Modeling Attacker "Danger" Based on Classification and Cyber-Physical System Security*

Oleksandr Umanskiy[1,*,†], Olha Korol[1,†], Oleksandr Kushnerov[2,†], Oleksandr Laptiev [3,†] and Andrey Sharapata [4,†]

[1] National Technical University "Kharkiv Polytechnic Institute", Kyrpychova 2 61002 Kharkiv, Ukraine
[2] Sumy State University, Kharkivska 116 40007 Sumy, Ukraine
[3] Taras Shevchenko National University of Kyiv, Volodymyrska Street 64/13 01601 Kyiv, Ukraine
[4] Kharkiv National Automobile and Highway University, Yaroslava Mudroho str. 25 61002, Kharkiv, Ukraine

Abstract

Assessing the threat level to cyber-physical systems (CPS) requires evaluating the capabilities of potential attackers. This thesis presents a model of attacker "danger" grounded in a detailed classification of attackers for industrial control systems (ICS) and CPS. We categorize both internal and external attackers and assign quantitative weight coefficients to their capabilities. A formal model is developed that incorporates attacker competence, resource availability, time to breach, attack likelihood, and motivation. We derive formulas for an attacker's danger level and its weight coefficient, and we provide tables of criteria for expert evaluation. Using the attacker classification, we map each attacker category to the technical levels of impact on ICS/CPS infrastructure. A methodology is proposed to determine an unknown attacker's category based on observed attack features. The results enable security analysts to rank attackers by danger level and to identify critical threats for each attacker category.

Keywords

impact levels, attacker modeling, cyber risk assessment, cyber-physical systems, threat assessment

1. Introduction

Modern ICS and CPS face a wide range of security threats. The feasibility and impact of each threat depend heavily on the capabilities of the adversary. It is thus essential to model the attacker's capabilities and danger level as part of threat analysis. The "danger" posed by an attacker is a function of factors such as the attacker's skills and resources ("competence"), available time, motivation, and the likelihood of successful exploitation of system vulnerabilities. By formally characterizing attacker capabilities and mapping them to potential system impacts, we can better prioritize security measures. This thesis develops an attacker classification and a quantitative danger model to support cybersecurity risk assessment for ICS/CPS environments.

2. Attacker Classification

We propose a comprehensive classification of attackers targeting ICS/CPS, including both insider and outsider categories. The classification defines attacker categories as follows [1]:

- ICS Insiders: including ICS Users (regular operators/users of the ICS or CPS), ICS Management (executives or administrators), ICS Staff (other internal employees), "At-Risk" Users (users in vulnerable positions or with risky access), Operational Personnel (engineering

Proceedings of the Workshop on Scientific and Practical Issues of Cybersecurity and Information Technology at the V international scientific and practical conference Information security and information technology (ISecIT 2025), June 09–11, 2025, Lutsk, Ukraine
* Corresponding author.
† These authors contributed equally.
oleksandr.umanskyi@cs.khpi.edu.ua (O. Umanskiy); korol.olha2016@gmail.com (O. Korol); o.kushnerov@biem.sumdu.edu.ua (O. Kushnerov); olaptiev@knu.ua (O. Laptiev); phd.sharapata@gmail.com (A. Sharapata)
0009-0006-7989-6285 (O. Umanskiy); 0000-0002-8733-9984 (O. Korol); 0000-0001-8253-5698 (O. Kushnerov); 0000-0002-4194-402X (O. Laptiev); 0000-0003-0823-9262 (A. Sharapata)

and control room staff), and Technical Support Personnel (maintenance and support staff). These are trusted insiders with varying levels of access.

- External Attackers: persons not employed by the ICS operator. These include Cyberterrorists, State Special Services (nation-state or intelligence actors), Hackers (skilled individual outsiders), Cybercriminals (organized cyber-crime groups), Competitors (industrial or corporate espionage agents), Organized Crime ("criminal" entities in the classification), and Vandals (opportunistic or hobbyist attackers).

Each attacker category is associated with certain capability levels. Insiders generally have legitimate access to the system but limited malicious resources, whereas external groups vary widely in resources and skills. For example, ICS users or staff may inadvertently or intentionally cause harm but typically lack advanced cyber skills, whereas cyberterrorists or state-sponsored actors possess extensive resources and expertise. This classification allows us to define a set of attacker categories {Hj}, and to map each category to the technical levels of impact on an ICS/CPS. The levels of impact correspond to layers of the system and network that an attacker can affect:

- H0: Technical channels (physical signal and wiring level)
- H1: Physical layer of the TCP/IP stack
- H2: Data link layer (TCP/IP)
- H3: Network layer (TCP/IP)
- H4: Transport layer (TCP/IP)
- H5: Malicious software (malware) level
- H6: Hardware backdoor (implanted device) level
- H7: Application layer (TCP/IP and software applications)
- H8: Information protection (security system) level

3. Attacker "Danger" Modelling

An attacker's danger level quantifies the risk they pose to a system. We define a formal model for attacker danger G_{CPS}^{ICS} that incorporates the attacker's category and capabilities. We express the attacker's danger as a function of their ability to carry out threats against system assets over time[2]. In particular, the model below considers the attacker's category i, their capability weights for ICS and CPS (β_i^{ICS} and β_i^{CPS}), the time T available for attack, the probability p_{rj} of realizing at least one threat to asset j, and the attacker's motivation r_{motiv}:

$$G_{CPS}^{ICS} = \left\{aid_i, \beta_i^{ICS} \in \left\{\beta_i^{ICS}\right\}, \beta_i^{CPS} \in \left\{\beta_i^{CPS}\right\}, p_{rj}, r_{\text{motiv}}, T\right\} \quad (1)$$

To instantiate this model, we must assign values to the various factors. We employ an expert evaluation approach to determine the weight coefficients and probabilities. Table 1 (below) summarizes the baseline quantitative values for each factor at five qualitative levels of attacker capability: critical, high, medium, low, and very low. At the critical level (the most dangerous attacker), all factors are set to 1 (indicating maximum threat likelihood, full motivation, daily attack frequency, and effectively unlimited resources). At the very low level (minimal attacker capability),

factors are as low as 0.001 (indicating negligible probability or resources). Intermediate levels (high, medium, low) are assigned scaled values (0.75, 0.5, 0.25) for each factor. These values serve as initial criteria for weighting an attacker's danger. For example, a "high"-capability attacker would have $p_{rj} = 0.75$ (a 75% chance to realize a given threat), $r_{\text{motiv}} = 0.75$ (high motivation), and so on. By using these baseline values, experts can estimate the weight coefficients β_i^{ICS} and β_i^{CPS} for each attacker category. In practice, an attacker category like Cyberterrorist might be rated "critical" on all factors, whereas an insider staff might be rated "low" or "medium" on technical resources and motivation (e.g. 0.25–0.5 range).

Table 1
Criteria and baseline values for expert evaluation of the attacker "danger" weight coefficient

Category	$\beta_i^{ICS} \in \{\beta_i^{ICS}\}$			$\beta_i^{CPS} \in \{\beta_i^{CPS}\}$				
	W_{cp}^{ICS}	T^{ICS}	W_{cash}^{ICS}	W_{cp}^{CPS}	T^{CPS}	W_{cash}^{CPS}	p_{rj}	r_{motiv}
Critical	1	1	1	1	1	1	1	1
High	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Medium	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
Low	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25
Very low	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001

We next define the attacker danger weight coefficient γ_{ICS}^{CPS} more explicitly. This coefficient combines the attacker's resource factors and time factors for both ICS and CPS contexts. We propose the following formula for the weight coefficient:

$$\gamma_{ICS}^{CPS} = \frac{1}{N}\sum_{i=1}^{N} \gamma_{ICS}^{CPS}{}_{i}, \tag{2}$$

$$\gamma_{ICS}^{CPS}{}_{i} = \left(\beta_i^{ICS} \cup \beta_i^{CPS}\right) \times p_{rj} \times r_{motiv}\,, \tag{3}$$

where $\beta_i^{ICS} = W_{cp}^{ICS} \cap W_{cash}^{ICS} \cap T^{ICS}$, and $\beta_i^{CPS} = W_{cp}^{CPS} \cap W_{cash}^{CPS} \cap T^{CPS}$ are the weight coefficients representing the attacker's effectiveness in the ICS domain and CPS domain respectively. These factors are assigned values based on attacker category: for instance, 1 for cyberterrorists (unlimited computing resources), 0.75 for state-sponsored attackers, 0.5$ for cybercriminal groups, 0.25 for ordinary criminals, competitors, or hackers, and 0.001for unsophisticated vandals. Similarly, the same scaled values for these categories. $T^{ICS}$and T^{CPS} denote the time windows required for a successful attack on ICS and CPS targets, respectively. We categorize time availability on a scale where 1 corresponds to attacks feasible within a day, 0.75 within a week, 0.5 within a month, 0.25 within a year, and 0.001 effectively unlimited time (for extremely slow, long-term attacks). Thus, Equation above accumulates the attacker's weighted capabilities in both the ICS and CPS realms. The coefficient γ_{ICS}^{CPS} is higher for attackers with greater resources and who can execute faster attacks, and it reflects the combined hazard posed to both ICS and CPS components of a system. These weight coefficients $G_{CPS}^{ICS}$are used in equation 1 to calculate the overall danger level.

4. Mapping Attacker Categories to Impact Levels.

Using the attacker classification and danger model, we can construct a mapping between attacker categories and the levels of impact {H_j}, on the system. Table 2 (below) presents this mapping. Each row corresponds to an attacker category, and each column H_0 – H_8 corresponds to a level of impact (defined in the Attacker Classification section). A value of "1" in the table means that attackers of that category are generally capable of executing attacks at that impact level, whereas "0" means they typically cannot. This assessment is based on the technical knowledge, access, and resources associated with each category.

Table 2
Mapping of attacker categories to levels of impact H0 - H8 on ICS/CPS infrastructure. A value of 1 indicates the attacker category can operate at that level; 0 indicates inability to significantly attack that layer

Category	Level of impact							
	H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7
ICS users	0	0	0	0	0	0	0	1
ICS management	1	1	0	0	0	0	1	1
ICS staff	0	0	0	0	0	0	0	1
At-risk users	0	0	0	0	0	0	0	1
Operational staff	1	1	1	1	1	0	1	0
Support staff	0	0	0	0	0	0	1	1
Cyberterrorists	1	1	1	1	0	1	1	0
State actors	1	1	1	1	1	1	1	1
Hackers	1	1	1	1	1	1	1	1
Cybercriminals	1	1	1	1	1	1	1	1
Competitors	1	1	1	1	0	1	1	0
Organized crime	1	1	1	1	1	1	1	0
Vandals	1	1	1	1	0	1	1	0

From Table 2, we observe clear distinctions between attacker types. For instance, ICS insiders such as regular users, staff, or at-risk users have limited impact—primarily at the application level H_7 and possibly affecting the information security systems H_8 through misuse of their access (indicated by "1" in columns H_7 and H_8, and "0" in lower-level columns for those categories). Management insiders show capability at physical and network-protocol levels (H_0, H_1) due to their broader access and authority, as well as at higher levels (H_6 – H_8) via directing insider actions (hence "1" for H_0, H_1, H_6 – H_8 for ICS Management). Operational staff (engineers/operators) can affect almost all ICS levels (H_0 – H_4 and H_6) but may not launch malware attacks (H_5) or directly manipulate security systems (H_8) without help. Technical support staff may install unauthorized devices (H_6) or misuse applications (H_7) but otherwise have little capability (zeros on most other levels).

External attackers, on the other hand, can often reach deeper into the tack. Cyberterrorists and state-sponsored actors are assessed as capable of attacks on all levels (1's across H_0 – H_8) — they have the expertise and resources to target everything from physical sabotage to advanced cyber intrusion. Hackers (skilled outsiders) can attack most network and software levels but might not easily breach dedicated security hardware or procedures (we note a 0 at H_8 for Hackers). Organized cybercriminal groups, competitors, and organized crime have substantial capabilities (many 1's), especially in network and software domains, but might lack the highest-level insider access or physical access in some cases (reflected by some 0's, e.g. at H_4 or H_7/ H_8 for those categories). Vandals are very limited, mainly capable of low-level physical disruption (H_0) or minor malware (H_5) vandalism, but not advanced attacks (hence 1's in only a couple of columns for Vandals). This categorical mapping helps an analyst infer likely attacker types from the nature of observed attacks: for example, an incident involving sophisticated malware and backdoor devices at multiple levels (H_5, H_6, H_7) would point to a high-capability external attacker (e.g. state actor or cybercriminal), whereas an attack confined to misuse of an HMI application (H_7) might point to an insider.

5. Attacker Category Determination Methodology

Given an unknown attacker and a set of detected threats, we can determine the most likely attacker category using the above classification and mapping. We propose a methodology that reduces attacker identification to an algorithm with the following steps:

1. Select Key Impact Features: Identify the characteristics of the attack in terms of impact levels H. Determine which levels in $\{ H_0,..., H_8 \}$ have been affected by the attacker (e.g. if the attack involved network traffic manipulation, H_3 and H_4 might be marked). These impacted levels serve as the classification features.

2. Determine the Threat Tuple: Based on the observed attack, form a tuple of realized threats across the different security dimensions (confidentiality, integrity, availability, etc.) according to a threat classifier. Essentially, enumerate which types of threats occurred (denial of service, data breach, sabotage, etc.).

3. Construct the Impact Vector: Using the threat tuple and a predefined set of critical threats (those with highest severity), construct a binary *impact vector* that indicates which critical threats have been realized for each asset j. This can be based on an evaluation of the product of importance coefficients of the attack and attacker (ensuring critical threats are weighted).

4. Identify Maximum Category: Starting from the lowest attacker category, compare the impact vector to the expected capabilities (Table 2) for each category. The first category whose capabilities fully explain the observed impact vector (or the highest category reached by the attack features) is identified as the maximum likely attacker category for this incident. This is done in increasing order of attacker "danger," ensuring we pick the smallest category that can account for all aspects of the attack.

Using this algorithm, we generate a list of critical threats relevant to the identified attacker category. For example, if the maximum category determined is Category 6: External Cybercriminal, we then focus on the critical threats that such attackers are known to pose and ensure those are mitigated. Furthermore, if we can eliminate certain attacker types (e.g. through insider background checks or external threat intelligence), we can reduce the maximum attacker category considered and thereby reduce the number of critical threats to address. This helps prioritize defence measures against the most probable attackers.

To automate the process of analyzing and categorizing text descriptions of threats, a neural network-based model was proposed. The basis was taken from the existing cybersecurity classifier [3], which is based on the synergistic threat model and provides extensive functionality for expert assessment, determination of synergy, hybridity and probability of threat impact. The key goal of the work was to create a system capable of classifying threats described by the user in arbitrary text form, according to eight defined categories: criticality level, security status, security services, nature of targeting, OSI level, social engineering threats, contour and category of the infrastructure object, returning the corresponding tuple of classes, without the need for prior standardization. It is assumed that the use of machine learning methods in combination with modern approaches to natural language processing will allow building an effective tool for assessing cyber threats, which will increase the practical value of the system for organizations that seek to respond quickly to risks and form priorities in protective measures. In addition to implementing the classification algorithm, the process of data preparation is valuable, especially in conditions of limited data quantity, which allows testing modern data science techniques for this purpose, complementing the existing cybersecurity classifier [3].

The initial data was an analysis table of 220 banking sector threats obtained from the cybersecurity classifier [3], which contained text descriptions of threats and corresponding eight-component class tuples, as well as coefficients and ratings of security services. Initially, the data were grouped by threat tuple, so it was necessary to regroup them by description, because the task was to group them by it. The set did not contain empty or corrupted values, but the threats did not necessarily contain all classes of each of the components of the tuple. Given the limited volume of the initial set, 220 unique threats, for each of which an "original" column was added, which is

insufficient for training effective machine learning models capable of generalizing on unknown examples, were applied using the data augmentation technique [4]. This process involves artificially increasing the training set by creating modified copies of existing data. Paraphrasing and synonymy techniques were used to increase and diversify the training sample, in particular with the involvement of models such as “bart-paraphrase”, “paraphrase-MiniLM”, “Rephrase” from the Hugging Face platform [5]. The technique of translation through intermediate languages (first into any language, and then back, sometimes several times or through different languages and translators), implemented using the “deep_translator” library, turned out to be less effective due to the specificity of the subject area. Threat descriptions often use professionalisms and abbreviations, and sometimes are already exhaustive, which made it difficult to obtain natural and meaningful translation results, since professionalisms often do not have synonyms even in other languages or are borrowings. In contrast, the paraphrasing technique, which uses synonymy at the level of word combinations or sentences together with a variation of grammatical forms, gave much better results. After manual analysis and cleaning of augmented data, removal of duplicates, correction of descriptions, enclosing descriptions with commas in quotes for correct reading of CSV, the final dataset consisted of 1078 threat descriptions, where each original threat was represented by 3-5 parallel formulations. Working with tabular data was carried out using the Pandas library.

The next step was to develop a classification model. An environment was prepared using Jupyter and key Python libraries [6–8], such as pandas for data processing, sklearn [6] for building machine learning models, numpy for numerical calculations, matplotlib for visualization, re for working with regular expressions, and hashlib and functools for optimization through caching. The training and test data were split using the “original” column to ensure that all 220 unique threats in the test sample were represented by one of their descriptions, with 17–25% of the data being test, while the remaining descriptions were used for training, ensuring that all original threats were covered. The categorical features of the target variables were encoded in a numeric format (from 0 to n-1), and an uneven distribution of the data across classes was found, which could affect accuracy, as the model would not be able to learn to assign a threat to a class it had not seen.

For classification, a data processing pipeline was created, which included three main components.

Figure 1: Model structure

The first component, TextCleaner, was responsible for preprocessing text descriptions: normalization (conversion to lower case, removal of digits, special characters, and punctuation) and lemmatization – bringing words to their original dictionary form. For lemmatization of Ukrainian-language texts, the Stanza NLP library [7] developed by Stanford University was used, which uses neural network approaches for morphological analysis, identification of parts of speech, and construction of dependencies between words, ensuring accurate lemmatization. To optimize performance, a two-level caching system for lemmatization results was implemented (on disk in the lemma_cache.pkl file and in RAM). The second component of the pipeline was the TF-IDF vectorizer from Scikit-learn [6], which converted texts into numerical vectors, where each document is represented by a set of numbers reflecting the weight of each word. TF, term frequency, shows how often a word occurs in a document, and IDF, inverse document frequency, reduces the weight of common words and strengthens unique ones. The vectorizer also used a list of Ukrainian stop words

to remove terms that did not carry a significant semantic load for classification. The third component was the MultiOutputClassifier from Scikit-learn [6], which used RandomForestClassifier as the base algorithm for simultaneous prediction of eight independent classes, chosen due to its resistance to overfitting and good compatibility with TF-IDF. Model training and hyperparameter optimization were performed using GridSearchCV from Scikit-learn [6], which uses cross-validation to find the best combination of parameters to maximize accuracy. The variation of parameters was manually sorted, and only the best parameters were retained in the final version. The weight of the keywords “threat”, “danger”, “risk” was also manually changed to reduce their impact on classification, since they are present in almost every description.

The trained model was evaluated using standard classification metrics such as accuracy, completeness, and F1-measure, as presented in the classification reports. Analysis of the results revealed high accuracy on the training data, but also indicated signs of overfitting and the impact of class imbalance on quality for individual categories. High precision with significantly different response also indicated overfitting. The uncertainty matrices and learning curves further confirmed these observations.

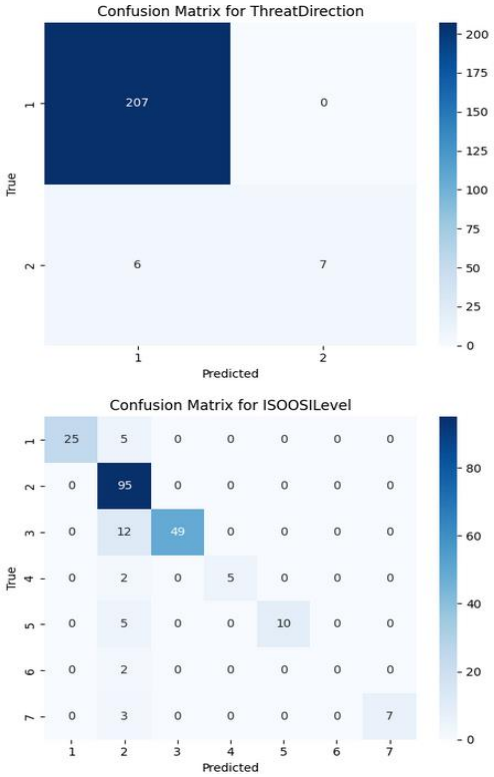

Figure 2: Deviation matrices

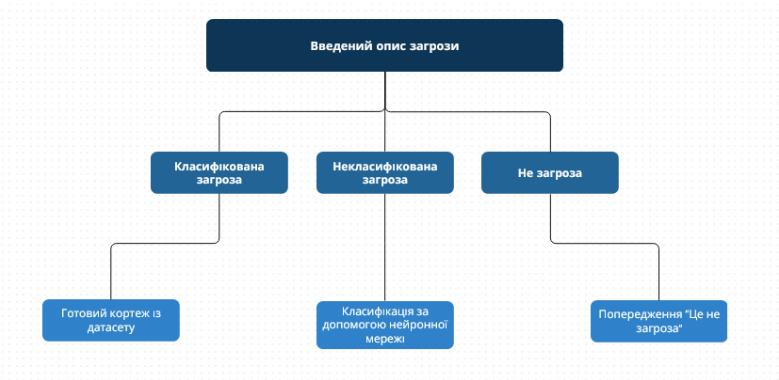

Figure 3: Classifier operation diagram

The learning curves showed that the training accuracy is high from the beginning (indicating insufficient diversity of the training data), while the accuracy on cross-validation is significantly lower, although it tends to increase, indicating that the model may not have enough data to correctly classify at the beginning, but the situation is improving. This indicates that the most likely option to improve the accuracy of the model is to increase and diversify the dataset, in particular by introducing new expert assessments, since the specificity of the threat description vocabulary limits the effectiveness of simple synonymy.

At the final stage, an integrated threat classification system was implemented in the form of the ThreatClassifier class. This system includes a preliminary stage of analyzing the description entered by the user for similarity with the threat descriptions already available in the database using the cosine similarity of vector representations obtained after identical data processing as for the machine learning model. First, the stored serialized objects are loaded: the trained model, encoding, dataframes. Then the lemmatizer and TF-IDF vectorizer process the input data. The vector of the entered description is compared with all the original vectors, and the similarity fraction is compared with the thresholds for categorization: classified threat, unclassified (requires model processing) or "not a threat".

This approach avoids overburdening the model for known threats and identifies irrelevant descriptions. For user interaction, a simple console interface was developed to demonstrate the system: the ThreatClassifier object performs tasks through the predict_threat() function, which allows for modular implementation.

In the course of the research, a cyber threat classification system was successfully developed and implemented, which effectively uses natural language processing methods and machine learning algorithms. The basis for training was data obtained from an existing threat classifier, which contained a previously conducted expert assessment of descriptions. The first and essential stage was thorough data preprocessing, which is a fundamental component of any data science project. A number of methods were applied to prepare the source data for effective use in the process of training the model. Faced with the key problem of the limited volume of the initial data set, data augmentation techniques were investigated and successfully tested, in particular paraphrasing using models from the Hugging Face platform, which allowed to significantly increase the volume and representativeness of the training sample. After that, the augmented data underwent a detailed review and cleaning stage from redundant or irrelevant content, thus ensuring the quality of the input information for the model.

6. Conclusion

We have developed a structured approach to quantify attacker risk ("danger") by integrating a detailed attacker classification with a formal danger modelling. By assigning weight coefficients to attacker capabilities and mapping attacker categories to technical impact levels, security practitioners can evaluate the threat level more systematically. The attacker classification model allows the formation of tailored threat sets depending on attacker capability. The formal danger model (Equations 1-3) provides a quantitative estimate of risk based on measurable criteria (resources, time, probability, motivation), grounded in expert-defined values (Table 1). The mapping in Table 2 and the identification algorithm offer a practical method to infer the likely attacker profile behind observed cyber incidents and to focus on mitigating the most critical threats for that attacker category. This approach enhances threat analysis for ICS and CPS by linking technical security events to adversary models, ultimately improving preventive defence by anticipating attacker behaviour and capabilities.

In the course of the research, a cyber threat classification system was successfully developed and implemented, which effectively uses natural language processing methods and machine learning

algorithms. The basis for training was data obtained from an existing threat classifier, which contained a previously conducted expert assessment of descriptions. The first and essential stage was thorough data preprocessing, which is a fundamental component of any data science project. A number of methods were applied to prepare the source data for effective use in the process of training the model. Faced with the key problem of the limited volume of the initial data set, data augmentation techniques were investigated and successfully tested, in particular paraphrasing using models from the Hugging Face platform, which allowed to significantly increase the volume and representativeness of the training sample. After that, the augmented data underwent a detailed review and cleaning stage from redundant or irrelevant content, thus ensuring the quality of the input information for the model.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] O. Shmatko, S. Balakireva, A. Vlasov, N. Zagorodna, O. Korol, O. Milov, O. Petrov, S. Pohasii, K. Rzayev, and V. Khvostenko, "Development of methodological foundations for designing a classifier of threats to cyberphysical systems," Eastern-European Journal of Enterprise Technologies, vol. 3, no. 9 (105), pp. 6–19, 2020, doi: 10.15587/1729-4061.2020.205702.

[2] S. Yevseiev, M. Karpinski, O. Shmatko, N. Romashchenko, T. Gancarczyk, and P. Falat, "Methodology of the Cyber Security Threats Risk Assessment Based on the Fuzzy-Multiple Approach," in Proc. 19th Int. Multidiscip. Sci. GeoConf. SGEM, Albena, Bulgaria, 2019, vol. 19, issue 2.1, pp. 445–452, doi: 10.5593/sgem2019/2.1/S07.057.

[3] Cybersecurity Classifier [Electronic resource]: https://skl.khpi.edu.ua/threat-analysis

[4] What is data augmentation? [Electronic resource]:https://www.ibm.com/think/topics/data-augmentation

[5] Hugging Face [Electronic resource]:https://huggingface.co/

[6] Scikit-learn: Machine Learning in Python [Electronic resource]:https://scikit-learn.org/stable/index.html

[7] Stanza: A Python Natural Language Processing Toolkit for Many Human Languages [Electronic resource]:https://arxiv.org/abs/2003.07082

[8] Python Documentation [Electronic resource]:https://docs.python.org