ICMS Chair
Axis 2: Intrusion and abnormal behavior detection

PhD Proposal :
"AI Assisted Automotive Threat Intelligence"

P. Mozharovskyi[1], S. Hommes[2], and V. T. Nguyen[1]
pavlo.mozharovskyi@telecom-paris.fr
stefan.hommes@zf.com
van-tam.nguyen@telecom-paris.fr

[1]Telecom Paris / IP Paris,
[2]ZF

1. Context of automotive threat intelligence

The current threat intelligence for deeply embedded, safety-critical automotive products is a time-consuming process which is not easy to scale: both structured and unstructured information sources have to be taken in account and the relevance for specific products needs to be evaluated. An example of such a task would be identification of product after-production (sophisticated and unexpected) defects, testified by first user experience formulated as a non-formal (also to be verified) but often publicly-available feedback.

As a natural consequence of the commonly growing necessity in automatization of the threat intelligence, we see potential in improving this process by including an (partially generative) AI based-proposal mechanism to support cybersecurity analysts with the tasks of *duplicate detection*, *relevance judgement*, and *prioritization*. While previous developments in generative modelling [1] testified global success, followed by, *e.g.*, [2] and [3], to name but a few, contemporary development of large language models only amplified their application domains, underlying potential use in threat intelligence.

With *product monitoring* being in the main scope of this thesis proposal, the practical task at hand here is the *vulnerability analysis* during long term support, *i.e.*, it is expected that the product has been put to series production with all state of the art measures applied and all known vulnerabilities at that point fixed. The next issue to be resolved is now to monitor various sources of new vulnerabilities which affect the product that is already on the market; usually, if something important is detected, various sources report about the same vulnerabilities. It is for this reason that the *duplicate detection* shall be tackled in first order, as we can see below.

2. Addressing methodological challenges

In the current PhD-proposal, we suggest to tackle the methodological challenges of the automotive threat intelligence subsequently addressing the following *four tasks*:

2.1. Duplicate detection. Often, different intelligence sources (such as blogs, forums, newspaper articles, AUTO-ISAC, ...) report the same events from researchers, hackers, *etc.* An AI-enforced system could recognize these duplicates by using, *e.g.*, NLP-based approach [4]

and tag the cybersecurity information accordingly; ideally, duplicate detection should be realizable without product specific training information (see also [5]).

To address this challenge, a novel methodology including attention-based mechanism [6] shall be developed; both commercial and free-to-use implementations shall be considered [7]. The expected output of the developed tool should constitute an embedding of the report at hand into a meaningful metric/linear space.

2.2. Relevance judgement (for certain products). In threat intelligence, analysts need to provide judgements whether an event is relevant for their company or for a certain product provided by the company. Clearly, such relevance decisions would require product specific training, to be implemented and employed in the following way:

(a) An embedding-constructing mechanism (usually a neural network) should be developed, pre-trained on existing events (see, *e.g.*, [8,9]). This can in fact be taken over from the previous task. Further, this network needs to be fine-tuned for task-specific situations [10, 11].

(b) On the inference stage, out of an event (description), an embedding shall be created, treatable as an observation in the Euclidean space (that is a point).

(c) Comparison-computation—in sense of a relevant measure, *e.g.*, statistical data depth function [12,13]—shall characterize semantic closeness of the observation to one of the predetermined groups; see also [14] and [15] for exact and approximate computation.

(d) Probabilistic calibration of the decision (1st and 2nd type errors) made, *e.g.*, based on testing methodology available for these types of methods [16].

2.3. Prioritization. Frequently, analysts need to prioritize certain events (both critical and non-critical). Such a relevance judgement would also require product-specific training data, and probably a somewhat adjusted learning process. Again, an embedding, taken over from the previous approach, *i.e.*, already calibrated one can either be readily used, or a potential re-calibration might be needed.

To ensure relevant operator-assistance, particularly important at the beginning while constituting a part of the data-collection process, an explanation is necessary for the human in charge. A rather simple model at the end, *e.g.*, explanation approaches like FLINT [17] can be used, that are expected to provide the explanation of why some news are judged relevant or not themselves or in comparison with the others. Then an operator/expert could verify/question/correct (thus continuing the data collection process) the model's decision. Such verification should be fast because of the high-level representation. Furthermore, features can be optionally chosen to be complex, *e.g.*, attributed to a human-understandable concept, as it is the case in [18] or [19].

2.4. Extraction of a threat phrasing. In many situations, a (perhaps local in time) combination of tokens can be of high priority while possessing a threat. This can represent a phrasing that refers to a specific product without obvious thematic tagging or any other meta-data identifier. With the available developed machinery, extraction of such tokens' ensembles should be done as an extension, possibly-chronologically-at the end of the current project.

3. Data and learning

3.1. Availability of training data. Currently, for product development, ZF is typically processing a small two-digit number of cybersecurity events per day (most of them not relevant). Some of them are public information (internet sources which the system could parse

automatically), some would be private disclosures. Before starting the thesis, we will judge whether this amount of information would be seen as sufficient to continuously train an AI-based proposal system.

As a start, and to show a general proof of concept, instead of training with automotive specific data, publicly available databases like CVE/NVD (in combination with CVSS scoring) could be taken into account. Further, AI-assisted labelling can be implemented, including existing on the web opinions, labelling by operator/expert after receiving pre-processed information, which for example could also include explainability elements for potential correction by the expert; and thus the network can immediately improve.

3.2. Continuous learning. Since both the threat landscape and the product spectrum that analysts have to monitor are evolving, the system should be continuously learning by making proposals to the operators/experts and monitoring whether the operators would agree with the proposals or choose other options instead. For hard to decide cases, judgements from multiple experts for the same cybersecurity event could be taken into account, and the experts could optionally provide ratings about how certain they are with their relevance and prioritization decisions.

A possible extension of this approach would be to use the data accumulated by the assistant system to autonomously *search for similar related threat data* on the web, *i.e.*, to actively gather cybersecurity information based on learned preferences of the analysts, if necessary exploiting the results of *task 2.4*.

4. Working plan:

While the working plan is ambitious, it is expected to maintain necessary intensity to satisfy the objectives and subsequently de-risk the technology to be developed:

- **1st year:**

(a) Overview of the relevant literature, including domains of threat intelligence, generative AI, NLP-embeddings, and relevant statistical and visualization tools (*e.g.*, *t*-SNE [20], …, *etc.*).
Note: In view of the very rapid contemporary development of the field, a non-negligible portion of a PhD student's time shall be dedicated to this durable activity.

(b) Determination of the main scope regarding the data to be used and the thematic of the first data set.

(c) In close collaboration with experts from both academia and industry, constitution of the (first) data set itself, as well as development of the mechanism of its automatic augmentation with newly-arriving data.

(d) Training of the first transformers towards achieving the 1st objective.

- **2nd year:**

First of all, completion of a full-scale data set as well as establishment of a well-working mechanism of data enrichment should be prioritized and ensured in this year.

Further, gradual advancement on *tasks 2.1*, *2.2*, and *2.3* is normally expected, underlined with publication(s) in high-level thematic and/or general journal(s).

- **3rd year:**

While first three tasks are expected to arrive to their completion, if the time allows, *task 2.4* shall be addressed.

An open-source implementation of the developed methodology should be accomplished, exemplified by (and using), *e.g.*, [21], [22], [23]. This shall be an expected output along with the corresponding publications.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, 27, 2672–2680.

[2] Karras, T., Laine, S., Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

[3] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T. (2020). Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33, 12104–12114.

[4] Mikolov, T., Chen, K., Corrado, G.S., Dean, J. (2013). Efficient estimation of word representations in vector space. In: *International Conference on Learning Representations*.

[5] Pennington, J., Socher, R., Manning, C. (2014). GloVe: Global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 31, 6000–6010.

[7] *Introducing Meta Llama 3: The most capable openly available LLM to date.* https://ai.meta.com/blog/meta-llama-3/.

[8] Kusner, M., Sun, Y., Kolkin, N., Weinberger, K. (2015). In: *International Conference on Machine Learning*, 37, 957–966.

[9] Clark, E., Celikyilmaz, A., Smith, N.A. (2019). Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2748–2760.

[10] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, 4171–4186.

[11] Zhang, J., Zhao, Y., Saleh, M., Liu, P. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In: *International Conference on Machine Learning*, 119, 11328–11339.

[12] Mosler, K. and Mozharovskyi, P. (2022). Choosing among notions of multivariate depth statistics. *Statistical Science*, 37(3), 348-368.

[13] Zuo, Y. & Serfling, R. (2000), General notions of statistical depth function, *The Annals of Statistics*, 28(2), 461–482.

[14] Dyckerhoff, R. and Mozharovskyi, P. (2016). Exact computation of the halfspace depth. *Computational Statistics and Data Analysis*, 98, 19–30.

[15] Dyckerhoff, R., Mozharovskyi, P., and Nagy, S. (2021). Approximate computation of projection depths. *Computational Statistics and Data Analysis*, 157, 107166.

[16] Malinovskaya, A., Mozharovskyi, P., and Otto, P. (2024). Statistical process monitoring of artificial neural networks. *Technometrics*, 66(1), 104–117.

[17] Parekh, J., Mozharovskyi, P., and d'Alché-Buc, F. (2021). A framework to learn with interpretation. In: *Advances in Neural Information Processing Systems*, 34, 24273–24285.
[18] Ribeiro, M.T., Singh, S., Guestrin, C. (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*, 1135–1144.
[19] Alvarez-Melis, D., Jaakkola, T. (2018). Towards robust interpretability with self-explaining neural networks. In: *Advances in Neural Information Processing Systems*, 32, 7775–7784.
[20] Van der Maaten, L., Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86), 2579−2605.
[21] Pokotylo, O., Mozharovskyi, P., and Dyckerhoff, R. (2019): Depth and depth-based classification with R-package ddalpha. *Journal of Statistical Software*, 91(5), 1-46.
[22] Codes for 'Framework to learn with interpretation': https://github.com/jayneelparekh/FLINT.
[23] Python-library 'data-depth': https://data-depth.github.io/ (https://pypi.org/project/data-depth/).