

УДК 004.855.6

DOI: [10.26102/2310-6018/2023.41.2.005](https://doi.org/10.26102/2310-6018/2023.41.2.005)

## Анализ состояния телекоммуникационных сетей с использованием графов знаний и управляемого автоматического машинного обучения

И.А. Куликов<sup>1</sup>✉, Н.А. Жукова<sup>2</sup>, М. Тяньсин<sup>3</sup>

<sup>1</sup>Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»,  
Санкт-Петербург, Российская Федерация

<sup>2</sup>Санкт-Петербургский федеральный исследовательский центр РАН,  
Санкт-Петербург, Российская Федерация

<sup>3</sup>Школа искусственного интеллекта Университета Джиллина, Чангчун,  
Китайская Народная Республика  
[i.a.kulikov@gmail.com](mailto:i.a.kulikov@gmail.com)✉

**Резюме.** В последнее время в качестве модели телекоммуникационных сетей и для хранения данных об их состоянии используются графы знаний. Графы знаний позволяют объединить в рамках одной модели множество частных моделей информационных систем, эксплуатируемых операторами, что делает возможным совместный анализ данных из разных источников и, как следствие, обеспечивает повышение эффективности решения задач управления сетью. Граф знаний дает возможность решать сложные прикладные задачи. Наполнение графа знаний требует обработки больших объемов сырых данных. Для их обработки требуется использовать алгоритмы машинного обучения, что при построении таких моделей затруднено ввиду изменений конфигураций современных сетей во времени, что требует частой перенастройки алгоритмов машинного обучения. Кроме того, сами по себе алгоритмы автоматизированного машинного обучения имеют высокую вычислительную сложность. Цель исследования – разработать подход, обеспечивающий возможность использования автоматизированного машинного обучения (AutoML) для анализа поступающих от сети оперативных данных за счет использования возможностей мета-майнинга для управления выбором алгоритмов машинного обучения и подбором гиперпараметров. Был использован метод определения состояния телекоммуникационной сети с использованием управляемого машинного обучения и мета-майнинга с последующим построением модели сети в виде графа знаний. Был разработан подход, позволяющий обеспечить управляемое машинное обучение при построении моделей телекоммуникационных сетей в виде графа знаний, обладающий сниженной вычислительной сложностью за счет уменьшения числа алгоритмов-кандидатов, подаваемых на вход AutoML. Приведены постановка и решение задачи обработки данных, поступающих от телекоммуникационной сети, представлено описание системы мониторинга, основанной на использовании предлагаемого подхода. Применение подхода проиллюстрировано на примере решения задачи определения состояния сети оператора кабельного ТВ.

**Ключевые слова:** граф знаний, AutoML, телекоммуникационная сеть, мета-обучение, мета-майнинг.

**Для цитирования:** Куликов И.А., Жукова Н.А., Тяньсин М. Анализ состояния телекоммуникационных сетей с использованием графов знаний и управляемого автоматического машинного обучения. *Моделирование, оптимизация и информационные технологии*. 2023;11(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1319> DOI: 10.26102/2310-6018/2023.41.2.005

# Analysis of telecommunication networks state using knowledge graphs and controlled automatic machine learning

I.A. Kulikov<sup>1</sup>, N.A. Zhukova<sup>2</sup>, M. Tianxing<sup>3</sup>

<sup>1</sup>*Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, the Russian Federation*

<sup>2</sup>*Saint Petersburg Federal Research Centre of the Russian Academy of Sciences (SPCRAS), Saint Petersburg, the Russian Federation*

<sup>3</sup>*School of Artificial Intelligence at Jilin University, Changchun, the People's Republic of China*  
*[i.a.kulikov@gmail.com](mailto:i.a.kulikov@gmail.com)*

**Abstract.** Nowadays, knowledge graphs are used as a model of telecommunication networks and for storing data on their state. Knowledge graphs make it possible to combine within one model many particular models of information systems used by operators, which allow joint analysis of data from various sources and, as a result, increase the efficiency of solving network management tasks. Knowledge graph helps to solve complex problems. Filling the knowledge graph requires processing large amounts of raw data. For their processing, it is necessary to use machine learning algorithms, which is difficult when building such models due to the fact that the configurations of modern networks change over time, which requires frequent reconfiguration of machine learning algorithms. In addition, automated machine learning algorithms have a high computational complexity. The purpose of the research is to develop an approach that makes it possible to employ automated machine learning (AutoML) to analyze live data coming from the network by means of metamining capabilities to control the choice of machine learning algorithms and the selection of hyperparameters. The method of determining the state of a telecommunications network using both managed machine learning and metamining, followed by building a network model in the form of a knowledge graph, was utilized. An approach has been developed to provide controlled machine learning when building models of telecommunication networks in the form of a knowledge graph, which has a reduced computational complexity by decreasing the number of candidate algorithms supplied to the AutoML input. The statement and solution of the problem of classifying the state of the vehicle according to the data coming from the network are given; a description of the monitoring system based on the use of the proposed approach is presented. The application of the approach is illustrated by the example of solving the task of determining the state of cable TV operator's network.

**Keywords:** knowledge graph, AutoML, telecommunication network, meta-learning, meta-mining.

**For citation:** Kulikov I.A., Zhukova N.A., Tianxing M. Analysis of telecommunication networks state using knowledge graphs and controlled automatic machine learning. *Modeling, Optimization and Information Technology*. 2023;11(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1319> DOI: 10.26102/2310-6018/2023.41.2.005 (In Russ.).

## Введение

При эксплуатации телекоммуникационных сетей (ТС) операторы с использованием систем мониторинга получают разнообразные данные о состоянии сетевых устройств в режиме реального времени. На основе этих данных операторы решают задачи анализа состояния и управления ТС. К таким задачам относятся анализ распределения сетевого трафика, поиск неисправностей сетевых устройств, анализ режимов работы оборудования для принятия решений об изменении конфигурации сетей и пр. В последнее время в качестве модели ТС и для хранения данных о состоянии ТС, поступающих от систем мониторинга, используются графы знаний (ГЗ) [1, 2]. Модели в виде ГЗ позволяют объединить в рамках одной модели множество частных моделей информационных систем, эксплуатируемых операторами, что делает возможным

совместный анализ данных из разных источников и, как следствие, обеспечивает повышение эффективности решения задач управления сетью. В значительном числе случаев в состав ГЗ ТС входят следующие частные модели:

1. графовые модели, описывающие статическую структуру сети [3];
2. графовые модели безопасности сетей (графы атак) [4-6];
3. графовые модели прав доступа пользователей к данным [7-9];
4. графовые модели пользовательских интерфейсов [10, 11].

Наполнение частных моделей данными осуществляется за счет сбора и обработки результатов измерений параметров, доступных для мониторинга. До настоящего времени решение задач сбора и обработки данных мониторинга продолжает вызывать существенные сложности из-за большого количества параметров и высокой скорости их изменения во времени. Количество таких параметров на каждом устройстве составляет от десятков до сотен в зависимости от типа сетевого устройства. Кроме этого, производительность графовых хранилищ снижается при значительном росте их размера [12].

Для анализа параметров, как правило, применяются методы машинного обучения (МО) [13], что частично решает проблему. Однако, существует ряд особенностей ТС, как источника данных о состоянии сетевых устройств, которые не позволяют использовать традиционный подход по выбору алгоритмов машинного обучения и оптимальных значений их гиперпараметров (<https://wiki.loginom.ru/articles/hyperparameters.html>) на этапе проектирования системы. К таким особенностям относится свойство ТС, как динамического объекта, менять свою конфигурацию во времени, кроме того, при изменении условий функционирования ТС могут меняться диапазоны значений параметров мониторинга, а при изменении состава устройств или их программного обеспечения могут меняться и перечни параметров устройств, подлежащих мониторингу. Предусмотреть все изменения в конфигурации и условиях эксплуатации сети при подборе оптимальных алгоритмов и параметров для обработки данных мониторинга при проектировании не представляется возможным. Одним из решений в этой ситуации является применение методов автоматизированного машинного обучения (AutoML) [14], обеспечивающих автоматический подбор как алгоритмов МО, так и оптимальных значений их гиперпараметров. Однако существующие решения для AutoML требуют больших временных затрат, поскольку они производят поиск лучшего алгоритма среди большого числа существующих алгоритмов МО. Это ограничение не позволяет оперативно изменять используемые алгоритмы и определять оптимальные значения гиперпараметров в случаях, когда из-за изменений конфигурации или условий работы сети ранее применяемые алгоритмы оказываются неспособными решать задачи обработки данных с приемлемой точностью.

В статье предлагается для управления выбором методов обработки данных ТС и определения оптимальных значений гиперпараметров совместно использовать ряд новых технологий области машинного обучения, в частности, Meta-Learning [15] и Meta-Mining [16], позволяющих сократить число алгоритмов-кандидатов на входе AutoML, что позволяет значительно снизить вычислительную сложность методов автоматизированного МО. Во втором разделе статьи приводятся описания технологий машинного обучения, предлагаемые для использования при анализе данных моделей ТС. В третьем разделе рассматриваются вопросы построения и анализа моделей ТС на основе графов знаний. В четвертом разделе представлен подход к построению графов знаний телекоммуникационных сетей, обогащенных данными о состоянии сетевых устройств, полученных с использованием автоматического машинного обучения. Архитектура систем мониторинга ТС, предусматривающая построение моделей ТС в виде графов знаний и ее анализ приведены в пятом разделе. В шестом разделе представлены

результаты сравнения управляемого AutoML, предусматривающего управление выбором алгоритма и значений его гиперпараметров при решении задачи классификации состояния устройств ТС по данным систем мониторинга с известными реализациями AutoML. В заключении сформулированы основные результаты работы и определены дальнейшие направления проведения исследований.

## Материалы и методы

**Современные технологии машинного обучения для построения ГЗ. Автоматическое машинное обучение.** Технология AutoML дает возможность специалистам прикладных предметных областей, не обладающим достаточной квалификацией в области машинного обучения, применять МО на практике. Первым программным средством AutoML, получившим широкое распространение, стал инструмент Auto-WEKA [17], который использует байесовскую оптимизацию для выбора и настройки алгоритмов машинного обучения. Auto-sklearn обеспечивает такие же возможности, используя библиотеку scikit-learn [18], кроме того, в нем предусмотрено применение мета-обучения, позволяющего осуществить быстрый старт при поиске лучших алгоритмов МО для обработки заданной выборки, который основывается на поиске ранее обработанных похожих наборов данных, а также обеспечена возможность создания ансамблей. Наконец, H2O AutoML [19] позволяет оптимизировать выбор алгоритмов из H2O путем совместного использования лучших решений, найденных в результате случайного поиска. Существенным ограничением для применения существующих фреймворков и библиотек AutoML на практике является их высокая вычислительная сложность, что обусловлено необходимостью перебора многих алгоритмов МО и многих комбинаций значений гиперпараметров.

**Мета-обучение.** Мета-обучение, или обучение обучению (Meta-learning) [20] предполагает применение методов машинного обучения к метаданным о результатах прошлых экспериментов, что позволяет конструировать процессы обработки новых данных на основе накопленного опыта.

Технологиями AutoML предусматривается возможность использовать мета-обучение для определения гиперпараметров для различных наборов данных. В частности, библиотека auto-sclearn использует мета-обучение для обеспечения так называемого «теплого старта» AutoML, когда выбор алгоритмов и гиперпараметров производится среди вариантов, хорошо себя показавших для близких наборов данных.

**Мета-майнинг.** Мета-майнинг (Meta mining) определяется как интеллектуальный анализ процессов Дата-майнинг (DM), управляемый одновременно метаданными и коллективным опытом майнеров данных, которые формализованы в виде онтологии DM и базы знаний (определения терминов «мета-майнинг» и «дата-майнинг» представлены в работах [21, 22]). Мета-майнинг позволяет выбирать алгоритмы и строить процессы обработки данных. В отличие от мета-майнинга, цели мета-обучения более конкретные, такие как определение параметров алгоритмов. Так в онтологиях DM могут размещаться знания о том, какие алгоритмы могут быть лучшими для данных, обладающих теми или иными характеристиками. Такие онтологии могут применяться и для алгоритмов машинного обучения. В этом случае становится возможным использовать мета-майнинг для снижения вычислительной сложности AutoML за счет кратного сокращения числа алгоритмов, которые рассматриваются как кандидаты при поиске лучших алгоритмов МО для обработки заданной выборки. Для определения кандидатов применяется технология мета-майнинга, при этом онтологии мета-майнинга формируются в результате применения технологий meta learning.

**Построение моделей ТС на основе графов знаний.** При построении моделей ТС в форме графов знаний предлагается использовать метод, рассмотренный в [22, 23]. Данный метод подразумевает использование алгоритмов индуктивного и дедуктивного синтеза. При индуктивном синтезе частные модели систем, функционирующих в составе телекоммуникационной сети, объединяются в единую иерархическую модель в виде графа знаний с учетом требований пользователей к составу модели и оперативных данных, поступающих от сети. Число уровней модели определяется иерархической структурой систем, входящих в ее состав. Для верификации актуальности состояния построенной модели используется дедуктивный синтез, который позволяет определить момент необходимости перестройки модели при изменении текущей конфигурации сети или условий ее функционирования. Рассматриваемый метод предусматривает объединение графовых моделей отдельных систем, входящих в ТС, в единую многоуровневую модель, а также обеспечивает возможность построения графов знаний, содержащих как статические данные о структуре сети, так и динамические данные мониторинга. Модель статических данных графа знаний телекоммуникационной сети представлена на Рисунке 1.

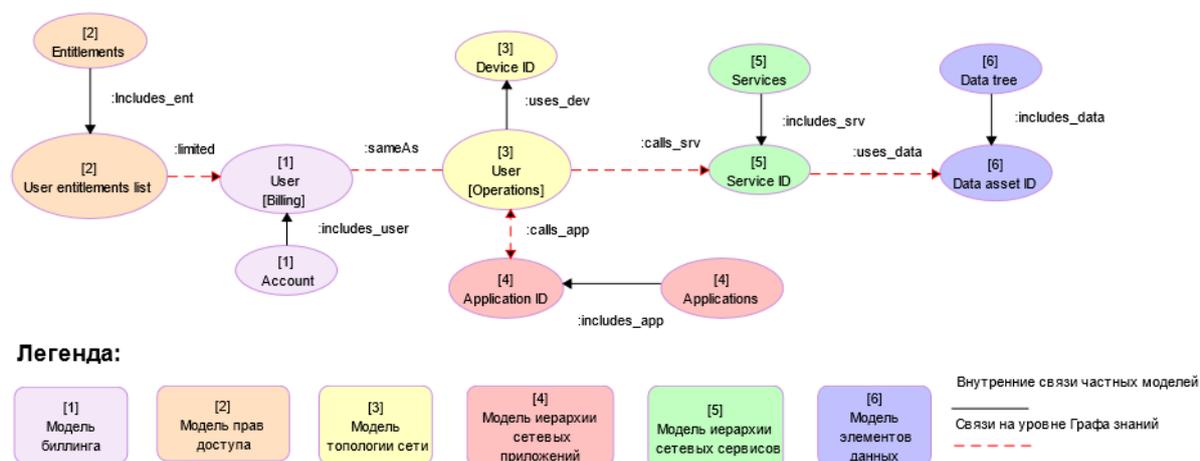


Рисунок 1 – Модель статических данных графа знаний  
Figure 1 – Model of knowledge graph static data

В состав статических данных ГЗ входят следующие:

- 1) биллинговая модель (Billing model);
- 2) модель прав доступа (Access rights model) (перечни прав доступа с точки зрения пользователей сети);
- 3) модель топологии сети (Network topology model);
- 4) модель иерархии сетевых приложений (Network applications hierarchy model);
- 5) модель иерархии сетевых сервисов (Network service hierarchy model);
- 6) модель элементов данных (Data model).

Динамические данные, входящие в состав ГЗ, включают:

- 1) данные традиционных систем мониторинга;
- 2) данные из log файлов сетевых устройств;
- 3) данные о действиях пользователей.

Динамические данные содержат отметку времени возникновения событий, поэтому для их описания используется структура Statement about Statement, представленная на Рисунке 2.

Структура данных, представленная на Рисунке 2, включает следующие элементы:

1. Requested node – элемент телекоммуникационной сети, который вызвал событие, зафиксированное при мониторинге.
2. Request ID – идентификатор события мониторинга.
3. Timestamp – отметка о времени возникновения события мониторинга.
4. is\_requested\_with (запрашивается вместе с) – тип связи между узлом идентификатора запроса и данными оператора.
5. Rdf:predicat – предикат события мониторинга (в зависимости от онтологической модели возможно несколько значений для предиката события мониторинга).
6. Object node – узел статической модели; объект мониторинга, участвующий в событии (например, сетевой сервис).
7. Subject node – статический узел модели; субъект мониторинга, участвующий в событии (например, элемент данных, связанный с объектом мониторинга).

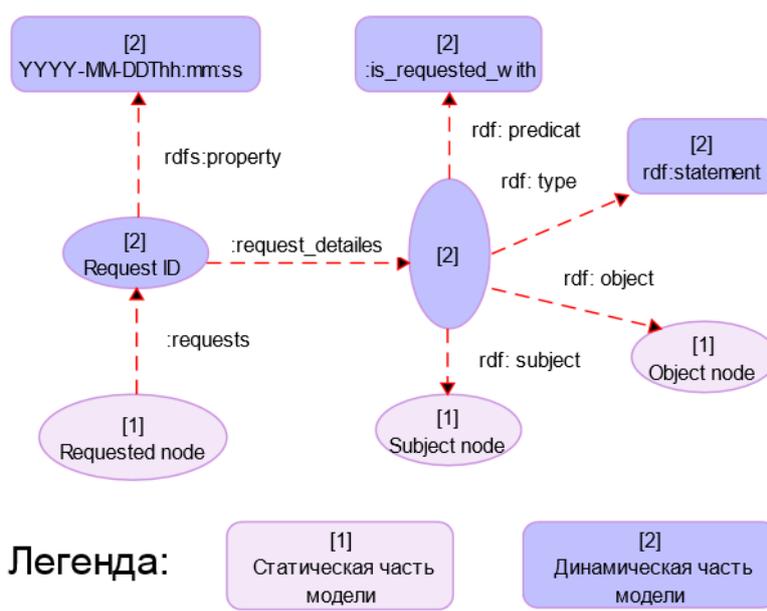


Рисунок 2 – Модель динамических данных графа знаний  
Figure 2 – Model of knowledge graph dynamic data

Динамические данные мониторинга интегрируются в модель графов знаний следующим образом:

1. Объект Request ID связан соответствующим предикатом с субъектом, статическим узлом графа знаний, вызвавшим событие мониторинга (пользователь, сервис, приложение и т. д.).
2. Пустой узел динамической модели данных Statement определяет триплет в статической модели (связи `rdfs:subject`, `rdfs:object`, `rdfs:predicat`), который в зависимости от характера события мониторинга определяет субъект события: Сервисы, Приложения, Элементы данных и т. д., тип связи события с объектом события (например, использует элемент данных) и объект события (ссылка на связанный элемент данных).

Модели статических и динамических данных для графа знаний используют единую онтологию. Граф знаний, построенный с использованием стандарта RDF, позволяет использовать стандартный язык запросов SPARQL для получения данных из графа знаний. Рассматриваемый метод применим для построения моделей различных телекоммуникационных сетей операторов, предоставляющих услуги связи, доступ к

приложениям и данным для конечных пользователей, использующих различные типы связи.

## Результаты

**Предлагаемый подход.** В значительном числе случаев для построения графа знаний ТС требуется предварительный анализ данных, поступающих от сети. Для анализа данных используются алгоритмы машинного обучения, однако ТС имеет динамическую структуру и ее параметры меняются во времени, что требует периодического замены используемых алгоритмов машинного обучения и переопределения их гиперпараметров. Для оптимизации процесса управления выбором алгоритмов и настройки их гиперпараметров целесообразно использовать методы AutoML, однако в своей базовой реализации они имеют высокую вычислительную сложность, что затрудняет их применение при обработке данных динамических ТС. Для преодоления этого ограничения при автоматическом выборе алгоритмов обработки данных о ТС при построении ГЗ предлагается применять технологии мета-майнинга, что обеспечивает снижение вычислительной сложности автоматического машинного обучения за счет сокращения числа кандидатов, рассматриваемых при поиске лучших алгоритмов. Для применения мета-майнинга авторами были расширены существующие онтологии DM&ML (от англ. Data mining and Meta learning ontology – онтология дата-майнинга и мета-обучения), которые содержат описания сложных взаимосвязей между задачами, данными и алгоритмами для различных этапов процесса DM&ML. Для возможности обработки данных конкретных предметных областей предусмотрено слияние онтологий DM&ML с онтологиями предметных областей, в которых размещаются описания характеристик наборов обрабатываемых данных. Для предметной области телекоммуникационных сетей разработаны ряд доменных онтологий, с которыми производится слияние онтологии DM&ML для решения поставленной задачи:

- 1) онтология NDL (язык описания сети) [24];
- 2) онтология домена телекоммуникационных услуг (TSDO) [25];
- 3) онтология беспроводной сети 3G [26];
- 4) мобильная онтология [27];
- 5) онтология для оптических транспортных сетей (OOTN) [28];
- 6) онтология, принятая в «Open Mobile Network» [29];
- 7) онтология TOUCAN (ToCo) [30].

Для описания знаний о процессе DM&ML предлагается использовать три онтологии, которые строятся путем интеграции существующих онтологий DM&ML и расширения их новыми сущностями (Рисунок 3).

*Базовая онтология DM.* Базовая Онтология DM строится за счет объединения и расширения существующих онтологий DM&ML (DMOP, DDT, DDM и DMWF) и построения онтологии «INPUT», которая используется для описания характеристик данных и решаемых задач. Базовая онтология DM описывает общие знания области DM&ML, включая алгоритмы, процессы, характеристики, цели и т. д.

*Онтология характеристик набора данных DM.* Онтология характеристик набора данных DM предназначена для описания характеристик наборов данных, требующих применения DM&ML для их обработки. Онтология построена на основе существующих онтологий DM&ML (OntoDT и стандарт IAO) и параметров, характеризующих особенности наборов данных. Онтология включает классы других онтологий и новые классы, предназначенные для представления характеристик наборов данных (Рисунок 4):

1. Класс «Data representational model» из IAO описывает основные типы единиц данных. Классы «Characterizing operation» и «Datatype generator» определяют доступные операции над типами данных.

2. Класс «Quality» из OntoDT определяет свойство типа данных («Datatype property») и свойство агрегации данных («Aggregate generator property»). «Dataset» определяет наборы данных DM, в то время как «Data collection» может быть любым вариантом агрегации некоторых единиц данных, таких как «Triple» или «Array». «Datatype property» описывает диапазон допустимых значений, порядок следования, точность, количество элементов. «Aggregate generator property» – однородность, уникальность, рекурсивность, структурированность и т. д.

3. Характеристики набора данных определяются с использованием их размерности, статистических метрик и метрик теории информации.

4. Класс «AlgorithmCharacteristics» описывает производительность алгоритмов DM, а также устойчивость к пропущенным значениям, шуму, определяет задачи, для решения которых алгоритмы могут применяться, например, двухклассовые, многоклассовые. Класс «AlgorithmCharacteristics» связан с классом «DatasetCharacteristics» свойством «suitableFor» и используется для выбора алгоритма.

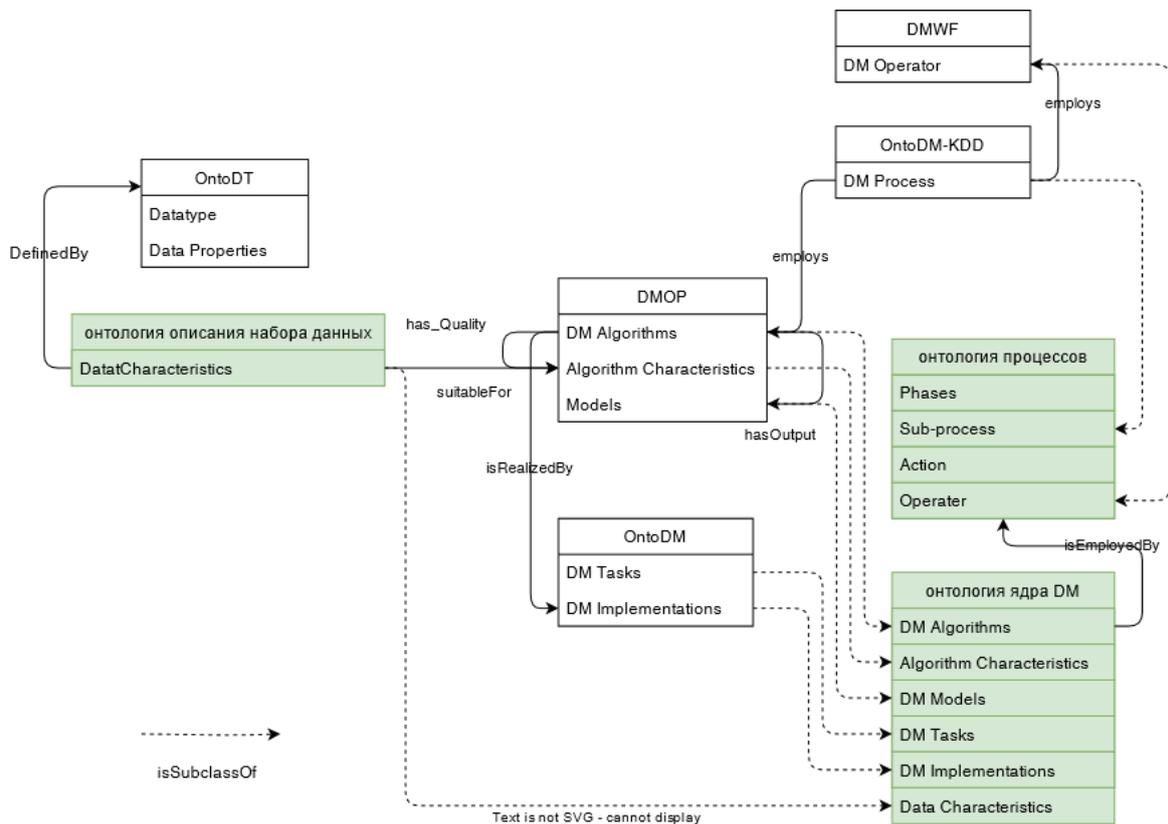


Рисунок 3 – Описание знаний о процессе DM&ML  
Figure 3 – Description of knowledge about DM&ML process

*Онтология процесса DM.* Онтология процессов DM имеет иерархическую структуру, что позволяет снизить вычислительную сложность построения процессов DM&ML. В онтологии выделяется четыре уровня представления процессов: уровень процессов, определенных в стандарте CRISP-DM (класс «CRISP-DM process») [31], уровень подпроцессов (класс «Subprocess»), уровень выполняемых действий (класс «Action») и уровень операторов (класс «Operator»). Описания процессов на первых трех

уровнях являются абстрактными. На нижнем, четвертом уровне, процессы описываются в виде операторов, которые могут быть выполнены программными системами. Онтология разработана в рамках CRISP-DM фреймворка. Между классами «CRISP-DM process», «Subprocess», «Action», «Operator» установлены отношения. Структура иерархической онтологии процесса DM представлен на Рисунке 5.

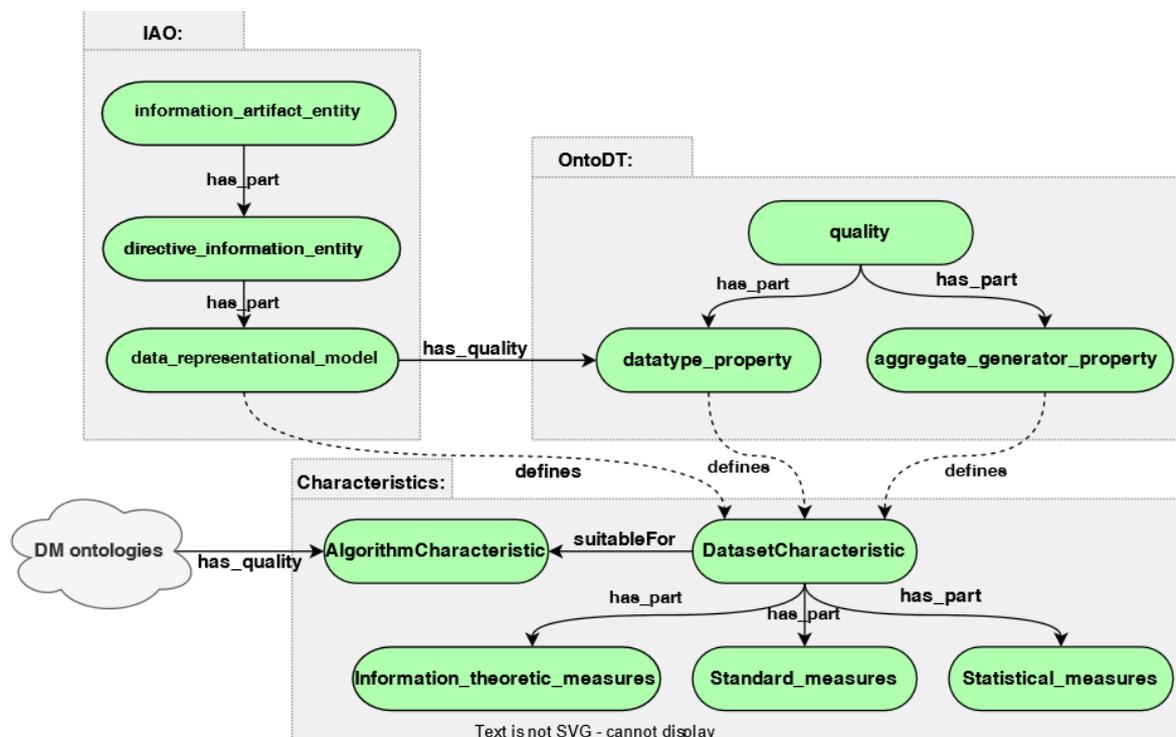


Рисунок 4 – Классы, используемые при описании онтологий  
Figure 4 – Classes used for ontologies description

Класс «CRISP-DM process» определяет общие фазы процесса DM. Свойство «has subprocess» используется для детализации фаз процессов за счет их представления в виде подпроцессов («Subprocess»). Для каждого подпроцесса определены входные и выходные данные. Свойство «has postprocessor» определяет порядок выполнения фаз и подпроцессов. «Actions» – используется для описания действий, которые необходимо выполнить для реализации каждого подпроцесса. С помощью «Operators» описывается реализация действий. Порядок выполнения действий и операторов определяется ограничениями, накладываемыми на их вход и выход. Для обеспечения выбора алгоритма классы, используемые для описания иерархических процессов, связываются с «Data Characteristics» и «Algorithm Characteristics». Предусловия и постусловия для процессов определены в «MetaData». Рассматриваемая онтология предполагает использование онтологий OntoDM-KDD [32] и DMWFontology. OntoDM-KDD обеспечивает абстрактное описание процессов DM&ML, DMWF предоставляет описание операторов и описание входных и выходных объектов (которые могут являться структурами данных, моделями, и т. д.).

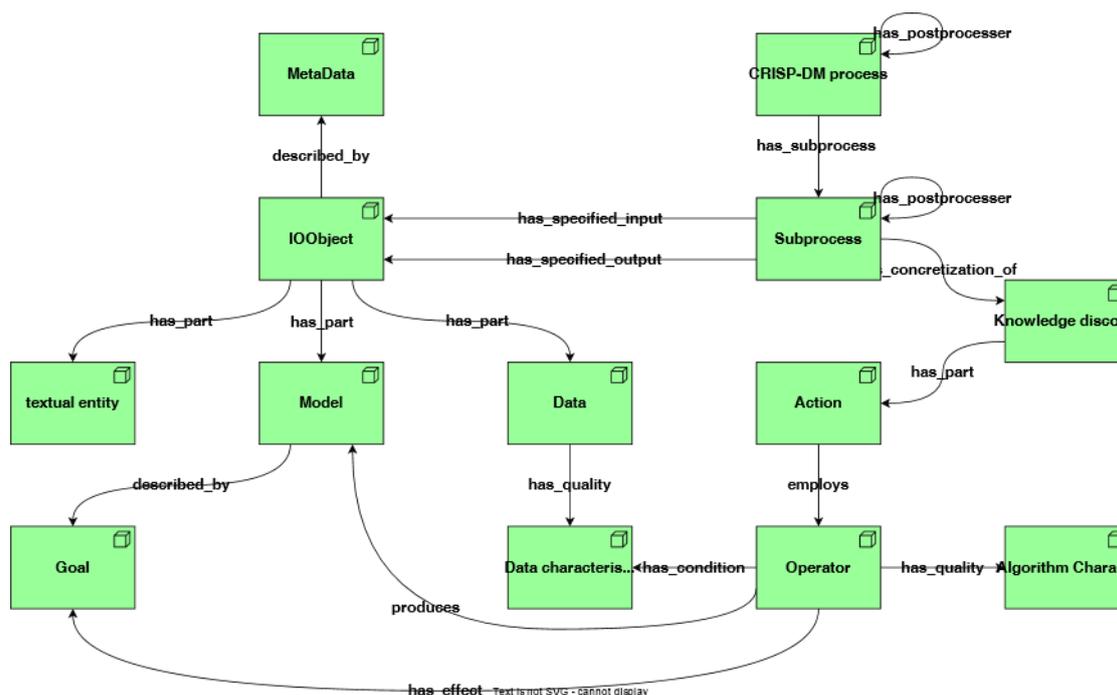


Рисунок 5 – Структура иерархической онтологии процесса DM  
Figure 5 – DM process hierarchical ontology structure

*Построение доменно-ориентированной DM&ML онтологии.* Построение процессов обработки данных требует применения знаний DM&ML и знаний предметной области. Онтологии DM&ML используются для описания экспертизы DM&ML и включают описания наборов данных DM&ML, алгоритмов DM&ML и процессов DM&ML. Доменные онтологии используются для описания характеристик наборов данных предметных областей. Онтология характеристик наборов данных как онтология верхнего уровня поддерживает построение онтологии предметной области для описания доменных наборов данных. Слияние двух онтологий на основе маппинга базовой онтологии DM на доменную онтологию позволяет построить доменную онтологию для обработки данных предметной области с применением алгоритмов машинного обучения.

**Структура системы мониторинга.** Структурная схема системы мониторинга на основе графа знаний представлена на Рисунке 6.

Система мониторинга содержит следующие компоненты:

I. Ядро системы мониторинга. Ядро включает:

1. Сервер приложений, обеспечивающий реализацию бизнес-логики функционирования системы в целом, включая взаимодействие с другими компонентами, шину передачи данных, функции обмена сообщениями и хранения файлов.
2. Динамический REST API, предоставляющий интерфейс для запросов внешних систем.
3. Набор адаптеров для запроса данных от внешних систем (систем мониторинга, ИТ-систем оператора и т. д.).
4. Веб-интерфейс для пользователей и администраторов системы.
5. Сервис построения отчетов с возможностью представления отчетов в WEB-интерфейсе или отправки их внешним потребителям.
6. Служба регистрации событий системы.
7. База данных SQL, предназначенная для хранения динамических данных мониторинга, которые в граф знаний не включаются.

## II. Граф знаний, включающий:

1. Хранилище данных RDF, соответствующее стандарту SPARQL 1.1. Этот компонент является ключевым элементом решения, содержащим триплеты графов знаний (статические и динамические компоненты). Компонент поддерживает функции добавления / удаления триплетов и поиска в RDF-хранилище. Хранилище также включает модуль анализа данных. В нем размещаются как статические, так и динамические графовые данные.

2. Репозиторий онтологий, хранящий копии всех онтологических моделей, на которых основан граф знаний. Используемые стандарты для описания данных и онтологии: RDF, RDFS (<https://www.w3.org/TR/rdf-schema/>), OWL (<https://www.w3.org/OWL/>).

3. Динамический REST API, поддерживающий интерфейс для взаимодействия с внешними системами, в частности, с ядром системы мониторинга.

## III. ИТ-системы оператора, предоставляющие статические данные для графа знаний. В рамках предлагаемой системы мониторинга рассматриваются следующие ИТ-системы оператора:

1. ИТ-система для управления сетевой инфраструктурой, предоставляющая данные о топологии сети, сетевых устройствах, сетевых сервисах, сетевых приложениях, доступных данных и правах доступа.

2. Биллинговая система, предоставляющая данные о пользователях, их устройствах, личных счетах, тарифах и платежах.

3. CRM-системы, предоставляющие данные об истории взаимодействия оператора с пользователем.

## IV. Традиционные системы мониторинга (могут использоваться в качестве источников агрегированных данных мониторинга). Традиционные системы мониторинга включают в себя:

1. Сервер приложений, поддерживающий бизнес-логику системы.

2. Динамический сервис REST API, предоставляющий интерфейс для взаимодействия с внешними системами, в частности, с ядром системы мониторинга.

3. Собственное SQL-хранилище данных сетевого мониторинга.

## V. Собственные системные агенты, поставляющие данные мониторинга ТС для ядра системы мониторинга.

## VI. Система построения и поддержания в актуальном состоянии графовой модели ТС, включающая следующие компоненты:

1. Модуль построения процессов обработки данных ТС.

2. AutoML модуль, позволяющий использовать автоматизированное машинное обучение при построении графа знаний телекоммуникационной сети.

3. DM&ML онтология, позволяющая сократить число кандидатов, рассматриваемых при выборе алгоритмов обработки данных согласно предложенного авторами подхода.

В качестве исходной информации для анализа при помощи алгоритмов машинного обучения используются следующие данные:

1. Значения параметров сетевых устройств (свой перечень параметров для каждого из типов сетевых устройств).

2. Значения принятого / переданного трафика.

Сообщения об ошибках.

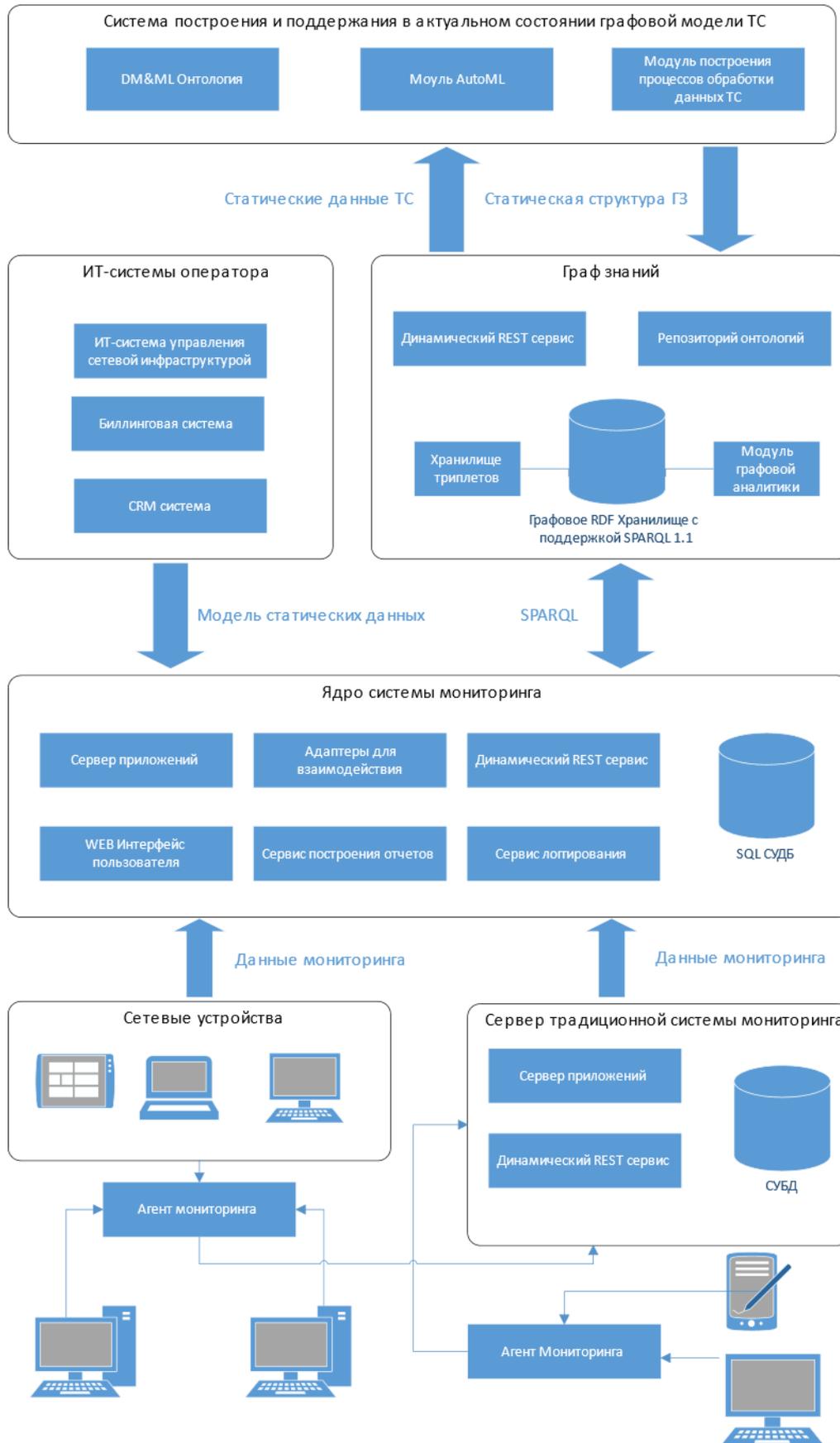


Рисунок 6 – Структурная схема системы мониторинга на основе графа знаний  
Figure 6 – Monitoring system flowchart based on knowledge graph

### Обсуждение

Для оценки предложенного подхода решается задача классификации состояния отдельных устройств ТС по данным, поступающим от системы мониторинга сети с использованием протокола SNMP. Полученные результаты классификации состояния сетевых устройств добавляются в модель ТС в виде графа знаний как статистические данные. Для решения задачи классификации состояния устройств телекоммуникационной сети был выбран набор данных AN-SNMP Dataset 2016 [33], представляющий собой 4998 записей, состоящий из значений 34 параметров сетевых устройств для различных режимов их функционирования, с точки зрения осуществления различного рода сетевых атак. Решение задачи классификации состояния каждого из устройств позволяет определить его состояние в каждый момент времени и зафиксировать его в динамической части модели ТС в виде графа знаний.

Исследование предложенного подхода проводилось с использованием трех библиотек AutoML (AutoWEKA, H2O, Auto-Sklearn). Сравнивалась производительность (время работы), обеспечиваемая при применении предложенного подхода, и производительность существующих библиотек AutoML. В обоих случаях для оптимизации гиперпараметров и расчета показателей эффективности использовались сами библиотеки AutoML. В экспериментах онтология DM&ML применялась для сокращения набора алгоритмов, которые рассматривались библиотеками в качестве кандидатов на лучшие алгоритмы. В состав библиотеки AutoWEKA входит 40 базовых алгоритмов классификации, 6 базовых алгоритмов классификации включены в состав библиотеки H2O, 16 базовых алгоритмов классификации поддержаны в библиотеке Auto-Sklearn. Применение мета-майнинга позволило существенно сократить число алгоритмов, среди которых библиотеками AutoML выполнялся поиск при определении лучшего алгоритма, в частности, для AutoWEKA до 3-6 алгоритмов, для H2O до 1-2 алгоритмов и для auto-sclearn до 2-4 алгоритмов. Для всех тестовых наборов данных лучший алгоритм, выбранный библиотеками AutoML, был среди алгоритмов в наборы кандидатов, выбранных с использованием предлагаемого подхода. Метрики алгоритмов классификации, выбранных с использованием предложенного подхода после оптимизации гиперпараметров были как минимум не хуже показателей лучших алгоритмов, отобранных AutoML. Время работы предложенного подхода рассчитывалось как сумма времени запроса к онтологии AutoML и времени оптимизации гиперпараметров для выбранного набора кандидатов, далее полученное время сравнивалось со временем работы библиотек AutoML (Таблица 1). Результаты показали, что использование предложенного подхода позволило сократить время работы в 0,6–15 раз для AutoWEKA, в 3-6 раз для H2O и в 3,75-7,5 раза для Auto-Sklearn.

Таблица 1 – Результаты применения предложенного подхода при решении задачи классификации текущего состояния телекоммуникационной сети по данным системы мониторинга

Table 1 – Results of applying the proposed approach for solving the problem of telecommunications network current state classification according to data obtained from the monitoring system

AutoML библиотека	S <sub>i</sub>	S <sub>r</sub>	T <sub>i</sub> , мин.	T <sub>q</sub> , мин.	T <sub>r</sub> , мин.
AutoWEKA	40	4 ± 2	15	0,1 ± 0,03	12 ± 11
H2O	6	2 ± 1	15	0,085 ± 0,015	4 ± 1.5
Auto-slearn	16	3 ± 1	15	0,085 ± 0,015	3 ± 1

Где  $S_i$  – исходное число алгоритмов, среди которых выполняется поиск лучшего,  $S_r$  – сокращенное число алгоритмов, рассматриваемых при поиске,  $T_i$  – время поиска лучшего алгоритма при рассмотрении всех алгоритмов,  $T_q$  – время запроса к DM&ML онтологии,  $T_r$  – время обработки для  $S_r$ .

Данные о состоянии элементов ТС добавляются в модель в виде ГЗ следующим образом (данные в формате RDF/XML, <https://www.w3.org/TR/rdf-syntax-grammar/>):

```
<rdf:Description rdf:about='http://127.0.0.1/tnmo/TN_Event_1/'>
  <rdf:type>tnmo:TN_Event</rdf:type>
  <tnmo:event_timestamp
rdf:datatype='http://www.w3.org/2001/XMLSchema#datetime'>2022-06-
30T06:08:26</tnmo:event_timestamp>
  <tnmo:has_event_type>tnmo:device_state</tnmo:has_event_type>
  <tnmo:event_details>
  <rdf:Description>
  <rdf:type>rdf:statement</rdf:type>
  <rdf:predicat>tnmo:device_security_state</rdf:predicat>
  <rdf:subject><rdf:Description
rdf:about='http://127.0.0.1/tnmo/Device_52/'></rdf:Description></rdf:subject>
  <rdf:object>normal</rdf:object>
  </rdf:Description>
  </tnmo:event_details>
  <tnmo:tn_event>
  <rdf:Description rdf:about='http://127.0.0.1/tnmo/Device_52/'></rdf:Description>
  </tnmo:tn_event>
  </rdf:Description>
```

Для описания данных о состоянии ТС использовалась онтология систем мониторинга TNMO (от англ. Telecommunication Network Monitoring Ontology – онтология систем мониторинга телекоммуникационных сетей [34]).

Результаты применения предлагаемого подхода для набора данных AN-SNMP Dataset 2016 доступны по адресу <https://github.com/kulikovia/AutoML>)

### Заключение

Предложен новый подход, позволяющий применять управляемое автоматизированное машинное обучение при построении моделей телекоммуникационных сетей по данным систем мониторинга. Для снижения вычислительной сложности выбора алгоритмов обработки данных предложено применять технологию мета-майнинга, за счет чего достигается сокращение числа алгоритмов, рассматриваемых при поиске лучших алгоритмов существующими библиотеками AutoML. Представленный подход позволяет преодолеть ограничения по применению существующих библиотек AutoML, в частности, длительное время работы библиотек. В рамках исследования предлагаемого подхода была проведена сравнительная оценка скорости решения задачи определения состояния отдельных устройств ТС по данным от системы мониторинга для последующего построения модели ТС в формате графа знаний с использованием предлагаемого подхода, предусматривающего управление выбором алгоритмов машинного обучения, и без него.

Работы, планируемые на будущее: 1) Расширение возможностей онтологий за счет добавления знаний о гиперпараметрах и вариантов задания их значений; 2) Обеспечение простой интеграции с существующими библиотеками DM&ML и AutoML

в рамках предлагаемой онтологии DM&ML для выбора лучших алгоритмов; 3) Разработка онлайн-сервиса DM&ML для свободного использования; 4) Определение состояния всей сети или ее сегмента с использованием предлагаемого подхода.

### СПИСОК ИСТОЧНИКОВ

1. Krinkin K., Vodyaho A., Kulikov I., Zhukova N. Models of Telecommunications Network Monitoring Based on Knowledge Graphs. *9th Mediterranean Conference on Embedded Computing (MECO)*. 2020;1–7. DOI: 10.1109/MECO49872.2020.9134148.
2. Krinkin K., Kulikov I., Vodyaho A., Zhukova N. Architecture of a Telecommunications Network Monitoring System Based on a Knowledge Graph. *26th Conference of Open Innovations Association (FRUCT)*. 2020;231–239. DOI: 10.23919/FRUCT48808.2020.9087429.
3. Wählisch M. Modeling the network topology. In: *Wehrle K., Güneş M., Gross J. (eds) Modeling and Tools for Network Simulation*. Berlin, Springer; 2010. 565 p. DOI: 10.1007/978-3-642-12331-3\_22.
4. Lallie H.S., Debattista K., Bal J. A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review*. 2020;35. DOI: 10.1016/j.cosrev.2019.100219.
5. Barik M., Sengupta A., Mazumdar C. Attack graph generation and analysis techniques. *Defence Science Journal*. 2016;66(6):559–567. DOI: 10.14429/dsj.66.10795.
6. Ou X., Singhal A. Attack graph techniques. In: *Quantitative Security Risk Assessment of Enterprise Networks. SpringerBriefs in Computer Science*. New York, Springer; 2012. 41 p. DOI: 10.1007/978-1-4614-1860-3\_2.
7. Sandhu R. A perspective on graphs and access control models. *Graph Transformations, ICGT 2004. Lecture Notes in Computer Science*. 2004;3256. DOI: 10.1007/978-3-540-30203-2\_2.
8. Lawall A., Schaller T., Reichelt D. Resource management and authorization for cloud services. *Proceedings of the 7th International Conference on Subject-Oriented Business Process Management. ACM*. 2015;18:1–8. DOI: 10.1145/2723839.2723864
9. Ionita C., Osborn S. Privilege administration for the Role Graph Model. *Research Directions in Data and Applications Security. IFIP – The International Federation for Information Processing*. 2003;128. DOI: 10.1007/978-0-387-35697-6\_2.
10. Lumertz P.R., Ribeiro L., Duarte L.M. User interfaces metamodel based on graphs *Journal of Visual Languages & Computing*. 2016;32:1–34. DOI: 10.1016/j.jvlc.2015.10.026.
11. Arrue M., Vigo M., Abascal J. Including heterogeneous web accessibility guidelines in the development process. *Engineering Interactive Systems. EHCI 2007. Lecture Notes in Computer Science*. 2008;4940. DOI: 10.1007/978-3-540-92698-6\_37.
12. Bizer C., Schultz A. The Berlin SPARQL benchmark. *Int. J. Semantic Web Inf. Syst.* 2009;5(2):1–24. DOI: 10.4018/jswis.2009040101.
13. Manna A., Alkasassbeh M. Detecting network anomalies using machine learning and SNMP-MIB dataset with IP group. *2nd International Conference on new Trends in Computing Sciences (ICTCS)*. 2019;1–5. DOI: 10.1109/ICTCS.2019.8923043.
14. Hutter F., Kotthoff L., Vanschoren J. (eds) Automated machine learning. *The Springer Series on Challenges in Machine Learning*. Cham, Springer; 2019. 220 p. DOI: 10.1007/978-3-030-05318-5.
15. Vanschoren J. Meta-Learning. In: *Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning*. Springer, Cham; 2019. 220 p. DOI: 10.1007/978-3-030-05318-5\_2.

16. Wirth R., Hipp J. CRISP-DM: Towards a standard process model for data mining. *Proc. of the 4th int. conf. PAKDD*. 2000;1:29–39.
17. Thornton C., Hutter F., Hoos H., Leyton-Brown K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *KDD*. 2012. DOI: 10.1145/2487575.2487629.
18. Feurer M., Klein A., Eggenberger K., Springenberg J., Blum M., Hutter F. Auto-sklearn: efficient and robust automated machine learning. In: *Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning*. Cham, Springer; 2019. 220 p. DOI: 10.1007/978-3-030-05318-5\_6.
19. LeDell E. H2O AutoML: scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning*. 2020.
20. Hutter, F., Kotthoff, L., Vanschoren, J. (eds) *Automated Machine Learning. The Springer Series on Challenges in Machine Learning*. Cham, Springer; 2019. DOI: 10.1007/978-3-030-05318-5\_2.
21. Wirth R., Hipp J. CRISP-DM: Towards a standard process model for data mining. *Proc. of the 4th int. conf. PAKDD*. 2000;1:29–39.
22. Krinkin K., Vodyaho A., Kulikov I., Zhukova N. Method of multilevel adaptive synthesis of monitoring object knowledge graphs. *Applied Sciences*. 2021;11(14):6251. DOI: 10.3390/app11146251.
23. Krinkin K., Vodyaho A., Kulikov I., Zhukova N. Deductive synthesis of networks hierarchical knowledge graphs. *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*. 2021;12(3):32–48. DOI: 10.4018/IJERTCS.2021070103.
24. van der Ham J.J. *A semantic model for complex computer networks: the network description language*. Thesis. Citeseer; 2010. 154 p.
25. Qiao X., Li X., Fensel A., Su F.. Applying semantics to Parlay-based services for telecommunication and Internet networks. *Open Computer Science*. 2011;1(4):406–429. DOI: 10.2478/s13537-011-0029-6.
26. Cleary D., Danev B., O'Donoghue D. *Using ontologies to simplify wireless network configuration*. FOMI; 2005.
27. Villalonga C., Strohbach M., Snoeck N., Sutterer M., Belaunde M., Kovacs E., Zhdanova A.V., Goix L.W., Droegehorn O. Mobile ontology: Towards a standardized semantic model for the mobile domain. *International Conference on Service-Oriented Computing*. 2007;248–257. DOI 10.1007/978-3-540-93851-4\_25.
28. Barcelos P.P.F., Monteiro M.E., Simoes R. de M., Garcia A.S., Segatto M.E.V. Ootn-an ontology proposal for optical transport networks. *IEEE ICUMT*. 2009;1–7. DOI: 10.1109/ICUMT.2009.5345459.
29. Uzun A., Kupper A. OpenMobileNetwork: extending the web of data by a dataset for mobile networks and devices. *ACM ICSS*. 2012;17–24. DOI: 10.1145/2362499.2362503.
30. Zhou Q., Gray A.J.G., McLaughlin S. ToCo: An ontology for representing hybrid telecommunication networks. *The Semantic Web. ESWC 2019. Lecture Notes in Computer Science*. 2019;11503. DOI: 10.1007/978-3-030-21348-0\_33.
31. Wirth R., Hipp J. Crisp-dm: Towards a standard process model for data mining. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. 2000;1:29–39.
32. Panov P., Soldatova L., Dzeroski S. Ontodm-kdd: ontology for representing the knowledge discovery process. *Int. Conf. on Discovery Science*. 2013;126–140. DOI: 10.1007/978-3-642-40897-7\_9.
33. Alkasassbeh M. *AN-SNMP Dataset*. 2016. DOI: 10.13140/RG.2.2.26384.30721.

34. Kulikov I., Vodyaho A., Stankova E., Zhukova N. Ontology for knowledge graphs of telecommunication network Monitoring systems. *Computational Science and Its Applications – ICCSA 2021. ICCSA 2021. Lecture Notes in Computer Science*. 2021;12956. DOI: 10.1007/978-3-030-87010-2\_32.

## REFERENCES

1. Krinkin K., Vodyaho A., Kulikov I., Zhukova N. Models of Telecommunications Network Monitoring Based on Knowledge Graphs. *9th Mediterranean Conference on Embedded Computing (MECO)*. 2020;1–7. DOI: 10.1109/MECO49872.2020.9134148.
2. Krinkin K., Kulikov I., Vodyaho A., Zhukova N. Architecture of a Telecommunications Network Monitoring System Based on a Knowledge Graph. *26th Conference of Open Innovations Association (FRUCT)*. 2020;231–239. DOI: 10.23919/FRUCT48808.2020.9087429.
3. Wählisch M. Modeling the network topology. In: *Wehrle K., Güneş M., Gross J. (eds) Modeling and Tools for Network Simulation*. Berlin, Springer; 2010. 565 p. DOI: 10.1007/978-3-642-12331-3\_22.
4. Lallie H.S., Debattista K., Bal J. A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review*. 2020;35. DOI: 10.1016/j.cosrev.2019.100219.
5. Barik M., Sengupta A., Mazumdar C. Attack graph generation and analysis techniques. *Defence Science Journal*. 2016;66(6):559–567. DOI: 10.14429/dsj.66.10795.
6. Ou X., Singhal A. Attack graph techniques. In: *Quantitative Security Risk Assessment of Enterprise Networks. SpringerBriefs in Computer Science*. New York, Springer; 2012. 41 p. DOI: 10.1007/978-1-4614-1860-3\_2.
7. Sandhu R. A perspective on graphs and access control models. *Graph Transformations, ICGT 2004. Lecture Notes in Computer Science*. 2004;3256. DOI: 10.1007/978-3-540-30203-2\_2.
8. Lawall A., Schaller T., Reichelt D. Resource management and authorization for cloud services. *Proceedings of the 7th International Conference on Subject-Oriented Business Process Management. ACM*. 2015;18:1–8. DOI: 10.1145/2723839.2723864
9. Ionita C., Osborn S. Privilege administration for the Role Graph Model. *Research Directions in Data and Applications Security. IFIP – The International Federation for Information Processing*. 2003;128. DOI: 10.1007/978-0-387-35697-6\_2.
10. Lumertz P.R., Ribeiro L., Duarte L.M. User interfaces metamodel based on graphs *Journal of Visual Languages & Computing*. 2016;32:1–34. DOI: 10.1016/j.jvlc.2015.10.026.
11. Arrue M., Vigo M., Abascal J. Including heterogeneous web accessibility guidelines in the development process. *Engineering Interactive Systems. EHCI 2007. Lecture Notes in Computer Science*. 2008;4940. DOI: 10.1007/978-3-540-92698-6\_37.
12. Bizer C., Schultz A. The Berlin SPARQL benchmark. *Int. J. Semantic Web Inf. Syst.* 2009;5(2):1–24. DOI: 10.4018/jswis.2009040101.
13. Manna A., Alkasassbeh M. Detecting network anomalies using machine learning and SNMP-MIB dataset with IP group. *2nd International Conference on new Trends in Computing Sciences (ICTCS)*. 2019;1–5. DOI: 10.1109/ICTCS.2019.8923043.
14. Hutter F., Kotthoff L., Vanschoren J. (eds) Automated machine learning. *The Springer Series on Challenges in Machine Learning*. Cham, Springer; 2019. 220 p. DOI: 10.1007/978-3-030-05318-5.
15. Vanschoren J. Meta-Learning. In: *Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning*. Springer, Cham; 2019. 220 p. DOI: 10.1007/978-3-030-05318-5\_2.

16. Wirth R., Hipp J. CRISP-DM: Towards a standard process model for data mining. *Proc. of the 4th int. conf. PAKDD*. 2000;1:29–39.
17. Thornton C., Hutter F., Hoos H., Leyton-Brown K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *KDD*. 2012. DOI: 10.1145/2487575.2487629.
18. Feurer M., Klein A., Eggenberger K., Springenberg J., Blum M., Hutter F. Auto-sklearn: efficient and robust automated machine learning. In: *Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning*. Cham, Springer; 2019. 220 p. DOI: 10.1007/978-3-030-05318-5\_6.
19. LeDell E. H2O AutoML: scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning*. 2020.
20. Hutter, F., Kotthoff, L., Vanschoren, J. (eds) *Automated Machine Learning. The Springer Series on Challenges in Machine Learning*. Cham, Springer; 2019. DOI: 10.1007/978-3-030-05318-5\_2.
21. Wirth R., Hipp J. CRISP-DM: Towards a standard process model for data mining. *Proc. of the 4th int. conf. PAKDD*. 2000;1:29–39.
22. Krinkin K., Vodyaho A., Kulikov I., Zhukova N. Method of multilevel adaptive synthesis of monitoring object knowledge graphs. *Applied Sciences*. 2021;11(14):6251. DOI: 10.3390/app11146251.
23. Krinkin K., Vodyaho A., Kulikov I., Zhukova N. Deductive synthesis of networks hierarchical knowledge graphs. *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*. 2021;12(3):32–48. DOI: 10.4018/IJERTCS.2021070103.
24. van der Ham J.J. *A semantic model for complex computer networks: the network description language*. Thesis. Citeseer; 2010. 154 p.
25. Qiao X., Li X., Fensel A., Su F.. Applying semantics to Parlay-based services for telecommunication and Internet networks. *Open Computer Science*. 2011;1(4):406–429. DOI: 10.2478/s13537-011-0029-6.
26. Cleary D., Danev B., O'Donoghue D. *Using ontologies to simplify wireless network configuration*. FOMI; 2005.
27. Villalonga C., Strohbach M., Snoeck N., Sutterer M., Belaunde M., Kovacs E., Zhdanova A.V., Goix L.W., Droegehorn O. Mobile ontology: Towards a standardized semantic model for the mobile domain. *International Conference on Service-Oriented Computing*. 2007;248–257. DOI 10.1007/978-3-540-93851-4\_25.
28. Barcelos P.P.F., Monteiro M.E., Simoes R. de M., Garcia A.S., Segatto M.E.V. Ootn-an ontology proposal for optical transport networks. *IEEE ICUMT*. 2009;1–7. DOI: 10.1109/ICUMT.2009.5345459.
29. Uzun A., Kupper A. OpenMobileNetwork: extending the web of data by a dataset for mobile networks and devices. *ACM ICSS*. 2012;17–24. DOI: 10.1145/2362499.2362503.
30. Zhou Q., Gray A.J.G., McLaughlin S. ToCo: An ontology for representing hybrid telecommunication networks. *The Semantic Web. ESWC 2019. Lecture Notes in Computer Science*. 2019;11503. DOI: 10.1007/978-3-030-21348-0\_33.
31. Wirth R., Hipp J. Crisp-dm: Towards a standard process model for data mining. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. 2000;1:29–39.
32. Panov P., Soldatova L., Dzeroski S. Ontodm-kdd: ontology for representing the knowledge discovery process. *Int. Conf. on Discovery Science*. 2013;126–140. DOI: 10.1007/978-3-642-40897-7\_9.
33. Alkasassbeh M. *AN-SNMP Dataset*. 2016. DOI: 10.13140/RG.2.2.26384.30721.

34. Kulikov I., Vodyaho A., Stankova E., Zhukova N. Ontology for knowledge graphs of telecommunication network Monitoring systems. *Computational Science and Its Applications – ICCSA 2021. ICCSA 2021. Lecture Notes in Computer Science*. 2021;12956. DOI: 10.1007/978-3-030-87010-2\_32.

#### ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Куликов Игорь Александрович**, соискатель, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», Санкт-Петербург, Российская Федерация.

*e-mail:* [i.a.kulikov@gmail.com](mailto:i.a.kulikov@gmail.com)

ORCID: [0000-0002-2532-5579](https://orcid.org/0000-0002-2532-5579)

**Igor Aleksandrovich Kulikov**, Degree-seeking Applicant, Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg, the Russian Federation.

**Жукова Наталья Александровна**, кандидат технических наук, доцент, Санкт-Петербургский федеральный исследовательский центр РАН, Санкт-Петербург, Российская Федерация.

*e-mail:* [nazhukova@mail.ru](mailto:nazhukova@mail.ru)

ORCID: [0000-0001-5877-4461](https://orcid.org/0000-0001-5877-4461)

**Natalia Aleksandrovna Zhukova**, Candidate of Technical Sciences, Associate Professor, Saint Petersburg Federal Research Centre of the Russian Academy of Sciences (SPCRAS), Saint Petersburg, the Russian Federation.

**Тяньсин Мань**, кандидат технических наук, ассистент, Школа искусственного интеллекта Университета Джилина, Чангчун, Китайская Народная Республика.

*e-mail:* [mantx@jlu.edu.cn](mailto:mantx@jlu.edu.cn)

ORCID: [0000-0003-2187-1641](https://orcid.org/0000-0003-2187-1641)

**Tianxing Man**, PhD, Tech., Postdoc, School of Artificial Intelligence at Jilin University, Changchun City, the People’s Republic of China.

*Статья поступила в редакцию 05.02.2023; одобрена после рецензирования 18.03.2023; принята к публикации 14.04.2023.*

*The article was submitted 05.02.2023; approved after reviewing 18.03.2023; accepted for publication 14.04.2023.*