Combining Rule-based System and Machine Learning to Classify Unsupervised Semi-natural Language Data

Zafar Hussain, Jukka K. Nurminen, Tommi Mikkonen

Department of Computer Science, University of Helsinki

Introduction: In this research, we have studied command-lines from the perspective of cyber security. Computer vulnerabilities can be exploited in a variety of ways. Malicious actors may use a specific exploit, a secret pathway to enter a computer system, or a misconfiguration in one of the system components. In most of these attacks, malicious actors aim to run malicious programs through command-lines. One way to detect malicious activities on a machine is by analyzing the structure of command-lines. The detection can be based on a combination of different methods from rule engines to more advanced machine learning methods. These methods compare a new command-line to existing ones and classify it as similar or not-similar to any existing groups of command-lines. This helps` in creating clusters of similar command-lines and identifying them as safe or malicious. As rule-based and Machine Learning (ML) approaches have different, distinct strengths, an attractive option is to use their combination . To classify the command-lines, we study a neuro-symbolic[1, 2, 3] (hybrid) approach combining a rule-based system with an ML system

Key Observations:

- Use a rule-based system to convey expert knowledge to train an ML system for unsupervised classification
- Apply the proposed neuro-symbolic approach to classify semi-natural language data of command-line commands in the perspective of cyber security
- Implementation of the approach and measurement of its performance with different NLP models

Architecture: The overall architecture of the workflow is shown in Fig 1. First, we visualized the unsupervised command-lines in a graph structure to understand their hierarchy. Then the set of rules are used to create a rule-based system. Using these rules, we compared pairs of command-lines one by one, to classify them as similar or not-similar. This classification results in the supervised data, to be used for the ML system. In the next step, we study three different ML systems. In the last step, experts take a sample of an ML system's output and detect wrongly classified command-lines. These commands will be classified correctly according to the expert opinions and will be fed back to the ML system for re-training.

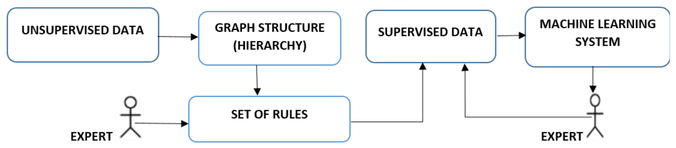

Figure 1: Step-wise Overview of the Workflow

Data Analysis: We collected data from one machine (Windows 10 OS). This data consists of the command-line commands used on the machine, and processes invoked by the system. The processes include opening a browser, creating a document, launching tools, and many more GUI operations. A total of 12262 commands were collected. Each command is fragmented into chunks of tokens. Each token is assigned a label such as COMM, SUBCOMM, FLAG, PARAM, etc. To understand the hierarchy of these tokens, we used graph approach. The tokens and labels are assigned as Nodes and edges respectively. Each token is connected to its parent token with its specific label. Following are the two command-lines which are compared against each other and a graph is drawn . as shown in Figure 1.

- C:\\WINDOWS\\system32\\svchost.exe –k LocalService –p –s WebClient
- C:\\WINDOWS\\system32\\svchost.exe -k LocalSystemNetworkRestricted –p –s SysMain

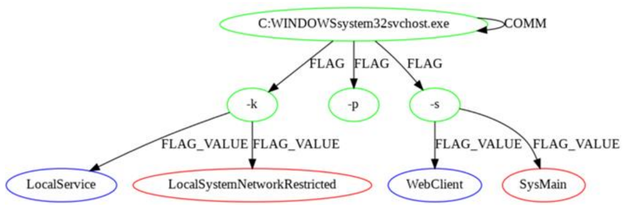

Figure 2: Graph representation of two commands

Methods: First, we built a rule-based system with the help of expert opinions and a set of rules. The expert opinions are used to create a reference The reference table consists of five columns, COMM, SUBCOMM, FLAG, PARAM, and OUTPUT as shown in Table 1. The first four columns can have one of the three possible values. Value 0 indicates that tokens are mismatched, whereas 1 shows a match. In the absence of a token, the value is set to be -1. The column OUTPUT can have one of the two possible values, similar, or not-similar, depending on the other four columns' values.

Table 1 Reference Table Sample

COMM	SUBCOMM	FLAG	PARAM	OUTPUT
1	1	1	0	*similar*
1	0	–1	1	*not-similar*
1	1	–1	1	*similar*
1	-1	0	–1	*not-similar*

We compared each command with its next 10 commands (creating a data-set that looks the same), and with 100 random commands (to diversify the data-set. Applying this algorithm on 12262 commands, over 1.209 million records of the supervised form were created. This supervised data is used for the machine learning system.

As a next step of the hybrid approach, we experimented with three machine learning models, a classical document classifier (logistic regression model), a DL document classifier using transformer, and a DL sentence-pair classifier using transformer.

Results: Out of 1.2 million records, a mere 100,000 records were used for the training and evaluation of the models. Table 2 shows the evaluation results of the three models. The baseline model wrongly predicted 1115 records as not-similar, whereas DL document classifier wrongly predicted 44 records as not-similar. The best result is achieved by the sentence-pair classifier, which wrongly predicted only 6 records as not-similar out of 20,000.

Table 2: Machine Learning Models' Comparison

Model	No. of Records	True Negative	True Positive	False Negative	False Positive	MCC
Baseline-Logistic Regression	20,000	14028	4800	1115	61	0.859
DL Document Classification	20,000,	14072	5784	44	100	0.982
DL Sentence Pair Classification	20,000	14262	5721	6	36	0.994

For the remaining 1.1 million records, 1000 random numbers were generated and used as indices to select the pair of commands for evaluation. We executed this experiment 100 times. The results in Figure 3 show the number of the wrong prediction out of 1000.

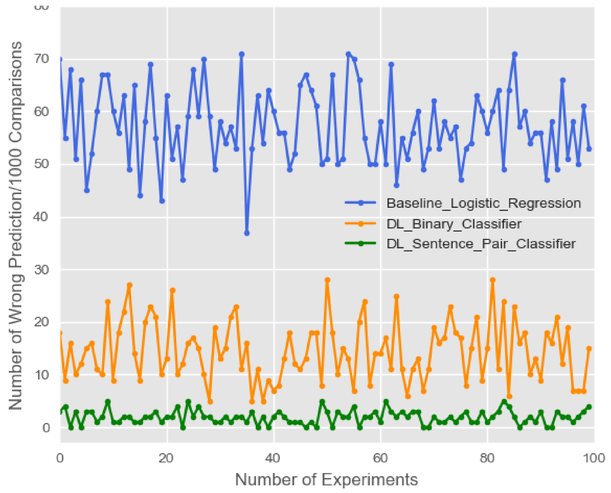

Figure 3: Comparison of Baseline model and DL Models

To verify the generalization of models, we tested the three models against unseen data. We selected 75 commands randomly from Stack Overflow5 and made combinations of each command with the other 74 commands. This gave us a total of 2640 pairs of unseen commands for testing.

Table 3: Comparison of the three models' performance on unseen data

Models	Accuracy
Baseline Logistic Regression	0.576
DL Document Classifier	0.943
DL Sentence Pair Classifier	0.983

Conclusions: We studied a hybrid approach of a rule-based system and machine learning system to classify command-line commands. Since the commands are in unsupervised form, the rule-based system transformed them into supervised data. This supervised data is used for the ML systems to learn the set of rules and classify the commands into similar or not-similar classes. The results achieved with the hybrid approach not only solve this complex problem but can be also replicated with a domain-specific set of rules for any semi-natural language unsupervised data.

References:

[1] Dean A. Pomerleau, Jay Gowdy, and Charles E. Thorpe. 1991. Combining artificial neural networks and symbolic processing for autonomous robot guidance. Engineering Applications of Artificial Intelligence 4, 4 (1991), 279–285

[2] Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, and Pascal Hitzler. 2021. Neuro-Symbolic Artificial Intelligence: Current Trends. arXiv:2105.05330 [cs] (May 2021). http://arxiv.org/abs/2105.05330 arXiv: 2105.05330.

[3] Julio Villena-Román, Sonia Collada-Pérez, Sara Lana-Serrano, and J. González. 2011. Hybrid Approach Combining Machine Learning and a Rule-Based Expert System for Text Categorization. FLAIRS Conference (2011).