DOI 10.36622/1682-7813.2024.27.3.002
УДК 349:004.056

НЕЙРОСЕТЕВОЙ СЕРВИС РЕГЛАМЕНТАЦИИ МЕР ПРОТИВОДЕЙСТВИЯ КИБЕРАТАКАМ (ЧАСТЬ I)

Г.А. Остапенко, А.П. Васильченко, А.А. Остапенко, А.А. Ноздрюхин, А.Г. Остапенко, И.Л. Батаронов, А.С. Кривошеин

Рассматривается нейросетевая реализация сервиса автоматизированного противоборства в части регламентации мер противодействия компьютерным атакам. Проведен анализ существующих чат-ботов, применимых для нейросетевой реализации модуля генерации регламентирования мер противодействия сетевым вторжениям в автоматизированных информационных системах. Разработаны методическое обеспечение организации машинного обучения языковой модели и соответствующая алгоритмизация, включающая в себя взамодействие с модулем нейросетевой идентификации, процедуру определения ip-адресов, функцию приёма данных об атаках и уязвимостях, регламентацию мер противодействия сетевым вторжениям, операцию вывода мер, процедуру корректировки мер, функцию подготовки базы знаний для машинного обучения и процедуру машинного обучения языковой модели. Предлагаемое методическое и алгоритмическое обеспечение каталогизирует новые методы агрегации данных, необходимые для обучения локальной языковой модели, предъявляет регламентацию мер противодействия сетевым вторжениям.

Ключевые слова: кибератака, уязвимость, сетевое вторжение, нейросетевая реализация модуля, языковая модель, машинное обучение.

Введение

Наряду со стремительным ростом компьютерных атак с момента создания компанией OpenAI языковой модели GPT-3.5 особую популярность приобрели нейросети, позволяющие автоматизировать процесс принятия решений на основе анализа крупного массива входных данных. Поэтому применение данной технологии при адекватной настройке под конкретные проектные нужды способно кратно увеличить эффективность регламентации мероприятий по противодействию сетевым вторжениям [1, 2]. В данном случае адаптация должна проходить с учетом имеющихся банков знаний об информационных атаках и уязвимостях, на основе которых осуществляется машинное обучение. Именно в этом контексте представляется актуальным настоящее исследование, посвященное нейросетевой реализации модуля генерации регламентирования мер противодействия сетевым вторжениям в автоматизированных информационных системах (АИС). Все выявленные аналоги подобной системы, позволяющие обеспечить взаимодействие с пользователем на основе чата с хранящейся историей и использующие в своей реализации ту или иную языковую модель, можно разделить на две большие группы:

1) чат-боты (на основе искусственного интеллекта), использующие в своей основе API, предоставляемое компанией OpenAI [3-12],

2) чат-боты собственной реализации [13-15].

Первый вариант чат-ботов был реализован на основе GPT-моделей OpenAI, поэтому возможность их использования сопряжена с наличием у пользователя аккаунта OpenAI с подпиской на ChatGPT-4. Еще одним существенным недостатком для пользователей, проживающих на территории России, является необходимость подключения VPN, что вносит дополнительные задержки при интеграции подобных чат-ботов в локальную сеть предприятия. Перечень выявленных аналогов первой категории с описанием их функциональности приведен в табл. 1.

Таблица 1

Обзор аналогов, работающих посредством OpenAI API

Название аналога	**Описание функциональных возможностей**
MagicUnprotect	GPT-агент, взаимодействующий с базой данных Unprotect для информации о методах уклонения от вредоносного программного обеспечения [3]
MitreGPT	Нейросеть, отвечающая на запросы пользователя, ориентируется на тактики и техники, взятые из базы данных MITRE ATT & CK [4]
ATT&CK MATE	GPT-агент, который взаимодействуетс базой знаний о тактиках и техниках киберпреступников, содержащей при этом и компенсирующие меры ATT & CK [5]
Cyber gpt	Нейросетевой бот, осуществляющий консультации по вопросам кибербезопасности [6]
Soc copilot	Помощник, выдающий рекомендации по обеспечению безопасности на основе ключевых слов [7]
CVEs	Взаимодействие с базой CVE для поиска существующих уязвимостей [8]
SpamGuard tutor	Нейросетевой помощник по обнаружению и предотвращению спама [9]
Threat modelcompanion	Бот, ориентированный на выявление и устранение угроз безопасности посредством выдачи пользователю интеллектуальных подсказок [10]
Prompt Injectiondetector	GPT-агент, используемый для выдачи подсказок о том, как действовать при попытках осуществления различного рода инъекций, в качестве выходных данных использует JSON-формат [11]
Owasp LLM advisor	Консультант в области кибербезопасности, использующий рекомендации OWASP [12]

Существуют и другие боты (на основе искусственного интеллекта), главным преимуществом которых по сравнению с аналогами табл. 1 является возможность их бесплатного использования. К числу подобных аналогов относятся следующие чат-боты:

− Threat Intel Bot, представляющий собой инструмент искусственного интеллекта, специализирующийся на предоставлении последней и наиболее полной информации об угрозах и методах противодействия им [13];

− Systems Security Analyst, выступающий в роли системного аналитика безопасности, основная цель которого обеспечить помощь пользователю в выявлении, анализе и смягчении киберугроз и уязвимостей в сетевых системах [14];

− CyberGuard, являющийся специализированным нейроинструментом, разработанным с целью предоставления помощи по вопросам кибербезопасности в виде интеллектуальных подсказок. Он в первую очередь ориентирован на домашнихпользователей и малый бизнес [15].

Исследование вышеперечисленных аналогов позволяет сделать вывод о наличии следующих, существенных для выбранной тематики противоречий между:

− методическим несовершенством подхода в выдаче интеллектуальных подсказок пользователю (заключающемся в отсутствии агрегации данных между базами знаний атак и уязвимостей; недостатках стандартизации вывода регламентов противодействия; невозможности встраивания перечисленных аналогов в инфраструктуру локальной сети предприятия; недостаточной изолированности системы; зависимости от технологий крупных компаний) и объективной необходимостью исправления

в реальной нейросетевой реализации этих недостатков,

− отсутствием массива данных, необходимых для машинного обучения нейросетевого модуля регламентации мер противодействия сетевым вторжениям (с учетом входной информации о вторжении в форме сценария, содержащего сведения об атаке и эксплуатируемой уязвимости, и выходной,

– конкретному мероприятию по обеспечению безопасности) и потребностью в наличии необходимого и достаточного массива для максимизации адекватности предлагаемых интеллектуальных подсказок.

В связи с вышеизложенным объектом исследования является пространство функционирования АИС, подвергающихся массированным сетевым вторжениям.

Предмет исследования видится в технических решениях нейросетевой реализации модуля генерации регламентирования мер противодействия сетевым вторжениям.

Цель исследования заключается в повышении защищенности за счет разработки и применения нейросетевого модуля регламентации мер противодействия сетевым вторжениям.

Для разрешения вышеизложенных противоречий и достижения намеченной цели необходимо обеспечить решение следующих задач.

1. Разработка методического и алгоритмического обеспечения модуля регламентации мер противодействия сетевым вторжениям, включая: формирование архитектуры сервиса; принципов взаимодействия с другими модулями в рамках общей автоматизированной системы обнаружения и идентификации сетевых вторжений; выдачу мер противодействия, с формулированием требований к реализуемому инструментарию.

2. Программная реализация модуля регламентации мер противодействия с учетом: потребности в создании изолированности проектируемой системы; необходимости стандартизации вывода интеллектуальных подсказок с возможностью их корректировки по запросам пользователя; агрегации знаний о сценариях атак и дополнительного машинного обучения используемой в рамках модуля нейросети (для более корректного регламентирования мероприятий по противодействию сетевым вторжениям), включая демонстрацию эффективности спроектированного нейросетевого инструментария (с учетом ввода требований администратора безопасности по корректировке выданных мер).

Методическое обеспечение машинного обучения языковой модели

Ввиду того, что языковая модель saiga_mistral_7b_lora обучалась на довольно скудной базе данных в сравнении с такими моделями как GPT-4 или Cloude-3, то генерация ответов под специфику выдачи регламентов противодействия сетевым вторжениям будет некорректной. Поэтому модель необходимо обучить новой информации, а именно [1-8]:

− знаниям об атаках, их названиям и идентификаторам согласно CAPEC;

− знаниям о существующих уязвимостях;

− знаниям о сценариях атаки, включающим сведения об атаке и уязвимостях, которые она эксплуатирует;

− знаниям о мерах противодействия на каждый сценарий, причем меры противодействия включают в себя регламенты как реагирования, так и ликвидации последствий;

− формату выдачи ответов, включающих в себя меры реагирования и мерыликвидации последствий.

Все техники дополнительного обучения крупных языковых моделей принято называть fine-tuning. На данный момент существует несколько основных техник дополнительного обучения fine-tuning, к их числу относятся:

1) полный fine-tuning;

2) fine-tuning с оптимизацией параметров:

a. LoRa;

b. Prompt Tuning.

Полный fine-tuning предусматривает обновление всех весов модели, в результате чего создается новая версия модели. Важно отметить, что, как и предварительное обучение, полный fine-tuning требует

достаточного количества памяти и вычислительных ресурсов, в связи с чем такой вариант дополнительного обучения не рекомендуется использовать на реальных проектах.

В отличии от полного fine-tuning, при которой каждый вес модели обновляется во время контролируемого обучения, методы fine-tuning с оптимизацией параметров обновляют лишь небольшое подмножество параметров языковой модели. В свою очередь существует два варианта реализации неполного обучения весов модели. Prompt Tuning, заключаются в добавлении токенизированного датасета в слой языковой модели, который отвечает за обработку запросов пользователя. По сути, эта техника является своеобразным аналогом концепции RAG, при которой необходимая информация для ответа подается в сам запрос. Только в данном случае эта информация уже будет записана в нейросеть, в связи с чем решается проблема быстрого заполнения контекстного окна. Однако не всех датасетах и моделях такая техника работает качественно, в связи с чем появилась необходимость во введении новый методики LoRA, которая развивает математические идеи, использующиеся в promt tuning, и встраивает прошедшие токенизацию датасеты в более глубокие слои языковой модели, как это показано на рис. 5. Такой подход позволяет дополнить модель большим количеством данных, нежели чем при promt tuning. При этом качество выдаваемых ответов улучшится. Данная концепция не вмешивается в исходный дизайн модели и не требует такого количества вычислительных ресурсов, как при полном fine-tuning, поэтому наиболее предпочтительна для использования в реальных проектах.

Таким образом, целесообразно использовать концепцию дополнительного машинного обучения LoRA.

Алгоритмизация взаимодействия с модулем нейросетевой идентификации вторжений

Исходя из архитектуры модуля регламентации мер противодействия сетевым вторжениям выделяются три базисных элемента, обеспечивающие стабильности и правильность работы всей системы. К их числу относятся:

− сервис взаимодействия с модулем идентификации сетевых вторжений;

− сервис веб-приложения администратора безопасности;

− сервис машинного обучения локальной языковой модели.

При этом каждый сервис имеет ряд требований, предъявляемых к его функционированию, исходя из которых и должно проектироваться алгоритмическое обеспечение.

Базисным компонентом ядра реализуемой системы, является сервис взаимодействия с модулем идентификации сетевых вторжений. Основной его задачей является обеспечение бесперебойной связи с модулем идентификации сетевых вторжений. Исходя из этого вытекает ряд требований, которые должно быть предъявлены к его работе:

− обеспечение возможности выбора сетевого интерфейса, подключенного в локальную сеть;

− правильное определение ip-адресов всех работающих в локальной сети автоматизированных информационных систем;

− обеспечение возможности обновления списка ip-адресов;

− настройка своевременного приема пакетов с данными о сценарии атаки врежиме реального времени;

− проверка целостности и подлинности полученных данных;

− извлечение информации из принятого пакета без потери полезныхданных;

− передача информации о сценарии атаки, полученной в результате парсинга, для дальнейшего манипулирования в рамках веб-приложения;

− обеспечение высокого уровня надежности и отказоустойчивости;

− сохранение целостности данных в случае системного сбоя.

Выполнение поставленных требований обеспечивается за счет разделения сервиса на два подмодуля, один из которых должен определять список ip-адресов хостов в локальной сети, а другой на основание этого списка – обеспечить получение

полезных данных о сценарии атаки и передачу извлеченных сведений в веб-приложение. В связи с этим алгоритмическое обеспечение данного сервиса должно быть логически разделено на две части, обеспечивая структуризацию функционирования подмодулей обнаружения списка IP-хостов и приема данных из локальной сети. Причем, помимо взаимодействия с модулем идентификации сетевых вторжений, для успешного функционирования системы необходимо обеспечить связьмежду двумя подмодулями. В рамках описываемого сервиса, подмодуль приема данных из локальной сети (ППДИЛС) должен принимать полезные сведения только с IP-адресов тех хостов, информация о которых была записана в файл подмодулем определения списка IP (ПОС IP).

В общем виде схема взаимодействия двух подмодулей сервиса взаимодействия с модулем идентификации сетевых вторжений показана на рис. 1.

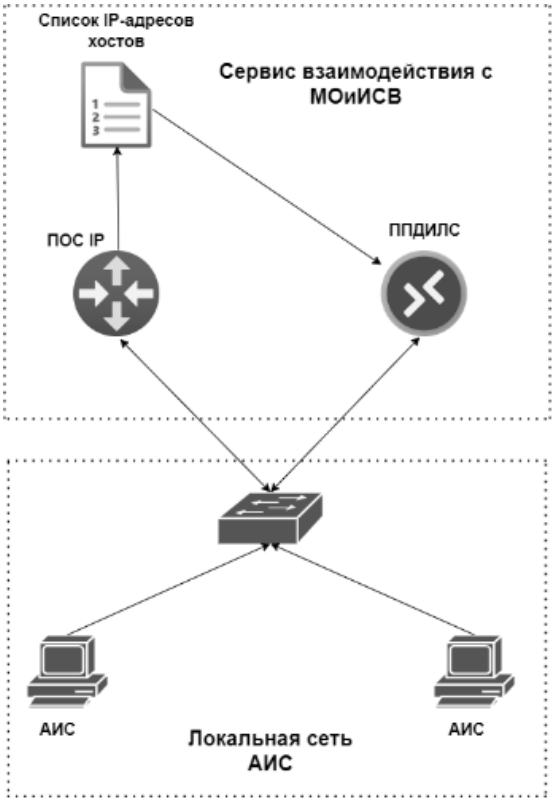

Рис. 1. Структурная схема сервиса взаимодействия с модулем идентификациисетевых вторжений

Процедура определения списка ip-адресов хостов в локальной сети

Алгоритмическое обеспечение ПОС IP, представляет собой эффективную схему автоматизированной идентификации IP-адресов хостов, работающих в рамках рассматриваемой локальной сети. Автоматизация данного подхода обеспечивает предоставление возможности администратору безопасности самостоятельно выбрать сетевой интерфейс, по которому его ПК соединяется слокальной сеть АИС. Создание такой вариативности позволяет исключить случай, при котором определение IP-адресов пойдет по loopback-интерфейсам устройства. Блок-схема алгоритма данного подмодуля приведена на рис. 2.

Результат на выходе будет передан ППДИЛС, чтобы обеспечить прием данных лишь с конкретных хостов, а также в веб-приложение, для автоматического вывода списка чатов для каждой АИС.

Функция приема данных об атаках и уязвимостях

Данный алгоритм необходим для правильного функционирования ППДИЛС. Он составляет основу взаимодействия между модулем идентификации сетевых вторжений и модулем регламентации мер противодействия. В связи с этим его алгоритм должен быть максимально проработан, покрывая при этом все требования, описанные для сервиса взаимодействия с МОиИСВ. Блок-схема алгоритма данного подмодуля приведена на рис. 3.

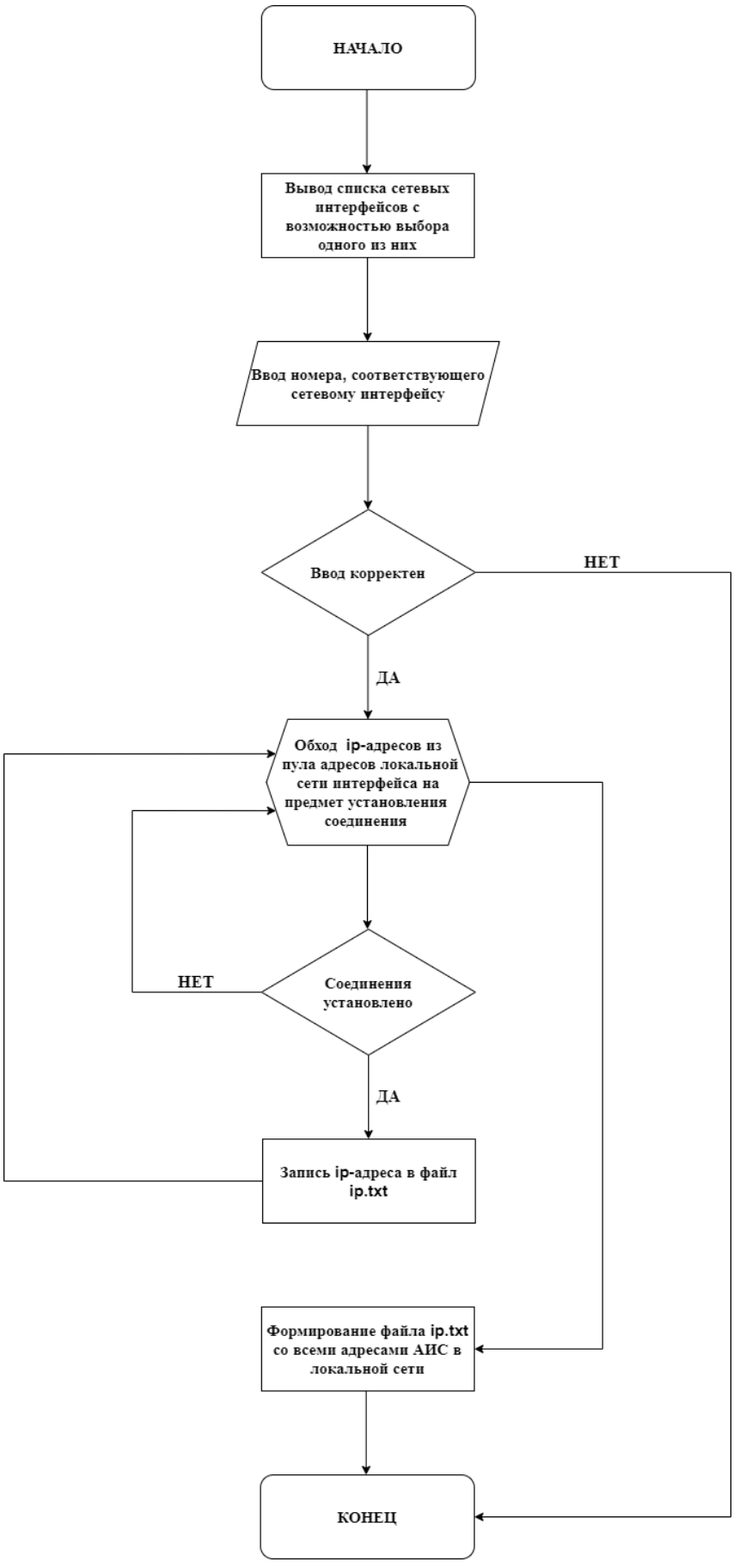

Рис. 2. Алгоритм определения списка ip-адресов хостов в локальной сети

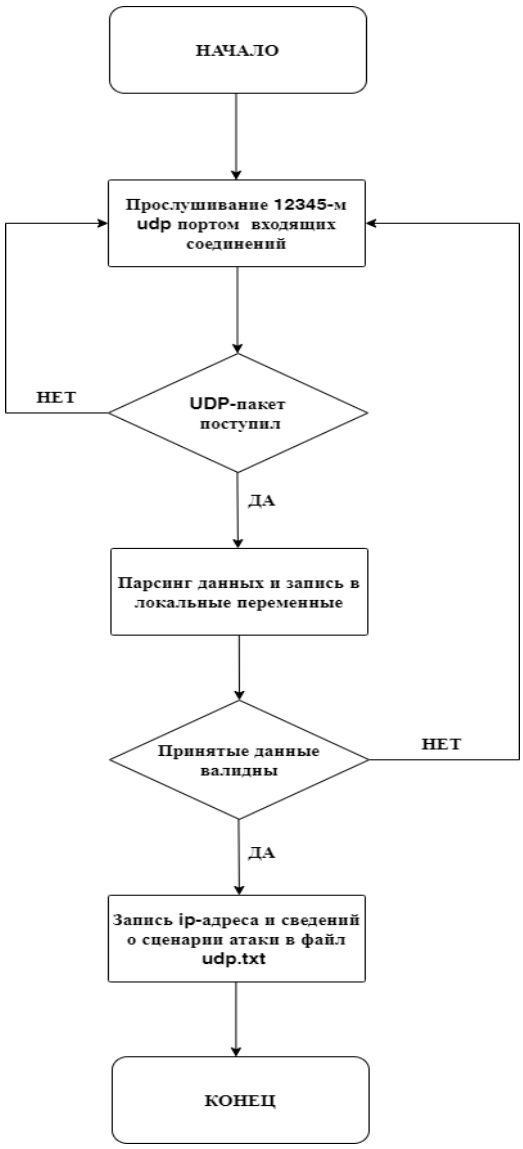

Рис. 3. Алгоритм приема данных о сценарии атаки

Ниже описан алгоритм приема данных о сценарии атаки, включающем идентификатор атаки согласно CAPEC и семейства уязвимостей, обнаруженные на хосте, с которого производилась отправка пакета:

1. Данный подмодуль на начальномэтапе обеспечивает прослушивание UDP-сокета, ассоциированного с портом 54321, на прием входящих соединений из пула адресов локальной сети.

2. В случае прихода UDP-пакета происходит парсинг данных и запись в локальные переменные. Соответственно отсутствие пакетов сопровождается уходом в ожидание потока, запущенного на предыдущем этапе.

3. Данные, прошедшие извлечение и записанные в локальные переменные, проходят проверку на валидность. В случае, если входящий UDP-пакет не соответствует структуре, описанной в разделе 2, либо данные об атаке или семейству уязвимостей некорректны, поток отбраковывает полученные сведения, локальные переменные очищаются и происходит переход в прослушивание входящих соединений.

4. В случае валидности принятых данных происходит запись ip-адреса и сведений сценария атаки в файл udp.txt в виде «[ip-адрес хоста]+++идентификатор атаки| уязвимость$_1$| уязвимость$_2$|… уязвимость$_N$». Это необходимо для более удобной манипуляции с обработанными данными на стороне веб-приложения.

Таким образом, на выходе ППДИЛС выдает сформированный на основание данных принятого UDP-пакета файл, содержащий информацию об ip-адреса хоста, накотором произошла атака, а также сведения о сценарии этой атаки. Этот файлпередается на вход веб-приложения, где уже происходит формирование запросов и вывод мер противодействия на основании принятой информации.

Алгоритмизация работы веб-приложения

Следующим элементом модуля регламентации мер противодействия сетевым вторжениям, обеспечивающим непосредственное взаимодействие пользователя с языковой моделью, является сервис веб-приложения администратора безопасности.

Для успешного функционирования общей автоматизированной системы данный сервис должен выполнять следующие требования:

– своевременный прием информации, обработанной сервисом взаимодействия с модулем;

– идентификация сетевых вторжений и информации о сценарии атаки;

– вывод мер противодействия в формате регламентов реагирования и ликвидации последствий;

− автоматическая выдача мер противодействия на основании принятого сценария атаки;

− обеспечение возможности доработки мер противодействия с учетом требований администратора;

− обеспечение многопоточного чата для конкретизации выданных мер по каждому хосту в отдельности;

− формирование и отправка запросов в языковую модель;

− прием ответов от языковой модели, с учетом того, что она развернута насервере;

− при отсутствии сигнала об атаке обеспечение возможности выдавать меры противодействия на основании сценария, введенного пользователем.

Исходя из требований, предъявляемых к данному сервису, делается вывод о том, что сервис должен работать, по сути, в двух режимах:

− в режиме ожидания прихода сигнала об атаке и выдаче мер на основании сценария, записанного в полученном пакете данных;

− в режиме, при котором администратор безопасности имеет возможность самостоятельно произвести ввод информации о сценарии атаки и получить на ее основании меры реагирования и ликвидации последствий.

В связи с чем в основе функционирования данного сервиса лежат два алгоритма, обеспечивающие возможность работы в двух описанных выше режимах. Опишем более подробно эти алгоритмы.

Операция вывода мер противодействия при поступлении сигнала об атаке

Данный алгоритм представляет собой эффективный способ связи между сервисом взаимодействия с модулем идентификации сетевых вторжений и сервисом веб-приложения администратора безопасности. Причем вывод мер противодействия при таком варианте осуществляется на основание данных
о сценарии атаки, взятых из UDP-пакета, и дальнейшее хранение истории сообщений проводится с учетом выведенных данных после поступления сигнала об атаке мероприятий. Тем самым обеспечивается возможность функционирования веб-приложения администратора безопасности в первом режиме работы. Стоит отметить, что такой вариант имеет большое значение в рамках функционирования автоматизированной системы обнаружения, идентификации сетевых вторжений и регламентации мер противодействия. Это связано с тем, что данные передаются от одного модуля к другому и дальнейшая работа исследуемого элемента системы происходит на основании принимаемых сведений о сценарии атаки. В совокупности такой механизм позволяет обеспечить автоматизацию проектируемой системы при ее развертывании в конкретной локальной сети, включающей хосты со своей спецификой.

Блок-схема алгоритм вывода мер противодействия при поступлении сигнала об атаке приведена на рис. 4, 5.

Ниже описан алгоритм вывода мер противодействия при поступлении сигнала об атаке:

1. Сначала запускается поток, прослушивающий UDP-соединения (ППДИЛС), который производит запуск сервиса приема данных при поступлении соответствующего пакета. В противном случае ожидание приема продолжается.

2. В результате работы сервиса приема данных появляется файл с временными данными udp.txt, который содержит сведения о сценарии атаки.

3. Файл, описанный в предыдущем пункте, передается на вход бекэнд-частивеб-приложения для чтения сведений о сценарии атаки и записи их в локальные переменные.

4. На основание значений, которые были записаны в локальные переменные, формируется запрос в языковую модель по средством API (сведения из файла udp.txt дополняются заранее заготовленными словосочетаниями для обеспечения лексической грамотности запроса).

5. Сформированный запрос подается на вход нейросетевой реализации.

6. Локальная языковая модель обрабатывает входящий запрос и отправляет ответ. Однако в случае возникновения сбоя программа уйдет в бесконечный цикл

ожидания. Для этого используется механизм оценки времениожидания ответа, и в случае, если ответ не был получен в течение заданного времени, алгоритм завершит свое выполнение.

Рис. 4. Блок-схема алгоритма вывода мер противодействия при поступлении сигнала об атаке

7. При успешном получении ответа от нейросетевой реализации, его данные сохраняются в локальный массив, хранящий историю чата. Причем он дополняется пометкой «assistant», что необходимо для того, чтобы отличать запросы администратора безопасности от ответов локальной языковой модели.

8. Происходит извлечение данных из локальных переменных и передача их на фронт-энд часть веб-приложения и последующий вывод списка мер реагирования и ликвидации последствий на экран.

Рис. 5. Продолжение блок-схемы алгоритма вывода мер противодействия при поступлении сигнала об атаке

Таким образом, результатом работы данного алгоритма является вывод списка мер противодействия, сформированного на основании сценария атаки, считанного из файла udp.txt, на экран веб-приложения, что позволяет администратору безопасности проводить их дальнейшую корректировку, алгоритм которой описан в следующем разделе.

Процедура корректировки мер противодействия

Ввиду требования к веб-приложению, согласно которому должна быть обеспечена возможность корректировки мер противодействия, сформирован данный алгоритм, который основа служит основой для реализации как первого, так и второго

режима работа. В первом случае он уместен после выполнения алгоритма вывода мер противодействия при поступлении сигнала об атаке, а во втором - может применятьсясразу, принимая входные запросы пользователя о сценарии атаки, при этом формируя дальнейшую корректировку на первом ответе.

Блок схема данного алгоритма изображена на рис. 6.

Ниже предлагается описание алгоритма корректировки мер противодействия:

1. Ввиду того, что данный алгоритм описывает работу веб-приложения, то концом его выполнения можно считать закрытие приложения.

2. В связи с первым пунктом, веб-приложение постоянно ждет либо закрытия, либо ввода запроса пользователем.

3. В случае поступления события «ввод данных» происходит запись этого запроса в локальную историю чата с добавлением роли «user».

4. На основание последнего сообщения из истории чата формируется запрос к локальной языковой модели и посредством API осуществляется его отправка.

5. После принятия запроса пользователя нейросетевая реализация формирует ответ. Как и в предыдущем алгоритме здесь предусмотрено ожидание ответа в течении определенного времени, дабы избежать вечного таймаута в результате ошибки на стороне нейросети.

6. В случае превышения времени ожидания программа выводит ошибку на экран.

7. Если же ответ был получен в временные рамки, то происходит его запись в локальную историю чата. При этом текстовым данным присваивается роль «assistant»

8. Последним шагом является вывод на экран сообщения (либо корректного ответа нейросетевой реализации, либо отчета об ошибке), после чего веб-приложение переходит к первого шагу, ожидая новый ввод пользователя или закрытие.

Таким образом, данный алгоритм позволяет обеспечить бесперебойную работу веб-приложения, являясь при этом основой для работы пользователя в двух режимах.

Алгоритмизация машинного обучения языковой модели

Использование локальной языковой модели несет за собой некоторые недостатки. Одним из самых серьезных является серьезные затраты по ресурсам системы и слабая обученность модели. Ввиду того, что будет использоваться языковая модель с 7 миллиардами параметров, возникает необходимость в ее обучении под конкретную специфику. Подобные модели содержат лишь логику, позволяющую выдавать текст на определенном языке без грамматических и синтаксических ошибок, и нуждаются в проведении дополнительного машинного обучения под цели регламентации мер противодействия сетевым вторжениям. Для достижения этих целей регламентации была сформирована база данных в формате xlsx файла, содержащая следующие колонки:

− идентификатор атаки (согласно CAPEC);

− название атаки;

− уязвимость;

− меры реагирования на регистрируемые инциденты, порожденные парой «атака-уязвимость»;

− меры ликвидации последствий инцидентов, порожденных парой «атака-уязвимость».

Пример из первичной базы данных для машинного обучения представлен на рис. 7.

Однако требования базы знаний для машинного обучения предусматривают наличие полей, содержащих сведения о поведении модели, запросе пользователя и ответе нейросети. Первичная база знаний не соответствует этим критериям, в связи счем нуждается в переформатировании.

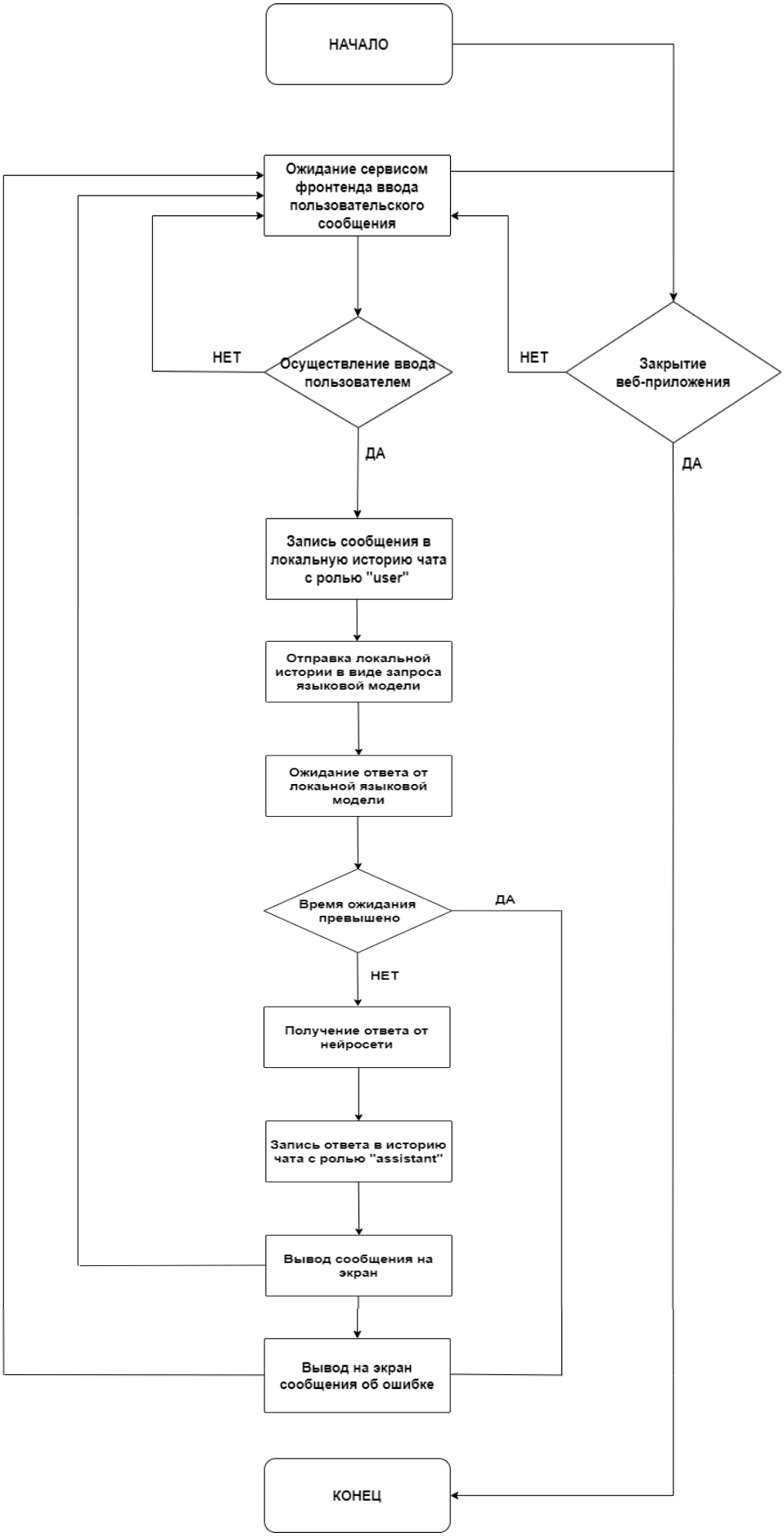

Рис. 6. Алгоритм корректировки мер противодействия

Идентификатор атаки	Название атаки	Уязвимость	Меры реагирования на регистрируемые инциденты, порожденные парой атака-уязвимость	Меры ликвидации последствий инцидентов, порожденных парой атака-уязвимость
CAPEC-59	Фальсификация учетных данных сеанса посредством прогнозирования	CVE-2023-25742	1. Проверить подлинность всех идентификаторов во время выполнения 2. Использовать тайм-аут сеанса для всех сеансов 3. Заблокировать доступ для учетных данных, которые могли быть скомпрометированы.	1. Использовать надежный источник случайности для создания идентификатора сеанса 2. Не использовать информацию, доступную пользователю, для генерации идентификатора сеанса (например, время) 3. Шифровать идентификатор сеанса
CAPEC-10	Переполнение буфера через переменные среды	CVE-2023-31066	1. Заблокировать подозрительные соединения. 2. Настроить веб-фильтры. 3. Изолировать вредоносный код. 4. Использовать auditd для мониторинга выполнения команд. 5. Использовать ps и kill для остановки процессов. 6. Отключить сервис cron для предотвращения запуска вредоносных задач. 7. Удалить запланированные задачи и задания. 8. Заблокировать IP-адреса злоумышленника. 9. Ограничить доступ к скомпрометированному узлу на время атаки.	1. Обеспечить анализ и мониторинг сетевого трафика. 2. Проверить все загруженные файлы на наличие вредоносного кода. 3. Обновить Apache InLog до версии 1.7.0 и выше. 4. Откатить изменения. 5. Использовать AIDE для проверки целостности. 6. Отследить все запланированные задачи. 7. Включить системы в безопасном режиме. 8. Удалить все лишние процессы из автозагрузки. 9. Восстановить уничтоженные данные из резервных копий.
CAPEC-10	Переполнение буфера через переменные среды	CVE-2022-2503	1. Дать доступ к системе только доверенным администраторам. 2. Провести аудит всех загруженных модулей ядра и удалить неподтвержденные. 3. Изолировать подозрительные машины в отдельный VLAN или сетевой сегмент. 4. Использовать инструменты мониторинга выполнения команд, такие как auditd, для отслеживания выполнения подозрительного кода. 5. Отключить подозрительные службы и задачи через systemctl или Task Scheduler, а также провести аудит всех служб автозагрузки. 6. Удалить все обнаруженные бекдоры и восстановить систему из чистых резервных копий. 7. Ограничить использование команд sudo и su. 8. Сменить пароли всех привилегированных учетных записей. 9. Ограничить доступ к логам только доверенных учетных записям. 10. Восстановить системные логи из резервных копий. 11. Ограничить доступ к конфиденциальным данным с использованием DLP-систем. 12. Использовать команды taskkill в Windows или kill в UNIX-подобных системах для остановки процессов вредоносного кода. 13. Заблокировать IP-адреса злоумышленника. 14. Ограничить доступ к скомпрометированному узлу на время атаки.	1. Настроить журналирование загрузки модулей ядра и отправку уведомлений о подозрительной активности. 2. Восстановить систему из чистых резервных копий, при необходимости. 3. Настроить систему контроля целостности файлов, такие как AIDE или Tripwire, для мониторинга изменений. 4. Провести аудит всех задач cron и автозагрузки для выявления изменений. 5. Настроить мониторинг изменений конфигураций систем с использованием OSSEC. 6. Настроить мониторинг использования привилегированных учетных записей и их действий. 7. Настроить использование двухфакторной аутентификации для привилегированных учетных записей. 8. Провести аудит всех привилегированных учетных записей и их действий. 9. Настроить защиту логов от изменения с использованием команды "chattr +i". 10. Настроить оповещения при попытках изменения логов через системы SIEM, такие как Splunk или ELK Stack, для немедленного уведомления о подозрительной активности. 11. Установить систему контроля целостности логов, такую как Wazuh, которая будет отслеживать изменения в логах и уведомлять о подозрительных действиях. 12. Провести анализ утечки данных и уведомить всех пострадавших о факте утечки. 13. Восстановить данные из резервных копий и усилить меры защиты, такие как шифрование данных. 14. Настроить оповещения о попытках передачи данных на внешние IP-адреса, что позволит оперативно выявлять и заблокировать утечки данных. 15. Включить системы в безопасном режиме. 16. Удалить все лишние процессы из автозагрузки. 17. Восстановить уничтоженные данные из резервных копий.

Рис. 7. Пример оформления первичной базы знаний для машинного обучения

Функция подготовки базы знаний для машинного обучения

Функция подготовки базы знаний для машинного обучения реализует алгоритм, преобразующий первичные сведения об атаках, уязвимостях и регламентах противодействия во вторичные, содержащие требуемые параметры машинного обучения. Следует отметить, что наиболее предпочтительным вариантом формата БД является так называемый «неправильный json». Он предусматривает наличие трех ролей:

– system – роль, описывающая форматы ответов и поведение нейросети, чтонужно для функционирования в рамках определенной сферы;

– user – роль, формирующая запрос пользователя;

– bot – роль, необходимая для выдачи ответа на запрос пользователя с учетом роли, описанной в system.

Причем эти роли адаптированы под json, формат, то есть каждая порция данныхимеет следующий вид: «{“system”:”текст”, “user”:”текст”, “bot”: “текст”}». В качестве разделителя между каждым элементом вторичной базы знаний выступает запятая. Пример вторичной базы знаний для машинного обучения приведен на рис. 8.

```
{"system": "поведение нейросети", "user": "запрос пользователя 1", "bot":
  "ответ модели 1"},
{"system": "поведение нейросети", "user": "запрос пользователя 2", "bot":
  "ответ модели 2"},
{"system": "поведение нейросети", "user": "запрос пользователя 3", "bot":
  "ответ модели 3"},
{"system": "поведение нейросети", "user": "запрос пользователя 4", "bot":
  "ответ модели 4"},
{"system": "поведение нейросети", "user": "запрос пользователя 5", "bot":
  "ответ модели 5"},
{"system": "поведение нейросети", "user": "запрос пользователя 6", "bot":
  "ответ модели 6"},
{"system": "поведение нейросети", "user": "запрос пользователя 7", "bot":
  "ответ модели 7"},
{"system": "поведение нейросети", "user": "запрос пользователя N", "bot":
  "ответ модели N"}
```

Рис. 8. Пример преобразованный под валидный формат для машиногообучения базы знаний

Для конвертации данных xlsx в json с учетом описанных выше ролей был разработан алгоритм подготовки базы знаний для машинного обучения, блок-схема которого изображена на рис. 9, 10.

Ниже приведено подробное описание данного алгоритма:

1. На вход алгоритма необходимо подать сведения о сценариях атак и мерах противодействия, оформленные в виде файла .xlsx с колонками, описанными в начале данного раздела.

2. Осуществляется проверка валидности формата входного файла, с учетом правильности его названия, которое должно соответствовать ML.xlsx, и верного наименования соответствующих колонок.

3. В случае некорректного формата скрипт завершает свою работу.

4. При правильной подготовке файла ML.xlsx происходит заход в цикл для извлечения полезных данных и формирования файла ML.json.

5. Каждая порция данных проходит проверку валидности, и в случае неудачицикл переходит на следующую итерацию.

6. При полном соответствии порции данных формату, описанному на рис. 9, происходит извлечение данных из строки таблицы и формирование json- подобного объекта с полями user, bot, system.

7. Сначала в поле system записывается заранее заготовленный сценарий поведения.

8. Затем в поле user осуществляется запись запроса пользователя на основание данных «Идентификатор атаки», «Название атаки», «Уязвимость».

9. Последним этапом формирования json-подобной сущности является запись в поле bot ответа, который содержит данные о мерах реагирования и ликвидации последствий, взятые из соответствующих колонок ML.xlsx.

10. После окончания создания json-подобного объекта происходит его запись в файл ML.json.

11. В результате выполнения цикла создается новый файл ML.json, в который записаны меры противодействия для сценариев атак.

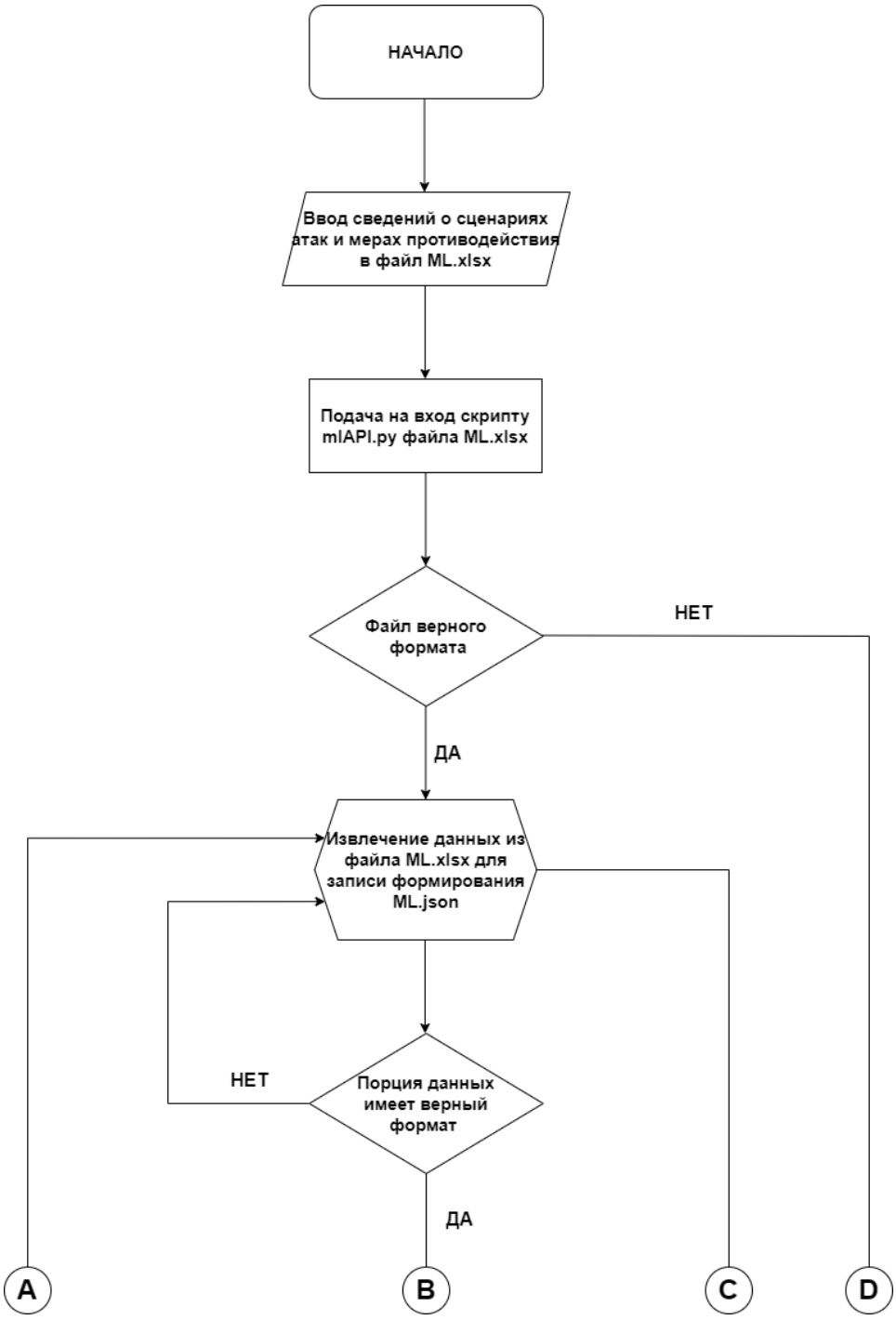

Рис. 9. Блок-схема алгоритма подготовки базы знаний для машинного обучения

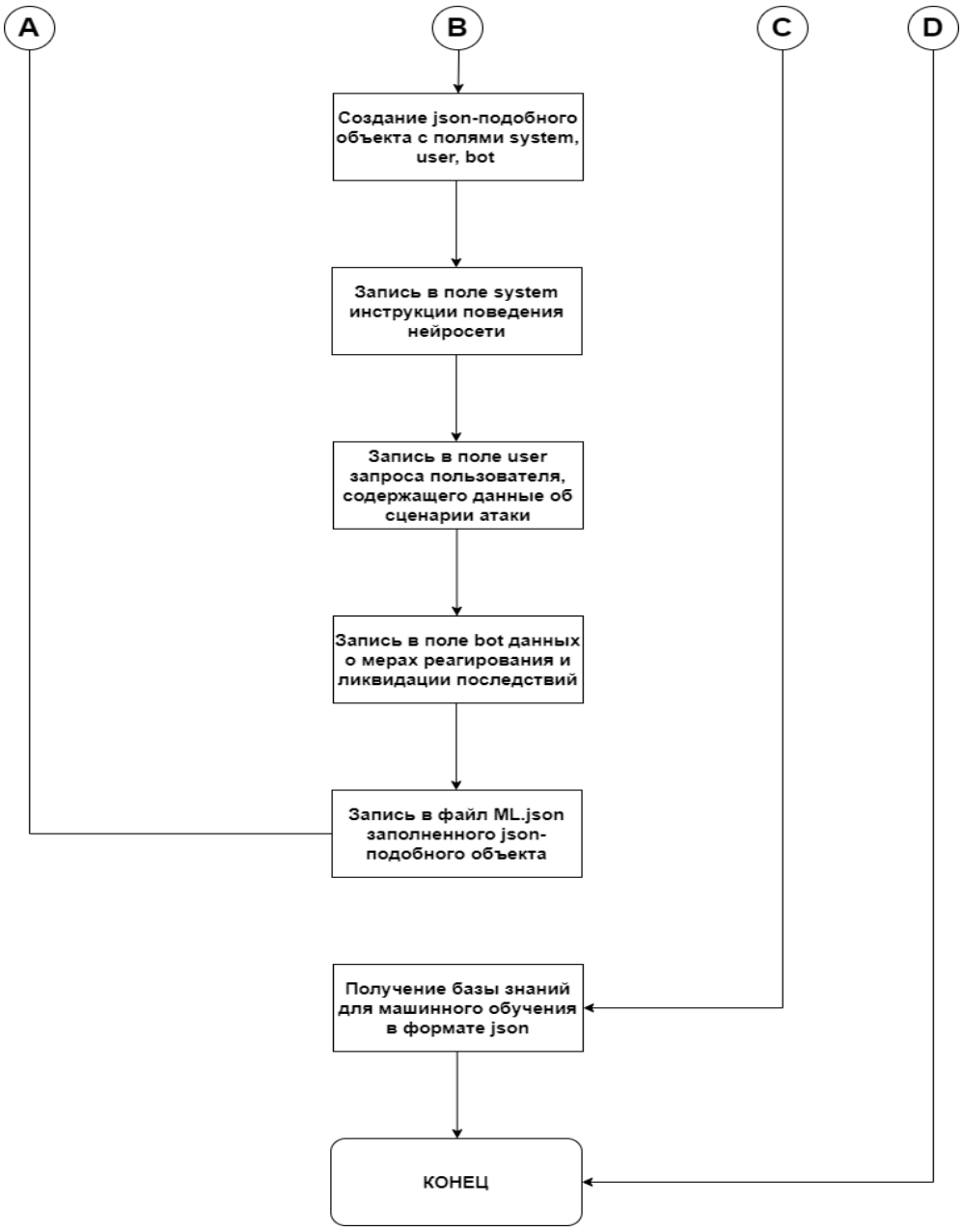

Рис. 10. Продолжение блок-схемы алгоритма подготовки базы знаний для машинного обучения

Процедура машинного обучения языковой модели

Без процедуры машинного обучения невозможно нормальное функционирование модуля регламентации мер противодействия, ввиду того что в противном случае необученная специфике информационной безопасности нейросетевая модель будет выдавать однообразные интеллектуальные подсказки, которые не смогут никак помочь администратору безопасности справиться с натиском кибератак, направленных на инфраструктуру локальной сети. Блок-схема данного алгоритма приведена на рис. 11-13.

Перед подробным описанием алгоритма стоит сделать замечание о том, что машинное обучение является достаточно затратным процессом в плане потребляемыхресурсов, их требуется как правило больше, чем для функционирования языковой модели. Из-за этого машинное обучение необходимо проводить на специализированном сервере либо же использовать платформы, предоставляющие мощности GPU и CPU, такие как Kaggle, Google Collab и другие. С учетом этих требований опишем алгоритм машинного обучения:

1. Если пропустить этап выбора платформы, на которой будет запускаться

машинное обучение, то начальным действием является загрузка с сайта HuggingFace языковой модели saiga_mistral_7b_lora.

2. В случае наличия необходимых мощностей GPU происходит установка значения устройства в «cuda». При таком раскладе обучение будет обеспечено за счет мощности видеокарты.

3. В случае отсутствия GPU, происходит установка значения устройства в «CPU». В таком случае обучение будет обеспечено за счет мощности процессора.

4. В независимости от выбора устройства машинного обучения происходит установка конфигурационных параметров для загруженной языковой модели.

5. На следующем этапе загружается датасет машинного обучения, имеющий формат «неправильный json».

6. В результате загрузки необходимой базы знаний необходимо произвести токенизацию системных полей датасета, в данном случае ими будут являться system, user и bot.

7. Далее происходит токенизация запросов и ответов, а также информации, описывающей поведение модели, подаваемых на вход нейросети.

8. Затем необходимо настроить параметры непосредственно машинного обучения, такие как количество эпох, оптимизатор, размер батча и другие.

9. На следующем шаге создается экземпляр входных данных, который будет передан в модель как параметр в функции запуска машинного обучения.

10. Происходит старт машинного обучения.

11. В результате запуска обучения могут возникнуть различные ошибки, такие как недостаток памяти и др.. В случае неудачного исхода необходимо будет повторить все шаги заново, начиная от загрузки языковой модели с сайта HuggingFace.

12. В случае успешного проведения машинного обучения локальной языковой модели будет получен файл .pt.

13. Для дальнейшего развертывания обученной языковой модели в LM Studio необходимо произвести ее преобразование в формат gguf, что и происходит на данном этапе.

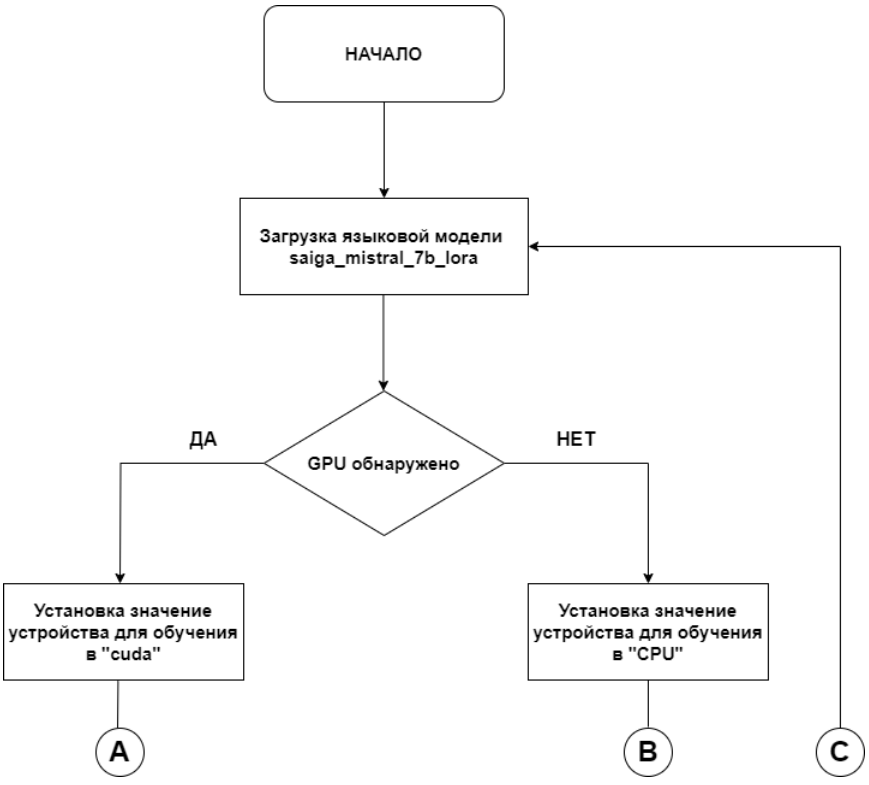

Рис. 11. Начало блок-схемы алгоритма машинного обучения

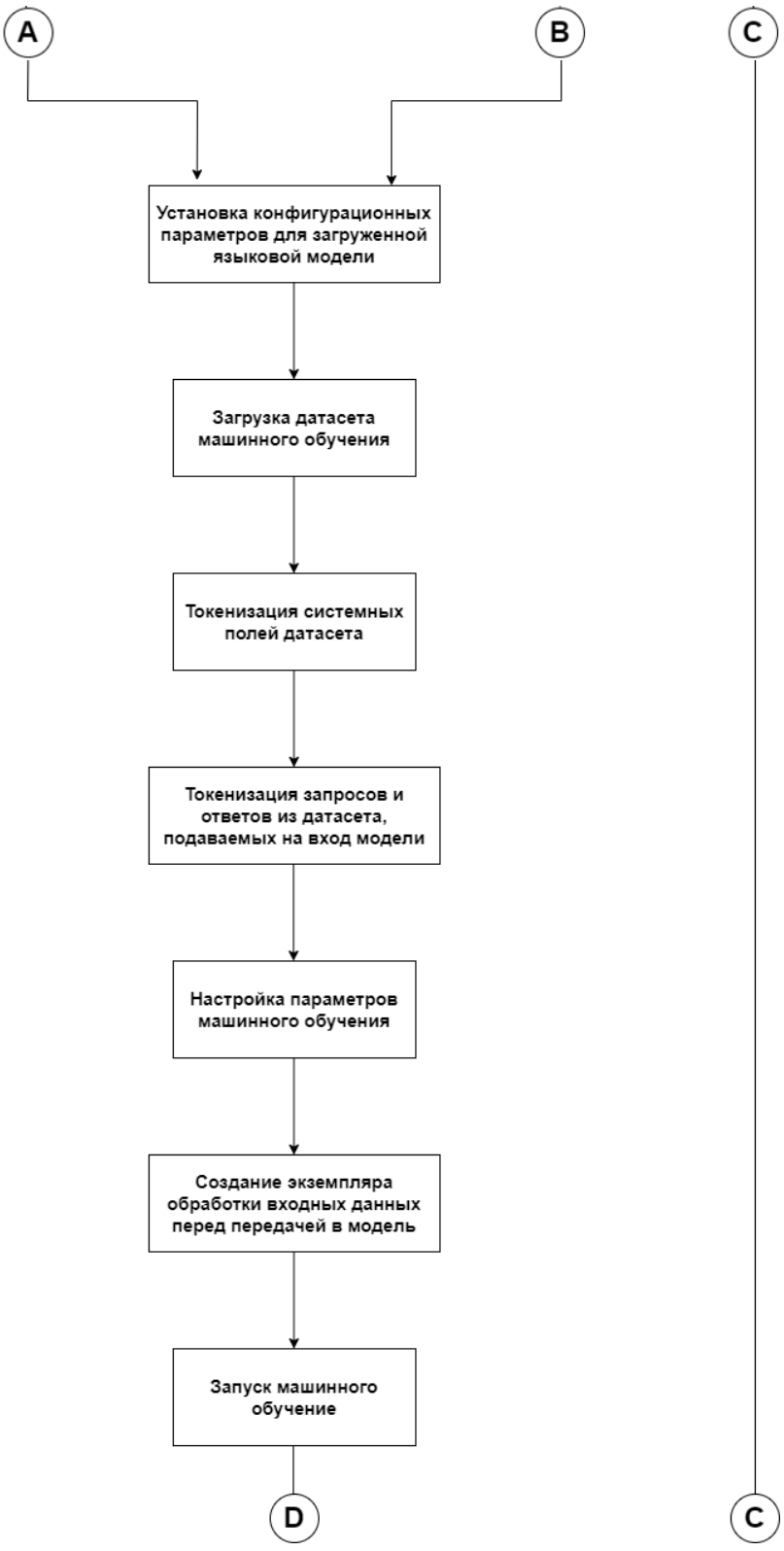

Рис. 12. Продолжение блок-схемы алгоритма машинного обучения

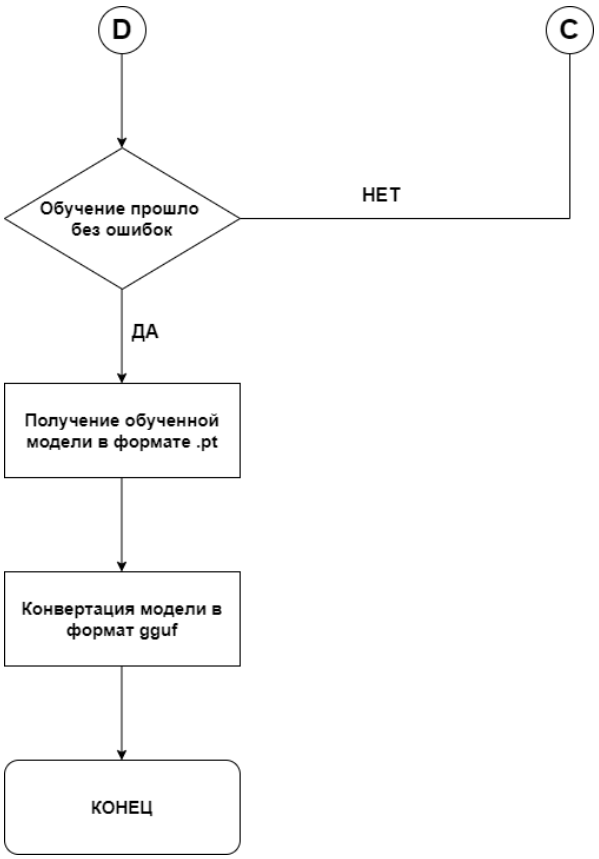

Рис. 13. Окончание блок-схемы алгоритма машинного обучения

На вход данному алгоритму подается сформированная база знаний машинного обучения в формате «неправильный json». На выходе получаем прошедшую дополнительное обучение языковую модель в формате gguf, что позволяет произвести ее развертывание и дальнейшее взаимодействие с веб- приложением посредством API.

Заключение

В работе были получены методическое и алгоритмическое обеспечение (описывающие: общую архитектуру проектируемого модуля регламентации мер противодействия сетевым вторжениям; взаимодействие с другимимодулями в рамках общей системы; принципы работы сервиса корректировки данных и веб-приложения, составляющего ядро этого сервиса) − обладающие следующей новизной:

− сформированное, в отличие от аналогов, обеспечение: каталогизируетновые методы агрегации данных, необходимые для обучения локальной языковой модели; задает архитектуру модуля регламентации мер противодействия с учетом обеспечения его изолированности и возможности встраиваться в локальные сети со своей спецификой; реализует алгоритмы взаимодействия веб-приложения с возможностью дополнительного машинного обучения языковой модели, что позволяет обеспечить большую валидность выдаваемых ответов;

− методическое обеспечение позволяет сформировать алгоритмическую базу для создания модуля регламентации мер противодействия сетевым вторжениям, описав его взаимодействие с другими модулями. В свою очередь алгоритмическое обеспечение является основой для создания программной реализации веб- приложения модуля регламентации мер противодействия наряду с проведением дополнительного машинного обучения языковой модели;

− предложенная архитектура, наряду с моделью машинного обучения локальной языковой модели, могут быть развиты специалистами по информационной безопасности для проектирования схожих систем, где предложенное алгоритмическое обеспечение может послужить в качестве

основы для создания программных реализаций на основе чат-ботов, использующих в своем ядре локальные языковые модели.

Список литературы

1. Остапенко Г.А. Нейросетевые задачи и компетенции проектной деятельности по созданию защищённых автоматизированных информационных систем / Г.А. Остапенко, А.П. Васильченко // Информация и безопасность. 2023. Т. 26.Вып. 4. С. 579-586.

2. MagicUnprotect // whatplugin.ai URL: https://www.whatplugin.ai/gpts/magicunprotect (дата обращения: 06.06.2024).

3. MITREGPT // TAAFT URL: https://theresanaiforthat.com/gpt/mitregpt/ (дата обращения: 06.06.2024).

4. ATT&CK Mate //whatplugin.ai URL: https://www.whatplugin.ai/gpts/attandck-mate (дата обращения: 06.06.2024).

5. Awesome-GPT-Agents // GtHub URL: https://github.com/fr0gger/Awesome- GPT-Agents (дата обращения: 06.06.2024).

6. Soc-copilot // GtHub URL: https://github.com/lewiswigmore/soc-copilot (датаобращения: 06.06.2024).

7. CVEs // whatplugin.ai URL: https://www.whatplugin.ai/plugins/cves-20bbd (дата обращения: 06.06.2024).

8. SpamGuard Tutor // TAAFTURL: https://theresanaiforthat.com/gpt/spamguard-tutor/#:~:text=SpamGuard Tutor is a GPT,mechanism to deliver these services (дата обращения: 06.06.2024).

9. Threat Model Companion // AllGPTs URL: https://allgpts.co/tools/threat- model-companion/ (дата обращения: 06.06.2024).

10. Prompt Injection Detector // AllGPTs URL: https://allgpts.co/tools/prompt- injection-detector/ (дата обращения: 06.06.2024).

11. OWASP LLM Advisor // TAAFT URL: https://theresanaiforthat.com/gpt/owasp-llm-advisor/ (дата обращения: 06.06.2024).

12. Threat Intel Bot-Бесплатная продвинутая кибербразведка // YesChat URL: https://www.yeschat.ai/ru/gpts-ZxX0quFr-Threat-Intel-Bot (дата обращения: 06.06.2024).

13. Systems Security Analyst-Expert Cybersecurity Insights // YesChat URL: https://www.yeschat.ai/gpts-2OToEiy6nT-Systems-Security-Analyst (дата обращения: 06.06.2024).

14. CyberGuard AI: Your Digital Shield-Digital Security Enhancement // YesChatURL: https://www.yeschat.ai/ru/gpts-2OToSkh3jy-CyberGuard-AI-Your-Digital-Shield (дата обращения: 06.06.2024).

15. Тарифы и оплата Gigachat // Сбербанк URL: https://developers.sber.ru/docs/ru/gigachat/api/tariffs (дата обращения: 13.05.2024).

Финансовый университет при Правительстве Российской Федерации
Financial University under the Government of the Russian Federation

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 17.07.2024

Информация об авторах

Остапенко Григорий Александрович – д-р техн. наук, профессор, Финансовый университет при Правительстве Российской Федерации, e-mail: ost@fa.ru

Васильченко Алексей Павлович – аспирант, Финансовый университет при Правительстве Российской Федерации, e-mail: rainichek@yandex.ru

Остапенко Александр Алексеевич – аспирант, Воронежский государственный технический университет, e-mail: alexostap123@gmail.com

Ноздрюхин Александр Александрович – студент, Воронежский государственный технический университет, e-mail: sfrvvv@yandex.ru

Остапенко Александр Григорьевич – д-р техн. наук, профессор, заведующий кафедрой, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Батаронов Игорь Леонидович– д-р физ.-мат. наук, профессор, заведующий кафедрой, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com
Кривошеин Александр Сергеевич – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

NEURAL NETWORK SERVICE FOR REGULATING MEASURES TO COUNTER CYBER ATTACKS (PART I)

G.A. Ostapenko, A.P. Vasilchenko, A.A. Ostapenko, A.A. Nozdryukhin, A.G. Ostapenko, I.L. Bataronov, A.S. Krivoshein

The neural network implementation of the automated warfare service is considered in terms of regulating measures to counter computer attacks. An analysis of existing chatbots applicable for the neural network implementation of the module for generating regulation of measures to counter network intrusions in automated information systems was carried out. Methodological support for organizing machine learning of a language model and the corresponding algorithmization have been developed, including interaction with the neural network identification module, the procedure for determining IP addresses, the function of receiving data on attacks and vulnerabilities, regulation of measures to counter network intrusions, the operation of outputting measures, the procedure for adjusting measures, the function of preparing a knowledge base for machine learning and the procedure for machine learning of a language model. The proposed methodological and algorithmic support catalogues new methods of data aggregation necessary for training a local language model and presents regulations for measures to counter network intrusions.

Keywords: cyberattack, vulnerability, network intrusion, neural network, neural network implementation of the module, language model, machine learning.

Submited 17.07.2024

Information about authors

Grigoriy A. Ostapenko – Dr. Sc. (Technical), Professor, Financial University under the Government of the Russian Federation, e-mail: ost@fa.ru
Alexsey P. Vasilchenko – graduate student, Financial University under the Government of the Russian Federation, e-mail: rainichek@yandex.ru
Aleksandr A. Ostapenko – graduate student, Voronezh State Technical University, e-mail: alexostap123@gmail.com
Aleksandr A. Nozdriuhin – student, Voronezh State Technical University, e-mail: sfrvvv@yandex.ru
Aleksandr G. Ostapenko – Dr. Sc. (Technical), Professor, Head of the Department, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com
Igor L. Bataronov – Dr. Sc. (Physical and Mathematical), Professor, Head of the Department, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com
Alexander S. Krivoshein – student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com