# Interpretable Identification of Cybersecurity Vulnerabilities from News Articles

**Pierre Frode de la Foret**
École Nationale Supérieure des Mines de Paris
1 Rue Claude Daunesse, Sophia Antipolis, France
pierre.frode_de_la_foret@mines-paristech.fr

**Stefan Ruseti**
University Politehnica of Bucharest
313 Splaiul Independentei, Bucharest, Romania
stefan.ruseti@upb.ro

**Cristian Sandescu**
CODA Intelligence
60 Mircea Vulcanescu, Bucharest, Romania
cristian.sandescu@codaintelligence.com

**Mihai Dascalu**
University Politehnica of Bucharest
313 Splaiul Independentei, Bucharest, Romania
mihai.dascalu@upb.ro

**Sebastien Travadel**
École Nationale Supérieure des Mines de Paris
1 Rue Claude Daunesse, Sophia Antipolis, France
sebastien.travadel@mines-paristech.fr

## Abstract

With the increasing adoption of technology, more and more systems become target to information security breaches. In terms of readily identifying zero-day vulnerabilities, a substantial number of news outlets and social media accounts reveal emerging vulnerabilities and threats. However, analysts often spend a lot of time looking through these decentralized sources of information in order to ensure up-to-date countermeasures and patches applicable to their organisation's information systems. Various automated processing pipelines grounded in Natural Language Processing techniques for text classification were introduced for the early identification of vulnerabilities starting from Open-Source Intelligence (OSINT) data, including news websites, blogs, and social media. In this study, we consider a corpus of more than 1600 labeled news articles, and introduce an interpretable approach to the subject of cyberthreat early detection. In particular, an interpretable classification is performed using the Longformer architecture alongside prototypes from the ProSeNet structure, after performing a preliminary analysis on the Transformer's encoding capabilities. The best interpretable architecture achieves an 88% F2-Score, arguing for the system's applicability in real-life monitoring conditions of OSINT data.

## 1 Introduction

With the increasing number of cybersecurity attacks, institutions need to join forces for the prevention and detection of cyberthreats. Malicious entities target companies, but also individuals and public institutions (including health organisations), with an overall expected cost of 6000 bn $ per year (Morgan, 2020). For example, the number of requests on the French malware assistance plat-



*Proceedings of Recent Advances in Natural Language Processing*, pages 428–436
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_049

form has tripled between 2018 and 2019[1], and this trend has worsened during the pandemic (Pinhasi and Huseman, 2021) due to the increased use of technology, the advent of remote working, coupled with an exponential growth of IoT. In their endless race against hackers, security experts need to detect zero-day vulnerabilities and understand new methods employed by hackers to exploit them (Lewis, 2018). In addition, both institutions and individuals need localized expertise applicable for their environment, given its specificity in terms of employed technologies.

Large companies often have their dedicated entity, the Security Operations Center (SOC), whose experts survey Open-Source Intelligence (OSINT) data to identify new emerging vulnerabilities (Dionísio et al., 2019); however, lower-sized entities and individuals should have this information also readily available. In addition, each data source (and there are a lot of possible venues – e.g., more than 40 major blogs and newspapers are reported by Feedspot (2021) ) has its own targeted fields and technologies of main interest, as well as its subjectivity in expressing the breath and impact of the vulnerability. Manual searches in multiple sources is an overall tedious process, and the wide scattering of news feeds makes the day-by-day surveillance a daunting task; as such, an automated detection of zero-day vulnerabilities from OSINT data becomes a necessity (Le Sceller et al., 2017).

Our goal is to provide an automated filtering of daily news feeds to identifying emergent cybersecurity threats as an *initial screening* for security experts. We emphasize from the beginning the importance of recall, namely it is critical not to disregard potential threats. Moreover, model interpretability is an important dimension of the analysis in order to provide a preliminary grounding for the model's decisions. Hence, our research question is: To what extent can automated systems detect cybersecurity threats in news articles while ensuring interpretable results?

The paper is structured as follows. The second section introduces related work on real-time identification of threats and interpretable Natural Language Processing (NLP) approaches, while the third section introduces our approach, composed of a deep analysis of the context and the design of an interpretable model. The fourth section presents results alongside performance metrics, followed by discussions and conclusions.

[1] https://www.cybermalveillance.gouv.fr/tous-nos-contenus/actualites/rapport-activite-2020

## 2 Related Work

In this section, we review relevant related research on real-time threat identification and interpretable models in NLP. The abundance of OSINT data brought by Twitter has enabled SOCs to develop cyber-threat intelligence with increasing performance, while exploring tweets in different manners (e.g., by CVE(Sabottke et al., 2015) or by account(Dionísio et al., 2019)). Nonetheless, to our knowledge, existing studies focus on performance and do not consider interpretability, which seems crucial for such a critical task.

### 2.1 Real-time Threat Identification

Several papers (Attarwala et al., 2017; Dionísio et al., 2019; Le Sceller et al., 2017) introduced Twitter-based approaches to design a pipeline for threat detection and, more generally, semantic analysis. Dionísio et al. (2019) start from a set of customers, whose experts chose the Twitter cybersecurity accounts to monitor. From these accounts, tweet texts are collected using a keyword-based selection. They rely on a Convolutional Neural Network (CNN) to select only interesting tweets, while also performing named entity recognition, and check performance using True Positive and Negative Rate. A comparison between the dates from first tweet disclosing the threat to the release on the national vulnerability database of a CVSS provides insights on the ability of Twitter to become an efficient cyberthreat detection platform.

Le Sceller et al. (2017) query tweets based on keywords. Their aim is to detect and characterize cybersecurity events using only texts from tweets. A preliminary taxonomy is built to understand how main cybersecurity keywords interact, when taking decisions. The collected texts are embedded using TF-IDF and a clustering algorithm allows for up-to-date unsupervised grouping of tweets. The clustering is applied in a dynamic manner to avoid obscolescence, as the keyword search is adapted according to relations and co-occurrences of words from tweets considered interesting upstream. These new keywords for the search are proposed to an external human actor, who takes the decision of adding them or not. Thus, the up-to-dateness of the architecture is guaranteed manually by experts, who

leave a trace of their analysis of new trends within the model.

Abdullah et al. (2018) directly use data from new articles to detect cyberthreats. After crawling various news websites, the authors define manually, together with experts, a certain number of features defining cyberthreats, such as the threat actors, the name of the cyberattack, or the jeopardized domain. The authors create a dictionary for each feature, in which occurrences of each word corresponding to this feature are stored with their context. Using this dictionary, a Conditional Random Field enables the detection of cyberattacks from sentences. Finally, Latent Semantic Analysis supports the categorization of articles, following the type of cyberattack they shed light upon.

### 2.2 Interpretability in NLP

Two classes of methods tackling interpretability coexist in the literature, namely: a) a posteriori methods, which take as input a model as such, and try to explain its decisions, and b) interpretable architectures whose interpretability is taken into account in their design.

With regards to a posteriori methods, several papers (Ribeiro et al., 2016; Sundararajan et al., 2017) have studied ways to determine and visualize the isolated influence of each variable from the input of a model. Ribeiro et al. (2016) designed LIME (Local Interpretable Model-agnostic Explanations), a tool which enables the local visualization (i.e., in the input space) of the influence of each interpretable component on the decision. The model whose decision we want to explain is locally approximated by a simple-and thus interpretable-classifier. In order to make these local explanations global, a fixed number of explanations are chosen a using Submodular Pick to render, as well as possible, the use of the features by the algorithm. These explanations are then aggregated into a global result. They also introduce metrics on the evaluation of a model's trustworthiness using their explanation system, and emphasize the importance of an oracle to assess the quality of explanations.

Sundararajan et al. (2017) bring a clear formalization on the problem of the attribution of deep network prediction to its input features. They emphasize the two conditions for a good attribution system, namely: a) *sensitivity* (i.e., if the input differs by 1 input component from the reference input, the attribution should be non-zero), and b) *implementation invariance* (i.e., if two architectures produce the same output for the same input, their attributions should be identical). They explain why state-of-the-art methods (especially gradients) do not meet these two criteria, and propose a method that complies – integrated gradient. Their method consists in integrating the gradient along component $i$ to get the attribution of input $i$.

Interpretability by attribution is a method that can be applied to any architecture, but as a consequence it does not consider the underlying mechanics for an architecture to explain the decision. A posteriori methods can also specialise in one type of architecture. In the particular case of the Transformer architecture (Vaswani et al., 2017) based on stacking self-attention layers, visualization tools like Bertviz (Vig, 2019) can be used to underpin an explanation of the model. Nonetheless, Brunner et al. (2019) warn on over-interpretation, when trying to explain self-attention. They especially argue that self-attention scores become a very complex mixture of interwoven words, while going deeper into the architecture, as a token is only responsible on average for 7.5% of the second-layer self-attention gradient. With this in mind, Chefer et al. (2020) do not restrict the influence of tokens to attention scores, but compute relevance and gradients back through the entire architecture, so as to compute the influence of each token on the decision.

The second approach considers interpretability by design. For example, Ming et al. (2019) provide an interpretable architecture for text classification that considers as implementation an RNN encoder; nevertheless, the model can consider any encoder. The idea for their architecture is to learn embeddings which well represent a special class of articles - i.e., "prototypes", and to cover as well as possible the latent space while relating to these articles. Our model builds on top of this architecture and additional details are provided in section 3.3.

## 3 Method

### 3.1 Corpora

Our aggregated corpus for training and evaluating our models consists of 1600 news articles from two collections of labeled articles that were obtained using two different approaches: Iorga et al. (2020) introduce a corpus of 1000 news articles on cybersecurity manually labeled by experts from news outlets, and Iorga et al. (2021) consider 600 more

articles that were extracted from selected Tweeter accounts. The distribution in terms of length is displayed in Figure 1; as it can be observed, more than half of the articles exceed the usual length of 512 tokens acceptable by most pretrained language models.

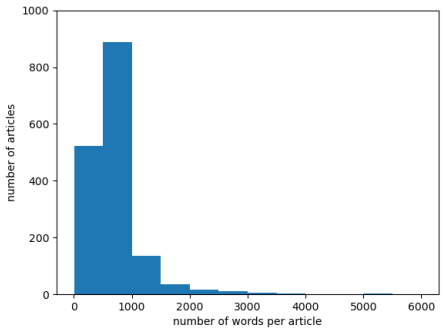


Figure 1: Length of articles

While accounting for the interpretability of our model, a more in-depth analysis of the articles from this aggregated dataset was required. An overwhelming majority of relevant articles disclose new vulnerabilities, either directly or through the description of an attack (or campaign). k-Means clustering was used to confirm this duality, while considering only relevant articles. Silhouette scores were computed for different number of clusters, which were afterwards visualized and cross-checked by hand to observe the split between attacks and explicit vulnerabilities. The optimal silhouette score was 0.15 for 2 clusters, followed by values lower than 0.13 for a higher number of clusters. As such, the duality is also confirmed while inspecting the most frequent tokens for each cluster (see Table 1).

| |
|---|
| attack, user, researcher, vulnerability, device, attacker, security, malware, data, malicious |
| vulnerability, security, flaw, attacker, user, cve, update, code, version, windows |

Table 1: Keywords grouped by cluster.

A higher number of clusters would have put forward specific threats like password leaks, which represent a minority of articles and are often linked to attacks. Next, we need to consider the permissiveness of the classification: even when the technology compromised by the cyberthreat is very specific, the automated pipeline needs to label the corresponding articles as relevant. Moreover, the article needs to include specific details about the identified vulnerabilities, for example the corresponding attack vector. However, these details may represent a very limited part of the overall article.

While considering the subtle difference between the introduced topics in cybersecurity articles and the previous observations, the model needs to process the article as a whole, with corresponding inter-dependencies at word, sentence, or even paragraph levels. As such, bag-of-words approaches, such as Multinomial Naive Bayes(MNB, (Kibriya et al., 2004)), though easily interpretable by the user, will only make the most obvious decisions.

Besides the previous aggregated corpus, a second considerably larger and unlabeled corpus was also collected. The creation of this second dataset meets a need for creating an embedding space for the cybersecurity domain. For this purpose, recursive scrapping was used on more than 20 news websites to collect webpages, which leads to a corpus of 65.8 million tokens and a vocabulary of 63 thousand words.

### 3.2 Data Pre-processing

Each article was pre-processed. First, accents, symbols, IP addresses, links, contractions, and residual dots were removed. Second, numbers were replaced by a # to report their presence. The dataset was then randomly split into a train set of 1000 articles and a test set of 600 articles using a stratification approach. It is important to emphasize the importance of recall, as we do not want to turn a "blind eye" on potential threats. Thus, the metric to be optimized for classification is F2-Score.

### 3.3 Interpretable Model

With the aim of designing an interpretable classifier, we combined the ProSeNet (Ming et al., 2019) architecture with Longformer (Beltagy et al., 2020) as an encoder for the entire articles.

ProSeNet considers a special layer, named the prototype layer, that takes the hidden state (in output of the encoder) as input, and its output is given to a standard classifier (dense layers and a sigmoid, in our case). The principle of the prototype layer is to compute the similarities ($sim(x, y) = \frac{<x,y>}{||x||*||y||}$) between the hidden state and learned vectors $p_1, ..., p_k$. These $k$ vectors (i.e., prototypes) are defined in the same latent space as the hidden state; as such, these vectors, modified by backpropagation during the epoch, are projected at the end

of epoch in the latent space of their nearest neighboring articles. If the dataset is large enough to cover the latent space, the training that includes this projection stabilizes, and the user ends up with a classification within which only the comparison between the input document and these *k* prototypes matters for final prediction.

This construction of the decision is similar to human selection: when deciding, for example, if an article is relevant or not, it is natural to relate to similar, previously seen, articles. Further comparisons informs users that, if they want to make reasonable decisions, they should consider a wide range of articles. Thus, the number of prototypes being fixed, what an expert would expect from these prototypes is for each of them to be representative for various types of articles and to properly cover the entire possible semantic space of articles.

Nonetheless, the similarities computed with all prototypes have to be aggregated in making the final binary decision. This is where the method reaches its limits in terms of interpretability, as there is no telling how the classifier mixes the information of proximity to the prototypes in order to take its decision.

When looking at state-of-the-art work on automated text classification, the Transformer-based models stand out, pushing the boundaries of RNNs (LSTMs or GRUs cells) in terms of long-term dependencies encoding. Nonetheless, the quadratic-order computation of self-attention limits the number of tokens allowed as input. A frequently employed solution is to use only part of the text, or make several predictions (e.g., each centered on a paragraph) that are aggregated by means of voting for the final decision. However, the vulnerability in the article might end up being either completely neglected or erased for classification, if its description place in the article is limited. Therefore, the article should be considered as a whole to mitigate the risk of neglecting important details.

The Longformer (Beltagy et al., 2020) architecture overcomes the limits of classic Transformer models by changing the computation of self-attention. Instead of performing $number\,of\,words^2$ operations, a window size $w$ is chosen, and relations are measured between the word and a sliding window of size $w$ around it. Beltagy et al. (2020) introduce three types of self-attention patterns distributed among the 12 encoding layers of the Longformer. The *sliding window attention* one has a dilation rate of 1 and it allows the efficient capturing of local information. This local information can then be aggregated thanks to *dilated sliding-window self-attention* patterns to capture more global information and to increase the receptive field. *Global+sliding windows* are mainly used in the final layers to provide task specific tokens.

In our classification task, a single global self-attention is computed in the last encoding layer of the Longformer. The resulting architecture is presented Figure 2.

As the current implementations of ProSeNet are not adapted to our problem, we decided to adapt the implementation of Meyer (2019). The project is released as open-source and is available on Github[2].

### 3.4 Training Hyper-parameters

The Longformer has a window size of 128 tokens, whereas the other parameters are the default ones. The usual learning rate scheduler for Transformers was used for all training episodes, and a weighted binary cross-entropy was chosen to counterbalance the 2/3-1/3 irrelevant-relevant ratio. The classical Longformer model (i.e., without ProSeNet, with the usual classifier at the end) was trained both with and without pretraining. Pretraining is achieved on masked language modelling using the unlabelled cybersecurity dataset.

The interpretable architecture (Longformer+ProSeNet) relies on 15 prototypes. A first challenge was to make the training stable. The training of the entire architecture takes place as follows: we initialize the encoder with the weights of the previously trained Longformer as standalone, and we freeze it for the first epochs to stabilize training. Then, we unfreeze the layers and set a small learning rate for the last epochs. Four projections are computed during training.

### 3.5 On the Role of ProSeNet

Moreover, we were interested on the importance of the additional ProSeNet layer in terms of data separation between relevant and irrelevant articles. With this goal in mind, we scrutinized the influence of the different training steps (pretrained only, pretrained and finetuned) on the action of Longformer on data.

Nonetheless, when considering the distribution of embeddings in the latent space (see Figure 3),

[2]https://github.com/readerbench/IRVIN

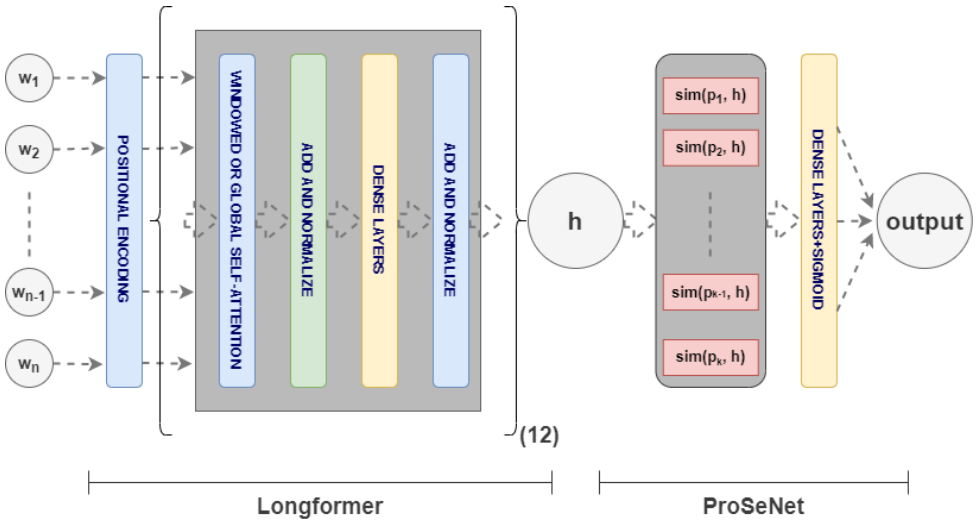


Figure 2: The architecture of the proposed Longformer+ProSeNet neural model.

only two groups emerge; reality is much more complex, as some irrelevant articles are not even linked to cybersecurity at all. This finer-grained characterization of the latent space is precisely what we expect ProSeNet to shed light upon. We fall in line with Ming et al. (2019) on the importance of diversity and sparsity for prototypes; however, it is also important to avoid having prototypes in the part of space where relevant and irrelevant articles coexist without a genuine separation or coherence, because of the lack of data. Prototypes thus need to represent a clearly relevant or clearly irrelevant part of the space. Indeed, they need to be representative for groups of articles, but if these groups are not clearly labeled as relevant or irrelevant, this will lead to a decrease in performance.

In terms of explainability, we emphasize the need for simplicity, as stated in the original study (Ming et al., 2019). Prototypes need to be represented briefly, without losing information required to identify them - as such, we simply present the articles by their title to the user.

## 4 Results

Table 2 reports the performance of various configurations, including a very simple interpretable baseline - a Multinomial Naive Bayes classifier trained using TF-IDF embeddings, while considering only the top 350 most relevant tokens chosen by feature importance.

In terms of the standard Longformer, the best performance was reached with pretraining, highlighting that additional cybersecurity context provides a significant boost in recall. Transformer-based models completely outperform MNB, particularly in terms of recall (98% for Transformer, only 62 % for MNB), but at the heavy cost of interpretability. ProSeNet was designed to tackle this problem, but as we can see in Table 2, classification performance slightly deteriorates, with a 4% decrease in F2-Score when compared to the standard Longformer with pre-training.

## 5 Discussions

An encountered issue is that only 11 prototypes were actually selected because of duplicates (the links for the selected papers are presented in Table 3). These duplicates appeared during projection, despite adding a loss favoring the variety between prototypes to the weighted binary cross-entropy for training. To understand the emergence of duplicates, we have to remember that the amount of data might not be sufficient to cover well the entire latent space; thus, two prototypes in a poorly covered area are likely to be projected on the same article.

While taking a closer look at the articles, our first intuition (after having studied and tediously labelled the entries), is that the selected article seem to cover quite well the different cases. The balance between relevant and irrelevant articles is fulfilled. Articles 1, 2, 3, and 7 present general studies or miscellaneous events which are linked to cybersecurity, but not at all to cyberthreat detection. Articles 4 and 8 study a cyberattack (without the disclosure of any vulnerability) and a legal case involving a cybercrime group; these are irrelevant. In contrast, relevant articles directly show vulnerability disclosure (articles 5, 9, and 10) or through cybercrime

Figure 3: Projections with (right) and without (left) finetuning. Relevant articles are blue, irrelevant articles are pink

| | Accuracy | Precision | Recall | F2-Score | Interpretable |
|---|---|---|---|---|---|
| MNB | 0.84 | **0.92** | 0.62 | 0.66 | *Yes* |
| Longformer (no pre-training) | **0.87** | 0.76 | 0.95 | 0.90 | No |
| Longformer (pre-trained) | 0.86 | 0.73 | **0.98** | **0.92** | No |
| Longformer+ProSeNet | **0.87** | 0.78 | 0.91 | 0.88 | *Yes* |

Table 2: Comparison between different architectures.

(articles 6 and 11). The security experts responsible for the initial datasets also confirmed our intuition. Nonetheless, it is important to acknowledge that the evaluation of interpretability has not been achieved by a representative assembly of cybersecurity experts or members of SOCs, even though this is the minimum expected if we want organizations to trust our model. Still, the analysis provided by the experts grounded an encouraging overview in terms of sparsity, diversity, and simplicity (see 3.5).

In order to further support the degree to which the selected prototypes cover the latent space, Figure 4 adds the prototypes to the latent space. At first sight, their distribution does not reflect the previously mentioned diversity, as the articles from the region of indecision from the encoder are mostly not covered by prototypes. This further consolidates our previous argument, i.e. the lack of data, which causes prototypes to be obviously labeled articles, and thus hinders performance.

Moreover, affinity propagation was also been implemented, with the aim of finding a finer-grained split: 54 clusters are identified, which are way too many for only 600 articles. On closer inspection, the grouping was done according to the company affected by the vulnerability. An approach based on keyword mining for each article would be interesting to further explore.

## 6 Conclusions and Future Work

In this study we introduce a state-of-the-art architecture based on Longformer and ProSeNet to create an interpretable pipeline to automatically label emerging cybersecurity threats. The similarity with the human decision process, coupled with a balanced performance on rather small dataset (i.e., recall - the most important metric - reaches 91%, while precision is at 78%), argue for the model's adequacy. Thus, in response to the initial research question, our architecture provides an efficient filter, while also ensuring interpretability.

The architecture is in place, but at a crossroads. First, further data collection is required in order to extend the training dataset. Once a substantial number of articles are collected, the quality of our pipeline will be re-assessed, while also including security experts to scrutinize the explanations of our model. This is a long-term endeavour in which

| Index | Title | Link | Label |
|---|---|---|---|
| 1 | IRS offers grants for software to trace privacy-focused cryptocurrency trades | https://cutt.ly/mnSVJ5R | irrelevant |
| 2 | How to build up cybersecurity for medical devices | https://cutt.ly/InSVLGO | irrelevant |
| 3 | The Comprehensive Compliance Guide | https://cutt.ly/cnSVCPK | irrelevant |
| 4 | Ransomware gang with $42 million laundering caught by Ukraine | https://cutt.ly/xnSVBW0 | irrelevant |
| 5 | New Highly-Critical SAP Bug Could Let Attackers Take Over Corporate Servers | https://cutt.ly/knSVMGv | relevant |
| 6 | 18-Byte ImageMagick Hack Could Have Leaked Images From Yahoo Mail Server | https://cutt.ly/WnSV0Dy | relevant |
| 7 | The cybersecurity skills shortage is getting worse | https://cutt.ly/InSV2LW | irrelevant |
| 8 | InvisiMole Hackers Target High-Profile Military and Diplomatic Entities | https://cutt.ly/ynSV3k9 | irrelevant |
| 9 | Google Discloses 20-Year-Old Unpatched Flaw Affecting All Versions of Windows | https://cutt.ly/bnSV4Ck | relevant |
| 10 | Google Android RCE Bug Allows Attacker Full Device Access | https://cutt.ly/lnSV6AR | relevant |
| 11 | MediaTek Bug Actively Exploited, Affects Millions of Android Devices | https://cutt.ly/WnSBwpx | relevant |

Table 3: Sample prototypes (using Longformer+ProSeNet).

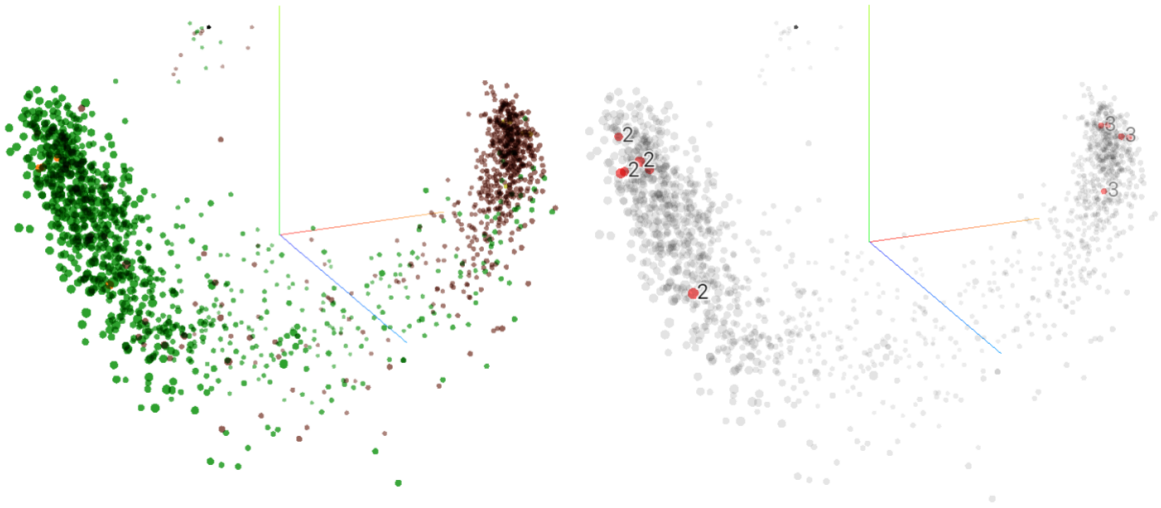


Figure 4: Projections of the two classes (left), and the corresponding prototypes (right). Relevant articles are in brown, irrelevant articles in green, whereas prototypes are in pink

the experts would be asked to chose the closest prototype out of 3 candidates for a given article. Second, we aim to implement a finer-grained clustering of the articles and identify trending topics or sub-domains. In addition, we aim to include a customizable filter that will enable SOCs to select themes for relevant articles, as well as targeted applications, thus accounting for their deployed infrastructure.

## Acknowledgments

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number 2PTE/2020, YGGDRASIL – "Automated System for Early Detection of Cyber Security Vulnerabilities" and by the internal UPB program Proof of Concept.

## References

Mohamad Syahir Abdullah, Anazida Zainal, Mohd Aizaini Maarof, and Mohamad Nizam Kassim. 2018. Cyber-attack features for detecting cyber threat incidents from online news. In *2018 Cyber Resilience Conference (CRC)*, pages 1–4. IEEE.

Abbas Attarwala, Stanko Dimitrov, and Amer Obeidi. 2017. How efficient is twitter: Predicting 2012 us presidential elections using support vector machine via twitter and comparing against iowa electronic markets. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 646–652. IEEE.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, and Roger Wattenhofer. 2019. On the validity of self-attention as explanation in transformer models. *arXiv preprint arXiv:1908.04211*.

Hila Chefer, Shir Gur, and Lior Wolf. 2020. Transformer interpretability beyond attention visualization. *arXiv preprint arXiv:2012.09838*.

Nuno Dionísio, Fernando Alves, Pedro M Ferreira, and Alysson Bessani. 2019. Cyberthreat detection from twitter using deep neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Feedspot. 2021. Top 45 cyber security news websites for information security pros. Feedspot. https://blog.feedspot.com/cyber_security_news_websites/.

D. Iorga, D.-G. Corlatescu, O. Grigorescu, Sandescu C., M. Dascalu, and R. Rughinis. 2021. Yggdrasil – early detection of cybernetic vulnerabilities from twitter. In *23rd Conference on Control Systems and Computer Science*. IEEE.

D. Iorga, D.-G. Corlatescu, O. Grigorescu, C. Sandescu, M. Dascalu, and R. Rughinis. 2020. Early detection of vulnerabilities from news websites using machine learning models. In *19th RoEduNet Conference: Networking in Education and Research*. IEEE.

Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2004. Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, pages 488–499. Springer.

Quentin Le Sceller, ElMouatez Billah Karbab, Mourad Debbabi, and Farkhund Iqbal. 2017. Sonar: Automatic detection of cyber security events over the twitter stream. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, pages 1–11.

James Andrew Lewis. 2018. *Economic Impact of Cybercrime*. Technical report, Center for Strategic and International Studies.

Ross Meyer. 2019. tf-prosenet. https://github.com/rgmyr/tf-ProSeNet.

Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. 2019. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 903–913.

Steve Morgan. 2020. Global cybercrime damages predicted to reach $6 trillion annually by 2021. In *Cybercrime Magazine*.

Zohar Pinhasi and Jim Huseman. 2021. Top cyber security experts report: 4,000 cyber attacks a day since covid-19 pandemic. CISION. https://www.prnewswire.com/news-releases/top-cyber-security-experts-report-4-000-cyber-attacks-a-day-since-covid-19-pandemic-301110157.html.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Carl Sabottke, Octavian Suciu, and Tudor Dumitras. 2015. Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1041–1056, Washington, D.C. USENIX Association.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, pages 5998–6008.

Jesse Vig. 2019. Bertviz: A tool for visualizing multihead self-attention in the bert model. In *ICLR Workshop: Debugging Machine Learning Models*.