УДК 004.89+004.021+004.45
DOI: 10.26102/2310-6018/2026.56.5.013

Математическое и алгоритмическое обеспечение для автоматического обновления правил в миварной экспертной системе с использованием больших языковых моделей

Л. Доу✉

Московский государственный технический университет им. Н.Э. Баумана, Москва, Российская Федерация

Резюме. Целью работы является разработка математического и алгоритмического обеспечения для автоматического обновления правил в миварной экспертной системе с использованием больших языковых моделей. В ходе этого исследования было разработано математическое и алгоритмическое обеспечение для динамического и автоматического обновления правил. Обеспечение используется для решения проблемы запаздывания обновления правил базы знаний традиционной миварной экспертной системы и длительного времени ручного обновления. Математическое и алгоритмическое обеспечение в этом исследовании основано на способности генерировать большие языковые модели, включая четыре алгоритма. Триггерный алгоритм на основе оценки достоверности; алгоритм генерации подсказок; алгоритм безопасной верификации; алгоритм введения правил. Научная новизна работы заключается в разработке четырех алгоритмов, обеспечивающих автоматическое динамическое обновление правил в миварной экспертной системе с использованием больших языковых моделей. Эксперименты показывают, что математическое и алгоритмическое обеспечение, использованное в этом исследовании, может эффективно улучшить способность в миварной экспертной системы автономно обновлять правила. Было обработано 95,8 % неизвестных сценариев, а точность генерации правил достигла 91,3 %. Цикл обновления базы знаний сократился с нескольких часов до нескольких секунд. Данное исследование доказывает преимущества использования LLM в качестве внешнего интеллектуального сервиса для реализации автоматического обновления правил базы знаний миварной экспертной системы.

Ключевые слова: мивар, экспертная система, большая языковая модель, автоматическое обновление базы знаний, искусственный интеллект, алгоритм.

Для цитирования: Доу Л. Математическое и алгоритмическое обеспечение для автоматического обновления правил в миварной экспертной системе с использованием больших языковых моделей. *Моделирование, оптимизация и информационные технологии.* 2026;14(5). URL: https://moitvivt.ru/ru/journal/article?id=2271 DOI: 10.26102/2310-6018/2026.56.5.013

Mathematical and algorithmic support for automatic rule updates in a mivar expert system using large language models

L. Dou✉

Bauman Moscow State Technical University, Moscow, the Russian Federation

Abstract. The aim of the work is to develop mathematical and algorithmic software for automatic updating of rules in the mivar expert system using large language models. In the course of this research, mathematical and algorithmic software was developed for dynamic and automatic updating of rules. The software is used to solve the problem of the delay in updating the rules of the knowledge base of the traditional mivar expert system and the long time required for manual updating. The mathematical and algorithmic support in this study is based on the ability to generate large language models, including four algorithms. Trigger algorithm based on confidence estimation; algorithm for generating hints;

algorithm for secure verification; algorithm for introducing rules. The scientific novelty of the work consists in the development of four algorithms that provide automatic dynamic updating of rules in a global expert system using large language models. Experiments show that the mathematical and algorithmic software used in this study can effectively improve the mivar expert systems ability to update rules autonomously. 95.8 % of unknown scenarios were processed, and the accuracy of rule generation reached 91.3 %. The knowledge base update cycle has been reduced from several hours to several seconds. This study proves the advantages of using LLM as an external intelligent service for implementing automatic updating of the rules of the knowledge base of the mivar expert system.

Keywords: mivar, expert system, large language model, automatic updating of the knowledge base, artificial intelligence, algorithm.

For citation: Dou L. Mathematical and algorithmic support for automatic rule updates in a mivar expert system using large language models. *Modeling, Optimization and Information Technology.* 2026;14(5). (In Russ.). URL: https://moitvivt.ru/ru/journal/article?id=2271 DOI: 10.26102/2310-6018/2026.56.5.013

Введение

Экспертные системы являются ключевой областью искусственного интеллекта [1] и играют важную роль в диагностике неисправностей [2] и поддержке принятия решений [3]. Благодаря своей уникальной объектно-ориентированной модели представления знаний [4], миварная экспертная система (МЭС) обеспечивает эффективный механизм рассуждения в таких сценариях [5], как динамическое планирование траектории [6]. Однако, как и многие традиционные экспертные системы [7], МЭС сталкивается с проблемой «узкого места в знаниях» [8]. В ответ на динамически изменяющиеся условия [9] это может привести к жесткости системы и задержке [10], ограничивая их потенциал в современных приложениях [11].

В последние годы крупные языковые модели совершили прорыв в понимании естественного языка и логическом мышлении, что открыло новые возможности для автоматизированной разработки знаний [12]. Магистр права обладает способностью обрабатывать естественный язык, генерировать код и выполнять логические рассуждения [13]. Это помогает извлекать знания из неструктурированного текста для создания правил. Объединение поколения LLM с МЭС для разработки безопасного и надежного механизма автоматического обновления является нерешенной проблемой.

МЭС использует XML-подобный формат для формализации логических связей между классами, параметрами, правилами и взаимосвязями. Преимущества миварной базы знаний заключаются в высокой читабельности и простоте рассуждения. КЭСМИ Wi!Mi – это программное обеспечение, используемое для рассуждения и являющееся ядром МЭС. КЭСМИ Wi!Mi показано на Рисунке 1.

Хадиев А. М. изучил архитектуру, принцип работы и фактическую реализацию машины миварной логического мышления [14]. В исследовании Коценко А.А. проанализировал использование миварной сети в формате двухсторонних и трехсторонних диаграмм в автоматизированных системах управления технологическими процессами [15]. В 2024 году Гаврилов Л.Я. и др. обсуждали возможность создания глобальной базы знаний для крупномасштабных языковых моделей [16]. Данилюком А.В. и Кимом Р.И. в 2023 году было подробно освещено применение MivarGPT в области искусственного интеллекта машиностроения для автоматического создания миварной базы знаний [17]. Du S.Q. и др. предложили модульную экспертную систему, основанную на большой языковой модели Tree-GPT, предназначенную для понимания и интерактивного анализа изображений лесов, полученных с помощью дистанционного зондирования [18]. Lammert Ja. и др. в журнале Journal of Clinical Oncology изучали использование крупномасштабных языковых

моделей в прецизионной онкологии, уделяя особое внимание поддержке принятия клинических решений посредством обучения под руководством экспертов [19]. Schuerkamp R., Ahlstrom H. и Giabbanelli PH.J. предложили экспериментальный метод, который использует крупномасштабные языковые модели и нечеткие когнитивные карты из исследований совместного моделирования для автоматического разрешения конфликтов между экспертными системами [20]. Awasthi Y.K. и др. провели сравнительное исследование для оценки практического применения двух инструментов искусственного интеллекта: ChatGPT и DeepSeek [21].

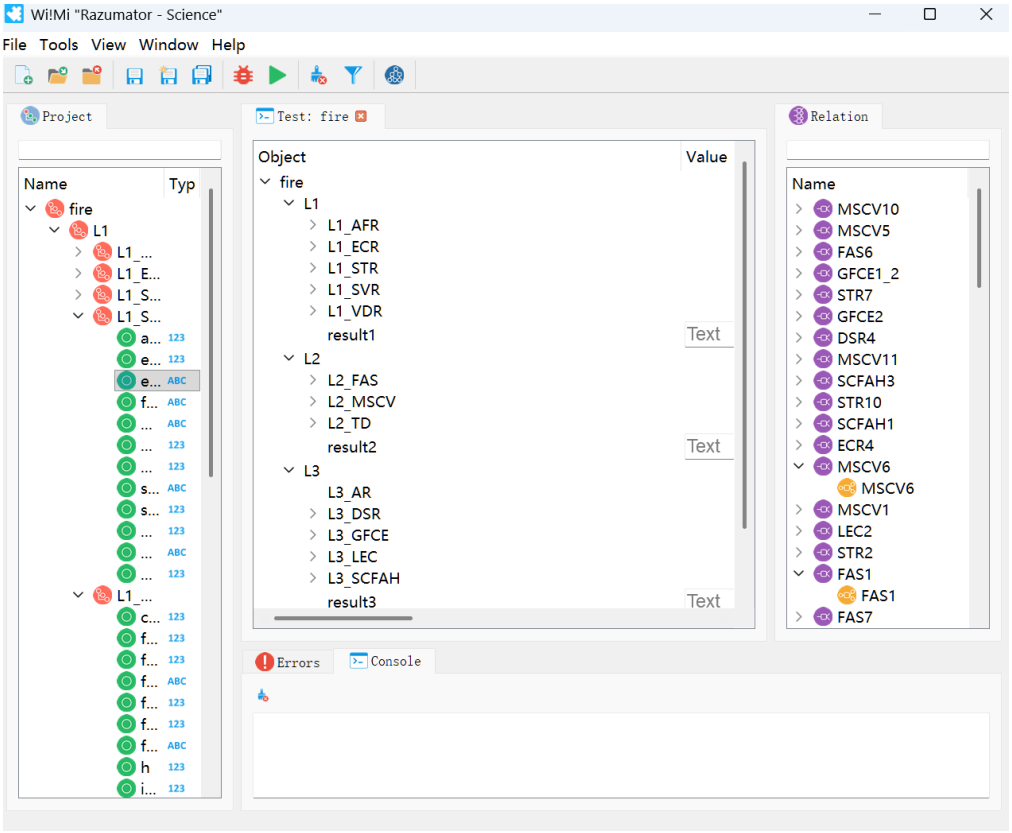

Рисунок 1 – Программное обеспечение КЭСМИ Wi!Mi
Figure 1 – Software KASMI Wi!Mi

Большая часть текущих исследований сосредоточена на создании псевдокода или простых правил для определенных скриптовых языков. Исследования представлений знаний со строгими грамматическими ограничениями по-прежнему недостаточны. Как обеспечить безопасную и точную интеграцию правил, сгенерированных LLM, с базой знаний МЭС, по-прежнему остается сложной проблемой.

Чтобы решить эту проблему, данное исследование направлено на фундаментальное преодоление этого ограничения. Его основной задачей является создание интегрированной системы математического и алгоритмического обеспечения. В этом исследовании используется крупномасштабная языковая модель в качестве внешнего интеллектуального сервиса для реализации автоматической эволюции знаний МЭС.

Научная новизна работы заключается в разработке четырех алгоритмов: триггерный алгоритм на основе оценки достоверности, алгоритм генерации подсказок, алгоритм безопасной верификации, алгоритм введения правил, обеспечивающих автоматическое динамическое обновление правил в МЭС с использованием больших

языковых моделей. Эти четыре алгоритма могут значительно повысить адаптивность и уровень интеллектуальности МЭС, позволяя ей более эффективно применяться в областях, где необходимо быстро реагировать на изменения окружающей среды. Это заложит прочную основу для разработки следующего поколения автономных интеллектуальных систем.

Материалы и методы

Создание предложенной в исследовании системы математического и алгоритмического обеспечения направлено на модель [6] для динамического обновления правил в МЭС. Основу данного обеспечения реализации метода [22] составляет автоматизированный замкнутый цикл, охватывающий весь процесс «восприятие-генерация-верификация-эволюция» и построенный на алгоритмах.

Основной процесс заключается в следующем: алгоритм запуска, основанный на оценке уверенности и ситуационной осведомленности, постоянно отслеживает работу системы. Когда он обнаружит, что база знаний не может охватить текущую сцену, он запустит цикл обновления и вызовет алгоритм генерации запроса. Алгоритм генерации подсказок отвечает за построение инструкций, которые точно управляют большой языковой моделью. LLM разрабатывает предварительные правила. Алгоритм проверки безопасности отвечает за анализ выходных данных LLM и выполнение многоуровневой логической проверки. Если правило будет проверено, оно будет добавлено в базу знаний.

Общая структура математического и алгоритмического обеспечения показана на Рисунке 2.

Рисунок 2 – Общая структура обеспечения
Figure 2 – The general structure of algorithmic support

Если мы определим динамическое обновление миварной базы знаний как процесс изменения состояния, мы можем предположить, что состояние системы в момент времени t равно KB_t. Когда система обнаруживает неизвестный входной параметр S, механизм обновления генерирует новый набор правил Δ.Таким образом, база знаний перешла в новое состояние KB_{t+1}:

$$KB_{t+1} = KB_t \cup \Delta. \quad (1)$$

Псевдокод алгоритма $\Lambda_{trigger}$ представлен ниже:

Алгоритм 1: Триггерный алгоритм на основе оценки достоверности ($\Lambda_{trigger}$)

Вход: Текущий сценарий S, порог достоверности θ

Выход: Булево значение (True - триггер сработал, False - нет) или информация о контексте Context

1: Вычислить общую текущую достоверность C ← C(S)
2: if C < θ then
3: Context ← Захватить текущее состояние среды
4: return (True, Context)
5: else
6: return False
7: end if

Алгоритм запуска, основанный на оценке достоверности и восприятии контекста, является основой для математического и алгоритмического обеспечения. Его основой является функция уверенности C(S). Функция достоверности генерирует общую оценку достоверности c ∈ [0,1]. Если c меньше динамического порога θ алгоритма, считается, что система столкнулась с неизвестным сценарием или недостатком знаний. Затем алгоритм автоматически обрабатывает входные параметры, промежуточные факты и окончательные выводы в текущем контексте.

$$C(S) = \frac{\Sigma(w_i \cdot conf_i)}{\Sigma w_i}, \quad (2)$$

где $conf_i$ – это степень достоверности i-го активированного правила при текущем сценарии S, а w_i – его весовой коэффициент, используемый для расчета общего уровня достоверности системы для сценария S.

Псевдокод алгоритма Λ_{prompt} представлен ниже:

Алгоритм 2: Алгоритм генерации подсказок (Λ_{prompt})

Вход: Контекст Context, набор метаинформации базы знаний Meta

Выход: Структурированная подсказка P

1: P ← ""
2: P ← P + "Ты — генератор правил для МЭС. Твоя задача –…\n "
3:
4: P ← P + "Информация о текущем сценарии:\n " + JSON.stringify(Context) + "\n"
5: P ← P + "Список доступных параметров (использовать строго в этом формате):\n n"
6: for each (name, id, type) in Meta do
7: P ← P + f"- {name}: {id} (тип: {type})\n"
8: end for
9: P ← P + "Вывод должен быть в следующем XML-формате:\n<rule id=.../>\n<relation id=...>...</relation>"
10: return P

Алгоритм генерации подсказок сочетает ограничения формата с метаинформацией из базы знаний. Входными данными алгоритма являются параметры, извлеченные из существующей базы знаний в контексте, предоставляемом алгоритмом запуска. Ядром алгоритма является высокоструктурированная функция шаблона подсказок, которая включает инструкции на естественном языке, структурированные примеры и строгие ограничения. Инструкции на естественном языке четко определяют задачи LLM. В структурированном примере представлен пример XML-кода, который

соответствует формату МЭС. Строгие ограничения требуют, чтобы LLM использовал указанный формат. Результатом работы алгоритма является запрос P. Этот алгоритм значительно сокращает количество ошибок, генерируемых LLM.

$$P = F_{prompt}(Context, Meta) = T_{role} + T_{template} + T_{constraints} + T_{example}, \quad (3)$$

где T_{role} – инструкция, определяющая роль LLM; $T_{template}$ – описание сценария, включающее конкретные данные из Context; $T_{constraints}$ – строгие ограничения формата, предписывающие использовать формат name:id параметров из Meta и выводить стандартный XML; $T_{example}$ – пример XML-правила, соответствующего спецификациям МЭС.

Псевдокод алгоритма Λ_{verify} представлен ниже:

Алгоритм 3: Алгоритм безопасной верификации (Λ_{verify})

Вход: Исходный вывод LLM R, набор метаинформации Meta, набор правил безопасности R_{safe}

Выход: Прошедшее верификацию DOM-дерево XML D_{valid} или сообщение об ошибке

```
try
    D ← XMLParser.parse(R)
catch ParseError
    R_fixed ← FixXML(R)
    D ← XMLParser.parse(R_fixed)
end try
for each ссылка на параметр ref in D do
   if ref not in Meta then
      return Error("Ошибка ссылки на параметр: " + ref)
    end if
end for
code_snippet ← D.relation.text
if MatchPattern(code_snippet, R_safe) == True then
    return Error("Логика кода нарушает правила безопасности")
end if
return D
```

Алгоритм безопасной верификации отвечает за очистку и проверку данных, полученных от LLM. Трехэтапный конвейер проверки V позволяет анализировать выходные данные LLM и выполнять многоуровневую логическую проверку безопасности.

$$V = V_{parse} \circ V_{logic} \circ V_{safety}, \quad (4)$$

где ∘ обозначает композицию операций; $V_{parse}(R)$ – функция парсинга и исправления формата; $V_{logic}(D)$ – функция проверки логической согласованности; $V_{logic}(D)$ = True тогда и только тогда, когда ∀ ref ∈ D, ref ∈ Meta; $V_{safety}(D)$ – функция проверки безопасности; $V_{safety}(D)$ = True, когда код в D не соответствует ни одному из опасных шаблонов в R_{safe}.

Псевдокод алгоритма Λ_{inject} представлен ниже:

Алгоритм 4: Алгоритм введения правил (Λ_{inject})

Вход: Проверенное DOM-дерево D_{valid}, текущая база знаний KB_t

Выход: Обновленная база знаний KB_{t+1} или сообщение об ошибке

```
KB_DOM ← LoadXML(KB_t)
rules_node ← KB_DOM.find("/rules")
```

3: $relations_{node} \leftarrow KB_{DOM}$.find("/relations")
4: $new_{rule} \leftarrow D_{valid}$.rule
5: $new_{relation} \leftarrow D_{valid}$.relation
6: if $new_{rule}.relation_{id}$!= $new_{relation}$.id then
7: $new_{relation}.id = new_{rule}.relation_{id}$
8: end if
9: $rules_{node}$.append(new_{rule})
10: $relations_{node}$.append($new_{relation}$)
11: $KB_{t+1} = KB_{DOM}$.toXML()
12: ExpertSystem.reload(KB_{t+1})
13: return KB_{t+1}

Алгоритм, который вводит правила и поддерживает согласованность базы знаний, отвечает за интеграцию проверенного правила Dvalid в существующую базу знаний KB. Он основан на точном позиционировании XPath и синхронизации Uuid. Алгоритм сначала точно определяет целевой узел, а затем вставляет в него сгенерированные правила. Алгоритм проверяет, совпадает ли идентификатор связи нового правила с идентификатором вставленного элемента связи, чтобы завершить обновление базы знаний.

$$KB_t \rightarrow KB_{t+1} = Inject(D_valid, KB_t). \quad (5)$$

Схема работы системы математического и алгоритмического обеспечения показана на Рисунке 3.

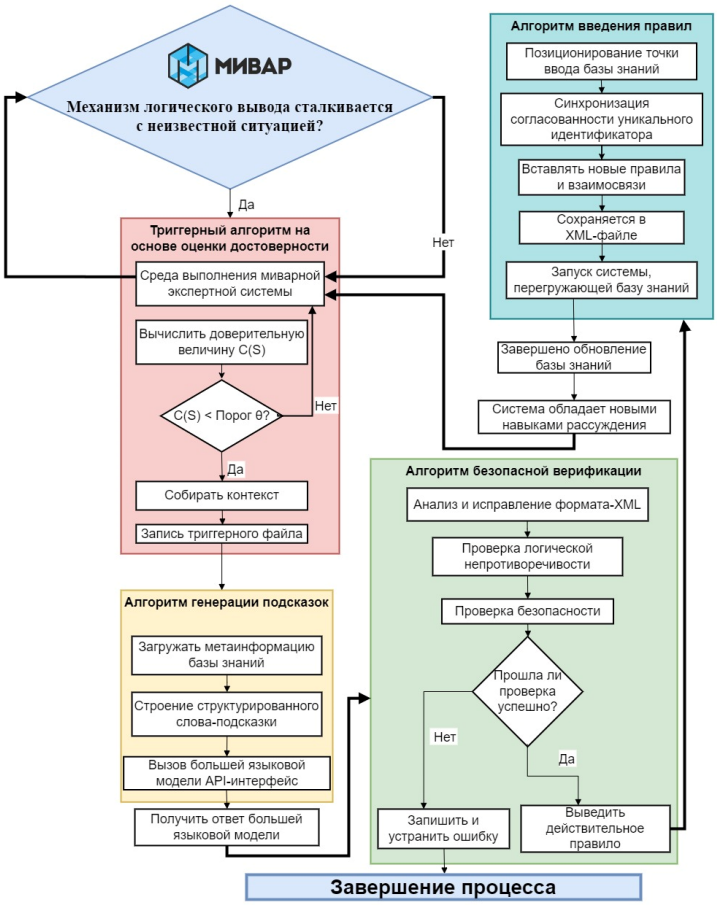

Рисунок 3 – Схема работы математического и алгоритмического обеспечения
Figure 3 – Scheme of operation of mathematical and algorithmic support

Для подтверждения эффективности математического и алгоритмического обеспечения были проведены некоторые эксперименты. Дана оценка общей производительности алгоритма обновления базы знаний МЭС в реальных условиях. Экспериментальной средой является экспертная система обнаружения пожара МЭС.

Чтобы имитировать реальное обновление правил, мы заранее удалили из базы знаний два правила: 20 сценариев теплового разгона для литиевых батарей и 30 сценариев тления для органической пыли.

Диапазон параметров тестового сценария:

Сценарий теплового разгона литиевых аккумуляторов: Начальное значение температуры w_1 составляет 25 ℃, которое линейно увеличивается до 72–85 ℃ в течение 60 секунд. После того, как концентрация дыма f_2 превысит температуру 60 ℃, она увеличивается с базового значения 50 мкг/м^3 до 220–300 мкг/м^3. Сигнал открытого пламени f_1: Постоянно поддерживается на уровне 0. Ожидаемое поведение системы: Должен быть запущен отчет о предупреждении первого уровня и выведено значение alert_level=1.

Сцена тления органической пыли: Начальное значение температуры w_1 составляет 22 ℃, и она медленно повышается до 48–58 ℃ в течение 300 секунд. co_sensor медленно накапливается с 15 до 120–180 частей на миллион по мере повышения температуры. Концентрация дыма f_2 поддерживается на уровне менее 100 мкг/кубический метр. Ожидаемое поведение системы: должен сработать ранний сигнал тревоги, и должно быть выведено значение alert_level=1.

В этом исследовании используется многомерная система количественных показателей для обеспечения объективной оценки. Показатель успешности процесса PSR измеряет надежность и стабильность всего процесса автоматизации. RA точности правил используется для точной оценки качества генерируемых знаний. Среднее время обновления AUT – это время, которое требуется от запуска обновления до перезапуска базы знаний. Скорость изменения стабильности системы SSCR сравнивает изменения в логической способности системы до и после обновления, чтобы предотвратить негативное влияние введения новых правил на стабильность системы. Эти четыре показателя служат основой для оценки практичности, эффективности и безопасности системы автоматического обновления.

Показатель успешности процесса PSR – это показатель успешности в замкнутом цикле от запуска до успешного обновления.

$$PSR = \frac{N_{success}}{N_{triggered}} \cdot 100\%, \tag{6}$$

где $N_{success}$ – количество сценариев, в которых база знаний была успешно обновлена, а $N_{triggered}$ – количество сценариев, в которых алгоритм запуска был успешно активирован.

Точность правил RA – это процент правил, успешно прошедших проверку.

$$RA = \frac{N_{accurate}}{N_{total}} \cdot 100\%, \tag{7}$$

где $N_{accurate}$ – количество правил, превышающее или равное набранному баллу, а N_{total} – общее количество сгенерированных правил.

Среднее время обновления AUT – это общее время, необходимое для динамического обновления. Это включает задержку при вызове LLM, локальное время расчета и время перезапуска системы.

$$AUT = \frac{\Sigma(T_{inject} - T_{trigger})}{N}, \tag{8}$$

где $T_{trigger}$ – временная метка, которая запускает алгоритм, а T_{inject} – временная метка, которая завершает перезагрузку базы знаний путем вставки алгоритма.

Скорость изменения стабильности системы SSCR – это изменение точности вывода системы после обновления.

$$SSCR = \frac{P_{post} - P_{pre}}{P_{pre}} \cdot 100\%, \quad (9)$$

где P_{pre} – это пропускная способность экспертной системы для вывода в стандартном тестовом примере до обновления, а P_{post} – это пропускная способность экспертной системы в том же тестовом наборе после обновления.

Результаты и обсуждение

Алгоритм запуска успешно сработал в 48 из 50 сцен. Из этих 48 сценариев 46 сценарийных правил были успешно сгенерированы и внедрены в конечном итоге, что составляет 95,8 % успеха процесса PSR. Из-за чрезмерного шума в данных датчика обнаружено 2 очага тления органической пыли, что привело к созданию непроверенных правил LLM.

Было оценено 46 успешно сгенерированных правил, из которых 42 получили четыре балла и более. Точность правила RA составляет 91,3 %. 3 правила, которые нуждаются в доработке, получили 3,5–4 балла. 1 правило получило 3 балла из-за чрезмерно сложной логики.

Среднее время обновления AUT составляет 52,3 секунды, σ=12,1 с. Среди них время вызова LLM и ответа на него составляет 38,2±10,5 секунды, что составляет 73,0 от общего времени. Проверка соблюдения этого правила заняла 5,1±1,3 секунды, что составило 9,8 % от общего времени. Перезапуск базы знаний занял 9,0±0,5 секунды, что составило 17,2 % от общего времени.

После всех 46 успешных обновлений пропускная способность системы осталась неизменной в 100 стандартных тестовых примерах. Коэффициент устойчивости системы SSCR равен нулю. Это показывает, что правило обновления не противоречит существующей базе знаний, подтверждая безопасность алгоритма реализации.

Общие результаты оценки эффективности системы приведены в Таблице 1.

Таблица 1 – Общие результаты оценки эффективности системы
Table 1 – Overall results of the system performance assessment

Показатель производительности	Значение	Записи
Всего тестовых сценариев	50	Включая термический выброс литиевой батареи (20) и тление органической пыли (30)
Успешные триггеры	48	Частота срабатывания 96 % (48/50)
Успешный процесс	46	Коэффициент успешности процесса (PSR): 95,8 % (46/48)
Сгенерированы точные правила	42	Точность правила (RA): 91,3 % (42/46)
Среднее время обновления AUT (секунды)	52,3	Стандартное отклонение (σ): ±12,1 с
Скорость изменения стабильности системы SSCR	Не изменялась	После обновления частота прохождения стандартного набора тестов не изменилась

Практическую механику автономной адаптации целесообразно рассмотреть на примере инцидента с аномальными показаниями термометрии. Входная переменная w_1 – 8,5, а w_2 – 2,1. Сначала активируется алгоритм Λtrigger. Λprompt преобразует

параметры окружающей среды в естественный язык и отправляет их в LLM, требуя, чтобы затвор был переведен в положение f_1=15. Параметры привязаны к их системному идентификатору. Правила в формате XML, сгенерированные LLM, отправляются в Λverify. Λverify проверяет грамматическую точность и безопасность правил. Λinject вставляет новые правила в базу знаний МЭС для автоматического обновления базы знаний.

Сгенерированные новые правила соответствуют физике. Логика вывода соответствует физической природе процесса компенсации аномалий. Установленный порог также соответствует фактическим требованиям. После интеграции новых правил стабильность логики рассуждений экспертной системы остается неизменной. Этот случай доказывает, что математическое и алгоритмическое обеспечение, предложенное в данном исследовании, позволяет сократить цикл автоматического обновления базы знаний с нескольких часов до нескольких секунд. Без ущерба для надежности системы улучшается ее адаптивная способность.

Заключение

В данном исследовании реализовано математическое и алгоритмическое обеспечение автоматического обновления правил в МЭС, а также решена проблема «узкого места знаний». Комбинация large language model LLM и МЭС реализует полный процесс от обнаружения неизвестных сценариев до внедрения новых правил. Эксперименты показали, что практичность и надежность математического и алгоритмического обеспечения высоки: вероятность успеха составляет 95,8 %, а точность правил – 91,3 %, что увеличивает время, необходимое для обновления правил, с нескольких часов до нескольких секунд.

Научная новизна разработки заключается в создании математического и алгоритмического обеспечения четырех алгоритмов для автоматического динамического обновления правил в МЭС с использованием больших языковых моделей. Разработанное математическое и алгоритмическое обеспечение доказывает, что LLM может быть безопасно интегрирована в системы символьного мышления, закладывая основу для нового поколения гибридных интегрированных интеллектуальных систем.

Практическая ценность результатов заключается в формировании фундамента для интеллектуального революционного развития МЭС. Основные принципы разработки алгоритмов могут быть распространены на другие области, обеспечивая широкую универсальность.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Варламов О.О. *Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство.* Москва: Радио и связь; 2002. 286 с.
2. Платонов Ю.Г. Анализ перспектив перехода информационных систем на сервисно-ориентированную архитектуру. *Проблемы информатики.* 2011;(4):56–65.
3. Шэнь Ц., Гун Ш., Варламов О.О., Адамова Л.Е., Баленко Е.Г. Динамическое планирование траектории робота на основе семантического обнаружения объектов с использованием миварной экспертной системы. *Проблемы искусственного интеллекта.* 2024;(4):164–176. https://doi.org/10.24412/2413-7383-2024-4-164-176
Shen Q., Gong Sh., Varlamov O.O., Adamova L.E., Balenko E.G. Dynamic robot path planning based on semantic object detection using mivar expert system. *Problems of*

Artificial Intelligence. 2024;(4):164–176. (In Russ.). https://doi.org/10.24412/2413-7383-2024-4-164-176

4. Коценко А.А. Разработка моделей миварного логического пространства для обеспечения трехмерного движения автономных роботов. В сборнике: *МИВАР '24: Сборник научных статей, 18–20 апреля 2024 года, Москва, Россия*. Москва: ИНФРА-М; 2024. С. 361–366.
   Kotsenko A.A. Development of models of mivar logic space to provide three-dimensional movement of autonomous robots. In: *MIVAR '24: Collection of scientific articles, 18–20 April 2024, Moscow, Russia.* Moscow: INFRA-M; 2024. P. 361–366. (In Russ.).
5. Чибирова М.О. Анализ подходов к построению систем поддержки принятия решений: Онтологии и Мивары. *Автоматизация и управление в технических системах.* 2014;(1-2):44–60.
   Chibirova M.O. Analysis of existing approaches in developing decision support systems: ontology and mivar nets. *Automation and Control in Technical Systems.* 2014;(1-2):44–60. (In Russ.).
6. Доу Л. Модель принятия решений для обнаружения пожаров на основе распознавания образов и миварной экспертной системы. *Системы управления и информационные технологии.* 2025;(3):59–65.
   Dou L. Decision-making model for fire detection based on pattern recognition and mivar expert system. *Control Systems and Information Technology.* 2025;(3):59–65. (In Russ.).
7. Доу Л. О разработке алгоритмического обеспечения для создания и обновления правил миварной экспертной системы на основе GPT. В сборнике: *МИВАР '25: Доклады, 17–19 апреля 2025 года, Москва, Россия.* Москва: ИНФРА-М; 2025. С. 415–417.
   Dou L. On the development of algorithmic software for creating and updating rules of a GPT-based mivar expert system. In: *MIVAR '25: Proceedings, 17–19 April 2025, Moscow, Russia.* Moscow: INFRA-M; 2025. P. 415–417. (In Russ.).
8. Hanney K., Keane M.T. The adaptation knowledge bottleneck: How to ease it by learning from cases. In: *Case-Based Reasoning Research and Development: Second International Conference on Case-Based Reasoning (ICCBR-97), 25–27 July 1997, Providence, RI, USA.* Berlin, Heidelberg: Springer; 1997. P. 359–370. https://doi.org/10.1007/3-540-63233-6_506
9. Доу Л. Создание мультимодальной системы обнаружения пожаров и применение в миварной экспертной системе. *Информация и образование: границы коммуникаций.* 2025;(17):223–227.
   Dou L. Creation of a multimodal fire detection system and application in the mivar expert system. *Information and Education: Borders of Communication.* 2025;(17):223–227. (In Russ.).
10. Zhang X., Sun W., Chen K., Jiang R. A multimodal expert system for the intelligent monitoring and maintenance of transformers enhanced by multimodal language large model fine-tuning and digital twins. *IET Collaborative Intelligent Manufacturing.* 2024;6(4). https://doi.org/10.1049/cim2.70007
11. Чибирова М.О. Сравнительный анализ миварного подхода с подходами, основывающимися на онтологиях и когнитивных картах. *Радиопромышленность.* 2015;(3):55–66.
    Chibirova M. Analysis of existing approaches: ontology, cognitive maps and mivar nets. *Radio Industry*. 2015;(3):55–66. (In Russ.).

12. Луцкович А.И., Васильев В.И., Вульфин А.М., Кириллова А.Д., Сулавко А.Е. Автоматизированная система анализа слабоструктурированных данных киберразведки с использованием больших языковых моделей. *Информационно-управляющие системы.* 2025;(2):50–67.
Lutskovich A.I., Vasilyev V.I., Vulfin A.M., Kirillova A.D., Sulavko A.E. Automated system for analyzing weakly structured threat intelligence data using large language models. *Information and Control Systems.* 2025;(2):50–67. (In Russ.).
13. Hu W., Xu Y., Li Y., et al. BLIVA: A Simple Multimodal LLM for Better Handling of Text-Rich Visual Questions. *Proceedings of the AAAI Conference on Artificial Intelligence.* 2024;38(3):2256–2264. https://doi.org/10.1609/aaai.v38i3.27999
14. Хадиев А.М. Разработка и практическая реализация миварной машины логического вывода. *Радиопромышленность.* 2015;(3):79–89.
Khadiev A.M. Development and practical realization of mivar inference machine. *Radio Industry.* 2015;(3):79–89. (In Russ.).
15. Коценко А.А. Анализ применения для АСУТП миварных сетей в формате двудольных и трехдольных графов. В сборнике: *МИВАР '24: Сборник научных статей, 18–20 апреля 2024 года, Москва, Россия.* Москва: ИНФРА-М; 2024. С. 432–438.
Kotsenko A.A. Analysis of application of mivar nets in the format of bipartite and tripartite graphs for automated process control systems. In: *MIVAR '24: Collection of scientific articles, 18–20 April 2024, Moscow, Russia.* Moscow: INFRA-M; 2024. P. 432–438. (In Russ.).
16. Гаврилов Л.Я., Уляшин В.В., Попов М.Ю. Исследование возможностей больших лингвистических моделей для создания миварных баз знаний. В сборнике: *ИИАСУ'24 – Искусственный интеллект в автоматизированных системах управления и обработки данных: Сборник статей III Всероссийской научной конференции: Том 2, 30 октября – 01 ноября 2024 года, Москва, Россия.* Москва: Издательский дом КДУ; 2025. С. 23–31.
Gavrilov L.Y., Ulyashin V.V., Popov M.Y. Research the possibilities of large linguistic models for creating multi-dimensional knowledge bases. In: *IIASU'24 – Artificial intelligence in management, control, and data processing systems: Proceedings of the III All-Russian scientific conference: Volume 2, 30 October – 01 November 2024, Moscow, Russia.* Moscow: Izdatel'skii dom KDU; 2025. P. 23–31. (In Russ.).
17. Данилюк А.В., Ким Р.И. О применении MivarGPT для автоматизации создания миварных баз знаний и машиностроительного искусственного интеллекта. В сборнике: *ИИАСУ'23 – Искусственный интеллект в автоматизированных системах управления и обработки данных: Сборник статей II Всероссийской научной конференции: Том 3, 27–28 апреля 2023 года, Москва, Россия.* Москва: Издательский дом КДУ; 2023. С. 568–573.
Danilyuk A.V., Kim R.I. On the application of MivarGPT for automating the creation of mivar knowledge bases and mechanical engineering artificial intelligence. In: *IIASU'23 – Artificial intelligence in management, control, and data processing systems: Proceedings of the II All-Russian scientific conference: Volume 3, 27–28 April 2023, Moscow, Russia.* Moscow: Izdatel'skii dom KDU; 2023. P. 568–573. (In Russ.).
18. Du S.Q., Tang S.J., Wang W.X., Li X.M., Guo R.Z. Tree-GPT: Modular Large language Model expert system for forest remote sensing image understanding and interactive analysis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* 2023;48(1):1729–1736. https://doi.org/10.5194/isprs-archives-xlviii-1-w2-2023-1729-2023

19. Lammert Ja., Dreyer T.F., Lörsch A.M., et al. Large language models for precision oncology: Clinical decision support through expert-guided learning. *Journal of Clinical Oncology.* 2024;42(16_suppl). https://doi.org/10.1200/jco.2024.42.16_suppl.e13609
20. Schuerkamp R., Ahlstrom H., Giabbanelli Ph.J. Automatically resolving conflicts between expert systems: An experimental approach using large language models and fuzzy cognitive maps from participatory modeling studies. *Knowledge-Based Systems.* 2025;313. https://doi.org/10.1016/j.knosys.2025.113151
21. Awasthi Y., Garikayi T., Fundisi L.T., Mukhalela B. A Comparative Study: Evaluating ChatGPT and DeepSeek AI Tools in Practice. *International Journal of Open Information Technologies.* 2025;13(5):67–70.
22. Доу Л. Метод обработки информации и изменения пороговых параметров правил в миварной экспертной системе с использованием больших языковых моделей. *Проблемы искусственного интеллекта.* 2025;(4):194–205. https://doi.org/10.24412/2413-7383-2025-4-39-194-205
Dou L. Method for information processing and modification of threshold parameters of rules in a mivar expert system using large language models. *Problems of Artificial Intelligence.* 2025;(4):194–205. (In Russ.). https://doi.org/10.24412/2413-7383-2025-4-39-194-205

ИНФОРМАЦИЯ ОБ АВТОРЕ / INFORMATION ABOUT THE AUTHOR

Доу Линхань, аспирант, Московский государственный технический университет им. Н.Э. Баумана, Москва, Российская Федерация.
e-mail: d923952505@gmail.com
ORCID: 0009-0007-7669-5004

Dou Linghan, Postgraduate, Bauman Moscow State Technical University, Moscow, the Russian Federation.

Статья поступила в редакцию 08.04.2026; одобрена после рецензирования 11.05.2026; принята к публикации 18.05.2026.

The article was submitted 08.04.2026; approved after reviewing 11.05.2026; accepted for publication 18.05.2026.