УДК 004.852
doi:10.31799/1684-8853-2025-6-28-41
EDN: KKSXFO

Научные статьи
Articles

Метаалгоритм управления процессами синтеза моделей машинного обучения

Н. А. Жукова[a]***,*** *доктор техн. наук, доцент, orcid.org/0000-0001-5877-4461, nazhukova@mail.ru*
В. Э. Ковалевский[a]***,*** *младший научный сотрудник , orcid.org/0000-0002-0414-906X*
[a]*Санкт-Петербургский Федеральный исследовательский центр Российской академии наук, 14-я линия В. О., 39, Санкт-Петербург, 199178, РФ*

Введение: *методы автоматизированного машинного обучения позволяют автоматизировать синтез моделей машинного обучения, адаптированных к обработке конкретных данных. Однако эти методы требуют значительных временных и вычислительных затрат.* ***Цель:*** *разработать метаалгоритм управления процессами синтеза моделей машинного обучения, позволяющий снизить вычислительную сложность автоматического построения моделей машинного обучения.* ***Результаты:*** *предложен общий метаалгоритм управления процессами синтеза моделей машинного обучения и частный алгоритм, предусматривающий ограничение пространства поиска за счет использования метаобучения. Предлагаемый частный алгоритм основан на использовании метасвойств данных и онтологии, содержащей правила выбора алгоритмов машинного обучения в зависимости от метасвойств обрабатываемых данных. Наполнение онтологии выполняется за счет предварительной обработки результатов ранее синтезированных моделей машинного обучения. Для частного алгоритма разработан алгоритм формирования обучающей выборки и алгоритм построения онтологии для сокращения пространства поиска. Проведенные экспериментальные исследования показали, что использование предложенного частного алгоритма позволило снизить время синтеза моделей машинного обучения на 41,12 %. Кроме того, у полученных моделей повысились значения достоверности (+0,54 %), полноты (+0,34 %) и AUC (+1,85 %).* ***Практическая значимость:*** *частный алгоритм, разработанный на основе метаалгоритма, помогает снизить вычислительную сложность процесса автоматического построения моделей машинного обучения, что облегчает применение машинного обучения в предметных областях, требующих оперативного построения и адаптации моделей машинного обучения к новым данным и новым задачам.*

Ключевые слова *— автоматизированное машинное обучение, синтез моделей машинного обучения, AutoML, метаобучение, онтологии для AutoML, управление синтезом моделей машинного обучения.*

Для цитирования: Жукова Н. А., Ковалевский В. Э. Метаалгоритм управления процессами синтеза моделей машинного обучения. *Информационно-управляющие системы*, 2025, № 6, с. 28–41. doi:10.31799/1684-8853-2025-6-28-41, EDN: KKSXFO
For citation: Zhukova N. A., Kovalevsky V. E. Meta-algorithm for the process control of complex machine learning model synthesis. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2025, no. 6, pp. 28–41 (In Russian). doi:10.31799/1684-8853-2025-6-28-41, EDN: KKSXFO

Введение

Область машинного обучения (МО) включает в себя многочисленные алгоритмы, которые позволяют извлекать полезную информацию из большого количества необработанных данных. Эти алгоритмы стали особенно востребованными, когда прогресс в вычислительных технологиях привел к значительному увеличению объемов собираемых и хранимых данных. У алгоритмов МО есть два типа параметров. Параметры первого типа — это параметры, значения которых настраиваются в процессе обучения модели. Так, например, для нейронной сети такими параметрами являются веса нейронной сети, значения которых меняются по мере обучения сети на данных. Параметры второго типа — это параметры, значения которых определяют саму структуру модели. Значение данных параметров устанавливается заранее и не меняется в процессе обучения. Параметры такого типа также называются гиперпараметрами. Для нейронной сети к таким параметрам относятся количество слоев сети, связи между ними и тип активационной функции, а, например, для метода опорных векторов такими параметрами являются ядро и значение константы C, определяющей соотношение между размером разделяющей полосы и суммарной ошибкой. Наличие большого количества алгоритмов МО и отсутствие универсального алгоритма, позволяющего эффективно обрабатывать различные наборы данных, требует в каждом конкретном случае выбирать наиболее подходящий алгоритм или комбинацию алгоритмов и настраивать их гиперпараметры. Следует отметить, что настройка гиперпараметров оказывает значительное влияние на эффективность работы алгоритмов.

Построение модели МО может быть выполнено как в ручном режиме [1, 2], когда пользователь самостоятельно конструирует модель из различных алгоритмов, выбирая и настраивая их, основываясь на собственном опыте, так и в автоматизированном режиме [3, 4], когда синтез моде-

ли осуществляется программными средствами. При создании модели вручную пользователь может применять рекомендательные системы, содержащие экспертные знания, а также знания, полученные из описаний ранее решенных задач. Выполняя запрос к такой системе, пользователь указывает метасвойства текущей задачи и получает рекомендации по алгоритму и значениям его гиперпараметров. Также для выдачи рекомендаций по построению модели может быть использован нейросетевой подход, при котором конструируется нейронная сеть, которая затем обучается на различных задачах [5].

При автоматизированном построении моделей МО применяются средства автоматизированного машинного обучения (Automated Machine Learning – AutoML) [4, 6], которые предполагают применение машинного обучения к самому себе. Данные методы рассматривают множество алгоритмов и их гиперпараметров как параметры, которые также могут быть настроены на основе обучающих данных, а построение модели рассматривается как поиск наиболее подходящего варианта в пространстве гиперпараметров. Существует несколько различных подходов к поиску моделей МО, среди которых поиск по сетке (Grid Search) [7], случайный поиск (Random Search) [8], байесовская оптимизация (Bayesian Optimization) [9], генетические алгоритмы [10]. В случае использования AutoML системы она подменяет собой эксперта, и сама конструирует модель МО. Пользователь может либо применить получившуюся модель, либо использовать ее как исходную рекомендацию и дополнительно настроить. К настоящему времени разработано значительное число AutoML-систем, построенных на основе различных подходов, среди которых наиболее широкое применение получили AutoWEKA [11], Auto-sklearn [12], TPOT [13], H2O [14] и AutoKeras [15]. Каждая из рассмотренных систем при поиске моделей использует лишь ограниченное подмножество алгоритмов машинного обучения, при этом рассматриваемые различными системами подмножества могут пересекаться.

Реализуемое в AutoML построение моделей МО требует значительных временных и вычислительных ресурсов из-за большого пространства поиска, обусловленного большим количеством алгоритмов и сложностью настройки их гиперпараметров. Для ускорения процесса поиска моделей в AutoML-системах исследователями разработаны методы метаобучения, которые используют данные о ранее решенных задачах для установки начальной точки поиска или ограничения пространства поиска. Метаобучение основывается на опыте решения предшествующих задач и предполагает сбор метаданных, описывающих предыдущие задачи и использованные для их решения модели. Такие метаданные включают в себя точные конфигурации алгоритмов, использованных для построения моделей, в том числе настройки гиперпараметров, полученные оценки моделей, а также свойства решенных задач, называемые метасвойствами. При решении новой задачи выбор алгоритмов и их гиперпараметров основывается на оценке сходства решаемой и предыдущих задач, такое сходство задач может быть оценено, например, как евклидово расстояние в пространстве метасвойств или как расстояние Кульбака – Лейблера [16].

Классификация методов метаобучения [17] включает в себя: обучение на основе оценок моделей, что может использоваться для рекомендации общих конфигураций и пространств поиска конфигураций, а также для переноса знаний из эмпирически схожих задач [18]; обучение на основе метасвойств задач, предусматривающее определение метасвойств задач для нахождения схожих между собой задач и построение метамоделей, которые описывают взаимосвязи между метасвойствами данных, использованными алгоритмами и полученными оценками [19]; обучение на основе существующих моделей [20], которое предполагает перенос параметров обученной модели между схожими задачами, например с помощью трансферного обучения [21], или использование существующих моделей для обучения с малым числом запусков [22].

Существует ряд исследований, направленных на применение онтологий для оптимизации поиска моделей машинного обучения. Так, в [23] предлагается онтология для описания существующих AutoM-систем и их возможностей, которая позволяет выбирать наиболее подходящую AutoML-систему при построении моделей МО. Однако данная онтология не содержит дополнительных знаний, которые могут использоваться для выбора алгоритмов и настройки их гиперпараметров. В работе [24] предлагается онтология, содержащая экспертные знания, в частности знания об алгоритмах, применяемых на различных шагах обработки данных и ассоциированных с ними гиперпараметрах. Онтология используется для выбора алгоритмов МО и построения моделей МО вместо поиска по пространству гиперпараметров. Проведено сравнение предлагаемого решения с системой TPOT на одном наборе данных. Однако в работе не рассматриваются вопросы совместного использования различных существующих AutoML-систем при выборе алгоритмов и построении моделей МО. В [25] предлагается онтология, содержащая знания, позволяющие выбирать алгоритмы построения признакового пространства в зависимости от свойств обрабатываемых данных. Проведены экспериментальные

исследования на 10 наборах данных, в которых осуществляется выбор алгоритмов отбора признаков, но используемые алгоритмы МО фиксированы. Также отсутствует сравнение предлагаемого решения с другими системами. В [26] авторами представляется онтология для семантического описания моделей МО, включающая знания о наборах данных, использованных для обучения моделей, областях применения моделей и алгоритмах МО и позволяющая рекомендовать ранее использованные модели МО. В [27] модели МО представляются в виде графов знаний, при построении моделей МО используется информация о ранее построенных моделях для схожих наборов данных. В экспериментальных исследованиях построение графов выполняется на основе данных из открытого репозитория OpenML. Данное решение может рассматриваться как один из возможных подходов к построению моделей МО и использоваться наряду с другими существующими решениями при выполнении синтеза моделей. Кроме того, графовое представление моделей может применяться как эффективное средство визуализации моделей для работы с моделями конечных пользователей.

Проведенный анализ показал, что разработанные к настоящему времени методы AutoML имеют высокую вычислительную сложность и требуют значительных затрат ресурсов. Существующие AutoML-системы используют для построения моделей лишь ограниченные подмножества алгоритмов МО, которые могут пересекаться. Для оптимизации процесса поиска моделей разработаны различные методы, направленные на решение проблемы высокой вычислительной сложности поиска алгоритмов и настройки их гиперпараметров, однако они являются разрозненными и не предполагают их использования в комбинации с другими методами, что приводит к сложностям применения AutoML при решении практических задач. Для накопления знаний, позволяющих оптимизировать процессы поиска моделей МО, могут использоваться онтологии. Имеются подходы к использованию онтологий для оптимизации МО, в которых применяются онтологии, созданные экспертами вручную, а также подходы, рассматривающие их внедрение как замену другим методам оптимизации. В экспериментальных исследованиях решений, разработанных на основе онтологического подхода, часто отсутствует сравнение с существующими методами, что затрудняет оценку их эффективности. Перечисленные ограничения создают существенные сложности для широкого применения методов МО на практике.

В представленной работе предлагается общий метаалгоритм управления процессом синтеза моделей МО и реализующий его частный алгоритм, позволяющий обеспечить эффективный поиск моделей за счет ограничения пространства поиска с применением метаобучения и онтологий.

Постановка задачи оптимизации синтеза модели машинного обучения

Модель МО M: $X \to Y$ представляет собой алгоритм с настроенными гиперпараметрами, преобразующий вектор признаков $\vec{\mathbf{x}} \in X$ в целевое значение $y \in Y$, например, в метку класса в случае решения задачи классификации. Обозначим фиксированный набор базовых алгоритмов как $A = \{A^{(1)}, A^{(2)}, \ldots, A^{(n)}\}$. Для каждого алгоритма $A^{(i)}$ задается свой вектор гиперпараметров $\vec{\boldsymbol{\lambda}} \in \Lambda_{A^{(i)}}$. Пусть $D = \{(\vec{\mathbf{x}}_1, y_1), \ldots, (\vec{\mathbf{x}}_w, y_w)\}$ обозначает множество из w наблюдений, состоящих из векторов признаков $\vec{\mathbf{x}}_i$, для которых известны соответствующие им целевые значения y_i. Обозначим модель МО, состоящую из алгоритма A с гиперпараметрами $\vec{\boldsymbol{\lambda}}$, построенную с использованием знаний K, как $M_{A,\vec{\boldsymbol{\lambda}},K}$. Пусть $L(\cdot, \cdot)$ обозначает функцию потерь. В этом случае потери для модели $M_{A,\vec{\boldsymbol{\lambda}},K}$ при обработке данных D могут быть оценены следующим образом:

$$\hat{R}(M_{A,\vec{\boldsymbol{\lambda}},K}, D) = \frac{1}{w}\sum_{i=1}^{w} L(M(\vec{\mathbf{x}}_i), y_i). \quad (1)$$

Задача синтеза модели МО состоит в том, чтобы, используя управляющее воздействие I, построенное с помощью знаний K, найти такой алгоритм и значения его гиперпараметров, которые минимизируют потери при обработке новых данных D_{new}:

$$(A, \vec{\boldsymbol{\lambda}})^* \in \arg\min_{A \in A, \vec{\boldsymbol{\lambda}} \in \Lambda} \hat{R}(M_{A,\vec{\boldsymbol{\lambda}},K}, I(K), D_{new}). \quad (2)$$

Для решения данной задачи необходимо построить такую функцию f, которая на основе множества доступных алгоритмов A и знаний K синтезирует модель МО для обработки D_{new}:

$$f : A \times K \xrightarrow{D_{new}} M_{A,\vec{\boldsymbol{\lambda}},K}. \quad (3)$$

Функция f должна обеспечивать достижение минимума потерь $\hat{R}$ при ограничениях на время выполнения T:

$$f \in \arg\min \hat{R}(T, D_{new});\ T(\hat{R}, D_{new}) \le T_{\max}. \quad (4)$$

Управление синтезом моделей МО

Общая схема процесса управления синтезом модели МО включает три блока: 1) наблю-

даемый объект, данные от которого поступают к двум подсистемам — подсистеме управления синтезом и подсистеме синтеза; 2) подсистему управления синтезом, формирующую управляющие воздействия; 3) подсистему синтеза, формирующую на основе полученных данных и управляющих воздействий модель МО. В общем виде предлагаемая схема управления представлена на рис. 1.

Обобщенная схема управления синтезом модели МО предполагает выполнение следующих шагов.

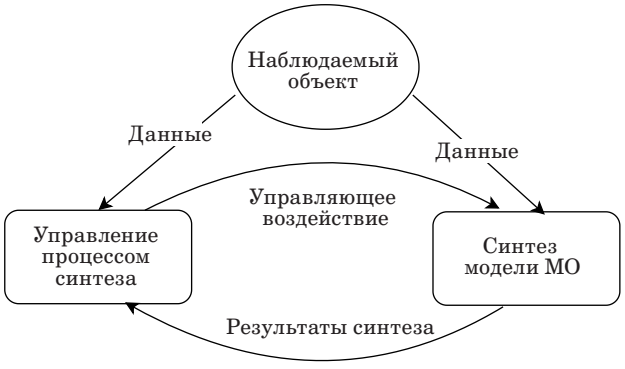
Шаг 1. При поступлении данных от наблюдаемого объекта подсистема управления процессом синтеза формирует управляющее воздействие на подсистему синтеза модели.

Шаг 2. Подсистема синтеза модели на основе управляющего воздействия и данных от наблюдаемого объекта синтезирует модель МО.

Шаг 3. Результаты применения модели передаются обратно в подсистему управления для оценки качества управляющего воздействия.

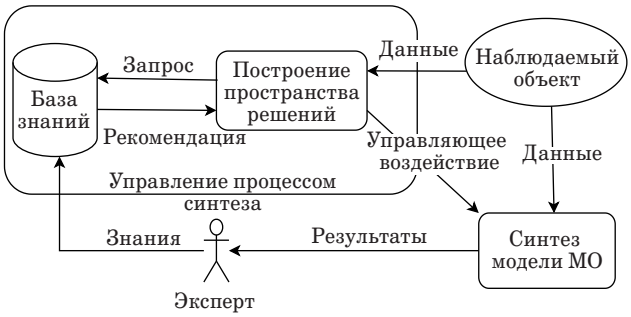
Подсистема управления процессом синтеза может быть реализована с помощью различных средств, в частности основана на использовании экспертных знаний, или, при наличии достаточно большой обучающей выборки ранее решенных задач, могут использоваться методы метаобучения.

При управлении процессом синтеза модели МО с помощью экспертных знаний реализуется схема управления, представленная на рис. 2. Подсистема управления процессом синтеза модели МО состоит из двух блоков: 1) базы знаний и 2) блока построения пространства решений. Эксперт на основе имеющихся у него знаний формирует базу знаний. Блок построения пространства решений отправляет запрос в базу знаний и на основе полученных от нее рекомендаций и данных, поступающих от наблюдаемого



■ ***Рис. 1.*** Обобщенная схема управления синтезом модели МО

■ ***Fig. 1.*** A generalized scheme of ML model synthesis control



■ ***Рис. 2.*** Управление процессом синтеза с использованием экспертных знаний

■ ***Fig. 2.*** A control of the synthesis process with the use of an expert knowledge

объекта, формирует управляющие воздействия. Подсистема синтеза строит модель МО на основе полученных данных и с учетом управляющих воздействий. Сформированная результирующая модель может быть проанализирована экспертом, который в случае необходимости вносит изменения в базу знаний.

Функция f для случая использования экспертных знаний при управлении процессом синтеза модели МО имеет вид

$$f_E : A \times I(K_E) \xrightarrow{D_{new}} M_{A,\bar{\lambda},K_E};$$

$$T(M_{A,\bar{\lambda},K_E}, \hat{R}) \le T_{\max}, \qquad (5)$$

где K_E — знания, основанные на экспертных знаниях E, используемые при формировании управляющего воздействия; $M_{A,\bar{\lambda},K_E}$ — модель, построенная с использованием знаний K_E.

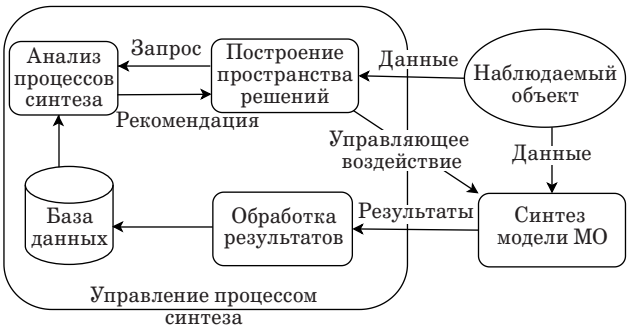
Процесс управления, основанный на экспертных знаниях, состоит из следующих шагов.

Шаг 1. Эксперт заполняет базу знаний на основе имеющегося у него опыта.

Шаг 2. Сформированная база знаний используется подсистемой управления процессом синтеза для выработки управляющих воздействий.

Шаг 3. Полученные в результате синтеза модели оцениваются экспертом. При необходимости эксперт вносит изменения в базу знаний.

При наличии накопленных данных о ранее решенных задачах управление синтезом может осуществляться на основе знаний, извлекаемых из накопленных данных, при этом для извлечения знаний могут использоваться различные методы, в частности методы метаобучения. Схема управления синтезом на основе знаний, извлекаемых из накопленных данных, представлена на рис. 3. Подсистема управления процессом синтеза модели МО в этом случае состоит из четырех блоков: 1) базы данных, в которой накапливаются данные о ранее решенных задачах; 2) блока анализа процессов синтеза, обрабатывающего



■ ***Рис. 3.*** Управление процессом синтеза на основе знаний, извлекаемых из данных о ранее решенных задачах

■ ***Fig. 3.*** A control of the synthesis process using the knowledge obtained from the data about previously solved tasks

данные о ранее построенных моделях МО для их последующего использования при выдаче рекомендаций; 3) блока построения пространства решений, отправляющего запрос блоку анализа и получающего от него рекомендации; 4) блока обработки результатов, преобразующего результаты решения новой задачи в форму, позволяющую их размещать в базе данных.

При такой схеме работы происходит постоянное автоматическое уточнение знаний, используемых при выдаче рекомендаций с каждым циклом обработки новых данных.

Функция f для случая использования знаний, автоматически извлекаемых из данных о ранее решенных задачах, имеет вид

$$f_{D_{prev}} : A \times I(K_{D_{prev}}) \xrightarrow{D_{new}} M_{A,\bar{\lambda},K_{D_{prev}}};$$

$$T(M_{A,\bar{\lambda},K_{D_{prev}}}, \hat{R}) \le T_{\max}, \qquad (6)$$

где D_{prev} – данные о ранее решенных задачах, используемые при формировании управляющего воздействия; $M_{A,\bar{\lambda},K_{D_{prev}}}$ – модель, построенная с использованием знаний, полученных при обработке D_{prev}.

Процесс управления, основанный на использовании автоматически извлекаемых знаний, состоит из следующих шагов.

Шаг 1. Блок анализа процессов синтеза получает информацию из базы данных о ранее решенных задачах, примененных алгоритмах и полученных результатах и использует эти данные для обучения.

Шаг 2. Блок построения пространства решений отправляет запрос к блоку анализа процессов и получает рекомендации по алгоритмам, которые могут обеспечить эффективное решение текущей задачи.

Шаг 3. На основе полученных рекомендаций формируется управляющее воздействие, которое передается в подсистему синтеза моделей.

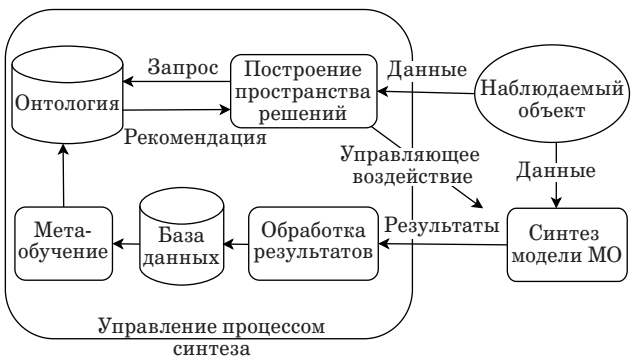
Шаг 4. Результаты синтеза передаются обратно в подсистему управления, где они обрабатываются и пополняют базу данных, которая далее используется в блоке анализа.

Частный алгоритм управления синтезом с применением метаобучения

В качестве частного алгоритма управления процессом синтеза, реализующего предложенный общий метаалгоритм, предлагается алгоритм, основанный на ограничении пространства поиска модели МО за счет метаобучения по накопленным данным и наполнения полученными знаниями базы знаний, представленной в форме онтологии. В этом случае подсистема управления процессом синтеза модели МО состоит из пяти блоков: 1) базы данных, содержащей информацию о ранее решенных задачах; 2) блока метаобучения, обрабатывающего обучающий набор из базы данных и формирующего на его основе базу знаний; 3) онтологии, хранящей выявленные взаимосвязи между метасвойствами обрабатываемых данных и примененными алгоритмами; 4) блока построения пространства решений, получающего данные от наблюдаемого объекта, отправляющего запрос к онтологии и формирующего на основе данных и полученных рекомендаций управляющее воздействие; 5) блока обработки результатов, преобразующего полученные результаты в форму, позволяющую их записывать в базу данных.

Схема управления синтезом с применением метаобучения представлена на рис. 4.

При применении метаобучения выполняется сравнение метасвойств данных, требующих обработки, и метасвойств ранее обработанных дан-



■ ***Рис. 4.*** Управление процессом синтеза на основе метаобучения и онтологии

■ ***Fig. 4.*** A control of the synthesis process using meta-learning and ontology

ных. В этом случае работа системы разбивается на два этапа: 1) этап предварительной подготовки и 2) этап работы с новыми данными.

На этапе предварительной подготовки выполняются следующие шаги.

Шаг 1. Задаются используемые метасвойства данных.

Шаг 2. Формируется база данных с обучающей выборкой, сформированной с учетом выбранных метасвойств.

Шаг 3. Обучающая выборка используется для метаобучения и формирования онтологии.

Этап работы с новыми данными состоит из следующих шагов.

Шаг 1. Блок построения пространства решений получает новые данные и выполняет запрос к онтологии, построенной на этапе предварительной подготовки.

Шаг 2. На основе полученных рекомендаций формируется управляющее воздействие на подсистему синтеза моделей.

Шаг 3. Подсистема синтеза моделей получает новые данные и управляющее воздействие и строит модель МО.

Шаг 4. Полученные результаты обрабатываются и пополняют базу данных новыми данными о решенной задаче.

При управлении синтезом модели МО с применением метаобучения и онтологии управление состоит в ограничении пространства поиска алгоритмов МО с применением правил, выявленных на основе обработки данных о ранее решенных задачах. Правила отображают взаимосвязи между метасвойствами данных и алгоритмами МО, подходящими для обработки данных с такими метасвойствами. Для построения правил возможно использование различных наборов метасвойств, ограничения на состав используемых метасвойств не накладываются. Определить метасвойства, формируемые для конкретного набора данных, можно с использованием метода выбора признаков [25] и др. Для ограничения пространства поиска определенными алгоритмами необходимо, чтобы применяемая AutoML система поддерживала использование подобного рода ограничений. К системам, позволяющим задавать ограничения на пространство поиска, относятся, в частности, Auto-sklearn и H2O.

Метаобучение для ограничения пространства поиска

Блоку метаобучения необходима обучающая выборка, на основе которой выявляются взаимосвязи между метасвойствами обрабатываемых данных и рекомендованными алгоритмами. При составлении обучающей выборки для метаобучения необходимо определить набор задач, которые будут решаться средствами AutoML. Для выбранных задач определяются значения метасвойств обрабатываемых данных и выполняется построение моделей МО. Полученные значения метасвойств и рекомендованные алгоритмы пополняют обучающую выборку. Процесс построения обучающей выборки приведен в алгоритме 1, где функция dataLoad используется для загрузки набора данных ds из репозитория, metaFeaturesExtract — для вычисления метасвойств данных data, а modelSearch производит поиск с помощью библиотеки AutoML модели МО, подходящей для обработки данных data за время, не превышающее T_{max}.

Алгоритм 1. createMetaDS(dsets, AutoML, T_{max}) — формирование обучающей выборки.

Входные данные: dsets — множество наборов данных; AutoML — используемая библиотека автоматизированного МО, T_{max} — максимальное время на поиск модели

Выходные данные: mfeat — набор метасвойств и рекомендуемых алгоритмов

```
1: mfeat = ∅
{Цикл для всех наборов данных ds из dsets}
2: repeat
3:    data = dataLoad(ds)
4:    meta = metaFeaturesExtract(data)
5:    algo = modelSearch (data, AutoML(T_max))
6:    mfeat = mfeat ∪ (algo ∪ meta)
7: until (ds ∈ dsets)
8: return mfeat
```

После получения обучающей выборки ее можно использовать для выявления связей между метасвойствами данных и рекомендованными для их обработки алгоритмами. Для этого строится дерево решений, в котором листьями являются выбранные алгоритмы, а узлами — используемые метасвойства. Переход от узла к левой ветке осуществляется в случае отсутствия данного метасвойства, а к правой — при его наличии. Полученный таким способом классификатор прост в интерпретации и может быть легко применен для добавления новых связей между элементами онтологии. Построение дерева решений и его использование для наполнения онтологии приведено в алгоритме 2, где функция treeCreation строит дерево решений на основе обучающей выборки mfeat, travelTree — рекурсивная функция, которая проходит по созданному дереву и формирует на его основе правила, размещаемые в онтологии.

Алгоритм 2. Построение дерева решений и наполнение онтологии правилами.

Входные данные: fileOnto — файл с онтологией, dsets — множество наборов данных; AutoML — используемая библиотека автомати-

зированного МО, T_{max} — максимальное время на поиск модели

Выходные данные: fileUpdated — файл, содержащий обновленную онтологию

1: mfeat = createMetaDS(dsets, AutoML, T_{max})
2: tree = treeCreation (mfeat)
3: fileUpdated = travelTree(tree, ∅, fileOnto)
4: **return** fileUpdated

Рекурсивный проход по дереву решений travelTree для записи всех путей и формирования на их основе правил для размещения в онтологии приведен в алгоритме 3, где функция getLeftNode возвращает левый подузел, getRightNode возвращает правый подузел, getFeature возвращает метасвойство, getValue возвращает значение, а extendOntology обновляет онтологию на основе файла с онтологией и сформированного пути.

Алгоритм 3. travelTree(node, list_of_nodes, file) — проход по дереву решений.

Входные данные: node — текущий узел, list_of_nodes — формируемый путь до листа, file — файл с онтологией

Выходные данные: file — файл с обновленной онтологией

1: left_node = getLeftNode(node)
2: right_node = getRightNode(node)
3: feature = getFeature(node)
4: **if** left_node == ∅ and right_node == ∅ **then**
5: list_of_nodes = list_of_nodes ∪ getValue(node)
6: file = extendOntology(file, list_of_nodes)
7: **end if**
8: **if** left_node != ∅ **then**
9: list_of_nodes = list_of_nodes ∪ (“No”+feature)
10: travelTree(left_node, list_of_nodes, file)
11: **end if**
12: **if** right_node != ∅ **then**
13: list_of_nodes = list_of_nodes ∪ feature
14: travelTree(right_node, list_of_nodes, file)
15: **end if**
16: **return** file

В алгоритме 3 осуществляется проход по дереву принятия решений, при этом происходит запись пути от корня до листа-алгоритма. При достижении листа весь сохраненный путь записывается в онтологию, содержащуюся в файле file, с помощью функции extendOntology, описанной в алгоритме 4, где функция getOntology читает онтологию из файла, pop извлекает из списка последний элемент (алгоритм), getAutoML читает из онтологии данные о AutoML системе, информация о которой будет дополнена, updateOntology обновляет онтологию для конкретного алгоритма и AutoML системы, а saveOntology сохраняет полученную онтологию в файл.

Алгоритм 4. extendOntology(file, list_of_nodes) — расширение онтологии.

Входные данные: file — файл с онтологией, list_of_nodes — путь до листа-алгоритма

Выходные данные: fileUpdated — файл с обновленной онтологией

1: onto = getOntology(file)
2: algo =pop(list_of_nodes)
3: info_to_add = getAutoML(onto)
{Цикл по всем узлам node в списке list_of_nodes}
4: **repeat**
5: info_to_add = info_to_add ∪ suitableFor(node)
6: **until** node ∈ list_of_nodes
7: onto = updateOntology(onto, algo, info_to_add)
8: fileUpdated = saveOntology(onto)
9: **return** fileUpdated

Вначале из сохраненного пути извлекается рекомендованный алгоритм, а затем все остальные части пути сохраняются как атрибуты данного алгоритма.

Обработка новых данных

При поступлении новых данных, для которых необходимо построить модель МО, вначале вычисляются значения их метасвойств, после этого на основе вычисленных метасвойств выполняется запрос к онтологии, возвращающей набор алгоритмов, которыми следует ограничить пространство поиска. Данные разбиваются на обучающую и тестовую выборки, и на основе обучающей выборки выполняется поиск модели МО, подходящей для обработки поступивших данных, с учетом ограниченного пространства поиска. После того как подходящая модель найдена, с помощью тестовой выборки вычисляются метрики качества полученной модели. Вычисленные метасвойства, информация о модели и значения ее метрик пополняют базу данных о решенных задачах, которая затем используется для обновления онтологии с помощью алгоритмов 3, 4. Процесс обработки новых данных приведен в алгоритме 5, где функция ontologyQuery с помощью метасвойств meta выполняет запрос к онтологии, записанной в файле fileOnto, функция splitData разбивает данные data на обучающую и тестовую выборки в соответствии с соотношением ratio, функция modelSearch производит поиск с помощью библиотеки AutoML модели МО, подходящей для обработки данных train_data за время, не превышающее T_{max} и с ограничением пространства поиска алгоритмами из множества algos_list, функция calcMetrics вычисляет значения метрик модели model с помощью тестовых данных test_data, функ-

ция updateDatabase обновляет базу данных database метасвойствами meta, информацией о модели model и ее метриках metrics, функция getDataFromDB извлекает из базы данных database данные mfeat, необходимые для обновления онтологии с использованием функции treeCreation.

Алгоритм 5. Обработка новых данных.

Входные данные: data — обрабатываемые данные; fileOnto — файл с онтологией, AutoML — используемая библиотека автоматизированного МО, ratio — соотношение разделения данных на обучающие и тестовые, T_{max} — максимальное время на поиск модели

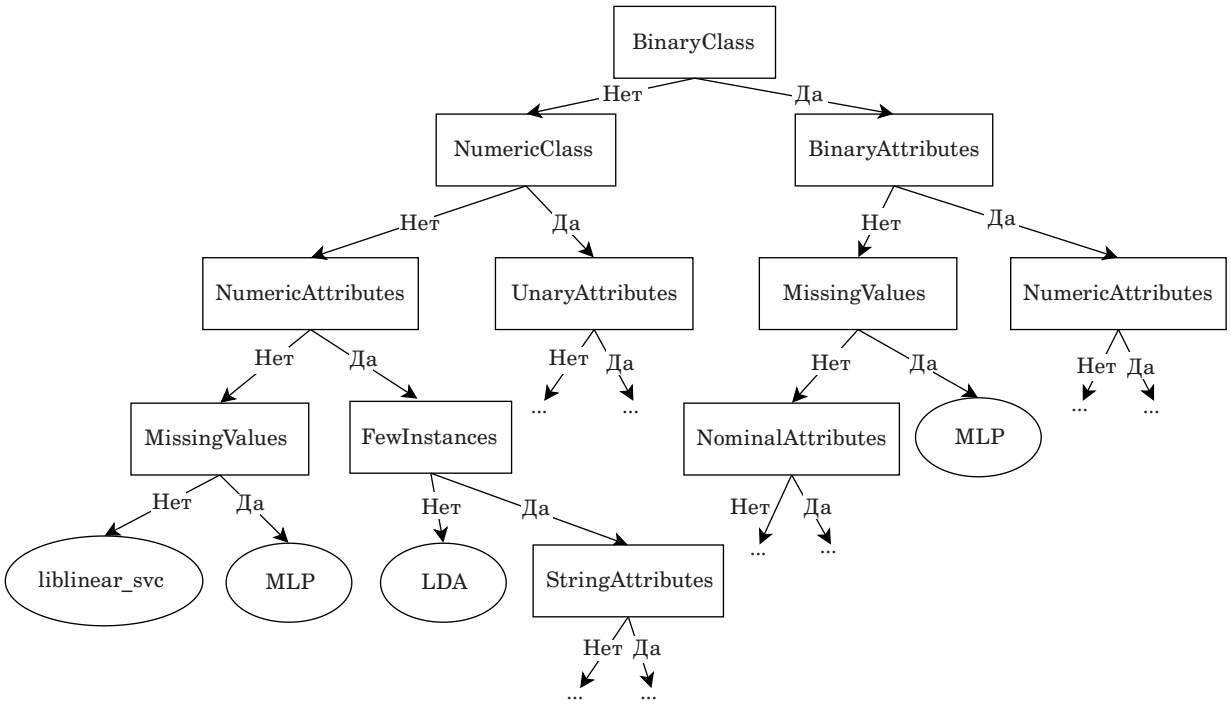
Выходные данные: fileUpdated — файл, содержащий обновленную онтологию

1: meta = metaFeaturesExtract(data)
2: algos_list = ontologyQuery(fileOnto, meta)
3: train_data = splitData(data, ratio)
4: test_data = data \ train_data
5: model = modelSearch (train_data, AutoML(algos_list, T_{max}))
6: metrics = calcMetrics(model, test_data)
7: database = updateDatabase(database, model, metrics, meta)
8: mfeat = getDataFromDB(database)
9: tree = treeCreation (mfeat)
10: fileUpdated = travelTree(tree, ∅, fileOnto)
11: **return** fileUpdated

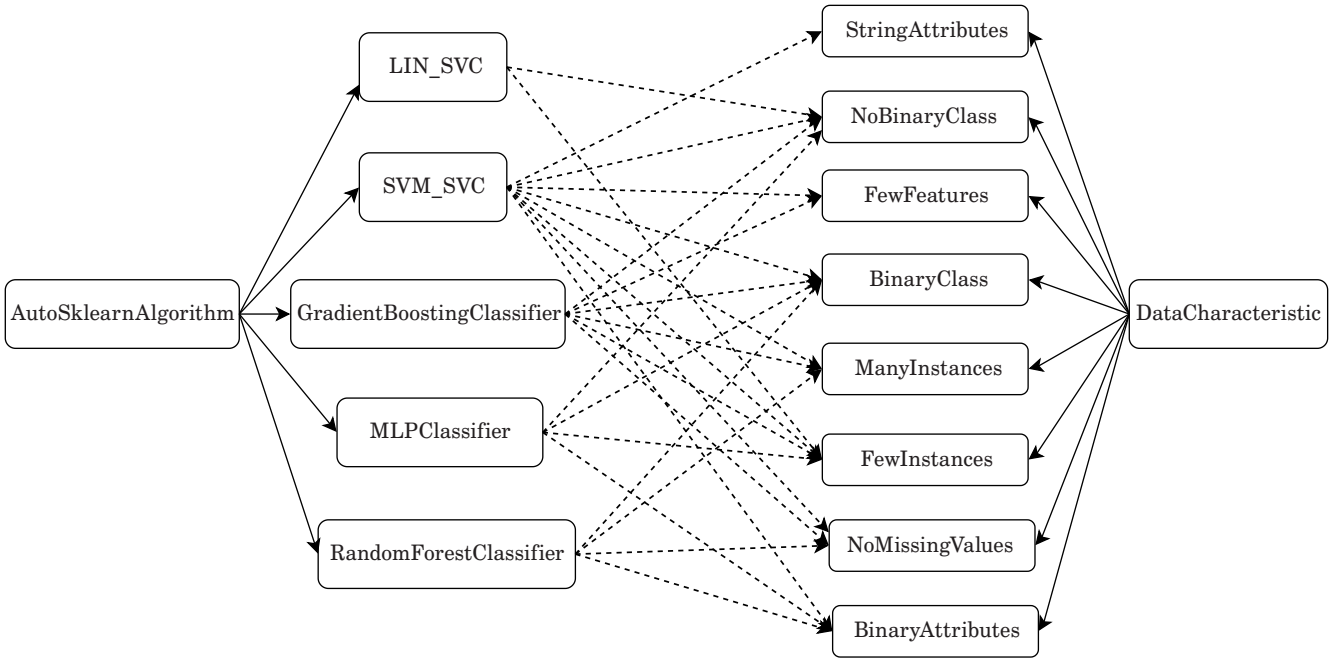
Экспериментальные исследования

Построение онтологии

При проведении экспериментальных исследований использовалась AutoML-система Auto-sklearn — решение, основанное на байесовской оптимизации и выполняющее поиск по 16 алгоритмам библиотеки алгоритмов МО Scikit-learn [28]. При формировании обучающей выборки для метаобучения и наполнения онтологии были использованы 35 наборов данных из открытого репозитория OpenML: letter, balance-scale, mfeat-fourier, mfeat-karhunen, mfeat-pixel, mfeat-zernike, breast-w, mushroom, optdigits, credit-approval, pendigits, sick, soybean, spambase, splice, tic-tac-toe, vehicle, waveform-5000, electricity, satimage, isolet, vowel, scene, monks-problems-1, JapaneseVowels, synthetic_control, irish, analcatdata_authorship, collins, ada_agnostic, gina_agnostic, KDDCup09_appetency, gas-drift, har, hill-valley. Загруженные наборы данных и библиотека Auto-sklearn были использованы для формирования обучающей выборки с помощью алгоритма 1 с вычислением следующих метасвойств: ManyInstances, FewInstances, ManyFeatures, FewFeatures, BinaryClass, NumericClass, NominalClass, StringClass, BinaryAttributes, MissingValues, NominalAttributes, NumericAttributes, UnaryAttributes, StringAttributes. Сформированная выборка была передана в алгоритм 2 для построения дерева решений. Часть сформированного дерева решений показана на рис. 5.



■ ***Рис. 5.*** Часть дерева решений, сформированного для выбора алгоритмов МО по метасвойствам данных
■ ***Fig. 5.*** A part of the decision tree created to determine a suitable ML algorithms based on meta-features of data



■ ***Рис. 6.*** Фрагмент онтологии, описывающей связи между алгоритмами МО и метасвойствами данных
■ ***Fig. 6.*** Part of the ontology describing the connection between algorithms and meta-features

С помощью редактора онтологий Protégé версии 5.6.4 построена онтология, включающая две независимые составляющие: 1) описания доступных в используемой AutoML-системе алгоритмов; 2) описания используемых метасвойств данных. Затем разработанная онтология и сформированное дерево решений использовались алгоритмами 3 и 4 для расширения онтологии [29] путем добавления связей между метасвойствами и алгоритмами МО. Фрагмент построенной таким образом онтологии приведен на рис. 6.

Синтез моделей МО
с использованием онтологии

Экспериментальные исследования были направлены на сравнительную оценку времени, требуемого для синтеза моделей МО с использованием онтологии и без, при этом также оценивались следующие показатели моделей: достоверность (accuracy), точность (precision), полнота (recall), F-мера (F-score) и площадь под ROC-кривой (AUC). Для проведения экспериментальных исследований использована онлайн-платформа Google Colab, предоставляющая виртуальный процессор Intel Xeon, RAM 13 ГБ, графический процессор (GPU) NVIDIA Tesla K80 с видеопамятью 16 ГБ и программное обеспечение Python v.3.10.12, Owlready2 v.2-0.47 и Auto-sklearn v.0.15. Для целей тестирования использовано 15 наборов данных из репозитория OpenML. Каждый тестовый набор данных был разбит на обучающую и тестовую выборки в соотношении 4:1. Над обучающими выборками выполнены следующие операции с использованием библиотеки Auto-sklearn:

1) поиск модели с ограничением времени поиска 15 мин;

2) поиск модели с ограничением времени поиска и ограничением точности;

3) поиск модели с ограничениями времени поиска, точности и пространства поиска. Для ограничения пространства поиска использовалась построенная онтология.

Для найденной в каждом случае модели были вычислены ее показатели, а также записаны количество времени, которое занял поиск, и рекомендованный алгоритм. Результаты проведенных экспериментов представлены в таблице.

Проведенные эксперименты показали, что среднее время поиска модели, ограниченного только временем работы 900 с, составило 896,44 с, среднее время поиска модели, ограниченного временем работы и точностью, составило 433,59 с, среднее время поиска модели, ограниченного временем работы, точностью и подмножеством алгоритмов, полученных путем запроса к онтологии по метасвойствам набора данных, составило 255,31 с. Таким образом, за счет сокращения пространства поиска с помощью запросов к онтологии время поиска было сокращено на 71,52 % по сравнению с поиском, ограниченным только временем работы, и на 41,12 % по сравнению с поиском, ограниченным временем и точностью. В восьми случаях из 15 алгоритм, предложен-

■ Метрики моделей, найденных в условиях ограничений на время поиска и (или) точности с использованием / без использования онтологии

■ Metrics of models found under time and/or accuracy constraints while limiting the search space with ontology and without

Набор данных	Ограничение по точности, %	Использование онтологии	Алгоритм	Достоверность	Точность	Полнота	F-мера	AUC	Время, с
Adult	Нет	Нет	gradient boosting	0,870	0,770	0,665	0,714	0,801	895,50
	0,14	Нет	gradient boosting	0,872	0,788	0,653	0,714	0,798	371,60
	0,14	Few Features	gradientboosting	0,872	0,788	0,653	0,714	0,798	286,34
Banking	Нет	Нет	gradientboosting	0,907	0,624	0,503	0,557	0,732	894,26
	0,09	Нет	gradient boosting	0,907	0,624	0,503	0,557	0,732	628,15
	0,09	Few Features	gradientboosting	0,907	0,624	0,503	0,557	0,732	227,43
Cars	Нет	Нет	gradientboosting	0,997	0,996	0,982	0,989	1	896,38
	0,01	Нет	mlp	0,991	0,990	0,960	0,974	0,999	115,97
	0,01	String Attributes	libsvm svc	0,986	0,965	0,974	0,969	0,999	18,273
Amazon	Нет	Нет	extra trees	0,950	0,953	0,996	0,974	0,566	904,52
	0,06	Нет	randomforest	0,947	0,949	0,997	0,972	0,527	30,168
	0,06	String Attributes	libsvm svc	0,946	0,946	1	0,972	0,5	27,777
Australian	Нет	Нет	ada boost	0,884	0,944	0,797	0,864	0,878	899,91
	0,15	Нет	ada boost	0,884	0,944	0,797	0,864	0,878	320,21
	0,15	Binary Class	random forest	0,877	0,912	0,813	0,860	0,872	103,82
blood	Нет	Нет	extra trees	0,787	0,68	0,415	0,515	0,671	896,01
	0,2	Нет	mlp	0,8	0,739	0,415	0,531	0,680	118,56
	0,2	Few Instances	mlp	0,8	0,739	0,415	0,531	0,680	112,55
kc1	Нет	Нет	mlp	0,879	0,613	0,328	0,427	0,647	895,46
	0,14	Нет	mlp	0,879	0,613	0,328	0,427	0,647	895,27
	0,14	Binary Class	mlp	0,870	0,543	0,328	0,409	0,642	726,62
christine	Нет	Нет	random forest	0,743	0,733	0,731	0,732	0,742	894,98
	0,26	Нет	random forest	0,743	0,733	0,731	0,732	0,742	284,20
	0,26	Many Instances	random forest	0,743	0,733	0,731	0,732	0,742	279,32
cnae9	Нет	Нет	sgd	0,926	0,929	0,922	0,923	0,996	894,19
	0,05	Нет	passive aggress	0,940	0,942	0,937	0,938	0,998	894,25
	0,05	Few Instances	liblinear svc	0,926	0,926	0,923	0,923	0,996	540,43
fabert	Нет	Нет	random forest	0,691	0,700	0,655	0,662	0,919	899,49
	0,32	Нет	random forest	0,691	0,700	0,655	0,662	0,919	41,934
	0,32	NoMissingValues	random forest	0,691	0,700	0,655	0,662	0,919	40,611
helena	Нет	Нет	k_nearestneighbors	0,179	0,089	0,086	0,086	0,539	895,39
	0,7	Нет	k_nearest_neighbors	0,179	0,089	0,086	0,086	0,539	895,49
	0,7	NoBinaryClass	liblinear svc	0,280	0,096	0,103	0,079	0,807	893,85
jannis	Нет	Нет	gradient boosting	0,712	0,638	0,543	0,559	0,872	893,60
	0,3	Нет	gradient boosting	0,712	0,638	0,543	0,559	0,872	223,33
	0,3	NoBinaryClass	gradient boosting	0,712	0,638	0,543	0,559	0,872	137,20
jasmine	Нет	Нет	random forest	0,812	0,751	0,940	0,835	0,811	895,76
	0,18	Нет	random forest	0,812	0,751	0,940	0,835	0,811	895,84
	0,18	Binary Attributes	random forest	0,814	0,753	0,940	0,837	0,813	97,186

■ *Окончание таблицы*
■ *The end of the Table*

Набор данных	Ограничение по точности, %	Использование онтологии	Алгоритм	Достоверность	Точность	Полнота	F-мера	AUC	Время, с
kr-vs-kp	Нет	Нет	gradient boosting	0,992	0,997	0,988	0,993	0,992	895,36
	0,01	Нет	gradient boosting	0,992	0,994	0,991	0,993	0,992	231,18
	0,01	String Attributes	libsvm svc	0,992	0,994	0,991	0,993	0,992	9,271
mfeat-factors	Нет	Нет	liblinear svc	0,965	0,968	0,965	0,966	0,999	895,85
	0,02	Нет	liblinear svc	0,965	0,968	0,965	0,966	0,999	557,75
	0,02	Few Instances	libsvm svc	0,965	0,967	0,965	0,966	0,993	329,04

ный поиском, ограниченным с использованием онтологии, был идентичен алгоритму, предложенному при поиске, ограниченном временем и точностью. Из этих восьми случаев в шести случаях значения метрик не изменились, в одном случае значения метрик, полученных с помощью поиска, ограниченного с использованием онтологии, улучшились, а в одном случае ухудшились. Изменение значений метрик при сохранении используемого алгоритма объясняется тем, что при ограничении пространства поиска меньшим количеством алгоритмов при поиске моделей выделяется больше времени на подбор гиперпараметров. Из семи случаев, когда поиском, ограниченным онтологией, был выбран другой алгоритм для построения модели, в одном случае улучшились значения достоверности (+56,79 %), точности (+7,11 %), полноты (+20,5 %), AUC (+49,91 %) и ухудшилось значение F-меры (–7,98 %), в двух случаях значения метрик не изменились, и в четырех случаях ухудшились значения достоверности (–0,73 %), точности (–1,99 %), F-меры (–0,65 %), AUC (–1,01 %) и улучшились значения полноты (+0,48 %). В среднем при поиске, ограниченном онтологией, улучшились значения достоверности (+0,54 %), полноты (+0,34 %) и AUC (+1,85%) и ухудшились средние значения точности (–1,21%) и F-меры (–0,45%). Ухудшение точности и F-меры может быть объяснено недостаточным объемом знаний в онтологии, используемой подсистемой управления синтезом моделей. При увеличении объемов обработанных данных и увеличении объема знаний можно ожидать улучшение значений метрик получаемых моделей МО.

Заключение

В рамках проведенного исследования был разработан общий метаалгоритм управления процессами синтеза моделей МО. Метаалгоритм обеспечил возможность использования базы знаний, наполняемой из различных источников, включая экспертные знания и знания, получаемые с применением существующих систем автоматизированного МО, для сокращения пространства поиска алгоритмов при синтезе моделей МО. Предложен частный алгоритм управления процессами синтеза на основе метаобучения и построения онтологии, реализующий общий метаалгоритм, который, как показали экспериментальные исследования, позволил ускорить поиск моделей в среднем на 41,12 % и улучшить средние значения трех из пяти метрик результирующих моделей.

Дальнейшая работа будет направлена на разработку других частных алгоритмов управления процессами синтеза моделей МО, определяемых в рамках общего метаалгоритма, на повышение эффективности предложенного частного алгоритма за счет использования дополнительных метасвойств и на реализацию предложенного частного алгоритма с использованием других систем автоматизированного МО.

Финансовая поддержка

Исследование выполнено при поддержке государственного бюджета, номер проекта FFZF-2025-0019.

Литература

1. **Горюнов М. Н., Мацкевич А. Г., Рыболовлев Д. А.** Синтез модели машинного обучения для обнаружения компьютерных атак на основе набора данных CICIDS2017. *Труды Института системного программирования РАН*, 2020, № 32, с. 81–94. doi:10.15514/ISPRAS–2020–32(5)–6
2. **Pei Y.** A comparative study of machine learning and automatic machine learning models for facial mask

recognition. *8th International Conference on Computer and Communication Systems (ICCCS)*, 2023, pp. 1047–1051. doi:10.1109/ICCCS57501.2023.10151333

3. **Kovalevsky V., Stankova E., Zhukova N., Ogiy O., Tristanov A.** *AutoML Framework for Labor Potential Modeling*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Science and Business Media Deutschland GmbH, 2023, pp. 87–98. doi:10.1007/978-3-031-36808-0_6
4. **Baratchi M., Wang C., Limmer S., van Rijn J. N., Hoos H., Bäck T., Olhofer M.** Automated machine learning: Past, present and future. *Artificial Intelligence Review*, 2024, no. 57. doi:10.1007/s10462-024-10726-1
5. **Salehin I., Islam Md. S., Saha P., Noman S. M., Tuni A., Hasan Md. M., Baten Md. A.** AutoML: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence*, 2024, no. 2, pp. 52–81. doi:10.1016/j.jiixd.2023.10.002
6. **Попова И. А., Ревунков Г. И., Гапанюк Ю. Е.** AutoML: исследование существующих программных реализаций и определение общей внутренней структуры решений. *Труды Института системного анализа Российской академии наук*, 2023, № 73, с. 43–54. doi:10.14357/20790279230106
7. **Radzi S. F. M., Karim M. K. A., Saripan M. I., Rahman M. A. A., Isa I. N. C., Ibahim M. J.** Hyperparameter tuning and pipeline optimization via Grid Search Method and Tree-Based AutoML in breast cancer prediction. *Journal of Personalized Medicine*, 2021, no. 11. doi:10.3390/jpm11100978
8. **Pokhrel P., Lazar A.** A comparison of AutoML hyperparameter optimization tools for tabular data. *The International FLAIRS Conference Proceedings*, 2023, no. 36. doi:10.32473/flairs.36.133357
9. **Karras A., Karras C., Schizas N., Avlonitis M., Sioutas S.** AutoML with bayesian optimizations for big data management. *Information*, 2023, no. 14. doi:10.3390/info14040223
10. **Pomsuwan T., Freitas A. A.** Genetic algorithm-based Auto-ML system for survival analysis. *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, ACM, New York, NY, USA, 2024, pp. 370–377. doi:10.1145/3605098.3635954
11. **Thornton C., Hutter F., Hoos H. H., Leyton-Brown K.** Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, 2013, pp. 847–855. doi:10.1145/2487575.2487629
12. **Feurer M., Eggensperger K., Falkner S., Lindauer M., Hutter F.** Auto-Sklearn 2.0: Hands-free AutoML via meta-learning. *Journal of Machine Learning Research*, 2022, no. 23, pp. 1–61. doi:10.5555/3586589.3586850
13. **Olson R. S., Moore J. H.** *TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning*. Automated Machine Learning: Methods, Systems, Challenges/ F. Hutter, L. Kotthoff and J. Vanschoren (Eds.). Springer International Publishing, Cham, 2019, pp. 151–160. doi:10.1007/978-3-030-05318-5_8
14. **LeDell E., Poirier S.** H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*, 2020, pp. 1–16.
15. **Jin H., Chollet F., Song Q., Hu X.** AutoKeras: An AutoML library for deep learning. *Journal of Machine Learning Research*, 2023, no. 24, pp. 1–6. doi:10.5555/3648699.3648705
16. **Jose S. T., Simeone O., Durisi G.** Transfer meta-learning: Information-theoretic bounds and information meta-risk minimization. *IEEE Transactions on Information Theory*, 2022, no. 68, pp. 474–501. doi:10.1109/TIT.2021.3119605
17. **Khan I., Zhang X., Rehman M., Ali R.** A literature survey and empirical study of meta-learning for classifier selection. *IEEE Access*, 2020, no. 8, pp. 10262–10281. doi:10.1109/ACCESS.2020.2964726
18. **Gomaa I., Mokhtar M. O. H., El-Tazi N., Zidane A.** SML-AutoML: A smart meta-learning automated machine learning framework. *Advances in Artificial Intelligence and Machine Learning*, 2024, no. 04, pp. 3071–3096. doi:10.54364/AAIML.2024.44176
19. **Kotlar M., Punt M., Radivojevic Z., Cvetanovic M., Milutinovic V.** Novel meta-features for automated machine learning model selection in anomaly detection. *IEEE Access*, 2021, no. 9, pp. 89675–89687. doi:10.1109/ACCESS.2021.3090936
20. **Dyrmishi S., Elshawi R., Sakr S.** A decision support framework for AutoML systems: A meta-learning approach. *International Conference on Data Mining Workshops (ICDMW)*, 2019, pp. 97–106. doi:10.1109/ICDMW.2019.00025
21. **Li Y., Shen Y., Jiang H., Bai T., Zhang W., Zhang C., Cui B.** Transfer learning based search space design for hyperparameter tuning. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, 2022, pp. 967–977. doi:10.1145/3534678.3539369
22. **Wang A., Zhang K., Wu H., Chen H., Wang M.** Meta-learning-integrated neural architecture search for few-shot hyperspectral image classification. *Electronics (Basel)*, 2025, no. 14. doi:10.3390/electronics14152952
23. **Humm B. G., Zender A.** An ontology-based concept for meta AutoML. *17th IFIP International Conference on Artificial Intelligence Applications and Innovations*, 2021, pp. 117–128. doi:10.1007/978-3-030-79150-6_10
24. **Aliev M. R., Baimuratov I. R.** Automation of machine learning pipeline design by an ontology as an integrative meta-learning model. *The XIII Majorov International Conference on Software Engineering and Computer Systems (MICSECS)*, 2021.
25. **Nayak A., Božić B., Longo L.** *An Ontological Approach for Recommending a Feature Selection Algorithm*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and

Lecture Notes in Bioinformatics). Springer Science and Business Media Deutschland GmbH, 2022, pp. 300–314. doi:10.1007/978-3-031-09917-5_20
26. **Kallab L., Mansour E., Chbeir R.** SML: Semantic machine learning model ontology. *24th International Conference Web Information Systems Engineering (WISE)*, 2023, pp. 896–911. doi:10.1007/978-981-99-7254-8_70
27. **Klironomos A.** Towards democratized machine learning: A semantic web approach. *Companion Proceedings of the ACM on Web Conference*, ACM, New York, NY, USA, 2025, pp. 697–700. doi:10.1145/3701716.3715280
28. **Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay É.** Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011, no. 12, pp. 2825–2830. doi:10.5555/1953048.2078195
29. **Ковалевский В. Э., Ман Т., Игнатов Д. И., Жукова Н. А., Куликов И. А.** *Auto-Onto — платформа автоматизированного машинного обучения с использованием онтологий*. https://github.com/DarkEol/AutoML/tree/main/AutoML-Ontology (дата обращения: 08.05.2025).

UDC 004.852
doi:10.31799/1684-8853-2025-6-28-41
EDN: KKSXFO

Meta-algorithm for the process control of complex machine learning model synthesis

N. A. Zhukova[a], Dr. Sc., Tech., Associate Professor, ORCID.ORG/0000-0001-5877-4461, nazhukova@mail.ru
V. E. Kovalevsky[a], Junior Researcher, orcid.org/0000-0002-0414-906X
[a]St. Petersburg Federal Research Center of the Russian Academy of Sciences, 39, 14th Line V.O., 199178, Saint-Petersburg, Russian Federation

Introduction: Automated machine learning methods allow automating synthesis of machine learning models adapted to specific data processing. However, these methods require significant time and computational costs. **Purpose:** To develop a meta-algorithm for the process control of synthesis of machine learning models, that would reduce the computational complexity of automated synthesis of machine learning models. **Results:** We propose a general meta-algorithm for the process control of synthesis of complex machine learning models and a specific algorithm that allows limiting the search space through meta-learning. The proposed specific algorithm is based on using meta-features of data and an ontology that contains rules for selecting machine learning algorithms depending on the meta-features of the processed data. The ontology is constructed by pre-processing the results of previously synthesized machine learning models. In addition, for the specific algorithm, we develop an algorithm for building a training set and an algorithm for constructing an ontology to reduce the search space. The experiments have shown that the use of the proposed specific algorithm reduces the time of the synthesis of machine learning models by 41.12%. Moreover, the obtained models have increased accuracy (+0.54%), recall (+0.34%) and AUC (+1.85%). **Practical relevance:** The specific algorithm developed on the base of the meta-algorithm allows reducing the computational complexity of the process of automated machine learning model synthesis and enables the application of machine learning in subject areas that require prompt construction and adaptation of machine learning models to new data and tasks.

Keywords — automated machine learning, synthesis of machine learning models, AutoML, meta-learning, ontologies for AutoML, control of machine learning model synthesis.

For citation: Zhukova N. A., Kovalevsky V. E. Meta-algorithm for the process control of complex machine learning model synthesis. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2025, no. 6, pp. 28–41 (In Russian). doi:10.31799/1684-8853-2025-6-28-41, EDN: KKSXFO

Financial support
This work was supported by the state budget, project No. FFZF-2025-0019.

References

1. Goryunov M. N., Matskevich A. G., Rybolovlev D. A. Synthesis of a machine learning model for detecting computer attacks based on the CICIDS2017 dataset. *Proceedings of the Institute for System Programming of the RAS*, 2020, no. 32, pp. 81–94 (In Russian). doi:10.15514/ISPRAS–2020–32(5)–6
2. Pei Y. A comparative study of machine learning and automatic machine learning models for facial mask recognition. *8th International Conference on Computer and Communication Systems (ICCCS)*, 2023, pp. 1047–1051. doi:10.1109/ICCCS57501.2023.10151333
3. Kovalevsky V., Stankova E., Zhukova N., Ogiy O., Tristanov A. *AutoML framework for labor potential modeling*. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Science and Business Media Deutschland GmbH, 2023, pp. 87–98. doi:10.1007/978-3-031-36808-0_6
4. Baratchi M., Wang C., Limmer S., van Rijn J. N., Hoos H., Bäck T., Olhofer M. Automated machine learning: Past, present and future. *Artificial Intelligence Review*, 2024, no. 57. doi:10.1007/s10462-024-10726-1
5. Salehin I., Islam Md. S., Saha P., Noman S. M., Tuni A., Hasan Md. M., Baten Md. A. AutoML: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence*, 2024, no. 2, pp. 52–81. doi:10.1016/j.jiixd.2023.10.002
6. Popova I. A., Revunkov G. I., Gapanyuk Y. E. AutoML: Examining existing software implementations and determining the overall internal structure of solutions. *Proceeding of the Institute for Systems Analysis of the Russian Academy of Science*, 2023, no. 73, pp. 43–54 (In Russian). doi:10.14357/20790279230106
7. Radzi S. F. M., Karim M. K. A., Saripan M. I., Rahman M. A. A., Isa I. N. C., Ibahim M. J. Hyperparameter tuning and pipeline optimization via Grid Search Method and Tree-Based AutoML in breast cancer prediction. *Journal of Personalized Medicine*, 2021, no. 11. doi:10.3390/jpm11100978

8. Pokhrel P., Lazar A. A comparison of AutoML hyperparameter optimization tools for tabular data. *The International FLAIRS Conference Proceedings*, 2023, no. 36. doi:10.32473/flairs.36.133357
9. Karras A., Karras C., Schizas N., Avlonitis M., Sioutas S. AutoML with bayesian optimizations for big data management. *Information*, 2023, no. 14. doi:10.3390/info14040223
10. Pomsuwan T., Freitas A. A. Genetic algorithm-based Auto-ML system for survival analysis. *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, ACM, New York, NY, USA, 2024, pp. 370–377. doi:10.1145/3605098.3635954
11. Thornton C., Hutter F., Hoos H. H., Leyton-Brown K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, 2013, pp. 847–855. doi:10.1145/2487575.2487629
12. Feurer M., Eggensperger K., Falkner S., Lindauer M., Hutter F. Auto-sklearn 2.0: Hands-free AutoML via meta-learning. *Journal of Machine Learning Research*, 2022, no. 23, pp. 1–61. doi:10.5555/3586589.3586850
13. Olson R. S., Moore J. H. *TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning*. In: *Automated Machine Learning: Methods, Systems, Challenges*. F. Hutter, L. Kotthoff and J. Vanschoren (Eds.). Springer International Publishing, Cham, 2019, pp. 151–160. doi:10.1007/978-3-030-05318-5_8
14. LeDell E., Poirier S. H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*, 2020, pp. 1–16.
15. Jin H., Chollet F., Song Q., Hu X. AutoKeras: An AutoML library for deep learning. *Journal of Machine Learning Research*, 2023, no. 24, pp. 1–6. doi:10.5555/3648699.3648705
16. Jose S. T., Simeone O., Durisi G. Transfer meta-learning: Information-theoretic bounds and information meta-risk minimization. *IEEE Transactions on Information Theory*, 2022, no. 68, pp. 474–501. doi:10.1109/TIT.2021.3119605
17. Khan I., Zhang X., Rehman M., Ali R. A literature survey and empirical study of meta-learning for classifier selection. *IEEE Access*, 2020, no. 8, pp. 10262–10281. doi:10.1109/ACCESS.2020.2964726
18. Gomaa I., Mokhtar M. O. H., El-Tazi N., Zidane A. SML-AutoML: A smart meta-learning automated machine learning framework. *Advances in Artificial Intelligence and Machine Learning*, 2024, no. 04, pp. 3071–3096. doi:10.54364/AAIML.2024.44176
19. Kotlar M., Punt M., Radivojevic Z., Cvetanovic M., Milutinovic V. Novel meta-features for automated machine learning model selection in anomaly detection. *IEEE Access*, 2021, no. 9, pp. 89675–89687. doi:10.1109/ACCESS.2021.3090936
20. Dyrmishi S., Elshawi R., Sakr S. A decision support framework for AutoML systems: A meta-learning approach. *International Conference on Data Mining Workshops (ICDMW)*, 2019, pp. 97–106. doi:10.1109/ICDMW.2019.00025
21. Li Y., Shen Y., Jiang H., Bai T., Zhang W., Zhang C., Cui B. Transfer learning based search space design for hyperparameter tuning. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, 2022, pp. 967–977. doi:10.1145/3534678.3539369
22. Wang A., Zhang K., Wu H., Chen H., Wang M. Meta-learning-integrated neural architecture search for few-shot hyperspectral image classification. *Electronics (Basel)*, 2025, no. 14. doi:10.3390/electronics14152952
23. Humm B. G., Zender A. An ontology-based concept for meta AutoML. *17th IFIP International Conference on Artificial Intelligence Applications and Innovations*, 2021, pp. 117–128. doi:10.1007/978-3-030-79150-6_10
24. Aliev M. R., Baimuratov I. R. Automation of machine learning pipeline design by an ontology as an integrative meta-learning model. *The XIII Majorov International Conference on Software Engineering and Computer Systems (MICSECS)*, 2021.
25. Nayak A., Božić B., Longo L. *An Ontological Approach for Recommending a Feature Selection Algorithm*. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Science and Business Media Deutschland GmbH, 2022, pp. 300–314. doi:10.1007/978-3-031-09917-5_20
26. Kallab L., Mansour E., Chbeir R. SML: Semantic machine learning model ontology. *24th International Conference Web Information Systems Engineering (WISE)*, 2023, pp. 896–911. doi:10.1007/978-981-99-7254-8_70
27. Klironomos A. Towards democratized machine learning: A semantic web approach. *Companion Proceedings of the ACM on Web Conference 2025*, ACM, New York, NY, USA, 2025, pp. 697–700. doi:10.1145/3701716.3715280
28. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay É. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011, no. 12, pp. 2825–2830. doi:10.5555/1953048.2078195
29. Kovalevsky V., Man T., Ignatov D., Zhukova N., Kulikov I. *Auto-Onto — platforma avtomatizirovannogo mashinnogo obuchenie s ispol'zovaniem ontologij* [Auto-Onto — A Framework for AutoML extended by use of ontologies]. 2024. Available at: https://github.com/DarkEol/AutoML/tree/main/AutoML-Ontology (accessed 8 May 2025).