# Development of a Hybrid LLM Agent Using Association Rules and the FP-Growth Algorithm to Predict MITRE ATT&CK Techniques

Konstantin D. Gorbunov, Sergei E. Ivanov

***Abstract*—This paper presents an algorithm that automates penetration testing of information systems through the introduction of an LLM-based agent. The algorithm constructs an attack vector at the level of techniques that adversaries may execute, expressed in the MITRE ATT&CK framework notation. The algorithm's accuracy is improved by incorporating information about related attacker techniques and by adding context about the target information system. Relationships between techniques were derived using association rules and the FP-Growth algorithm based on a dataset containing real-world cyberattack scenarios.**

***Keywords*—LLM, cybersecurity, penetration testing, FP-Growth, association rules.**

## I. Introduction

In the era of large-scale digitalization, organizations of all sizes, ranging from small enterprises to large businesses with distributed branches, increasingly migrate business processes online to automate workflows and optimize resource use. E-commerce platforms, warehouse management software, CRM systems for recruitment, and specialized applications for managing construction equipment are just a few examples of domains undergoing automation.

However, the growth of automation inevitably expands the number of potential vulnerabilities. Modern cyberattacks rarely occur as isolated events. They increasingly represent a carefully prepared sequence of attacker actions that can unfold over hours or even years [1]. During this time, adversaries systematically increase their privileges within an information system from reconnaissance and collecting details about architecture, operating system versions and installed modules, through initial intrusion and finally to privilege escalation.

Core challenge is that the main damage from a cyberattack often becomes apparent only at its final stage, while reconstructing the full chain of attacker actions is frequently not feasible [2]. At the same time, the business impact of cyberattacks can be critical, including financial losses, suspension of sales and partner interactions, and reputational risks.

In such conditions, the cybersecurity posture of information systems and especially preventive measures plays a decisive role. Penetration testing is one such measure, it imitates adversarial behavior to identify weak points and potential vulnerabilities [3]. In a typical pentest, security specialists build a test environment for the target system and, using specialized tools and scanners, attempt to discover exposed ports, APIs and directories that could enable a malicious scenario. The downside of this approach is the required expert time. For example, a standard website with modules such as web forms, self-registration, partner download center and news section can take a security team up to a month to test, followed by another month to fix and re-verify the issues. While companies are interested in rapid releases, an IBM study from 2009 indicates that the cost of an unfixed defect at deployment is 100 times higher than fixing it during testing [4].

Nevertheless, parts of the pentesting process can be automated by integrating LLMs. A recent study by XBOW shows that modern LLMs, when used within an agent platform, are increasingly effective at finding vulnerabilities compared to classical brute-force approaches [5]. The model can plan cyberattacks, select and run necessary tools, interpret results and adjust the vulnerability analysis direction. Through an iterative loop including hypothesis, testing, validation and result adjustment, the model more quickly finds complex vulnerabilities and scales better to real-world targets. However, the model remains limited by its training dataset, while attack techniques are constantly evolving. Although modern LLMs support web search, a NewsGuard report indicates that after this capability was introduced, the share of incorrect answers increased from 18% to 35% over the past year. Among the reasons are excessive trust in external sources and the growing share of AI-generated content [6]. In that case, a hybrid approach, combining an LLM with relevant data provided alongside the prompt, becomes especially required [7].

The vulnerability discovery process can be discovered at several levels, from setting the search direction to issuing concrete commands that exploit a flaw. This structure is

Manuscript received October 10, 2025.

K. D. Gorbunov is with the ITMO University, 49 Kronverksky Pr., 197101 St. Petersburg, Russia (corresponding author to provide e-mail: 264892@niuitmo.ru).

S. E. Ivanov is with the ITMO University, 49 Kronverksky Pr., 197101 St. Petersburg, Russia and SPbGMTU, 3 Lotsmanskaya St., 190121 (e-mail: serg_ie@mail.ru).

described by the MITRE ATT&CK framework, which organizes cyber threats as a matrix of tactics (categories of techniques) and the techniques themselves used by adversaries [8]. Tactics can be seen as directions or threat vectors (reconnaissance, initial access, persistence), while techniques are the methods of realizing the threat. Each technique in the matrix is assigned an identifier beginning with T. For example, exploitation of client-side application vulnerabilities has the identifier T1203. This study focuses on predicting techniques that an adversary may execute in the target system.

The main goal of this study is to develop a hybrid LLM agent that uses association rules to predict MITRE ATT&CK techniques. To achieve this goal, the study addresses the following tasks:

1. Review literature on LLMs and methods for constructing association rules;
2. Build a dataset of cyber incidents for algorithm validation;
3. Define the research methodology;
4. Analyze algorithm effectiveness and refine model parameters to achieve the enhanced results;
5. Validate the obtained findings.

## II. Literature Review

### A. Overview of LLMs

The first component of the hybrid technique prediction algorithm is LLM. In its basic form, LLM is a question-answer model that can provide relevant, context-aware responses due to extensive training data and deep contextual understanding. While LLMs typically return text, it is also possible to design prompts so that the model outputs a structured response, such as a JSON schema, suitable for downstream execution [9]. In vulnerability discovery, a pentester provides the model with inputs about the target system - known architecture, installed modules and versions, available APIs, and asks it to propose a strategy for further exploration of exploits. Alongside this information, the model also receives a set of available tools (functions) to investigate potential weaknesses, such as fetching a URL or checking a port's reachability. LLM then returns structure indicating which function should be called to test a particular exploit. This structure is passed to, for example, an orchestrator for execution, and the results are fed back to the model [10]. Multiple such iterations may occur.

Modern LLMs differ by many characteristics, including parameters count, context window, domain specialization, and deployment mode (open-source models that can run locally, and cloud-hosted models accessed via API). For this study, local deployment is important for practical use (e.g., in SIEM systems), in part due to Federal Law No. 152 "On Personal Data," which regulates storage and processing of personal information, and due to business requirements for scalability, flexibility and integration with existing systems.

Given these constraints, the following LLMs were selected:

1. DeepSeek-R1. A family of LLMs optimized for reasoning and program synthesis, showing strong performance on logical inference and code-generation benchmarks with moderate compute cost, supported by effective post-training and careful data filtering [11];

2. Gemma 3. A line of compact and medium-sized LLMs from Google focused on safe fine-tuning and embedding in resource-constrained applications. The models provide balanced performance for reasoning and code generation;

3. Qwen3. A multilingual and multimodal LLMs family from Alibaba spanning a wide range of parameter sizes and supporting orchestration tools, characterized by high scalability and competitive accuracy in agent scenarios.

### B. Overview of Association Rule Mining Methods

To make the prediction of likely attacker techniques more accurate, the LLM is supplied with the set of currently observed techniques along with a list of related techniques obtained via association rules. This problem is analogous to the classical market basket task, which examines relationships among co-purchased items and returns results in the form "if a customer buys item A, they are likely to buy item B" [12]. With historical data on cyberattack scenarios against information systems, association rules can derive conclusions of the form "if an attacker executed technique A, then technique B is likely to be the next step."

The most common approaches to association rule mining include the Apriori algorithm and FP-Growth, the latter using a frequent-pattern tree to improve efficiency [13, 14]. The following sections discuss each method to identify key strengths and weaknesses.

### C. The Apriori Algorithm

Apriori relies on the anti-monotonicity property [15] and constructs patterns iteratively: first frequent 1-itemsets are identified, then larger candidate sets are generated and verified by repeated database scans; the process continues until no new frequent sets are found.

Strengths include implementation simplicity and broad applicability. Limitations are high computational cost due to multiple scans, exponential growth of resource consumption, and notable slowdown as the number of items increases.

### D. The FP-Growth Algorithm

FP-Growth uses a frequent-pattern tree structure, avoiding iterative candidate generation as in Apriori [16]. The database is scanned only twice: first to build a compact prefix tree with items ordered by frequency, and then to extract frequent item sets via recursive analysis, from which associations are derived.

Advantages include high speed (only two passes) and memory savings, which are crucial for scalability on large datasets. Drawbacks include more complex implementation due to recursive procedures, the need to hold the tree in memory, and weaker compression when item distributions are highly imbalanced.

### E. Comparison of Apriori and FP-Growth

Empirical results in [17] show that for large datasets, FP-Growth is generally more efficient than Apriori in finding frequent itemsets.

Key advantages of FP-Growth over Apriori:

1. Execution time: at support thresholds below 0.3, FP-Growth is typically 3–5 times faster;

2. Memory: candidate generation in Apriori uses about 2–3 times more memory than the tree representation in FP-Growth;

3. Practical suitability: Apriori is easier to adapt to narrow, specialized scenarios, while FP-Growth excels in common association-mining tasks.

Based on these findings, FP-Growth was selected for further research.

## III. Research Methodology

Hardware used in the study:

1. CPU: Intel(R) Core(TM) i5-10600 @ 3.30GHz;
2. RAM: 16 GB;
3. GPU: Intel(R) UHD Graphics 630 (128 MB).

The set of LLMs was chosen according to business and compute constraints and present in Table 1.

Table 1. Set of chosen LLMs and its parameters.

| Model | Parameters | Context | Model size |
|---|---|---|---|
| DeepSeek-R1 | 1.5B | 128k tokens | 1.1 GB |
| Gemma 3 | 1B | 32k tokens | 815 MB |
| Qwen3 | 1.7B | 40k tokens | 1.4 GB |

Validation metrics:

1. Recall: share of cases where the next technique appears in the top N recommendations;
2. MRR: metric that accounts for the position of the correct technique in the ranked list;
3. Coverage: share of techniques in the predicted list that occur next in the scenario;
4. Request execution time.

Association rules were built from a dataset containing 2,500 real-world scenarios of current cyberattacks across several industries:

1. Industrial enterprises;
2. Healthcare institutions;
3. Educational institutions;
4. Financial companies;
5. Government organizations;
6. IT and telecom companies;
7. Retail and e-commerce companies.

The dataset was compiled based on analysis of cyber threat reports. Each row represents a sequence of techniques executed by adversaries. To normalize the data, each technique was mapped to a MITRE ATT&CK identifier. The dataset structure is illustrated on Table 2.

Row 1 provides an example of phishing with a malicious attachment; row 2 shows gaining access to company RDP services; row 3 covers exploiting client software vulnerabilities; row 4 shows persistence and execution of malicious commands via internal services; row 5 shows network penetration for credential theft; row 6 shows a denial-of-service attack; row 7 shows data exfiltration via web protocols.

Table 2. Dataset structure.

| ID | Tech 1 | Tech 2 | Tech 3 | Tech N |
|---|---|---|---|---|
| 1 | T1566.001 | T1204.002 | T1059.003 | None |
| 2 | T1190 | T1133 | T1078 | T1021.001 |
| 3 | T1189 | T1203 | T1105 | T1055 |
| 4 | T1569.002 | T1543.003 | None | None |
| 5 | T1210 | T1021.002 | T1570 | T1003.001 |
| 6 | T1499 | T1498.001 | T1565.003 | None |
| 7 | T1041 | T1071.001 | None | None |

The dataset was split 75% and 25% into training and test sets. The training set was used to generate association rules via FP-Growth; the test set was used for validation.

## IV. Interim Results

The baseline scenario evaluated model performance without FP-Growth. The model input contained the current state of the cyberattack scenario, and the expected output was a top-5 list of next techniques. The prompt template was “You are a system for predicting MITRE ATT&CK techniques that may be executed in an information system. Given: Previously executed techniques (chronological): [list]. Current technique: [tech ID]. Task: determine the top 5 next techniques after the current technique. Order techniques by decreasing relevance. Response requirements: Return strictly JSON of the form {"predictions": ["Txxxx", ...]} of length exactly 5.”. Results are shown in Table 3.

Table 3. Interim results for chosen LLMs.

| Model | Recall | MRR | Coverage | Time, s |
|---|---|---|---|---|
| DeepSeek-R1 | 0.42 | 0.18 | 0.58 | 30.5 |
| Gemma 3 | 0.38 | 0.15 | 0.51 | 2.2 |
| Qwen3 | 0.45 | 0.22 | 0.60 | 15.7 |

Qwen3 leads on Recall and Coverage due to higher parameter count and larger context, which improves sequence ranking but increases latency. MRR is higher for Qwen3 and DeepSeek-R1 owing to better placement of the correct technique. Times include tokenization and CPU inference without a discrete GPU; Gemma 3 is fastest due to a smaller model and context.

## V. Integrating FP-Growth into the LLM Agent

In the basic FP-Growth setup, hyperparameters were:

1. Support = 0.5
2. Confidence = 0.5

However, the hyperparameters have been optimized to achieve the best metrics, as well as to adapt the algorithm to the context of modeling attack vectors. An adaptive loop is used to compute Support. The updated formula is shown in Figure 1. Here N is the dataset length, and coefficient is iteratively decreased from 1 to 0 by 0.01 until the number of FP-tree elements is less than 17,000.

$$Support = (N * coefficient) / 20$$

Fig. 1. Support parameter calculation formula.

Employing the Optuna hyperparameter optimization framework, count 17,000 was identified as the maximal FP-tree size that ensures association rule mining within a single category (economic sector) completes in no more than 1 second.

Using the Optuna framework, the divisor in support’s formula was swept from 0 to 100 with a step of 1 and the

resulting metric values were evaluated. Table 4 reports the best metric values obtained for different divisor settings. The results indicate that the optimized configuration (divisor = 20) yields the highest metric scores.

Table 4. Comparison of metrics with different dividers.

| Divider | Accuracy, % | Precision, % | Recall, % |
|---|---|---|---|
| 14 | 85.2 | 77.5 | 81.3 |
| 18 | 87.9 | 81.8 | 85 |
| 20 | 89 | 84.7 | 87.2 |
| 45 | 87.1 | 83.6 | 84.1 |
| 54 | 85.6 | 81.2 | 80.5 |

These hyperparameters significantly reduce compute resources, minimize dependence of runtime on dataset size, and improve prediction accuracy.

After applying FP-Growth to the training set, associative links among techniques in cyber incidents were obtained. Figure 2 shows a bidirectional graph for T1078 (Valid Accounts), including T1018 (Remote System Discovery), T1190 (Exploitation of Public-Facing Application), and T1021 (Remote Services).

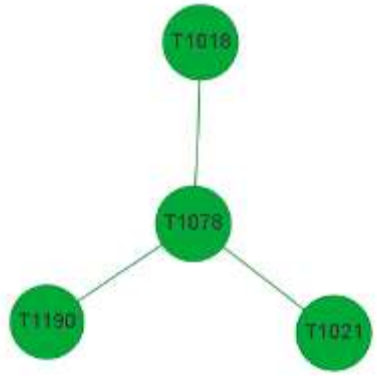


Fig 2. Techniques linked to T1078.

The model prompt was augmented with techniques related to the current ones, obtained via FP-Growth. Results are shown in Table 5.

Table 5. Results for chosen LLMs after integrating FP-Growth.

| Model | Recall | MRR | Coverage | Time, s |
|---|---|---|---|---|
| DeepSeek-R1 | 0.49 | 0.21 | 0.73 | 31.5 |
| Gemma 3 | 0.44 | 0.18 | 0.69 | 3.4 |
| Qwen3 | 0.51 | 0.25 | 0.78 | 18.2 |

Recall and MRR improved by 15–20% on average, while Coverage increased by 30%. Average request time also increased due to additional steps for finding and forming association rules.

## VI. Adding Target System Context to the Algorithm

Despite improved metrics, the model still did not account for features of the specific target system. As noted above, the dataset scenarios were grouped by industry. Industry information was added to the model request together with the current scenario and was also used when retrieving association rules. Results are shown in Table 6.

Table 6. Enhanced results after context augmentation.

| Model | Recall | MRR | Coverage | Time, s |
|---|---|---|---|---|
| DeepSeek-R1 | 0.61 | 0.27 | 0.87 | 31.7 |
| Gemma 3 | 0.55 | 0.24 | 0.82 | 3.8 |
| Qwen3 | 0.63 | 0.31 | 0.92 | 18.8 |

After adding industry information, Recall increased by up to 25%, MRR by 30%, and Coverage by 20%, while execution time remained almost unchanged.

## VII. Practical Application

To enable integration into existing cyber incident analytics systems, such as SIEM platforms, the solution was containerized. This isolates all modules and dependencies from the host OS. The solution can be further adapted, for example, by implementing a web service using FastAPI or a PowerShell tool that accepts command-line arguments.

## VIII. Conclusion

The following tasks were accomplished:

1. A literature review on LLMs and association rule construction methods was conducted;
2. A cyber incident dataset was compiled for algorithm validation;
3. A research methodology was defined;
4. Algorithm effectiveness was analyzed, and model parameters were refined to achieve better results;
5. The obtained results were validated.

The research core goal was achieved. We developed a hybrid LLM-agent algorithm that uses association rules to predict MITRE ATT&CK techniques, achieving over 60% accuracy for predicting the next technique in a cyberattack scenario, and over 90% probability that the predicted technique appears in the scenario. Among the models considered, Qwen3 (1.7B parameters) achieved the best quality metrics, while Gemma 3 (1.0B parameters) achieved the best latency.

## References

[1] A. O. Kalashnikov, E. V. Anikina, G. A. Ostapenko and V. I. Borisov, "The impact of new technologies on the information security of critical information infrastructure," *Information and Security*, Vol. 22. No. 2, 2019.

[2] S. Sarkar, "A Study on Cybersecurity Standards for Power Systems // Advanced Power System Standards and Practices," pp. 429–450, 2013, doi: 10.1007/978-3-031-20360-2_18.

[3] R. Beuran, "Cybersecurity Awareness Training," *Cutting-Edge Advances in Cybersecurity Education and Training*, pp. 153–170, 2025, doi: 10.1007/978-981-96-0555-2_8.

[4] How software testing saves billions. – Online resource. Available: https://tproger.ru/articles/ekonomim-milliardy-rol-testirovaniya-v-razrabotke-programmnogo-obespecheniya.

[5] XBOW Unleashes GPT-5's Hidden Hacking Power, Doubling Performance. Available: https://xbow.com/blog/gpt-5.

[6] Despite progress, neural networks more often produce fake answers. Available: https://www.gazeta.ru/tech/news/2025/09/06/26667170.shtml.

[7] S. V. Kuznetsov, D. A. Pelekhov and V. V. Novlyansky, "The role of artificial intelligence in detection and prevention of cyberattacks," *Science and Reality*, No. 2 (18), pp. 57–60, 2024.

[8] A. Sreejith and K. Swarup, "MITRE ATT&CK for Smart Grid Cyber-Security," *Smart Grid Security and Privacy*, pp. 59–73, 2024, doi: 10.1007/978-981-97-1302-8_5.

[9] D. E. Namiot and E. A. Ilyushin, "Architecture of LLM agents." International Journal of Open Information Technologies, Vol. 13, No. 1, pp. 64-74, 2025. Available: http://injoit.org/index.php/j1/article/view/2057.

[10] D. E. Namiot, "What LLM knows about cybersecurity," *International Journal of Open Information Technologies*, Vol. 13, No. 7, pp. 37-46, 2025. Available: http://injoit.org/index.php/j1/article/view/2214.

[11] A. V. Savkina, "Comparative analysis of free AI assistants: Poe, DeepSeek, GPT-3.5," *Vestnik Nauki i Obrazovaniya*, No. 7-2(162), pp. 15-19, 2025, doi: 10.24411/2312-8089-2025-10702.

[12] I. A. Olyanich, "Comparison of algorithms for constructing association rules based on a dataset of customer transactions," *Izvestia of Samara Scientific Center of the Russian Academy of Sciences*, No. 6-2, pp. 379-382, 2018. Available: https://www.ssc.smr.ru/media/journals/izvestia/2018/2018_6_379_382.pdf.

[13] H. Hery, A. Widjaja, "Analysis of Apriori and FP-Growth Algorithms for Market Basket Insights: A Case Study of The Bread Basket Bakery Sales," *Journal of Digital Market and Digital Currency*, No. 1, pp. 63–83, 2024, doi: 10.47738/jdmdc.v1i1.2.

[14] M. Rosadi and M. Hasibuan, "Comparison of Apriori and FP-Growth Algorithms in Analyzing Association Rules," *PIKSEL*, No. 12(2), pp. 399–408, 2024, doi: 10.33558/piksel.v12i2.9965.

[15] P. Majumdar, "Apriori Algorithm for Engineers," *Zenodo*, 2024, doi: 10.5281/zenodo.14566774.

[16] E. O. Khramshina and A. V. Prutzkow, "Association rules mining with three-dimensional data structure," *International Journal of Open Information Technologies*, Vol. 8, No. 8, pp. 8-12, 2020, Available: http://injoit.org/index.php/j1/article/view/972.

[17] S. Sarkar, S. Dey, S. Goswami, S. Bhunia, S. Mukhoty and S. Dutta, "Comparative Analysis of Performance in FP-Growth and Apriori Algorithm," *American Journal of Electronics & Communication*, No. 4, pp. 9–13, 2023, doi: 10.15864/ajec.4103.

K. D. Gorbunov is with the ITMO University, 49 Kronverksky Pr., 197101 St. Petersburg, Russia (corresponding author to provide e-mail: 264892@niuitmo.ru).

S. E. Ivanov is with the ITMO University, 49 Kronverksky Pr., 197101 St. Petersburg, Russia and SPbGMTU, 3 Lotsmanskaya St., 190121 (e-mail: serg_ie@mail.ru).