

Прогнозирование времени выполнения суперкомпьютерных заданий с применением методов машинного обучения

В.В. Баранцев ^{1✉}, А.В. Мокряков ², А.А. Прилипко ¹

¹ НИЦ «Курчатовский институт», г. Москва, 119334, Россия

² РГУ им. А.Н. Косыгина, г. Москва, 119071, Россия

Ссылка для цитирования

Баранцев В.В., Мокряков А.В., Прилипко А.А. Прогнозирование времени выполнения суперкомпьютерных заданий с применением методов машинного обучения // Программные продукты и системы. 2025. Т. 38. № 4. С. 568–577. doi: 10.15827/0236-235X.152.568-577

Информация о статье

Группа специальностей ВАК: 2.3.5

Поступила в редакцию: 08.04.2025

После доработки: 19.05.2025

Принята к публикации: 20.05.2025

Аннотация. Предметом представленного в статье исследования является применение методов машинного обучения для прогнозирования времени выполнения заданий в суперкомпьютерных системах. Планировщик суперкомпьютерных заданий составляет расписание их запусков на основе пользовательских оценок времени выполнения. При этом пользователи в большинстве случаев значительно превышают время выполнения своих заданий, чтобы исключить риск их принудительного завершения по истечении заказанного времени. Это приводит к построению неоптимального расписания и существенному снижению качества планирования заданий. Прогнозирование времени выполнения заданий позволит планировщику формировать более точное расписание. В качестве метода исследования использован сравнительный анализ моделей машинного обучения, включая деревья решений, метод *k*-ближайших соседей, случайный лес, градиентный бустинг, нейронные сети и широкое обучение. Обучение моделей проводилось на статистических данных о выполнении заданий на суперкомпьютере МВС-10П ОП. Дополнительно рассмотрены подходы, направленные на повышение качества прогнозов, включая методы кластеризации и классификации заданий. Результаты исследования позволили выявить специфику применения машинного обучения для прогнозирования времени выполнения заданий в условиях ограниченного и не всегда информативного набора признаков. Показано, что существующие методы машинного обучения обладают определенными ограничениями, связанными с устойчивостью моделей и риском переобучения. Вместе с тем полученные данные дают возможность наметить пути повышения точности прогнозирования. Практическая значимость работы заключается в возможности применения ее результатов для оптимизации планирования заданий в суперкомпьютерных системах за счет повышения точности прогноза времени выполнения заданий.

Ключевые слова: суперкомпьютер, машинное обучение, прогнозирование времени выполнения, распределение ресурсов, кластеризация, классификация, регрессионный анализ

Благодарности. Результаты получены в рамках государственного задания НИЦ «Курчатовский институт» по теме FNEF-2024-0016

Введение. Современные суперкомпьютеры функционируют в режиме коллективного пользования. Для эффективного управления этими ресурсами пользователи формулируют задания, включающие расчетные программы, исходные данные и требования к объему ресурсов и времени выполнения. Формируют расписание системы управления заданиями (СУЗ), оптимизируя использование ресурсов. Однако, если выполнение задания превышает запрошенный пользователем лимит времени, оно может быть автоматически завершено системой. Опасаясь прерывания, пользователи склонны значительно превышать лимиты времени. В результате система вынуждена резервировать избыточные ресурсы, что и приводит к снижению общей эффективности вычислительного процесса.

Завышение пользователями времени, необходимого для выполнения заданий, является одной из ключевых проблем. Это затрудняет пла-

нирование и распределение ресурсов, особенно для небольших заданий. Неточные прогнозы времени выполнения как в сторону завышения, так и занижения негативно влияют на производительность системы. В ряде работ была предпринята попытка повысить точность прогнозов с помощью машинного обучения. Например, в [1] предложена модель PREP, в которой вводится новый признак – путь выполнения задания, содержащий сведения о проекте, данных и параметрах. Такой подход позволил достичь точности прогнозирования до 88 %. В работе [2] этот метод был масштабирован на разные суперкомпьютерные системы, что подтвердило эффективность модели и ее применимость в реальных условиях, включая интеграцию в симулятор SLURM. В [3] предложен альтернативный подход, основанный на классификации вместо регрессии, позволяющий более эффективно избегать занижения времени выполне-

ния, используя кластеризацию и статистические поправки. Исследования показывают (например [4]), что достижение высокой точности прогнозирования ($R^2 \geq 0,8$) является критически важным для существенного повышения эффективности планирования заданий в системах с распределенными суперкомпьютерными центрами и для безопасного использования мягких лимитов времени администраторами систем [5]. Авторы [6] подчеркивают важность точного предсказания времени выполнения для заданий оркестрации в гетерогенных средах.

В работе [7] был проведен предварительный анализ данных, выявлены ключевые признаки, влияющие на время выполнения заданий, и изучены их корреляционные взаимосвязи. Настоящее исследование является логическим продолжением, и направлено на построение и сравнение моделей машинного обучения для более точного прогнозирования времени выполнения заданий. Основные методы исследования – оценка достижимой точности моделей на имеющемся наборе признаков, выявление ключевых ограничений и определение условий, при которых возможно кардинальное улучшение качества прогнозов. Для достижения цели исследуются различные архитектуры нейронных сетей, ансамблевые методы и подходы к предварительной обработке данных, включая кластеризацию и классификацию заданий по характерным признакам.

Обзор предшествующих работ

Исследование [7], которое основано на данных за 2022 год, продемонстрировало наличие корреляции между рядом параметров и временем выполнения заданий. Установлено, что наибольшую зависимость времени выполнения можно наблюдать при изменении таких параметров, как запрашиваемое пользователем время и количество вычислительных ядер. При этом выявлена отрицательная связь между временем выполнения и коэффициентом неиспользованного времени. Вместе с тем такие признаки, как идентификаторы пользователя и организации, а также временные характеристики (час и месяц), демонстрируют низкую информативность.

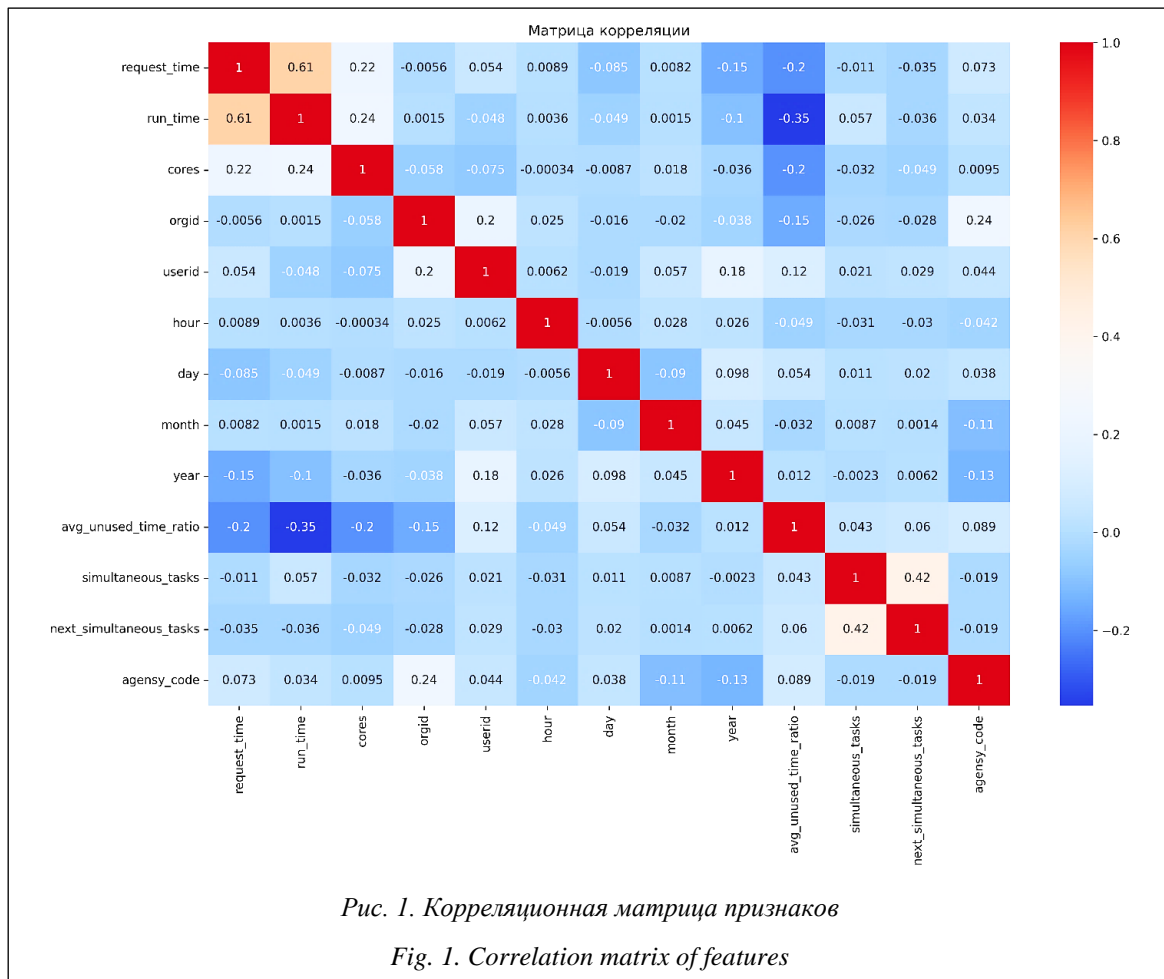
Данная работа развивает направления, наметенные в исследовании [8], в котором так же ставилась задача повышения точности прогноза времени выполнения суперкомпьютерных заданий с использованием методов машинного обучения. В работе [9] был представлен высокоточный предсказательный механизм, реализо-

ванный на суперкомпьютере, что акцентирует внимание на важности разработки максимально точных моделей. Исследование [10] сосредоточено на выявлении признаков, оказывающих наибольшее влияние на процесс выполнения заданий. Сделан вывод о том, что характеристики, описывающие поведение пользователей, обладают большей прогностической значимостью по сравнению с временными параметрами. Кроме того, персонализированные метрики, отражающие индивидуальные особенности использования системы, оказываются более полезными, чем простые идентификаторы пользователей или организаций. Если точность прогнозирования окажется недостаточной, возможно, потребуется дополнить набор признаков метриками, отражающими поведение пользователей и их индивидуальные особенности, как это предлагается в указанной работе.

Важный вклад в понимание практических аспектов интеграции прогнозных моделей в системы управления заданиями внесла работа [11], в которой исследуется потенциал оптимизации планирования при идеально точном прогнозе времени выполнения заданий. Авторы провели сравнительный анализ двух сценариев: использования времени, указанного пользователем, и реального времени выполнения, взятого из исторических данных (что эквивалентно прогнозу с нулевой ошибкой). Результаты симуляции на статистических данных суперкомпьютера МВС-10П ОП2 [12] показали, что применение точного времени выполнения позволяет уменьшить среднее время ожидания на 25 %, а приведенное время ожидания – на 50 %. Работа демонстрирует значительный потенциал оптимизации планирования заданий при условии достижения высокой точности прогнозирования, однако не предлагает конкретной модели машинного обучения, а лишь задает целевой ориентир для ее разработки. В отличие от [12], настоящее исследование направлено на оценку точности прогнозных моделей на основе методов машинного обучения.

Анализ параметров, влияющих на время выполнения заданий, и их взаимосвязей

В рамках исследования был проведен детальный анализ параметров, влияющих на фактическое время выполнения заданий (целевая переменная – `run_time`). Для выявления ключевых факторов, оказывающих наибольшее воздействие на целевую переменную, была построена корреляционная матрица признаков (рис. 1)



на основе выборки 32 459 заданий с 13 признаками. Этот метод позволил оценить степень взаимосвязи между всеми рассматриваемыми параметрами из [7], включая новый признак – количество одновременно запущенных или поставленных в очередь заданий одним пользователем (simultaneous_tasks).

Корреляционная матрица выявила, что наибольшее влияние на фактическое время выполнения оказывает запрашиваемое пользователями время (request_time). Среди остальных параметров выделяются количество запрашиваемых ядер (cores) и среднее соотношение неиспользуемого времени (avg_unused_time_ratio).

Подготовка и предварительная обработка данных

На основании выявленных взаимосвязей сформирована выборка, объединяющая категориальные и числовые параметры. Матрица признаков x имела размерность 32 459,20, а вектор целевой переменной y – 32 459. Для обеспечения сопоставимости данных применялось мас-

штабирование с использованием метода StandardScaler из библиотеки sklearn. Параметры (среднее и стандартное отклонение) рассчитываются только на тренировочной выборке, а затем применяются к валидационной и тестовой.

Последующий анализ важности признаков показал, что некоторые из них оказывают крайне слабое влияние на целевую переменную. В частности, пять параметров имели вклад менее 0,01, что указывало на их незначимость для прогнозирования. Благодаря исключению их из модели улучшилось ее качество.

Выборка была разделена на обучающую, валидационную и тестовую подвыборки в соотношении 72, 18 и 10 % соответственно. Это позволило обеспечить репрезентативность данных на всех этапах работы с моделями. Обучающая выборка составила 23 370 строк, валидационная – 5 843, тестовая – 3 246.

Предварительная обработка данных, включая фильтрацию малозначимых признаков, создала основу для применения алгоритмов машинного обучения, направленных на прогнозирование времени выполнения заданий.

Применение алгоритмов машинного обучения

Для решения задачи прогнозирования времени выполнения заданий были исследованы различные алгоритмы машинного обучения, включая деревья решений, метод k -ближайших соседей, случайный лес, градиентный бустинг, нейронные сети и широкое обучение (broad learning). Подробное описание теоретических основ и практических аспектов применения данных алгоритмов представлено в ряде работ. Так, в [13, 14] рассматриваются классические методы построения деревьев решений и непараметрической регрессии, включая метод k -ближайших соседей. В [15, 16] представлены ансамблевые подходы – случайный лес и градиентный бустинг. Обзор нейронных сетей приведен в [17], где подчеркивается значимость глубокого обучения для задач распознавания и прогнозирования. В [18] описывается метод broad learning, который является альтернативой глубоким нейросетям и позволяет строить модели с меньшими вычислительными затратами и более простой архитектурой. Каждый из подходов был реализован с учетом специфики регрессионной задачи, где целевой переменной выступало фактическое время выполнения.

На первом этапе была построена модель на основе нейронной сети со следующей архитектурой: три скрытых слоя с 256, 128 и 64 нейронами соответственно, с функцией активации ReLU. Между слоями применялись методы нормализации выходных данных и дропаут с вероятностью 20 % (отключение случайных нейронов) для предотвращения переобучения. Выходной слой состоял из одного нейрона без функции активации, что соответствует задаче регрессии. Для оптимизации использовался алгоритм Adam с параметром скорости обучения равный 0,001, в качестве функции потерь – среднеквадратичная ошибка (MSE), а в качестве метрики – средняя абсолютная ошибка (MAE). Обучение проводилось в течение 200 эпох с размером батча 64 и использованием 10 % данных для валидации.

Результаты показали, что модель достигла коэффициента детерминации $R^2 = 0,53$ (53 % дисперсии объясняется целевой переменной) на тестовой выборке и $R^2 = 0,64$ на обучающей, при этом ошибка прогнозирования составила 72 %. Анализ продемонстрировал, что качество модели может быть улучшено за счет модификации архитектуры сети, изменения па-

раметров обучения или использования других методов регуляризации.

Параллельно были исследованы классические алгоритмы машинного обучения. Дерево решений было настроено с максимальной глубиной 5 и минимальным числом образцов для разделения узла 2. Метод k -ближайших соседей использовал равномерные веса, а случайный лес состоял из 100 деревьев с глубиной каждого дерева 10. Градиентный бустинг был реализован с 200 деревьями, со скоростью обучения 0,05 и с глубиной деревьев 3. Модель широкого обучения использована в 10 нейронах на каждом из 10 окон, 10 нейронов в усиливающем слое, коэффициент масштабирования 0,5 и параметр регуляризации 2^{-30} . Результаты отражены в таблице 1.

Таблица 1

Результаты обучения моделей

Table 1

Model training results

| Алгоритм | Точность прогнозирования | | |
|------------------------|--------------------------|----------------|-----------|
| | R^2 на обучении | R^2 на тесте | Ошибка, % |
| Нейронная сеть | 0,64 | 0,53 | 72 |
| Дерево решений | 0,52 | 0,50 | 75 |
| k -ближайших соседей | 0,64 | 0,49 | 73 |
| Случайный лес | 0,81 | 0,58 | 65 |
| Градиентный бустинг | 0,72 | 0,56 | 67 |
| Широкое обучение | 0,42 | 0,41 | 89 |

Наилучшие результаты среди классических алгоритмов продемонстрировал метод случайного леса, достигнув $R^2 = 0,58$ при ошибке 65 %. Однако для всех моделей наблюдалось переобучение, выразившееся в значительном разрыве между качеством на обучающей ($R^2 > 0,7$) и на тестовой выборках. Дополнительная настройка гиперпараметров позволила улучшить показатель R^2 до 0,59 при ошибке 64 %, но проблема переобучения сохранилась.

Сравнительный анализ показал, что наилучшие результаты были достигнуты с использованием случайного леса и нейронной сети. Первый алгоритм продемонстрировал более высокую точность, в то время как второй – потенциал для дальнейшего улучшения за счет гибкости архитектуры и методов оптимизации.

Полученные результаты подчеркивают необходимость исследования дополнительных

подходов к улучшению качества прогнозирования. В частности, представляет интерес изучение методов, позволяющих предусмотреть внутреннюю структуру данных, таких как кластеризация и классификация.

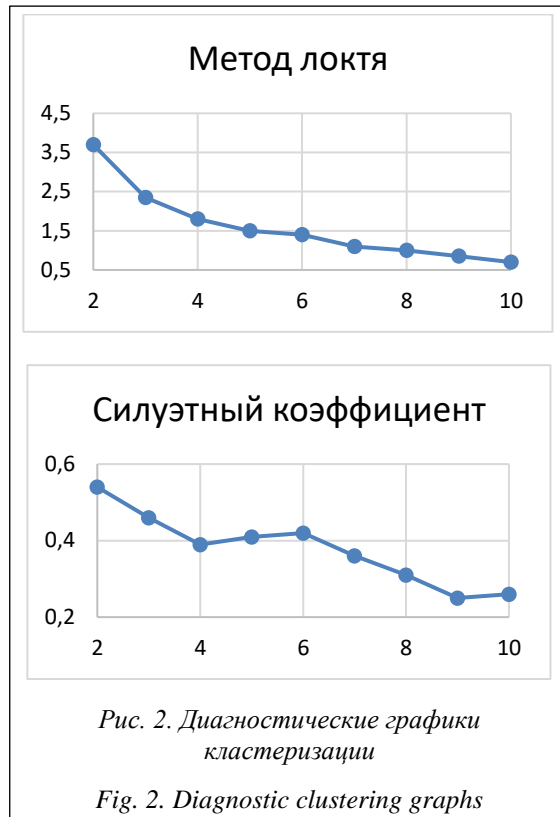
Исследование методов кластеризации

В рамках исследования потенциала улучшения качества прогнозирования времени выполнения заданий был проведен анализ методов кластеризации. Такой подход был направлен на выявление внутренней групповой структуры данных, которая могла бы быть использована для построения более точных прогнозных моделей. С целью объективного определения оптимального количества кластеров применялись метод локтя и оценка силуэтного коэффициента.

Подготовленная функция реализовала комплексный подход, последовательно применяя алгоритм k -ближайших соседей для разбиения данных на количество кластеров от 2 до 10. Для каждой конфигурации вычислялись два ключевых показателя: суммарная внутрикластерная дисперсия (для метода локтя) и силуэтный коэффициент. Метод локтя основан на анализе изменения дисперсии при увеличении числа кластеров, где оптимальное количество соответствует точке, после которой дальнейшее увеличение числа групп не приводит к существенному уменьшению дисперсии. Силуэтный коэффициент оценивает качество кластеризации, показывая степень схожести объектов внутри кластера и их отличия от объектов других кластеров.

Анализ полученных графиков (рис. 2) показал, что оптимальным является разделение данных на 3 кластера. Этот вывод подтверждается как положением локтя на графике дисперсии, свидетельствующим о замедлении ее снижения после трех кластеров, так и значением силуэтного коэффициента (0,47), которое начинает уменьшаться при дальнейшем увеличении числа групп.

Определение оптимального числа кластеров является важным этапом для последующего анализа, поскольку каждый кластер потенциально представляет собой группу заданий со схожими характеристиками, которые могут быть связаны с различными паттернами времени выполнения. В отличие от классификации, где классы заданы априори, кластеризация позволяет выявить естественную структуру данных.



Полученные кластеры были использованы в моделировании двумя основными способами. Первый подход предполагал добавление метки кластера в качестве дополнительного признака к исходным данным. Второй заключался в построении отдельной регрессионной модели прогнозирования времени выполнения для каждого кластера.

Проведенный анализ показал, что добавление кластерного признака в качестве дополнительного входного параметра не привело к значимому улучшению качества предсказаний. Несмотря на эксперименты с различным количеством кластеров (от 3 до 10), важность этого признака в модели оставалась крайне низкой (менее 1 % вклада), что свидетельствует о его нерелевантности для решения поставленной задачи. Статистический анализ распределения запрашиваемого пользователем времени (`request_time`) внутри трех кластеров (табл. 2) показал их высокую схожесть по ключевым статистическим показателям. Для создания кластеров и последующего регрессионного моделирования использовался весь набор признаков. Такой подход оправдан, поскольку алгоритм k -ближайших соседей оперирует многомерным пространством признаков, и исключение каких-либо параметров искусственно обедняет модель и не позволяет выявить слож-

Таблица 2

Распределение запрашиваемого времени внутри 3-х кластеров

Table 2

Distribution of the requested time within three clusters

| Кластер | Количество объектов | Среднее значение | Стандартное отклонение | Медианное значение | Максимальное значение |
|---------|---------------------|------------------|------------------------|--------------------|-----------------------|
| 1 | 11 212 | 13 163,73 | 24 524,49 | 1554 | 86 400 |
| 2 | 11 032 | 12 984,71 | 24 428,25 | 1 258 | 86 400 |
| 3 | 10 215 | 13 129,27 | 24 426,94 | 1 498 | 86 400 |

ные скрытые взаимосвязи между всеми характеристиками задания и его итоговой длительностью.

Детали результатов анализа добавления признака десяти кластеров следующие:

- средние значения запрашиваемого времени варьируются в узком диапазоне (11,824–20,341) при общем размахе данных до 86,400;
- стандартные отклонения остаются чрезвычайно высокими (~23,000–31,000) во всех кластерах;
- максимальные значения (86,400) присутствуют практически во всех кластерах;
- коэффициент детерминации по-прежнему не превышает значение 0,6, при этом процент ошибки повысился в среднем до 67.

Из такого подхода следует, что кластерный признак имеет низкую объясняющую способность.

Применение второго метода, заключавшегося в построении отдельных моделей для каждого из трех кластеров, также не привело к ожидаемому улучшению. Коэффициент детерминации R^2 на тестовой выборке варьировался в диапазоне 0,55–0,58, что существенно ниже показателей на обучающих данных (0,79–0,87). Значительный разрыв указывает на выраженное переобучение моделей, которое не только не было нивелировано кластерным подходом, но и в некоторой степени усугублено. Дальнейшее увеличение количества кластеров до 10 привело к дополнительному ухудшению качества прогнозирования. Внутри отдельных кластеров значение R^2 снижалось до 0,16, а в некоторых случаях достигало отрицательных значений (–0,9), что указывает на полную неадекватность моделей для соответствующих подгрупп данных.

На основании проведенного исследования можно заключить, что применение кластеризации на основе алгоритма k -ближайших соседей не оправдало возложенных ожиданий и не может быть рекомендовано для улучшения точности прогнозирования времени выполнения за-

даний в рамках данной постановки. Полученные результаты ставят под сомнение целесообразность использования предварительной кластеризации для решения данной прогностической задачи. Можно сделать вывод, что признаки, использованные для кластеризации, не позволяют выделить группы заданий с однородным временем выполнения. Соответственно, представляется логичным исследование альтернативных подходов к структурированию данных, в частности, методов классификации заданий по заранее определенным категориям.

Исследование методов классификации

В продолжение исследования методов повышения точности прогнозирования времени выполнения заданий был проведен комплексный анализ подхода, основанного на предварительной классификации заданий по величине запрашиваемого времени выполнения (`request_time`). Данный подход предполагал, что группировка заданий по категориям со схожими характеристиками может улучшить качество прогнозных моделей. Для предварительного разделения заданий на классы по значению `request_time` использовались квантили. Для последующего регрессионного моделирования внутри каждого класса использовался метод случайного леса с 200 деревьями и максимальной глубиной 15.

Экспериментальная работа включала два последовательных этапа. На первом осуществлялось добавление класса времени запроса в качестве дополнительного признака к исходному набору данных. Таблица 3 содержит статистику по классам переменных запрашиваемого и фактического времени (`request_time` и `run_time`, в таблице – `rq` и `rn` соответственно). Значения запрашиваемого времени (`rq`) представлены в стандартизированном виде.

Для формирования классов применялось разделение на 5 категорий по квантилям запрашиваемого времени, что позволило получить

сбалансированные группы с четкими различиями в среднем времени выполнения. Анализ статистики по классам показал, что задания с наименьшими значениями запрашиваемого времени (класс 1: $-0,91 \pm 0,72$) демонстрировали среднее время выполнения 509,93 секунды, в то время как задания с наибольшими значениями (класс 5: $1,58 \pm 36,53$) имели среднее время выполнения 41 923,57 секунды.

Добавление классового признака позволило достичь на тестовой выборке коэффициента детерминации $R^2 = 0,5976$ при ошибке 65 % от среднего значения, что превышает показатели базовой модели без классификации.

На втором этапе исследования для каждого класса построена отдельная прогнозная модель и были созданы индивидуальные наборы данных с персональным масштабированием признаков и обучением моделей.

Итоги отдельного моделирования (табл. 4) продемонстрировали существенную неоднородность качества прогнозирования в разных классах. Наилучшие результаты были достигнуты для заданий с минимальным запрашиваемым временем (класс 1), где коэффициент детерминации составил $R^2 = 0,6218$ при ошибке прогнозирования 51 %. Для заданий класса 2 показатель R^2 снизился до 0,5337 при ошибке 59 %. Наиболее низкое качество прогнозирования наблюдалось для заданий с наибольшим временем выполнения: класс 4 ($R^2 = 0,3956$, ошибка 71 %) и класс 5 ($R^2 = 0,2900$, ошибка 61 %).

Эксперименты с увеличением количества классов до 15 привели к неравномерному распределению данных по категориям и к хаотичным результатам. Хотя некоторые классы с малым объемом данных показали относительно высокие значения R^2 ($> 0,53$), подавляющая часть категорий, особенно с большим объемом данных, продемонстрировала низкое качество прогнозирования ($R^2 = 0,27-0,50$).

Сравнительный анализ двух подходов показал, что добавление классового признака в общую модель обеспечивает более стабильные результаты во всех категориях в отличие от отдельного моделирования. Отдельное обучение моделей для каждого класса показало преимущество только для заданий с малым временем выполнения, в то время как для длительных операций комбинированный подход оказался эффективнее. Напрашивается вывод, что для коротких заданий поведение более предсказуемо, а для длительных заданий большое влияние оказывают непредсказуемые факторы (например, очередь ввода-вывода, состояние сети, более сложная и разнообразная природа самих вычислений), которые не отражены в имеющихся признаках.

Во всех экспериментах сохранялся значительный разрыв между качеством на обучающей и тестовой выборках, что указывает на устойчивую проблему переобучения, не решаемую простым разделением данных на классы. Полученные результаты могут свидетельство-

Таблица 3

Статистика переменных внутри пяти классов

Table 3

Statistics of variables within five classes

| Класс | Среднее μ | Среднее σ | Отклонение σ | Количество объектов |
|-------|---------------|------------------|---------------------|---------------------|
| 1 | -0,91 | 509,93 | 724,60 | 7 923 |
| 2 | -0,76 | 1 788,05 | 2 356,92 | 6 395 |
| 3 | -0,45 | 6 588,07 | 7 152,58 | 5 235 |
| 4 | 0,67 | 16 463,76 | 21 159,87 | 6 523 |
| 5 | 1,58 | 41 923,57 | 36 532,86 | 6 383 |

Таблица 4

Результаты статистики и метрики классов

Table 4

Class statistics and metrics results

| Класс | Среднее μ | R^2 | Ошибка, % | Количество объектов |
|-------|---------------|--------|-----------|---------------------|
| 1 | 509,93 | 0,6218 | 51 | 7 923 |
| 2 | 1 788,05 | 0,5337 | 59 | 6 395 |
| 3 | 6 588,07 | 0,3989 | 61 | 5 235 |
| 4 | 16 463,76 | 0,3956 | 71 | 6 523 |
| 5 | 41 923,57 | 0,2900 | 61 | 6 383 |

вать как о недостаточной информативности имеющихся признаков для прогнозирования времени выполнения длительных заданий, так и о необходимости применения более сложных методов регуляризации или сбора дополнительных данных.

Заключение

В ходе проведенного исследования был выполнен анализ методов машинного обучения для прогнозирования времени выполнения заданий на суперкомпьютерах. На основе реальных данных была проведена сравнительная оценка эффективности различных алгоритмов, а также исследовано влияние на точность прогноза дополнительных подходов, включая кластеризацию и предварительную классификацию задач.

Полученные результаты показали, что использование машинного обучения в рассматриваемой задаче сталкивается с рядом ограниче-

ний, связанных с неполнотой исходных данных и недостаточной информативностью признаков. Проведенный сравнительный анализ подтвердил возможность выявления закономерностей, которые могут быть использованы для последующего совершенствования прогнозных моделей. Вместе с тем было выявлено, что для достижения более высоких показателей точности требуется дальнейшее развитие применяемых подходов.

Полученные результаты определяют перспективные направления исследований. Наиболее важным из них является расширение набора признаков для повышения информативности данных. Кроме того, представляется целесообразным исследование альтернативных методов кластеризации, применение продвинутых методов регуляризации для повышения обобщающей способности моделей, а также интеграция разработанных прогнозных моделей в системы управления заданиями для их валидации в реальных условиях.

Список литературы

1. Zhou L., Zhang X., Yang W. et al. PREP: Predicting job runtime with job running path on supercomputers. Proc. ICPP, 2021, art. 16. doi: 10.1145/3472456.3473521.
2. Yang W., Liao X., Dong D. et al. Exploring job running path to predict runtime on multiple production supercomputers. JPDC, 2023, vol. 175, pp. 109–120. doi: 10.1016/j.jpdc.2023.01.001.
3. Cui H., Takahashi K., Shimomura Y., Takizawa H. Clustering based job runtime prediction for backfilling using classification. Proc. JSSPP, 2024, pp. 40–59. doi: 10.1007/978-3-031-74430-3_3.
4. Shabanov B., Baranov A., Telegin P., Tikhomirov A. Influence of execution time forecast accuracy on the efficiency of scheduling jobs in a distributed network of supercomputers. In: LNTCS. Proc. PaCT, 2021, vol. 12942, pp. 318–332. doi: 10.1007/978-3-030-86359-3_25.
5. Klusacek D., Chlumsky V. Evaluating the impact of soft walltimes on job scheduling performance. In: LNTCS. Proc. JSSPP, 2019, vol. 11332, pp. 15–38. doi: 10.1007/978-3-030-10632-4_2.
6. Chupakhin A., Kolosov A., Bahmurov A. et al. Application of recommender systems approaches to the MPI program execution time prediction. Proc. MoNeTeC, 2020, pp. 1–7. doi: 10.1109/MoNeTeC49726.2020.9258345.
7. Баранцев В.В. Прогнозирование времени выполнения заданий с использованием методов машинного обучения // ИНТЕКС-2025: сб. матер. Всерос. науч. конф. молодых исследователей с междунар. уч. 2025. Ч. 4. С. 55–58.
8. Savin G.I., Shabanov B.M., Nikolaev D.S. et al. Jobs runtime forecast for JSCC RAS supercomputers using machine learning methods. Lobachevskii J. of Math., 2020, vol. 41, no. 12, pp. 2593–2602. doi: 10.1134/S1995080220120343.
9. Savin G.I., Lyakhovets D.S., Baranov A.V. Influence of job runtime prediction on scheduling quality. Lobachevskii J. of Math., 2021, vol. 42, no. 11, pp. 2562–2570. doi: 10.1134/S1995080221110196.
10. Рыбаков А.А., Шумилин С.С. Сравнение алгоритмов машинного обучения для предсказания времени работы пользовательских заданий в рамках оптимизации использования ресурсов суперкомпьютерного кластера МСЦ РАН // Тр. НИИСИ РАН. 2020. Т. 10. № 2. С. 4–13. doi: 10.25682/NIISI.2020.2.0001.
11. Шумилин С.С., Воробьев М.Ю. Использование симулятора планировщика заданий для оценки эффективности предсказания времени работы задания // Программные продукты и системы. 2022. Т. 35. № 1. С. 124–131. doi: 10.15827/0236-235X.137.124-131.
12. МСЦ РАН. Вычислительные ресурсы. URL: <https://www.jssc.ru/resources/hpc/> (дата обращения: 20.11.2024).
13. Rokach L., Maimon O. Data Mining with Decision Trees: Theory and Applications. Singapore, 2007, 264 p. doi: 10.1142/6604.
14. Altman N.S. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 1992, vol. 46, no. 3, pp. 175–185. doi: 10.1080/00031305.1992.10475879.
15. Breiman L. Random forests. Machine Learning, 2001, vol. 45, pp. 5–32. doi: 10.1023/A:1010933404324.
16. Friedman J.H. Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 2001, vol. 29, no. 5, pp. 1189–1232. doi: 10.1214/aos/1013203451.
17. Schmidhuber J. Deep learning in neural networks: An overview. Neural Networks, 2015, vol. 61, pp. 85–117. doi: 10.1016/j.neunet.2014.09.003.

18. Hou Z., Shen H., Feng Q. et al. Optimizing job scheduling by using broad learning to predict execution times on HPC clusters. *CCF Transactions on High Performance Computing*, 2024, vol. 6, pp. 365–377. doi: 10.1007/s42514-023-00137-z.

Software & Systems

doi: 10.15827/0236-235X.152.568-577

2025, 38(4), pp. 568–577

Supercomputer task runtime forecast using machine learning methods

Vitaliy V. Barantsev ^{1✉}, Aleksey V. Mokryakov ², Aleksey A. Prilipko ¹

¹ National Research Centre “Kurchatov Institute”, Moscow, 119334, Russian Federation

² A.N. Kosygin Russian State University, Moscow, 119071, Russian Federation

For citation

Barantsev, V.V., Mokryakov, A.V., Prilipko, A.A. (2025) ‘Supercomputer task runtime forecast using machine learning methods’, *Software & Systems*, 38(4), pp. 568–577 (in Russ.). doi: 10.15827/0236-235X.152.568-577

Article info

Received: 08.04.2025

After revision: 19.05.2025

Accepted: 20.05.2025

Abstract. The paper applies machine learning methods to predict job execution times in supercomputer systems. The supercomputer job scheduler creates launch schedules based on user-provided runtime estimates. Users typically overestimate their jobs' execution time to avoid the risk of forced termination once the allocated time expires. This results in suboptimal schedule construction and significantly reduces overall scheduling efficiency. Job execution time prediction will enable the scheduler to generate more accurate schedules. The authors employed a comparative analysis of machine learning models as their research method, including decision trees, k-nearest neighbors, random forest, gradient boosting, neural networks, and broad learning. Model training utilized statistical data from job executions on the MVS-10P supercomputer. The study additionally examined approaches for improving prediction quality, including job clustering and classification methods. The research results revealed specific characteristics of applying machine learning for job execution time prediction with limited and often uninformative feature sets. The paper demonstrates that existing machine learning methods possess certain limitations related to model stability and overfitting risks. At the same time, the obtained results make it possible to identify potential ways to improve prediction accuracy. The practical significance of the study lies in the possibility of using its results to optimize job scheduling in supercomputing systems by increasing the accuracy of runtime forecasts.

Keywords: supercomputer, machine learning, execution time prediction, resource allocation, clustering, classification, regression analysis

Acknowledgements. The results were obtained within the framework of the state assignment of the National Research Centre “Kurchatov Institute” under a state-funded research project FNEF-2024-0016

References

1. Zhou, L., Zhang, X., Yang, W. et al. (2021) ‘PREP: Predicting job runtime with job running path on supercomputers’, *Proc. ICPP*, art. 16. doi: 10.1145/3472456.3473521.
2. Yang, W., Liao, X., Dong, D. et al. (2023) ‘Exploring job running path to predict runtime on multiple production supercomputers’, *JPDC*, 175, pp. 109–120. doi: 10.1016/j.jpdc.2023.01.001.
3. Cui, H., Takahashi, K., Shimomura, Y., Takizawa, H. (2024) ‘Clustering based job runtime prediction for backfilling using classification’, *Proc. JSSPP*, pp. 40–59. doi: 10.1007/978-3-031-74430-3_3.
4. Shabanov, B., Baranov, A., Telegin, P., Tikhomirov, A. (2021) ‘Influence of execution time forecast accuracy on the efficiency of scheduling jobs in a distributed network of supercomputers’, in *LNTCS. Proc. PaCT*, 12942, pp. 318–332. doi: 10.1007/978-3-030-86359-3_25.
5. Klusacek, D., Chlumsky, V. (2019) ‘Evaluating the impact of soft walltimes on job scheduling performance’, in *LNTCS. Proc. JSSPP*, 11332, pp. 15–38. doi: 10.1007/978-3-030-10632-4_2.
6. Chupakhin, A., Kolosov, A., Bahmurov, A. et al. (2020) ‘Application of recommender systems approaches to the MPI program execution time prediction’, *Proc. MoNeTeC*, pp. 1–7. doi: 10.1109/MoNeTeC49726.2020.9258345.
7. Barantsev, V.V. (2025) ‘Job execution time prediction using machine learning methods’, *Proc. All-Russ. Sci. Conf. INTEX-2025*, (4), pp. 55–58 (in Russ.).
8. Savin, G.I., Shabanov, B.M., Nikolaev, D.S. et al. (2020) ‘Jobs runtime forecast for JSCC RAS supercomputers using machine learning methods’, *Lobachevskii J. of Math.*, 41(12), pp. 2593–2602. doi: 10.1134/S1995080220120343.
9. Savin, G.I., Lyakhovets, D.S., Baranov, A.V. (2021) ‘Influence of job runtime prediction on scheduling quality’, *Lobachevskii J. of Math.*, 42(11), pp. 2562–2570. doi: 10.1134/S1995080221110196.
10. Rybakov, A.A., Shumilin, S.S. (2020) ‘Comparison of machine learning algorithms for predicting the user job runtime in the framework of optimizing the use of resources of supercomputer cluster of JSCC RAS’, *Proc. SRISA RAS*, 10(2), pp. 4–13 (in Russ.). doi: 10.25682/NIISI.2020.2.0001.

11. Shumilin, S.S., Vorobev, M.Y. (2022) 'Using a job scheduler simulator to evaluate the effectiveness of job runtime prediction', *Software & Systems*, 35(1), pp. 124–131 (in Russ.). doi: 10.15827/0236-235X.137.124-131.
12. JSCC RAS. *Computing resources*, available at: <https://www.jscc.ru/resources/hpc/> (accessed November 20, 2024) (in Russ.).
13. Rokach, L., Maimon, O. (2007) *Data Mining with Decision Trees: Theory and Applications*. Singapore, 264 p. doi: 10.1142/6604.
14. Altman, N.S. (1992) 'An introduction to kernel and nearest-neighbor nonparametric regression', *The American Statistician*, 46(3), pp. 175–185. doi: 10.1080/00031305.1992.10475879.
15. Breiman, L. (2001) 'Random forests', *Machine Learning*, 45, pp. 5–32. doi: 10.1023/A:1010933404324.
16. Friedman, J.H. (2001) 'Greedy function approximation: A gradient boosting machine', *The Annals of Statistics*, 29(5), pp. 1189–1232. doi: 10.1214/aos/1013203451.
17. Schmidhuber, J. (2015) 'Deep learning in neural networks: An overview', *Neural Networks*, 61, pp. 85–117. doi: 10.1016/j.neunet.2014.09.003.
18. Hou, Z., Shen, H., Feng, Q. et al. (2024) 'Optimizing job scheduling by using broad learning to predict execution times on HPC clusters', *CCF Transactions on High Performance Computing*, 6, pp. 365–377. doi: 10.1007/s42514-023-00137-z.

Авторы**Баранцев Виталий Витальевич**¹,инженер-исследователь,
barantsev.vitalik@mail.ru**Мокряков Алексей Викторович**²,к.ф.-м.н., заведующий кафедрой,
mokryakov-av@rguk.ru**Прилипко Алексей Алексеевич**¹, к.ф.-м.н.,ведущий научный сотрудник,
aaprilipko@mail.ru**Authors****Vitaliy V. Barantsev**¹,Research Engineer,
barantsev.vitalik@mail.ru**Aleksey V. Mokryakov**²,Cand. of Sci. (Physics and Mathematics),
Head of Chair, mokryakov-av@rguk.ru**Aleksey A. Prilipko**¹,Cand. of Sci. (Physics and Mathematics)
Leading Researcher, aaprilipko@mail.ru¹ НИЦ «Курчатовский институт»,
г. Москва, 119334, Россия² РГУ им. А.Н. Косыгина,
г. Москва, 119071, Россия¹ National Research Centre "Kurchatov Institute",
Moscow, 119334, Russian Federation² A.N. Kosygin Russian State University,
Moscow, 119071, Russian Federation