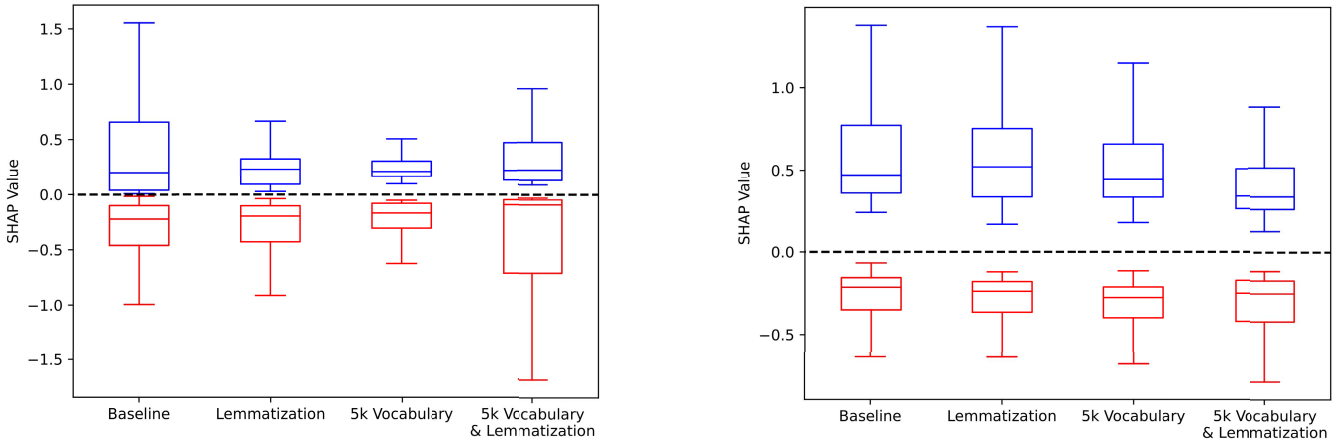


FIGURE 3. Bar plots of the SHAP value for Baseline, *Lemmatization*, 5,000-word addition (*5k Vocabulary*), and 5,000-word addition with *Lemmatization* (*5k Vocabulary & Lemmatization*) for the category *Attack Vector*, regarding *Network* class.

E. INTERPRETING CATEGORY CLASSIFICATION

We assess word importance in two distinct scenarios: 1) comparing the most relevant words, using different processing techniques, for a given category; and 2) assessing the variance of word importance towards/against binary and multi-class category prediction, given different processing techniques. In the first scenario, we compare word importance variance with text pre-processing and vocabulary addition in DistilBERT. Given the overall superiority of Lemmatization and 5,000-word addition (Table 4), these are the chosen approaches. We consider the four stages for comparison: 1) Baseline; 2) Lemmatization; 3) 5,000-word addition; and 4) 5,000-word addition with Lemmatization. For the remaining experiments, we will refer to each *word* of a description as a *token* to accurately represent the word translated into the tokenizer vocabulary. We evaluate token importance for the category *Attack Vector*, regarding the *Network* class. In this case, *Network* has a value of 1, and the remaining three classes have the value 0. Tokens with positive Shapley value influence *Network* classification, while negative ones are more relevant to the other three classes.

The results show a variance in token importance with text pre-processing and vocabulary addition. Starting in the **Baseline**, with no processing or vocabulary addition, *protocols*, *Matter*, and *remote* are tokens that, when in a description, influence the classification of the category towards Network. There is some logic behind said importance, given that *remote* and *protocols* are linked to network-related activities. The influence of *Matter* is linked to Mattermost, an open-source chat service, which was the target of multiple attacks. This shows that token importance might be influenced by specific network-related events. When we analyze tokens more associated with other classes (negative Shapley value), we observe that these are closely related to class definition (Local, Physical, and Adjacent) or associated with it (*infrastructure*). Adding **Lemmatization**, we observe the same tendency for tokens influential towards other classes but with increased importance. Furthermore, tokens linked to Network classification lose importance, aside from the specific network-related event of baseline (*Matter*). This suggests that token descriptions are more interpretably linked in not classifying Network than towards it, which could be due to class imbalance. Network is over 70% of Attack Vector classes, making it harder to distinguish tokens clearly associated with it, thus justifying the Lemmatization results. The addition of vocabulary (**5k Vocabulary**) heavily influences category classification, with new tokens being associated with Network classification: *libxaac*, *Mattermost*, and *man-in-the-middle*. *Libxaac* is an Android library with



(a) Binary Class Data Distribution

(b) Multi-Class Data Distribution

FIGURE 4. Boxplots summarizing the effect of applying text pre-processing techniques (Lemmatization) and vocabulary addition (5,000-word addition) for (a) binary class (*Attack Complexity*, *User Interaction*, and *Scope*) and (b) multiple class problems.

reported out-of-bound reading/writing errors, while *man-in-the-middle* is a type of network attack. The importance of these tokens is linked to specific network-related events (attacks, errors), which was also observed in the baseline. *Protocols* also increases in importance towards Network classification, which could be linked to their association with the added vocabulary. This shows that vocabulary addition shifts the focus of token importance heavily towards specific events, for Network classification. *Network-adjacent* (added by vocabulary addition) also gains importance in classifying other classes, given its relevance to dissociate Network from Adjacent. Complementing vocabulary addition with Lemmatization (**5k Vocabulary & Lemmatization**) diminishes the importance of tokens closely linked to Network (positive Shapley value), resurging the tendency observed with Lemmatization alone. The reduced importance of specific network-related events also greatly decreased token importance associated with it (*protocols*). Furthermore, the influence of added vocabulary was enhanced in *logon* (closely related to classes other than Network) and *network-adjacent*, while keeping high importance of tokens associated with other classes definition (*physical* and *local*). This result suggests that Lemmatization is necessary to obtain more coherent/explainable token importance, which ultimately translates into better model performance (as shown in Table 4).

The second considered scenario relates token importance when considering binary (*Attack Complexity*, *User Interaction*, and *Scope*) and multi-class categories, for the same processing approaches of the first scenario. For all categories, the highest proportion class per category was associated with the value 1, with the remaining being associated with 0. Fig. 4 displays the boxplots for the two cases considered, showing the data distribution (ignoring wildcard cases).

The analysis of binary boxplots indicates that using Lemmatization and vocabulary addition promotes a decrease in token importance variance in both towards (positive Shapley value) and against (negative Shapley value) the highest class. However, combining vocabulary addition with Lemmatization increases token importance variance, particularly for negative values. This translates into increased importance of tokens to categorize the least represented class. If almost all descriptions relate to a specific class, it may be more beneficial/discriminative to focus on tokens linked to the underrepresented class, which is the approach of the model in this case.

Analyzing multi-class boxplots shows that the variance of negative Shapley value remains nearly constant throughout the various text pre-processing methods. Comparatively to the binary classes, negative Shapley value refers to various classes and not simply one, which justifies the (low) variance observed for these cases. Relative to positive Shapley value, using vocabulary addition and its combination with Lemmatization tends to reduce the variance of token importance, achieving a similar variance to negative Shapley value tokens. In multi-class prediction, even when one class is more prevalent than others, the existence of tokens closely linked to specific categories is not as likely as in binary class prediction. For this reason, reducing the overall importance towards specific token importance classification translates into better results.

VI. CONCLUSION

The increasing number of threats and vulnerabilities in IT systems surpass the capability of professionals to handle them, potentially leading to company prejudice. This raises the need to prioritize vulnerabilities, typically achieved through CVSS metrics, via manual vulnerability description analysis. In this paper, we present a vulnerability dataset, from NVD data, and analyze the applicability of deep learning approaches, namely NLP methods, to aid in CVSS metric prediction via description interpretation. In our experiments, we also assess the importance of text processing and vocabulary addition in metric prediction while interpreting

it via Shapley value. Our results show that DistilBERT is a state-of-the-art model for CVSS metric prediction, with increased performance when combined with Lemmatization (text pre-processing) and 5,000 word-addition. Furthermore, this combination mitigates the effect of specific events in category prediction and leads to weighted word importance, particularly for binary categories, contributing to increased model accuracy. The presented dataset and model experiments serve as a comparable basis for future works in CVSS metric prediction, applicable for vulnerability handling/prioritization, which leads to increased usefulness and accuracy of the metric, benefiting system security and operational effectiveness.

REFERENCES

[1] P. Boden. (2016). *The Emerging Era of Cyber Defense and Cybercrime*. Accessed: Jul. 29, 2021. [Online]. Available: https://www.microsoft.com/security/blog/2016/01/27/the-emerging-era-of-cyber-defense-and-cybercrime/

[2] S. Morgan. (2020). *Cybercrime to Cost the World $10.5 Trillion Annually by 2025*. Accessed: Jul. 29, 2021. [Online]. Available: https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/

[3] vuldb.Com. (2021). *Vulnerability Database*. Accessed: Jul. 28, 2021. [Online]. Available: https://vuldb.com/?

[4] M. U. Aksu, M. H. Dilek, E. I. Tatli, K. Bicakci, H. I. Dirik, M. U. Demirezen, and T. Aykir, "A quantitative CVSS-based cyber security risk assessment methodology for IT systems," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2017, pp. 1–8.

[5] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Secur. Privacy*, vol. 4, no. 6, pp. 85–89, Nov./Dec. 2006.

[6] D. E. Mann and S. M. Christey, "Towards a common enumeration of vulnerabilities," in *Proc. 2nd Workshop Res. Secur. Vulnerability Databases*, West Lafayette, IN, USA: Purdue Univ., 1999, pp. 1–13.

[7] N. I. of Standards and Technology. (2021). *NVD—Vulnerability Metrics*. Accessed: Jul. 28, 2021. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss

[8] P. Johnson, R. Lagerstrom, M. Ekstedt, and U. Franke, "Can the common vulnerability scoring system be trusted? A Bayesian analysis," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 1002–1015, Dec. 2018.

[9] A. Feutrill, D. Ranathunga, Y. Yarom, and M. Roughan, "The effect of common vulnerability scoring system metrics on vulnerability exploit delay," in *Proc. 6th Int. Symp. Comput. Netw. (CANDAR)*, Nov. 2018, pp. 1–10.

[10] L. S. Shapley, *A Value for N-Person Games*. Princeton, NJ, USA: Princeton Univ. Press, 2016, ch. 17.

[11] A. A. Younis and Y. K. Malaiya, "Comparing and evaluating CVSS base metrics and Microsoft rating system," in *Proc. IEEE Int. Conf. Softw. Qual., Rel. Secur.*, Aug. 2015, pp. 252–261.

[12] H. Joh, "Software risk assessment for windows operating systems with respect to CVSS," *Eur. J. Eng. Technol. Res.*, vol. 4, no. 11, pp. 41–45, Nov. 2019.

[13] R. Wirtz and M. Heisel, "CVSS-based estimation and prioritization for security risks," in *Proc. 14th Int. Conf. Eval. Novel Approaches Softw. Eng.*, 2019, pp. 297–306.

[14] A. Ur-Rehman, I. Gondal, J. Kamruzzuman, and A. Jolfaei, "Vulnerability modelling for hybrid IT systems," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 1186–1191.

[15] A. Ur-Rehman, I. Gondal, J. Kamruzzaman, and A. Jolfaei, "Vulnerability modelling for hybrid industrial control system networks," *J. Grid Comput.*, vol. 18, no. 4, pp. 863–878, Dec. 2020.

[16] N. Mishra and R. Singh, "Taxonomy & analysis of cloud computing vulnerabilities through attack vector, CVSS and complexity parameter," in *Proc. Int. Conf. Issues Challenges Intell. Comput. Techn. (ICICT)*, vol. 1, Sep. 2019, pp. 1–8.

[17] M. P. Chase and S. M. C. Coley, "Rubric for applying CVSS to medical devices," MITRE Corp., McLean, VA, USA, Tech. Rep. HHSM-500-2012-00008I, Oct. 2020.

[18] B. Sheehan, F. Murphy, M. Mullins, and C. Ryan, "Connected and autonomous vehicles: A cyber-risk classification framework," *Transp. Res. A, Policy Pract.*, vol. 124, pp. 523–536, Jun. 2019.

[19] M. Frigault, L. Wang, S. Jajodia, and A. Singhal, "Measuring the overall network security by combining CVSS scores based on attack graphs and Bayesian networks," in *Network Security Metrics*. Cham, Switzerland: Springer, 2017, pp. 1–23.

[20] P. Cheng, L. Wang, S. Jajodia, and A. Singhal, "Refining CVSS-based network security metrics by examining the base scores," in *Network Security Metrics*. Cham, Switzerland: Springer, 2017, pp. 25–52.

[21] M. Ali Allouzi and J. I. Khan, "Identifying and modeling security threats for IoMT edge network using Markov chain and common vulnerability scoring system (CVSS)," 2021, *arXiv:2104.11580*.

[22] A. Khazaei, M. Ghasemzadeh, and V. Derhami, "An automatic method for CVSS score prediction using vulnerabilities description," *J. Intell. Fuzzy Syst.*, vol. 30, no. 1, pp. 89–96, Aug. 2015.

[23] K. Gencer and F. Başçiftçi, "The fuzzy common vulnerability scoring system (F-CVSS) based on a least squares approach with fuzzy logistic regression," *Egyptian Informat. J.*, vol. 22, no. 2, pp. 145–153, Jul. 2021.

[24] C. Elbaz, L. Rilling, and C. Morin, "Fighting N-day vulnerabilities with automated CVSS vector prediction at disclosure," in *Proc. 15th Int. Conf. Availability, Rel. Secur.*, Aug. 2020, pp. 1–10.

[25] A. Beck and S. Rass, "Using neural networks to aid CVSS risk aggregation—An empirically validated approach," *J. Innov. Digit. Ecosyst.*, vol. 3, no. 2, pp. 148–154, 2016.

[26] X. Liu, J. Ospina, and C. Konstantinou, "Deep reinforcement learning for cybersecurity assessment of wind integrated power systems," *IEEE Access*, vol. 8, pp. 208378–208394, 2020.

[27] S. E. Sahin and A. Tosun, "A conceptual replication on predicting the severity of software vulnerabilities," in *Proc. Eval. Assessment Softw. Eng.*, Apr. 2019, pp. 244–250.

[28] H. Chen, J. Liu, R. Liu, N. Park, and V. S. Subrahmanian, "VASE: A Twitter-based vulnerability analysis and score engine," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 976–981.

[29] L. Allodi, S. Banescu, H. Femmer, and K. Beckers, "Identifying relevant information cues for vulnerability assessment using CVSS," in *Proc. 8th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2018, pp. 119–126.

[30] K. B. Alperin, A. B. Wollaber, and S. R. Gomez, "Improving interpretability for cyber vulnerability assessment using focus and context visualizations," in *Proc. IEEE Symp. Vis. Cyber Secur. (VizSec)*, Oct. 2020, pp. 30–39.

[31] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL-HLT)*, Minneapolis, MN, USA, vol. 1, Jun. 2019, pp. 4171–4186.

[32] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[33] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[34] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *8th Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020.

[35] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced bert with disentangled attention," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Vienna, Austria, May 2021.

[36] T. Wolf, L. Debut, V. Sanh, and J. Chaumond, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[37] A. Paszke, S. Gross, F. Massa, and A. Lerer, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, 2012.

[39] S. Bird, E. Klein, and E. Loper, *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*. Sebastopol, CA, USA: O'Reilly Media, 2009.

[40] L. S. Shapley, "A value for n-person games," *Contrib. Theory Games*, vol. 2, no. 28, pp. 307–317, 1953.
[41] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4765–4774.
[42] S. E. A. Lundberg. (2021). *Shap (Shapley Additive Explanations)*. Accessed: Jul. 2, 2021. [Online]. Available: https://github.com/slundberg/shap
[43] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. EMNLP Workshop BlackboxNLP: Analyzing Interpreting Neural Netw. (NLP)*, 2018, pp. 353–355.
[44] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2016, pp. 2383—2392.

JOANA CABRAL COSTA received the bachelor's and master's degrees in computer science and engineering from the Universidade da Beira Interior (UBI), in 2019 and 2021, respectively, where she is currently pursuing the Ph.D. degree with a Fundação para a Ciência e a Tecnologia (FCT) Scholarship, in the field of computer vision and adversarial attacks.

TIAGO ROXO (Member, IEEE) received the bachelor's degree in computer science and engineering from the Universidade da Beira Interior (UBI), in 2019, where he is currently pursuing the Ph.D. degree with a Fundação para a Ciência e a Tecnologia (FCT) Scholarship, in the field of computer vision and artificial intelligence.

JOÃO B. F. SEQUEIROS received the bachelor's and master's degrees in computer science and engineering from the Universidade da Beira Interior, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree under the title "Towards a Framework for System and Attack Modeling and Mapping of Security Requirements for the Internet of Things." His dissertation focused on the development of a box for automated network-based security assessments. He has authored or coauthored several journals and conference papers mainly in the security field. His main research interests include network and application security, cryptography, cybersecurity, and the IoT security.

HUGO PROENÇA (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees, in 2001, 2004, and 2007, respectively. He is currently an Associate Professor with the Department of Computer Science, University of Beira Interior, and has been researching mainly about biometrics and visual-surveillance. He is a member of the Editorial Boards of the *Image and Vision Computing*, IEEE Access, and *International Journal of Biometrics*. He served as a Guest Editor for special issues of the *Pattern Recognition Letters*, *Image and Vision Computing*, and *Signal, Image and Video Processing* journals. He was the Co-ordinating Editor of the IEEE Biometrics Council Newsletter and the Area Editor (Ocular Biometrics) of the IEEE Biometrics Compendium journal.

PEDRO R. M. INÁCIO (Senior Member, IEEE) was born in Covilhã, Portugal, in 1982. He received the B.Sc. degree in mathematics/computer science and the Ph.D. degree in computer science and engineering from the Universidade da Beira Interior (UBI), Portugal, in 2005 and 2009, respectively. The Ph.D. work was performed in the enterprise environment of Nokia Siemens Networks Portugal S.A., through a Ph.D. Grant from the Portuguese Foundation for Science and Technology.

He has been a Professor of computer science at the UBI, since 2010, where he lectures subjects related with information assurance and security, programming of mobile devices and computer based simulation, to graduate and undergraduate courses, namely to the B.Sc., M.Sc. and Ph.D. programs in computer science and engineering. He is currently the Head of the Department of Computer Science, UBI. He is an Instructor of the UBI Cisco Academy. He is a Researcher of the Instituto de Telecomunicações (IT). He has about 40 publications in the form of book chapters and papers in international peer-reviewed books, conferences, and journals. He frequently reviews articles for IEEE, Springer, Wiley, and Elsevier journals. His main research interests include information assurance and security, computer-based simulation, and network traffic monitoring, analysis, and classification.

Dr. Inácio has been a member of the Technical Program Committee of International Conferences, such as the ACM Symposium on Applied Computing—Track on Networking. He was one of the chairs of WISARC 2016.