CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain

MARKUS BAYER, PEASEC, Technical University of Darmstadt, Darmstadt, Germany
PHILIPP KUEHN, PEASEC, Technical University of Darmstadt, Darmstadt, Germany
RAMIN SHANEHSAZ, PEASEC, Technical University of Darmstadt, Darmstadt, Germany
CHRISTIAN REUTER, PEASEC, Technical University of Darmstadt, Darmstadt, Germany

The field of cybersecurity is evolving fast. Security professionals are in need of intelligence on past, current and —ideally — upcoming threats, because attacks are becoming more advanced and are increasingly targeting larger and more complex systems. Since the processing and analysis of such large amounts of information cannot be addressed manually, cybersecurity experts rely on machine learning techniques. In the textual domain, pre-trained language models such as Bidirectional Encoder Representations from Transformers (BERT) have proven to be helpful as they provide a good baseline for further fine-tuning. However, due to the domain-knowledge and the many technical terms in cybersecurity, general language models might miss the gist of textual information. For this reason, we create a high-quality dataset[1] and present a language model[2] specifically tailored to the cybersecurity domain that can serve as a basic building block for cybersecurity systems. The model is compared on 15 tasks: Domain-dependent extrinsic tasks for measuring the performance on specific problems, intrinsic tasks for measuring the performance of the internal representations of the model, as well as general tasks from the SuperGLUE benchmark. The results of the intrinsic tasks show that our model improves the internal representation space of domain words compared with the other models. The extrinsic, domain-dependent tasks, consisting of sequence tagging and classification, show that the model performs best in cybersecurity scenarios. In addition, we pay special attention to the choice of hyperparameters against catastrophic forgetting, as pre-trained models tend to forget the original knowledge during further training.

CCS Concepts: • **Computing methodologies** → **Machine learning**; *Natural language processing*; • **Security and privacy** → **Intrusion/anomaly detection and malware mitigation;**

Additional Key Words and Phrases: Language model, cybersecurity BERT, cybersecurity dataset

[1] https://github.com/PEASEC/cybersecurity_dataset
[2] https://huggingface.co/markusbayer/CySecBERT

This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE and by the German Federal Ministry for Education and Research (BMBF) in the projects CYWARN (grant no. 13N15407) and CYLENCE (grant no. 13N16636).

Authors' address: M. Bayer (Corresponding author), P. Kuehn, R. Shanehsaz, and C. Reuter, PEASEC, Technical University of Darmstadt, Pankratiusstraße 2, 64289 Darmstadt, Germany; e-mails: bayer@peasec.tu-darmstadt.de, kuehn@peasec.tu-darmstadt.de, r.shanehsaz@posteo.de, reuter@peasec.tu-darmstadt.de.

ACM 2471-2566/2024/04-ART18
https://doi.org/10.1145/3652594

ACM Reference Format:
Markus Bayer, Philipp Kuehn, Ramin Shanehsaz, and Christian Reuter. 2024. CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain. *ACM Trans. Priv. Sec.* 27, 2, Article 18 (April 2024), 20 pages.
https://doi.org/10.1145/3652594

1 INTRODUCTION

Cybersecurity has become an increasingly critical concern in today's digital world, as the number of cyber attacks continues to rise and their consequences become more severe [1]. The COVID-19 pandemic,[3] the Russian war of aggression against Ukraine [2], and the resulting shifts in diplomatic and geopolitical dynamics have only intensified this development. As a result, there is a growing need for effective cybersecurity measures to prevent, detect, and respond to these threats. One of the key components of effective cybersecurity is the ability to analyze and understand the vast amounts of data generated by various sources, such as logs, network traffic, and threat intelligence reports. However, much of their analysis is still performed manually by security experts, which is a time-consuming and labor-intensive process.

To address these challenges, recent advancements in **Natural Language Processing (NLP)** and machine learning have shown promise in automating the analysis of cybersecurity-related data. For instance, **cyber threat intelligence (CTI)** [3], an important domain of cybersecurity, involves the collection and analysis of information about emerging threats and vulnerabilities. Information thereby is often disseminated in the form of **indicators of compromise (IOCs)** or unstructured natural language text in blog posts and news articles [4, 5]. Applying NLP techniques to CTI can help automate the extraction and understanding of relevant, evidence-based knowledge, thus significantly reducing the manual workload for experts [6].

With regard to underlying NLP mechanisms, word embedding methods that use a sparse vector space to represent words are prominent examples [7]. In this context, models such as **bidirectional encoder representations from transformers (BERT)** [8] have become the standard basis models in all machine learning tasks that involve natural language as input. These models are already pre-trained on a general level and can be adapted to the task at hand by so-called fine-tuning. However, research has shown that the full potential of such models cannot be realized when applied to domain-specific tasks [9–12]. This is intuitive because these models intend to cover as many domains as possible and, especially in normal-sized models, specific domain knowledge is lost due to capacity constraints or because the knowledge is not even included in the training data. To gain domain-specific knowledge, pre-trained models can be further trained on domain-specific corpora to achieve better results in this particular domain [10]. However, it must be ensured that catastrophic forgetting does not occur, which means that the model forgets its original knowledge.

Models trained on general domain corpora, such as Wikipedia, and without further training often reach their limits when applied to domain-specific tasks such as cybersecurity [13]. Their limitations can be explained by two primary reasons:

(1) Unfamiliarity with domain-specific terminology: General domain models may not have encountered specific vocabulary of the cybersecurity domain, such as names and designations of new vulnerabilities or unique threat actor groups. This lack of exposure can lead to reduced performance when analyzing cybersecurity texts, as the model may fail to recognize crucial information.
(2) Semantic ambiguity across domains: General domain models may fail to disambiguate words with different meanings in different contexts. For example, the word *virus* might be

[3] https://enterprise.verizon.com/en-gb/resources/articles/analyzing-covid-19-data-breach-landscape/

interpreted by a general model as referring to a biological disease rather than to a type of malware, which is the more relevant meaning in the context of cybersecurity [13].

In this article, we propose CySecBERT, a word embedding model based on BERT [8] for analyzing cybersecurity texts. Our aim is to enable state-of-the-art Natural Language Processing (NLP) for the security domain and to provide a model that is highly suitable for practical cybersecurity use cases and a solid base for further research in this field. By evaluating our resulting model on different tasks (i.e., intrinsic and extrinsic tasks) we ensure that it indeed enriches the cybersecurity domain. In a preliminary evaluation phase, we try to identify appropriate hyperparameters to minimize the problem of forgetting previously trained knowledge, which is then verified using a standard NLP benchmark. In this study, we pre-train a model on a thoroughly chosen cybersecurity corpus consisting of various datasets, such as scientific papers, X (formerly known as Twitter), webpages, and the national vulnerability database. A well-performing model for this use case may supersede high manual workload for researchers and experts. Although there are well-performing models for various very specific purposes in the cybersecurity domain [14, 15], the importance of a general cybersecurity model that can serve as a basis for all kinds of tasks is undeniable. The following contributions are made by this article:

- A pre-trained, general-purpose cybersecurity language model based on BERT, called CySecBERT **(C1)**.
- Experiments for hyperparameter tuning in light of catastrophic forgetting **(C2)**.
- An evaluation of CySecBERT based on several tasks tailored to the cybersecurity domain, including intrinsic and extrinsic tasks, as well as a general benchmark, to measure whether and to which degree the model forgets past knowledge **(C3)**.
- A comparison of our model to a related cybersecurity model and to the original BERT model, as well as a discussion about its shortcomings and potential improvements **(C4)**.

2 RELATED WORK

This section provides an overview of relevant work on the topic of BERT models. We thereby outline models adapted to different domains that have emerged following the publication of BERT. Moreover, we summarize work that already proposes BERT-like language models for the cybersecurity domain. Finally, we specify the research gap that we intend to fill in the scope of our research.

2.1 BERT Models in Different Domains

In various publications, researchers have demonstrated that it is possible to achieve good classification quality on domain-specific text corpora with pre-trained models such as BERT. Of particular interest for our research is the method of **domain-adaptive pre-training (DAPT)** [9], which describes the process of training an already pre-trained language model on a domain-specific, domain-dependent dataset and which is conducted in the same way as the pre-training. Hence, this technique differs from classical fine-tuning in that the model is not specialized for just one task, but serves as a building block for many tasks in the field. It has been implemented in several other domains since the introduction of BERT [10, 11, 16]. A prominent example is BioBERT, introduced by Lee et al. [10], in which BERT was adapted to a biomedical corpus. BioBERT was initialized with weights from Devlin et al. [8]'s BERT model and then pre-trained once again, this time with a large biomedical dataset, in which the dataset was more than five times larger than BERT's. Evaluating the resulting model with a subsequent fine-tuning process of three different biomedical text mining tasks, which are **Named Entity Recognition (NER)**, **Relation Extraction (RE)**, and **Question Answering (QA)**, Lee et al. [10] were able to largely outperform

Table 1. Overview of Relevant Existing BERT Models for Special Domains

Model/Paper	Domain/Use Case	Method	Model Base
BioBERT [10]	biomedical	PT (+ FT)	BERT
SciBERT [11]	scientific	PT (+ FT)	BERT
[9]	papers (bio. + CS), news, reviews	PT (+ FT)	RoBERTa
MalBERT [15]	malware	FT	BERT, RoBERTa, DistilBERT
CatBERT [14]	phishing	FT	DistilBERT
ExBERT [20]	exploit prediction	FT	BERT
[19]	CTI	FT	BERT
CyBERT [16]	cybersecurity	PT	BERT

The method explains whether the model was only fine-tuned (FT) or also pre-trained (PT).

BERT and previous state-of-the-art models on these aforementioned tasks. Models that address other domains present similar approaches. SciBERT [11], for example, focuses on scientific publications whereas DA-RoBERTa, introduced by Krieger et al. [17] covers media bias. Gururangan et al. [9] underpin our method of additional pre-training on BERT by yielding good results in their application of this approach on RoBERTa [18], a variant of BERT that uses the same transformer-based architecture. In contrast to studies such as BioBERT by Lee et al. [10], in which only a single domain at a time is considered, Gururangan et al. [9] covered several different domains.

Similarly, researchers have also explored BERT models for the cybersecurity domain. For example, Ranade et al. [16] propose a BERT model for this domain called CyBERT. Although their paper states that fine-tuning on BERT took place, in fact, they further pre-trained BERT for the cybersecurity domain. Fine-tuning is performed on top of this pre-trained cybersecurity model and is primarily used for application. In general, however, their research goal is similar to ours.

There are further cybersecurity BERT models that, however, are fine-tuned instead of subjected to continued pre-training, as is required for true DAPT. Hence, this makes them less suitable for other tasks of the cybersecurity domain. MalBERT [15] is a BERT-based model from the cybersecurity domain focusing on the detection of malicious software. Another security-related work is CatBERT, introduced by Lee et al. [14]. They replaced some transformer blocks with adapters and fine-tuned the BERT model for the detection of phishing emails. Mendsaikhan et al. [19] introduced a BERT-based Natural Language filter for identifying and classifying cyber threat–related information from publicly available information sources with high accuracy.

An overview of the approaches with their domains and how they are trained can be found in Table 1. These works are related to ours because the approach of adapting BERT to a specific domain is similar to our methodological framework and differs mainly in the specific target domain. All in all, the different pre-trained BERT approaches can be of use for our work as an orientation and for comparisons of our results to theirs with regard to performance. Notwithstanding the fact that BERT has achieved great results in various domains, the full potential for the cybersecurity domain has yet to be exploited.

2.2 Research Gap

The previously established research gap has led us to develop a model with the aim of achieving satisfactory performance for cybersecurity textual material in various tasks. BERT has already been transferred to different domains, resulting in domain-specific models (BioBERT [10], SciBERT [11]) and has been applied even to specific domains in the cybersecurity field, producing models such as MalBERT [15] or CatBERT [14]. There are also BERT models that are not specific to a particular cybersecurity domain and that can handle various different texts but are then fine-tuned for a particular task, as in the case of the BERT model from Mendsaikhan et al. [19]. In contrast to these models and associated development research, the proposed CySecBERT is able to handle many different sources and text forms, and is at the same time the basic building block for all cybersecurity-related tasks.

As outlined in the introduction, there are a multitude of research problems in the field of cybersecurity based on the essential aspect of information extraction. A solid method to address this can improve research in the cybersecurity field at a stroke. Furthermore, the outcome enables extensibility and the application of additional layers on top of the model, for instance, CRF [21], (Bi)LSTM, or both combined [22].

Ranade et al. [16] also address the delineated research gap to some extent. Unfortunately, no juxtaposition was made with the results of BERT as the baseline, but only a presentation of their model's outcome was given. We compare our CySecBERT with theirs, which is varied in the model training and the corpus [16]. Furthermore, in delimitation to their work, we evaluate a whole span of different cybersecurity tasks, ranging from classification to NER and clustering tasks and we include the results of BERT for comparison. We also take into account the phenomenon of catastrophic forgetting, in which the pre-trained model forgets its already acquired knowledge in the new training phase. This issue has not been targeted in other works of this area. The similarity in in the work of Ranade et al. [16] and our work results from the nature of the research task, which all the more underlines the importance of the approach. We understand that the attention given to this research gap is important and encouraging at the same time. Multiple works addressing a similar objective can be complementary and thus accelerate filling the gap in research. Nonetheless, our work is distinguishes itself from the work of Ranade et al. [16] at several points, including the evaluation step, the applied data, and overall by the extent of our work.

3 METHODOLOGY

This section provides a brief background on domain adaptive pre-training, including the planned training process, and presents the dataset used to adapt our proposed language model to the cybersecurity domain, the architecture of the model, and a pre-evaluation phase.

3.1 Domain Adaptive Pre-Training

DAPT of language models to a specific domain is a common method to achieve an advanced domain-specific language model (*cf.* Section 2). It has shown to increase model performance in several ways. Not only is model performance on downstream tasks increased, hence generating better evaluation results, but training time is also reduced for such tasks due to smaller datasets for the training process to achieve similar performance. These prospects lead us to expect that the cybersecurity domain will benefit greatly from a domain-adapted, pre-trained language model for every possible task, e.g., NER and relevance classification, to name a few.

We aim to adapt BERT to the cybersecurity domain based on domain-specific text corpora (*cf.* Section 3.2) [9]. Our DAPT pipeline is built with *Huggingface*[4] and *Weights and Biases.*[5] The final domain-adapted pre-trained model is based on `bert-base-uncased`. Likewise, the text

[4] https://huggingface.co/
[5] https://wandb.ai/

corpus is tokenized using the bert-base-uncased model. The training itself is done on the Lichtenberg Cluster.[6]

During the training phase, we try to mitigate the problem of catastrophic forgetting [23] by reducing the learning rate, the training steps, and the size of the dataset compared with BERT pre-training. In this way, the susceptibility to catastrophic forgetting should be greatly reduced because the new learning process is subordinated. Nevertheless, we test whether the problem also occurs with the created model by evaluating it on a non-cybersecurity task. While we expect no improvements, we want to analyze to what extent the old knowledge is altered.

3.2 Text Corpus

When creating the text corpus, we paid great attention to the quality of the data, as this quality transfers to the model [24]. The text corpus is composed of four different sub-corpora: (i) blog data, (ii) arXiv data, (iii) National Vulnerability Database (NVD) data, and (iv) X (Twitter) data. Our decision for this selection is based on the kind of information that is used by security professionals and on the fact that these sources are commonly used in recent publications regarding machine learning. Those sub-corpora vary considerably in their structure. The NVD contains short, objective, and precise language with semi-structured information, and X consists of short posts with objective, subjective, emotional, on- as well as off-topic, etc. content. arXiv encompasses long papers with highly educational language, and blog posts are typically longer articles with less formal language.

The blog posts build a solid foundation for different writing styles and practical information in information security, including vulnerability and exploit information, threat notifications [25], and foundational knowledge. We initially aimed for a set of 41 different domains to be crawled. During the automated crawling process, three of them either blocked our requests or contained so much advertising and cookie information that we decided to omit those sources. This resulted in 38 different blogs, which we crawled, such as troyhunt.com, darkreading.com, schneier.com, and krebsonsecurity.com, to name a few. We filtered duplicates and instances shorter than 300 characters[7] and filtered for English texts using fasttext [26]. This process resulted in a total of over 151k blog posts based on the initial list of 165k web pages.

Next, we use arXiv papers from the category *Cryptography and Security*[8] [10]. Due to errors during the text extraction process, we ignored papers of length lower than 3000 characters, resulting in over 16k papers.

Then, we use vulnerability descriptions of the NVD [27, 28]. Experts curate those texts,[9] which is why they need no further processing. Hence, we neither filter nor pre-process this information.

Finally, we use X as information source [29–31], where we crawled datasets containing the following keywords based on information we received from CERT members:

- *infosec OR security OR threat OR vulnerability OR cyber OR cybersec OR infrasec OR netsec OR hacking OR siem OR soc OR offsec OR osing OR bugbounty*

Additionally, we crawled dedicated datasets of data breaches, such as the Microsoft Exchange Server Data Breach. Of all the tweets collected, we only used those that were annotated as English by X. While the posts include retweets, we did not gather replies. Overall, we managed to crawl nearly 4M tweets with over 179M tokens in total.

[6] http://www.hhlr.tu-darmstadt.de/

[7] A randomly selected and manually inspected sample of corner case blog posts, e.g., shorter or longer posts, showed that short posts contained mostly advertisements or cookie notifications.

[8] For text extraction, we used opendetex for papers in tex format or textract for papers in pdf format.

[9] https://www.cve.org/ResourcesSupport/FAQs#pc_cve_recordscve_record_descriptions_created

Table 2. Statistics of the Number of Tokens and Entries Based on the Training Dataset

#Tokens	Min[10]	Max[10]	Sum	Median	Mean	Entries
Blogs	44	0.1M	169M	710	1.1k	151k
arXiv	533	0.7M	167M	8.2k	9.9k	16k
NVD	5	1.9k	12M	58	71	171k
X (Twitter)	1	500	179M	39	45	4M
Total	1	0.7M	528M	40	122	4.3M

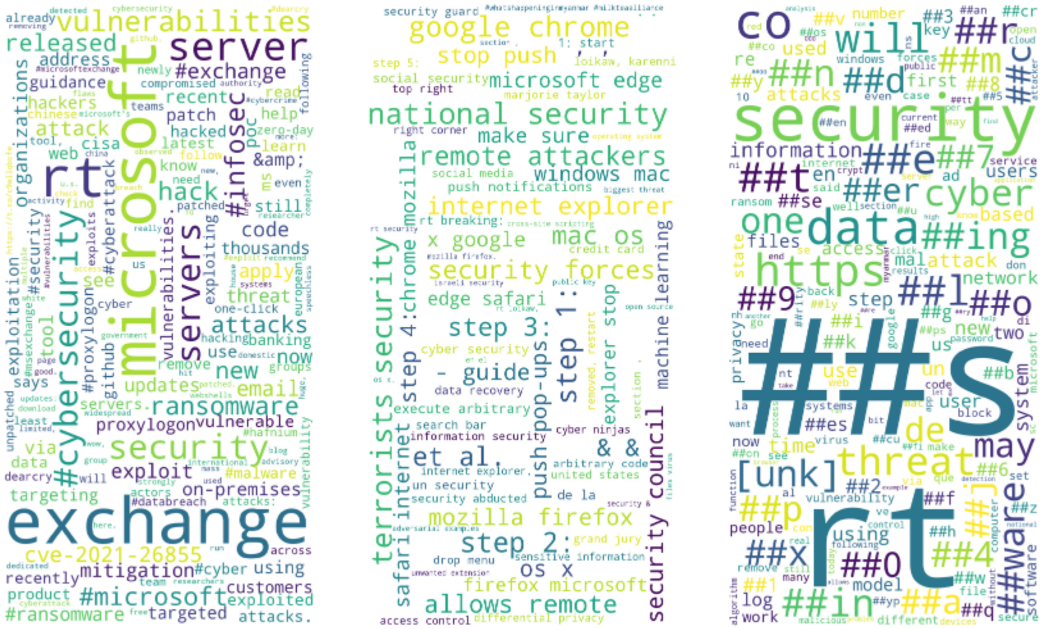

Fig. 1. Visualization of the training corpus through word clouds: (a) single words, (b) bigrams, and (c) tokenized input for the model.

We would like to emphasize that all collected entries have been minimally edited at most, as we want to have the most natural form in which the model will later function. A summary of all datasets is depicted in Table 2. To get an overview of the words and tokens contained in the dataset, we created word clouds for the dataset, which can be seen in Figure 1. The first image represents individual words, the second highlights bigrams, and the third illustrates the tokenized dataset. Across all word clouds, a prominent association with the security domain is evident.

3.3 Architecture

As explained before, we adapt the domain of the BERT model with further pre-training. The architecture is based on the transformer architecture, introduced by Vaswani et al. [32]. Transformers leverage the self-attention mechanism that enables the model to weigh the significance of each word in a sequence relative to the others, thereby capturing long-range dependencies and contextual information. They consist of encoders and decoders, but BERT only employs the encoder

[10] Minimal or maximal token per entry.

portion, i.e., it only encodes the text into an internal representation. During pre-training, BERT utilizes masked language modelling by randomly masking a certain percentage of words in the input sequence. The model is then tasked with predicting the masked words based on their surrounding context. This encourages BERT to develop a deep understanding of the language structure, as it has to infer the masked words using the available context.

From this follows the mathematical representation of the key components: given an input text which is transformed into a sequence of tokens, $X = \{x_1, x_2, \ldots, x_n\}$. From them, the token embeddings, i.e., fixed-size vector representations (with positional encodings), $E = \{e_1, e_2, \ldots, e_n\}$ are obtained. For each token, we compute the Query (Q), Key (K), and Value (V) vectors by projecting the embeddings using learned weight matrices WQ, WK, and WV:

$$Q_i = e_i * WQ \qquad K_i = e_i * WK \qquad V_i = e_i * WV. \tag{1}$$

Next, we calculate the attention scores for each token pair (i, j) (with d being the embedding dimension):

$$score(i, j) = \frac{Q_i * K_j^T}{\sqrt{d}}. \tag{2}$$

Then, we apply the softmax function to the scores to obtain attention weights:

$$weights(i, j) = e^{score(i,j)} / \sum_{k=1}^{n} e^{score(i,k)}. \tag{3}$$

Finally, we compute the self-attention output for each token by taking a weighted sum of the Value vectors:

$$output_i = \sum_{j=1}^{n} (weights(i, j) * V_j). \tag{4}$$

BERT uses multiple self-attention heads to capture different aspects of the context. Each head computes its own self-attention output $output_{h,i}$, which are concatenated and projected using another learned weight matrix WO:

$$multiheadoutput_i = [output_{1,i} \oplus output_{2,i} \oplus \ldots \oplus output_{H,i}] * WO. \tag{5}$$

Then, BERT applies a position-wise fully connected feed-forward network to the multi-head attention output, with W1, b1, W2, and b2 as learned weight matrices:

$$nnoutput_i = ReLU(multiheadoutput_i * W1 + b1) * W2 + b2. \tag{6}$$

Each encoder layer in BERT consists of the multi-head self-attention mechanism, followed by layer normalization, the position-wise feed-forward network, and another layer normalization:

$$outputmha_i = LayerNorm(multiheadoutput_i + e_i) \tag{7}$$

$$outputnn_i = LayerNorm(nnoutput_i + outputmha_i). \tag{8}$$

This process is repeated for each encoder layer in the model, as depicted in Figure 2.

While these mathematical representations describe the core components of the BERT architecture, including self-attention, multi-head attention, and the position-wise feed-forward network within the encoder layers, further details can be extracted from the work by Devlin et al. [8].

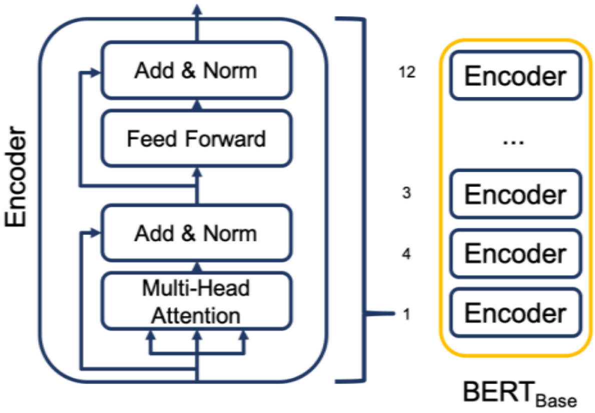

Fig. 2. Visualization of the BERT encoder stack [33].

3.4 Preliminary Evaluation: Catastrophic Forgetting

Catastrophic forgetting describes the phenomenon that an already trained model tends to catastrophically forget the previously trained knowledge when trained on further data [34]. If catastrophic forgetting occurs, our model might lose its ability to understand or interpret nuances and contexts that may not be explicitly covered in the cybersecurity domain but are still relevant for processing cybersecurity-related texts. (1) In real-world applications, data often contains a mix of domain-specific and general language. For instance, cybersecurity texts may include general language expressions, analogies, or references that require a broader understanding of language beyond just technical terms. (2) There are certain tasks for which the model has to handle in domain and out of the domain instances: for example, when the task is in an intersection of cybersecurity and another domain (such as law, policy, and management). A model exclusively trained on cybersecurity might struggle with texts that blend these disciplines. A model that retains its general language capabilities alongside its domain-specific knowledge is more robust and flexible [35]. Finally, the work of Rongali et al. [36] shows that avoiding catastrophic forgetting in domain-specific BERT models even improves the performance on the domain tasks. This illustrates the benefits of a well-balanced model that can handle both domain-specific and general language data effectively.

Research [8, 37] suggests that the second training should be more lightweight than the first so that the already trained knowledge is not overshadowed. In this regard, the three most important hyperparameters are the learning rate, which determines how much the new data updates the weights of the model; the epochs, i.e., how long the model is trained and how many updates are made; and, finally, the size of the dataset. The work of Sun et al. [37], for example, addresses the problem of catastrophic forgetting by examining different learning rates, demonstrating that a lower learning rate is best. The authors of the BERT paper, Devlin et al. [8], recommend training with only a few epochs. Accordingly, further training can often lead to overfitting of the data and, as already discussed, to catastrophic forgetting. This remark can also be transferred to the size of the datasets for DAPT, as they are very large and, correspondingly, many training updates are carried out. The difficulty is to find the right configuration of these parameteres to avoid catastrophic forgetting and at the same time to enable enough learning that the model achieves high quality in the new domain. Therefore, we test different configurations of the hyperparameters' learning rate, dataset size, and number of epochs in this preliminary evaluation. Since training a model already requires a considerable amount of time and resources (about 4 days on 4 NVIDIA V100 GPUs), we

Table 3. Pre-evaluation of Different CySecBERT Configurations Considering the Catastrophic Forgetting Task and the Performance in the Security Domain with the BoolQ Task (Accuracy) and the MSExchange Task (F1), Respectively

Configuration	BoolQ	MSExchange
LR 2×10^{-5}, Epochs 30, Data 10%	0.6752	**0.8869**
LR 1×10^{-4}, Epochs 30, Data 10%	0.6722	0.8620
LR 5×10^{-5}, Epochs 30, Data 10%	0.6700	0.8544
LR 2×10^{-5}, **Epochs 20**, Data 10%	0.6660	0.8860
LR 2×10^{-5}, **Epochs 40**, Data 10%	0.6614	0.8846
LR 2×10^{-5}, Epochs 30, **Data 5%**	**0.6785**	0.8816
LR 2×10^{-5}, Epochs 30, **Data 15%**	0.6716	0.8496

cannot perform an extensive hyperparameter search, such as the bio-inspired search by Ibor et al. [38]. Hence, we narrow down the range of configurations by taking into account the research findings and by orienting our approach towards the original BERT training process. The authors of BERT trained the model with a learning rate of 1×10^{-4} for about 40 epochs. We therefore propose the following configurations:

(1) Learning Rate: 1×10^{-4}, 2×10^{-5}, 5×10^{-5}
(2) Epochs: 20, 30, 40
(3) Data: 5%, 10%, 15%

For the other hyperparameters, we followed the original BERT approach, i.e., we used a weight decay of 0.01, a dropout rate of 0.1, 10 000 warm-up steps, and ADAM as the optimization algorithm [39]. We trained all models with a batch size of 64 on 4 NVIDIA Tesla V100 GPUs.

To be able to measure the degree of forgetting as well as the performance in the security domain, we evaluate the different models on the SuperGLUE task BoolQ and the security task MSExchange (a more detailed description of the datasets follows in Section 4.1).

The results are presented in Table 3. As can be seen, almost every configuration performs well in both tasks. The learning rate of 2×10^{-5} appears to be better in both tasks, which is in line with the results of Sun et al. [37]. Similarly, 30 epochs perform better on both tasks than the higher and lower configurations. However, with regard to the size of the dataset, the results are not as clear. While a dataset size of 10% of the BERT training dataset performs best on the MSExchange dataset, a dataset size of 5% of the original BERT training dataset seems to perform best on the BoolQ task. This is not surprising, as the degree of forgetting is lower with less new training data. Since the model with 10% still performs very well on the BoolQ task and because we weight the performance in the security domain more heavily, we decided to use the model with 10% of the original BERT training dataset. The training loss of this CySecBERT model can be seen in Figure 3. Accordingly, the loss decreases logarithmically and improves only very slowly after 300k steps.

4 EVALUATION

In this section, the evaluation process and the corresponding results are presented in detail. We provide a short description of the evaluation tasks in Section 4.1, followed by the presentation and interpretation of the results in Section 4.2.

4.1 Experiments and Tasks

Since our goal is to publish a model that is highly usable for the cybersecurity domain, we evaluate it against the current standard method of the domain (BERT) and against another cybersecurity

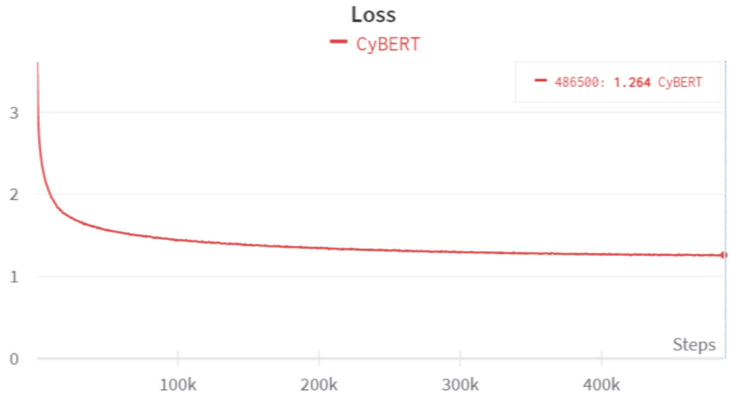

Fig. 3. Training loss of the CySecBERT model.

language model (CyBERT from Ranade et al. [16]). We use different types of tasks, i.e., intrinsic and extrinsic evaluation tasks, which are reasonably chosen for the field of cybersecurity. While the extrinsic tasks measure how well the trained model performs on downstream tasks, i.e., measuring real-world application, the intrinsic tasks measure the model itself without any kind of additional classifier, e.g., by measuring the representations of the model and by demonstrating an overall fit to the domain.

As intrinsic tasks, we apply a word similarity task using parts of the dataset by Mumtaz et al. [13] and an X dataset for clustering evaluation. The clustering dataset is based on a random sample of Log4j X posts. For this task, posts are converted into latent representations with different BERT models. The latent representation consists of the concatenation of the last four layers of the model output and of the mean values across all words in the post. A KMeans clustering algorithm with k-values from 5 to 9 is executed on the gathered and transformed posts. The evaluation scores are measured with the Silhouette Coefficient (the higher the better). This is an internal clustering metric that analyzes the clusters created and does not require gold labels. It is important because there can be many solutions and gold labels can be misleading in the case of clustering [40]. The cybersecurity word similarity dataset consists of words with their equivalents, all from the field of cybersecurity. Moreover, the dataset is an extension of the public cybersecurity word similarity dataset from Mumtaz et al. [13] and contains over 300 word pairs. For this extension of the original dataset, a cybersecurity expert followed the process of Mumtaz et al. [13] and added more cybersecurity words that can be considered similar. Normally, the word similarity evaluation is based on the cosine similarity of the word embeddings when static embeddings are used. The problem is that BERT is context dependent, which means that a clear word embedding cannot be given without context and the standard method of measuring word similarity is no longer suitable. For this reason, we have developed a new method for evaluating word similarity, in which the model predicts whether two given words are similar. Similar to the works on zero-shot learning, we create a meaningful cloze task consisting of a sentence with a masked word that the model fills in, which is implicitly the answer to the similarity question. The task is written in the following way:

"*Are word*$_1$ *and word*$_2$ *similar?* ***[MASK]***", where **[MASK]** can be either "Yes" or "No", which represent the masked words that the model has to fill.

Example: *"Are* ***virus*** *and* ***malware*** *similar?* ***[MASK]****"*

This trick of zero-shot learning means that no explicit classification model needs to be trained and the task remains intrinsic. In addition, we aim to show that the evaluated model not only predicts the similarity of each word to all other words but also detects when two words are not similar. We therefore randomly take word pairs from the dataset that are not similar and add them to the evaluation. This dataset is then used for evaluation and an F1 score is calculated for the model's predictions.

As part of the extrinsic tasks, we employ two cybersecurity classification tasks from Riebe et al. [31] and Bayer et al. [41]. In the first task, the classifier has to decide whether a Twitter post is related to the field of cybersecurity. In the second task, it has to predict whether a post might be relevant to experts in the field during a major cybersecurity event. Furthermore, we use the sequence tagging dataset by Kuehn et al. [27]. Sequence tagging is the task of finding specific words in a text, which requires that each word in a text is tagged, often as IOB: either *i*-nside, *o*-utside, or *b*-eginning, referring to the specific words being searched for. The dataset consists of several NER tasks (recognition of named entities) for predicting relevant details of NVD descriptions. We then chose the task of predicting the name and version of the software and the attack vector. We decided to employ these tasks because of their different performances in the work of Kuehn et al. [27] in order to analyze how well the models perform on varying levels of difficulties of the tasks.

We also evaluate CySecBERT and BERT on the SuperGLUE benchmark [42]. This is a common NLP benchmark, which we utilize to identify signs of catastrophic forgetting [23]. We assume that our cybersecurity model is not able to achieve a better or even equivalent result, as a certain degree of forgetting is acceptable and necessary for learning. Nevertheless, we expect the performance not to be too poor, as this would otherwise indicate that some basic knowledge would have been forgotten during the domain training phase.

Neural networks consist of many random hyperparameters that have to be fixed before each training and that are not transferable to other learning processes. As shown by Reimers and Gurevych [43], the frequent execution of a training process can be used to avoid incorrect inferences arising from a randomly better choice of hyperparameters. Accordingly, and following the experiments of Sanh et al. [44] and Liu et al. [18], all extrinsic experiments were performed five times and the mean values as well as the standard deviation are given in the respective tables.

4.2 Results

As stated in Section 4.1, we aim to evaluate CySecBERT, as well as BERT and CyBERT [16] on different tasks that are mainly located in the cybersecurity (CySec) domain, incorporating both intrinsic and extrinsic evaluation tasks. Additionally, we run the SuperGLUE task to test our model for catastrophic forgetting.

4.2.1 *Intrinsic Tasks.*

To measure the representation quality of the model, we evaluate it by applying two intrinsic tasks from CySec: clustering and word similarity.

Clustering. The results of the clustering task are given in Table 4. While the CyBERT model of Ranade et al. [16] only performs better than the baseline when forming 5 to 7 clusters, our approach performs better in every constellation according to the Silhouette Score. In fact, CySecBERT outperforms the cybersecurity model of Ranade et al. [16] by a considerable margin for each number of clusters, with consistent improvements ranging from +0.002 to +0.059 points. Our model shows the highest improvement when 9 clusters are formed. On the one hand, these results demonstrate that we can obtain more coherent clusters thanks to our trained language model. On the other hand, from a more general perspective, the results show that the model is more able to represent the given instances in a meaningful latent space.

Nevertheless, even better results could be expected if we used an approach such as Sentence-BERT by Reimers and Gurevych [45] for our model, as it has proven to be much more suitable for representing complete documents, such as tweets in our case.

Table 4. Silhouette Score of the First Intrinsic Task, Clustering the Data of a Log4J Dataset

# Clusters	BERT	CyBERT [16]	CySecBERT
5	0.114	0.141	**0.143**
6	0.115	0.124	**0.150**
7	0.118	0.133	**0.167**
8	0.125	0.117	**0.163**
9	0.130	0.113	**0.172**

The best values are bolded.

Table 5. Overview of the Results of the Word Similarity Task Where the Scores are Indicated by the F1 Score

Tasks	Word Similarity
BERT	0.4382
CyBERT [16]	0.4861
CySecBERT	**0.6382**

Table 6. Named Entity Recognition Score Based on Tagged Software Versions (SV), Software Names (SN), and Attack Complexities (AC) of NVD Descriptions

CVSS NER	SV	SN	AC
[27]*	0.8735 (-)	0.8584 (-)	- (-)**
BERT	0.9247 (0.0064)	0.8837 (0.0037)	0.3323 (0.0135)
CyBERT [16]	0.9298 (0.0019)	0.8834 (0.0029)	0.3336 (0.0214)
CySecBERT	**0.9302** (0.0066)	**0.8871** (0.0025)	**0.3472** (0.0116)

The results are given as F1 scores and the best values are bolded. *Showing the reported results of the work. **No comparable results available.

Word Similarity. The word similarity task results are displayed in Table 5. The baseline model has the worst performance, with an F1-score of 0.44. This is to be expected, as most domain-specific words were not or only very rarely included in the standard BERT training. Our CySecBERT model is clearly superior to the other two approaches, which is remarkable as it confirms the previous intrinsic results. However, we would like to point out that this task is different from other word similarity tasks as it does not reflect word similarities directly through the word representations, but rather by questioning the model in a cloze fashion (see Section 4.1).

4.2.2 Extrinsic Tasks. After we have already demonstrated that the model produces meaningful representations of cybersecurity-related vocabulary and data, we want to test whether our model is also comparably more suitable for real-world applications, i.e., for extrinsic tasks, the so-called *downstream tasks* of machine learning. The tasks that we have chosen for the cybersecurity domain are (i) NER, (ii) general relevance classification, and (iii) CTI classification.

NER. The results of the NER task are shown in Table 6. The table shows that the results of the original work of Kuehn et al. [27] are worse than those of the methods evaluated in our study. While the basic BERT model and the CyBERT model of Ranade et al. [16] are more similar to each other, e.g., in terms of **software naming (SN)**, our model consistently outperforms both. Only in the tagging of the **software version (SV)** does the CyBERT model of Ranade et al. [16] perform significantly better than the baseline BERT model, whereas our model improves this result. Hence, it could be speculated that the CyBERT training data from Ranade et al. [16] contained entries of software versions at a higher frequency than the normal BERT data. However, the CyBERT model deteriorates the results for **software names (SNs)**, which could indicate that either a large number of SNs are missing in high frequency in their dataset, or that they have been neglected due to errors in the training process. The highest improvements of our model can be seen in **attack complexity (AC)**, outperforming the CyBERT model of Ranade et al. [16] by 0.0136 points. Nevertheless, we perceive the results of this particular task as overall not very satisfactory. Reasons have already

Table 7. Classification Results of the MS Exchange and CySecAlert Dataset, Given as F1 Scores

	MS Exchange	CySecAlert
[31]*	-	0.8051 (-)
[41]*	0.8536 (0.0007)	-
BERT	0.8599 (0.0193)	0.8779 (0.0084)
CyBERT [16]	0.8766 (0.0153)	0.8647 (0.0095)
CySecBERT	**0.8869** (0.0026)	**0.8883** (0.0064)

The best values are bolded. *Showing the reported results of the work.

Table 8. Classification Results of the Few-Shot Experiments of the MS Exchange Dataset

Name	Model	Accuracy	F1
Baseline	BERT	0.4965 (0.0190)	0.5870 (0.1881)
ADAPET	BERT	0.6589 (0.0135)	0.6254 (0.0432)
[41]	CySecBERT	**0.7913** (0.0056)	**0.8063** (0.0027)

The best values are bolded.

been discussed by Kuehn et al. [27] and are mainly related to the problem of too little data in this task.

Relevance Classification (CySecAlert). In the first classification task of our experiments, the models are trained to predict whether an X post is related to the cybersecurity domain (see Table 7). This can be considered a general cybersecurity task, as the model only has to identify cybersecurity-related words. The original work by Riebe et al. [31] has the worst results, possibly due to the use of a classical machine learning method. It is particularly interesting to see that the Ranade et al. [16] model performs worse than the basic BERT model. Our model significantly improves the base model and the CyBERT model by 0.0104 and 0.0236 points in the F1 score, respectively. All models have a relatively low standard deviation, indicating that the fine-tuning process is stable across all runs.

Specialized CTI Classification (MS Exchange). The second classification task is about identifying specialized CTI where very specific words are needed to classify the instances. The results of this task are also presented in Table 7. Surprisingly, unlike in the previous tasks, the CyBERT model of Ranade et al. [16] is able to improve the baseline, showing that while it does not contribute improvements in the more general tasks, it could be beneficial in more specific tasks. There is a high improvement of our model compared with the baseline observation (+0.027), which we expected since this task focuses on very domain-dependent language and specific vocabulary. Moreover, although the CyBERT model of Ranade et al. [16] is advantageous for this task, our model still improves the results significantly by +0.0103 F1 points. It is also observable that our model has a significantly lower standard deviation than the other two models, which again indicates a very stable training process. Table 7 also contains the result of the work by Bayer et al. [41], which is very similar compared with the BERT baseline, but the best standard deviation.

We also included the few-shot learning task of this dataset in Table 8, in which only 32 instances are given as training data. CySecBERT is an integral part of the few-shot approach of Bayer et al. [41]. The results show a significant improvement over the baseline and over the state-of-the-art few-shot learning method ADAPET by Tam et al. [46]. This demonstrates how the CySecBERT model is incorporated and applied in further research, contributing to the advancements in other

Table 9. Results of the SuperGLUE Benchmark

	record	rte	wic	wsc	boolq	cb	copa	multirc	total
BERT	0.6416	0.5949	0.6476	0.5538	0.6760	0.3704	0.606	0.4067	0.6010
CyBERT [16]	0.6346	0.6173	0.5980	0.5337	0.6887	0.5676	0.615	0.4146	0.6065
CySecBERT	0.6137	0.5545	0.5887	0.5404	0.6752	0.5551	0.486	0.3915	0.5468

Indicated in the evaluation metric proposed in the benchmark.

disciplines. Details of the approach and further results can be extracted from the work of Bayer et al. [41].

4.2.3 Catastrophic Forgetting. In the final part of our evaluation, we revisit the problem of catastrophic forgetting. To this end, we evaluate our model with the SuperGLUE benchmark to test whether the model degrades the results considerably, which would indicate that the model has forgotten the initial knowledge acquired in the BERT training phase. The results of this task and a comparison to the BERT model is displayed in Table 9. As expected, we can see that our model reduces almost every task outcome. Nevertheless, the worsening of results is not an indication for catastrophic forgetting, as the differences are still relatively small, with a mean drop of about -0.05 points. This shows that although the model has lost some of its knowledge, most of it is still present. It is particularly interesting that the performance in the cb task has even increased and the result of the boolq task has remained almost the same.

In addition, we have again included the results of the CyBERT model by Ranade et al. [16], which, interestingly, performs almost the same as the BERT model. One could interpret this as an indication that the model did not suffer catastrophic forgetting during training. However, taking the results of the extrinsic and intrinsic domain tasks into account, we can conclude that their model might have learned very little overall, as it is seldom better, sometimes worse and most often on par with BERT. This can be attributed to the training process of Ranade et al. [16], who trained their model only on 17,000 documents with one epoch, whereas our model is trained on 4,300,000 documents with 30 epochs.

4.2.4 Conclusion. In the evaluation, we have shown that the model developed in this work is very well adapted to the cybersecurity context. The tasks have demonstrated that the CySecBERT model is able to outperform the BERT baseline and the CyBERT model by Ranade et al. [16] consistently across all cybersecurity tasks. We evaluated these models on intrinsic cybersecurity tasks in which we summarized how accurate the models represent documents and words in latent space. Based on these tasks, the fundamental quality of the language model has been assessed. In addition, we evaluated the three models using extrinsic cybersecurity tasks that demonstrate the practicality of the model in most real-world application contexts. Our model improves the results of these tasks by up to 0.027 F1 points compared with the other two models and it achieves its highest improvement on an in-depth cybersecurity task in which very specific language differences have to be considered. Moreover, we analyzed the phenomenon of catastrophic forgetting by evaluating our model on standard NLP tasks. Although we observed a deterioration in performance in these tasks, it is only within the expected range of decline. We concluded that the CyBERT model of Ranade et al. [16] may have been trained on a much too small corpus with too few training steps (17,000 documents and one epoch for CyBERT versus 4,300,000 documents and 30 epochs for our CySecBERT). We can say with confidence that our model is capable of handling a wide range of cybersecurity tasks while retaining the original language modelling knowledge.

5 DISCUSSION, CONCLUSION, AND OUTLOOK

In this work, we propose a novel state-of-the-art cybersecurity language model based on BERT [8]. We performed DAPT on this model with a sensibly chosen cybersecurity corpus. The corpus consists of a variety of source data structures, such as blogs, scientific papers, and X data. The data and the sources were selected to be appropriate for cybersecurity research and practice. The size of the dataset and the structure of the training process were chosen to prevent catastrophic forgetting on the one hand and, on the other hand, to enable enough learning for the model to contribute to the general field and to the specific niches of cybersecurity. We explored this through a pre-evaluation phase by tuning the hyperparameters in terms of catastrophic forgetting and domain quality. We then highlighted the performance of our model with a thorough main evaluation of various tasks and compared it with the BERT baseline as well as the current state-of-the-art cybersecurity language models. First, we evaluated the models on two intrinsic tasks, in which we demonstrate that the quality of our model improves in terms of the learned representation space, i.e., how well the cybersecurity-specific instances (words and texts) can be distinguished from each other. Tables 4 and 5 show the substantial performance increases achieved by our model. Second, we evaluated the model together with the other two models for cybersecurity-specific classification and NER tasks to exemplify the usefulness and practicality of the model in application contexts. Our model outperforms the other models in every task, which can be derived from Tables 6 and 7. The greatest improvement is observed in the special CTI classification dataset, suggesting that the model is particularly beneficial when dealing with very specific cybersecurity language that is not contained in the training dataset of the standard BERT model. Our evaluation concludes with a focus on catastrophic forgetting by which we assessed the performance of our model against a general NLP benchmark. While these results (Table 9) point out that our model does indeed degrade them overall, they also show that, as intended, there is no catastrophic forgetting and that the final model has combined much of its original knowledge with the new knowledge about cybersecurity.

While we are aware that the current state-of-the-art on research in language modeling and NLP generally tends to focus on larger language models such as GPT-3 by Brown et al. [47], we have chosen the BERT model on purpose. Most of the cybersecurity research and especially practice does not have the necessary resources to apply large language models. In most cases, the BERT model can still be considered the standard model in machine learning application contexts such as the cybersecurity domain. In this way, our work is most beneficial to the research landscape and to practice.

Practical and Theoretical Implications. Our work contributes to research and practice through a novel, state-of-the-art cybersecurity model called CySecBERT, which we have published. We also published the associated dataset so that it can contribute to further research. Thus, our work has several implications for practice and research:

A novel, state-of-the-art language model for cybersecurity that is useful for various tasks. With our research surrounding the model, we have aimed to find a solution to increase the performance of machine learning in as many cybersecurity language tasks as possible. Our model provides utility for a large number of tasks, as can be estimated based on the success in extrinsic task scores as well as inferred from intrinsic task scores. Thereby, they show that the representation space is better for the domain-dependent language with our model. With the release of our model, we are paving the way for better cybersecurity tools, as practitioners can easily use the new model in existing pipelines, for example, in alert aggregation [48], in detection of phishing websites [49, 50], or even in malware detection [51]. More advanced tools will then also be the result of new research derived from the model. These tools will have the potential to improve the results in various tasks by incorporating further research ideas already implemented as part of the research of Bayer et al. [41], for example, where the CySecBERT model has been integrated. This can be pursued on a smaller scale, in which the model is not the focus but serves as a foundation on which further techniques such as data augmentation, meaningful data selection, few-shot learning, or specific applications are built. Yet, it can also be done on a larger scale, in which the model is the subject of research, for example, by analyzing its results in explainable artificial intelligence approaches.

In our evaluation, we demonstrate that the CySecBERT model can serve as a substitute for other models in the field of cybersecurity by providing new state-of-the-art performances. However, the CySecBERT model will not be suitable as a replacement for every type of cybersecurity model. We expect that further work in this area will adopt our methodology and will train other models that may be even more specialized regarding a specific cybersecurity topic or that may have a much larger neuron size.

Evaluation of catastrophic forgetting in terms of learning rate, dataset size, and number of epochs. When training the CySecBERT model, we paid special attention to catastrophic forgetting phenomena in which the trained model loses its valuable initial knowledge. To ensure that the level of catastrophic forgetting is kept to a minimum, we first performed a preliminary evaluation of the hyperparameters' learning rate, dataset size, and number of epochs, as research in this area suggests that the second training process should not overshadow the first training [8, 37]. We narrowed down the space of possible hyperparameter constellations by these considerations and evaluated seven fully trained CySecBERT models on a standard NLP task and a cybersecurity task to measure the degree of catastrophic forgetting and the quality in the cybersecurity domain. Our results are in line with research and show that the training process should not be too heavyweight, e.g., by heavy updates due to a high learning rate, but also not too lightweight so that domain knowledge is not acquired, e.g., due to a smaller dataset. In the final section of our main evaluations, we evaluated the CySecBERT model on the SuperGLUE benchmark, which consists of standard NLP tasks and compared it with the BERT model and the CyBERT model of Ranade et al. [16]. While the results show that our model performs worse than the BERT model, this is to be expected since some level of catastrophic forgetting will always occur when further training the model towards a certain domain. However, it is interesting to note that the results of the CyBERT model of Ranade et al. [16] are very similar to the BERT model, which is due to the small training process that the CyBERT model originates from. While the CyBERT model is trained on only 17,000 documents with one epoch, our model is trained on about 4,300,000 documents with 30 epochs.

It is not clear whether these results, especially with regard to the hyperparameters, are generalizable to other language models, especially to very large language models such as GPT-3 and GPT-4. For an evaluation with a broader hyperparameter search, as executed with a bio-inspired, evolutionary method from Ibor et al. [38], for example, or for a focus on these very large language models, we expect research to be primarily concerned with catastrophic forgetting and with having the necessary extensive resources.

A cybersecurity dataset containing most relevant sources for the training process. The dataset was created with consideration given to including a variety of sources and textual types. This ensures that the model can be applied to a broad range of cybersecurity tasks. However, we anticipate future work analyzing the published dataset, for example, to enhance its quality based on specific criteria or to identify any unintended biases. The dataset can also serve as a basis for training other language models. Although we have deliberately chosen this size of the dataset for BERT training to prevent catastrophic forgetting, it might be useful to expand it, which can easily be done by collecting more data from the sources that we have already selected. It might also make sense to utilize larger language models that might achieve even greater performance or that might even use neural architecture search to find a suitable architecture for special use cases, as proposed by Shang et al. [52] or Okunoye and Ibor [53].

Ethical Considerations. We would like to emphasize that we did not explicitly focus on and analyze social biases in the data or the resulting model. While this may not be so damaging for most application contexts, there are certainly applications that depend heavily on these biases, and including any kind of discrimination can have serious consequences. As authors, we would like to express our warnings regarding the use of the model in such contexts. Nonetheless, we aim for an open-source mentality, observing the great impact it can have. Therefore, we transfer the thinking to the user of the model, drawing on the many previous discussions in the open-source community.

ACKNOWLEDGMENTS

We thank all anonymous reviewers of this work. The calculations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt.

REFERENCES

[1] Suzanne Widup, Dave Hylender, Gabriel Bassett, Philippe Langlois, and Alex Pinto. 2020. 2020 Verizon data breach investigations report. *Verizon* (05 2020). DOI: http://dx.doi.org/10.13140/RG.2.2.21300.48008

[2] Song Tae Eun. 2022. Cyber warfare in the Russo-Ukrainian war: Assessment and implications. *Institute of Foreign Affairs and National Security* (2022), 2.

[3] Eoin Hinchy. 2022. *Voice of the SOC Analyst*. Technical Report. Tines. 39 pages. Retrieved from https://www.tines.com/reports/voice-of-the-soc-analyst/

[4] Bhavna Soman. 2019. Death to the IOC: What's Next in Threat Intelligence. Retrieved December 28, 2020 from https://www.blackhat.com/us-19/briefings/schedule/#death-to-the-ioc-whats-next-in-threat-intelligence-15392 (2019).

[5] Thomas D. Wagner, Khaled Mahbub, Esther Palomar, and Ali E. Abdallah. 2019. Cyber threat intelligence sharing: Survey and research directions. *Computers & Security* 87 (Nov 2019), 101589. DOI: http://dx.doi.org/10.1016/j.cose.2019.101589

[6] Wiem Tounsi and Helmi Rais. 2018. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & Security* 72 (Jan 2018), 212–233. DOI: http://dx.doi.org/10.1016/j.cose.2017.09.001

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 — Workshop Track Proceedings*.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964* (2020).

[10] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[11] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676* (2019).

[12] Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. *arXiv preprint arXiv:1904.03323* (2019).

[13] Sara Mumtaz, Carlos Rodriguez, Boualem Benatallah, Mortada Al-Banna, and Shayan Zamanirad. 2020. Learning word representation for the cyber security vulnerability domain. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[14] Younghoo Lee, Joshua Saxe, and Richard Harang. 2020. CATBERT: Context-aware tiny BERT for detecting social engineering emails. *arXiv preprint arXiv:2010.03484* (2020).

[15] Abir Rahali and Moulay A. Akhloufi. 2021. MalBERT: Using transformers for cybersecurity and malicious software detection. *arXiv preprint arXiv:2103.03806* (2021).

[16] Priyanka Ranade, Aritran Piplai, Anupam Joshi, Tim Finin, et al. 2021. CyBERT: Contextualized embeddings for the cybersecurity domain. In *IEEE International Conference on Big Data*.

[17] Jan-David Krieger, Timo Spinde, Terry Ruas, Juhi Kulshrestha, and Bela Gipp. 2022. A domain-adaptive pre-training approach for language bias detection in news. *arXiv preprint arXiv:2205.10773* (2022).

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[19] Otgonpurev Mendsaikhan, Hirokazu Hasegawa, Yukiko Yamaguchi, Hajime Shimada, and Enkhbold Bataa. 2020. Identification of cybersecurity specific content using different language models. *Journal of Information Processing* 28 (2020), 623–632.

[20] Jiao Yin, MingJian Tang, Jinli Cao, and Hua Wang. 2020. Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description. *Knowledge-Based Systems* 210 (Dec. 2020), 106529. DOI: http://dx.doi.org/10.1016/j.knosys.2020.106529

[21] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using BERT-CRF. *arXiv preprint arXiv:1909.10649* (2019).

[22] Shaohua Jiang, Shan Zhao, Kai Hou, Yang Liu, Li Zhang, et al. 2019. A BERT-BiLSTM-CRF model for Chinese electronic medical records named entity recognition. In *2019 12th International Conference on Intelligent Computation Technology and Automation (ICICTA)*. IEEE, 166–169.

[23] Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*. Vol. 24. Elsevier, 109–165.

[24] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *Comput. Surveys* (June 2022), 3544558. DOI: http://dx.doi.org/10.1145/3544558

[25] Xiaojing Liao, Kan Yuan, Xiaofeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. 2016. Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In *Proceedings of the ACM Conference on Computer and Communications Security*, Vol. 24-28-Octo. ACM Press, New York, NY, 755–766. DOI: http://dx.doi.org/10.1145/2976749.2978315

[26] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. FastText.Zip: Compressing Text Classification Models. (2016). DOI: http://dx.doi.org/10.48550/arXiv.1612.03651

[27] Philipp Kuehn, Markus Bayer, Marc Wendelborn, and Christian Reuter. 2021. OVANA: An approach to analyze and improve the information quality of vulnerability databases. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*. ACM, 11. DOI: http://dx.doi.org/10.1145/3465481.3465744

[28] Ying Dong, Wenbo Guo, Yueqi Chen, Xinyu Xing, Yuqing Zhang, and Gang Wang. 2019. Towards the detection of inconsistencies in public security vulnerability reports. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 869–885.

[29] Haipeng Chen, Rui Liu, Noseong Park, and V. S. Subrahmanian. 2019. Using Twitter to predict when vulnerabilities will be exploited. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, New York, NY, 3143–3152. DOI: http://dx.doi.org/10.1145/3292500.3330742

[30] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. 2015. Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits. *Proceedings of the 24th USENIX Security Symposium* (2015), 1041–1056.

[31] Thea Riebe, Tristan Wirth, Markus Bayer, Philipp Kuehn, Marc-André Kaufhold, Volker Knauthe, Stefan Guthe, and Christian Reuter. 2021. CySecAlert: An alert generation system for cyber security events using open source intelligence data. In *International Conference on Information and Communications Security (ICICS)*.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[33] R. Evtimov, M. Falli, and A. Maiwald. 2020. Anti Social Online Behaviour Detection with BERT. (Feb 2020). Retrieved from https://humboldt-wi.github.io/blog/research/information_systems_1920/bert_blog_post/

[34] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. 2018. Measuring catastrophic forgetting in neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (April 2018). DOI: http://dx.doi.org/10.1609/aaai.v32i1.11651

[35] Lixin Su, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2020. Continual domain adaptation for machine reading comprehension. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1395–1404. DOI: http://dx.doi.org/10.1145/3340531.3412047 arXiv:2008.10874 [cs].

[36] Subendhu Rongali, Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. 2020. Improved pretraining for domain-specific contextual embedding models. CoRR abs/2004.02288, (2020). Retrieved from https://arxiv.org/abs/2004.02288

[37] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to Fine-Tune BERT for Text Classification? (Feb. 2020). http://arxiv.org/abs/1905.05583 arXiv:1905.05583 [cs].

[38] Ayei E. Ibor, Olusoji B. Okunoye, Florence A. Oladeji, and Khadeejah A. Abdulsalam. 2022. Novel hybrid model for intrusion prediction on cyber physical systems' communication networks based on bio-inspired deep neural network

structure. *Journal of Information Security and Applications* 65 (March 2022), 103107. DOI:http://dx.doi.org/10.1016/j.jisa.2021.103107

[39] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. (Jan. 2017). DOI:http://dx.doi.org/10.48550/arXiv.1412.6980 Number: arXiv:1412.6980 arXiv:1412.6980 [cs].

[40] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. Information overload in crisis management: Bilingual evaluation of embedding models for clustering social media posts in emergencies. In *European Conference on Information Systems 2021 Research Papers*. 19. https://aisel.aisnet.org/ecis2021_rp/64

[41] Markus Bayer, Tobias Frey, and Christian Reuter. 2022. Multi-Level Fine-Tuning, Data Augmentation, and Few-Shot Learning for Specialized Cyber Threat Intelligence. (July 2022). http://arxiv.org/abs/2207.11076 arXiv:2207.11076 [cs].

[42] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv:1905.00537 [cs]* (Feb. 2020). http://arxiv.org/abs/1905.00537 arXiv: 1905.00537.

[43] Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. DOI:http://dx.doi.org/10.18653/v1/d17-1035

[44] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[45] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. DOI:http://dx.doi.org/10.18653/v1/d19-1410

[46] Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and Simplifying Pattern Exploiting Training. (Sept. 2021). http://arxiv.org/abs/2103.11955 arXiv:2103.11955 [cs].

[47] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[48] Max Landauer, Florian Skopik, Markus Wurzenberger, and Andreas Rauber. 2022. Dealing with security alert flooding: Using machine learning for domain-independent alert aggregation. *ACM Transactions on Privacy and Security* 25, 3 (Aug. 2022), 1–36. DOI:http://dx.doi.org/10.1145/3510581

[49] Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorrie Cranor. 2011. CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security* 14, 2 (Sept. 2011), 1–28. DOI:http://dx.doi.org/10.1145/2019599.2019606

[50] Peng Yang, Guangzhen Zhao, and Peng Zeng. 2019. Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access* 7 (2019), 15196–15209. DOI:http://dx.doi.org/10.1109/ACCESS.2019.2892066

[51] Aleieldin Salem, Sebastian Banescu, and Alexander Pretschner. 2021. Maat: Automatically analyzing VirusTotal for accurate labeling and effective malware detection. *ACM Transactions on Privacy and Security* 24, 4 (Nov. 2021), 1–35. DOI:http://dx.doi.org/10.1145/3465361

[52] Ronghua Shang, Songling Zhu, Jinhong Ren, Hangcheng Liu, and Licheng Jiao. 2022. Evolutionary neural architecture search based on evaluation correction and functional units. *Knowledge-Based Systems* 251 (2022), 109206. DOI:http://dx.doi.org/10.1016/j.knosys.2022.109206

[53] Olusoji B. Okunoye and Ayei E. Ibor. 2022. Hybrid fake news detection technique with genetic search and deep learning. *Computers and Electrical Engineering* 103 (2022), 108344. DOI:http://dx.doi.org/10.1016/j.compeleceng.2022.108344

Received 5 December 2022; revised 18 December 2023; accepted 6 March 2024