УДК 004.8
DOI:10.25729/ESI.2025.40.4.003

Диагностирование и прогнозирование состояний технических и технологических объектов на основе ансамблевых технологий машинного обучения

Ломакина Любовь Сергеевна, Двитовская Алиса Николаевна, Корелин Кирилл Александрович

Нижегородский государственный технический университет им. Р.Е. Алексеева, Россия, Нижний Новгород, *avd.alisa@gmail.com*

Аннотация. В статье исследуется применение ансамблевых методов машинного обучения для диагностики и прогнозирования состояний технических и технологических объектов в условиях зашумленных данных, нелинейных зависимостей и высокой размерности признакового пространства. Актуальность работы обусловлена необходимостью повышения надежности промышленных систем за счет минимизации аварийных рисков и оптимизации эксплуатационных процессов. Традиционные подходы демонстрируют недостаточную точность в сложных сценариях, что мотивирует использование ансамблевых технологий, объединяющих прогнозы множества моделей для достижения устойчивых результатов. Основное внимание уделено методам Бэггинг, Бустинг и Стекинг, их математическому обоснованию и практической реализации. Проведен эксперимент с ансамблем на основе сверточных нейронных сетей (CNN) для классификации дефектов микроструктуры металла. Результаты показали рост точности прогнозирования при увеличении числа классов по сравнению с одиночной моделью, что подтверждает эффективность ансамблей в снижении дисперсии ошибок и коррекции смещения моделей. Предложенный подход демонстрирует потенциал для внедрения в промышленные системы, повышая надежность диагностики и безопасность эксплуатации сложных технических систем.

Ключевые слова: диагностика технических объектов, прогнозирование состояний, ансамблевые методы, бэггинг, бустинг, стекинг, сверточные нейронные сети

Цитирование: Ломакина Л.С. Диагностирование и прогнозирование состояний технических и технологических объектов на основе ансамблевых технологий машинного обучения / Л.С. Ломакина, А.Н. Двитовская, К.А. Корелин // Информационные и математические технологии в науке и управлении, 2025. – № 4(40). – С.26-37. – DOI:10.25729/ESI.2025.40.4.003.

Введение. Современные промышленные системы требуют надежных методов диагностики состояний технических объектов для предотвращения аварийных ситуаций и оптимизации процессов эксплуатации. Рост числа датчиков, генерирующих огромные объемы данных в режиме реального времени, а также необходимость прогнозирования редких, но катастрофических отказов, обусловливают потребность в переходе к интеллектуальным решениям.

Однако традиционные методы интеллектуального анализа данных зачастую демонстрируют недостаточную точность в условиях зашумленных данных, нелинейных зависимостей и высокой размерности признакового пространства.

В подобных ситуациях ансамблевые модели машинного обучения, объединяющие предсказания нескольких базовых алгоритмов, позволяют достичь более высокой точности и устойчивости по сравнению с одиночными моделями.

Вклад работы в развитие научной области заключается в исследовании эффективности применения ансамблевых методов машинного обучения для диагностирования и прогнозирования состояний объектов.

В статье рассматриваются принципы работы ансамблевых моделей машинного обучения, основное внимание уделено методам Бэггинга (Bagging), Бустинга (Boosting) и Стекинга (Stacking).

1. Основы ансамблевых методов машинного обучения. Ансамблевые методы машинного обучения [1] – это подход, при котором решения множества моделей объединяются для повышения общей точности и надежности итогового результата.

Основная цель ансамблевых методов – минимизировать риски переобучения, снизить влияние случайных ошибок и повысить устойчивость алгоритмов к зашумленным данным или выбросам. Ключевая идея ансамблевых методов заключается в том, что коллективное решение группы моделей превосходит предсказание отдельного алгоритма, если выполняются условия независимости ошибок и их диверсификации.

Математическое обоснование этой идеи заключено в теореме Кондорсе о жюри присяжных [2]. Эта теорема утверждает, что если каждый член жюри присяжных обладает независимым мнением по поводу рассматриваемой ситуации, и если вероятность того, что любой из них принимает правильное решение, больше 0,5, то вероятность того, что решение присяжных в целом окажется правильным, возрастает с увеличением количества членов жюри и стремится к 1. Если же вероятность быть правым у каждого из членов жюри меньше 0,5, то вероятность принятия правильного решения присяжными в целом уменьшается и стремится к 0 с увеличением количества присяжных. Теорема Кондорсе о жюри присяжных иллюстрируется рисунком 1.

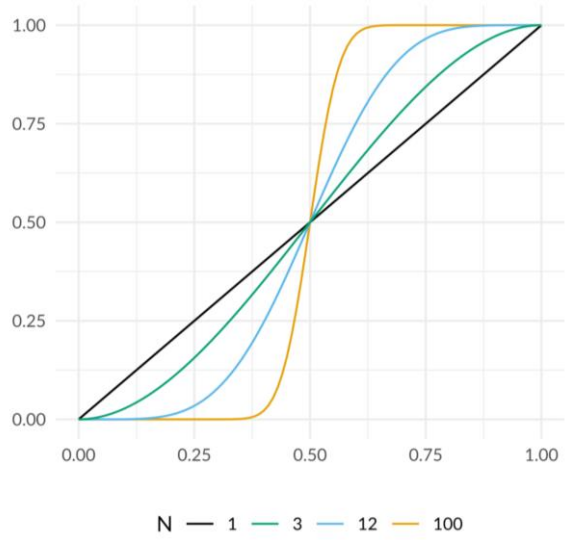

Рис. 1. График вероятностей успеха по теореме Кондорсе с параметрами по оси Y – общая вероятность успеха, по оси X – вероятность успеха члена жюри, N – кол-во членов жюри

На основании этой теоремы мы можем сделать вывод, что, имея несколько моделей с вероятностью успешного и независимого прогноза больше 0,5, можно объединить их результаты и тем самым достичь большей точности прогноза [3].

Еще одним математическим обоснованием идеи ансамблевых методов является неравенство Хёфдинга [4]. Согласно этому неравенству, при увеличении числа независимых моделей в ансамбле вероятность ошибочного прогноза уменьшается (экспоненциально для бинарной классификации), при условии, что все модели независимы, и каждая модель имеет вероятность ошибки меньше 0,5.

Однако стоит отметить, что на практике идеальная независимость недостижима: модели часто обучаются на пересекающихся данных или используют схожие алгоритмы, что ослабляет эффективность использования ансамблевых методов.

2. Основные типы ансамблей (Бэггинг, Бустинг, Стекинг)

2.1. Бэггинг (Bootstrap Aggregating). Бэггинг (Bootstrap Aggregating) [5] – метод ансамблирования, предложенный Лео Брейманом в 1996 году. Цель бэггинга – повышение

стабильности и точности моделей машинного обучения за счёт объединения прогнозов множества базовых алгоритмов, обученных на различных подвыборках исходных данных. Основная идея заключается в создании разнообразия моделей через бутстрэп-выборки (случайное извлечение объектов с возвращением), что позволяет компенсировать ошибки отдельных алгоритмов. Схема метода ансамбля Bagging представлена на рисунке 2.

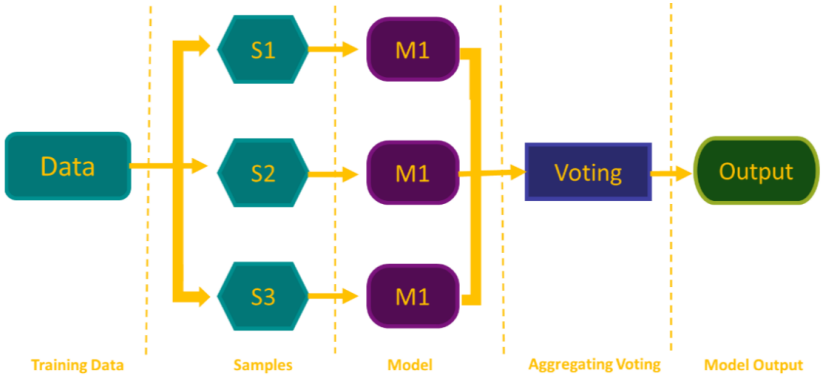

Рис. 2. Схема метода ансамбля Bagging

Основные этапы бэггинга:

1. Создание бутстрэп-выборок (Bootstrap Sampling). Из исходного обучающего набора данных случайным образом формируется несколько подвыборок с возвращением. Это означает, что каждая подвыборка имеет тот же размер, что и исходные данные (если размер исходного набора данных достаточно велик, иначе выбирается меньший размер относительной исходной выборки), а также некоторые объекты попадают в подвыборку несколько раз, а некоторые не попадают вообще.
2. Обучение базовых моделей. На каждой бутстрэп-выборке обучается отдельная модель. Чаще всего используются нестабильные модели, то есть их прогнозы сильно зависят от данных. Например, глубокие деревья решений или нейронные сети, что склонны к переобучению.
3. Агрегация прогнозов (Aggregation). Для агрегации в моделях бэггинга применяются два основных метода: голосование большинства, когда итоговый ответ выбирается как наиболее часто встречающийся класс среди прогнозов базовых моделей и усреднение, когда результатом становится среднее арифметическое предсказанных значений всех моделей.

Разновидности методов на основе бэггинга в основном отличаются друг от друга тем, как они формируют случайные подмножества обучающего набора:

- Pasting [6] – если случайные подмножества набора данных формируются, как случайные подмножества выборок.
- Random Subspaces [7] – если случайные подмножества набора данных формируются, как случайные подмножества признаков.
- Random Patches [8] – если базовые модели строятся на подмножествах как выборок, так и признаков.

Эффективность бэггинга объясняется двумя обстоятельствами:

- Компенсация ошибок: различия в обучающих подвыборках приводят к тому, что ошибки одних моделей нейтрализуются другими при агрегации.
- Работа с выбросами: выбросы могут не попадать в некоторые выборки, что повышает точность отдельных моделей.

На основании рассмотренных данных мы можем выделить основные преимущества метода бэггинг, а именно, снижение дисперсии, что актуально для моделей с высокой дисперсией (например, глубокие деревья), и параллелизм, так как обучение базовых моделей независимо, что ускоряет процесс (важно для нейросетей). Так же стоит отметить универсальность данного метода, однако, хоть он и применим к любым алгоритмам, но

максимальная эффективность достигается с сильными моделями (в отличие от бустинга, ориентированного на слабые модели).

2.2. Бустинг (Boosting). Бустинг [9] – это метод машинного обучения, основанный на идее ансамблирования, где несколько моделей, часто называемых “слабыми классификаторами”, объединяются в единую сильную модель. В отличие от других ансамблевых подходов, таких как бэггинг, в бустинге модели обучаются последовательно, а не параллельно. Каждая последующая модель фокусируется на исправлении ошибок, допущенных предыдущими моделями. Это достигается за счёт динамического перераспределения весов объектов в обучающем наборе: экземпляры, которые были неправильно классифицированы, получают больший вес, заставляя новые модели "концентрироваться" на сложных случаях. Схема метода ансамбля Boosting представлена на рисунке 3.

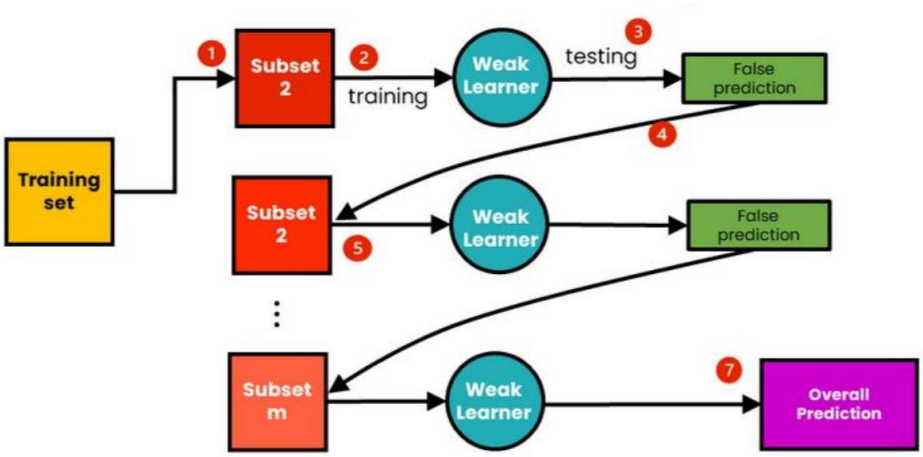

Рис. 3. Схема метода ансамбля Boosting

Стоит выделить две основные разновидности бустинга:

– Gradient Boosting [10]: Использует градиентный спуск для минимизации функции потерь. Каждая новая модель обучается на градиенте ошибки предыдущей. Популярные реализации: XGBoost, LightGBM, CatBoost.
– AdaBoost (Adaptive Boosting) [11]: Динамически изменяет веса объектов, увеличивая значимость ошибочно классифицированных экземпляров.

Главным преимуществом бустинга по сравнению с другими ансамблевыми моделями является уменьшение смещения. Бустинг эффективно снижает систематическую ошибку ансамбля, так как каждая новая модель корректирует недостатки предыдущих. Базовые алгоритмы часто выбираются с высоким смещением (например, неглубокие деревья), что позволяет компенсировать их слабость через последовательное обучение.

В тоже время важным недостатком бустинга является склонность к переобучению, особенно если ансамбль содержит слишком большое количество моделей или высокую скорость обучения. Поэтому данный метод сильно зависит от настройки гиперпараметров. Ключевые параметры, такие, как количество моделей, скорость обучения, требуют тщательного подбора. Так же стоит отметить, что обучение происходит последовательно, что исключает распараллеливание и увеличивает время работы.

2.3. Стекинг (Stacking). Стекинг [12] – это продвинутый метод ансамблирования машинного обучения, который комбинирует прогнозы нескольких разнородных моделей (базовых моделей) на уровне метамодели. Основная идея заключается в использовании предсказаний базовых алгоритмов в качестве входных данных для метамодели, которая обучается на этих данных, чтобы синтезировать итоговый прогноз с повышенной точностью.

В отличие от бэггинга и бустинга, где решения усредняются или взвешиваются, стекинг предполагает более гибкую стратегию, где метамодель “учится” оптимально объединять результаты базовых моделей. Схема метода ансамбля Stacking представлена на рисунке 4.

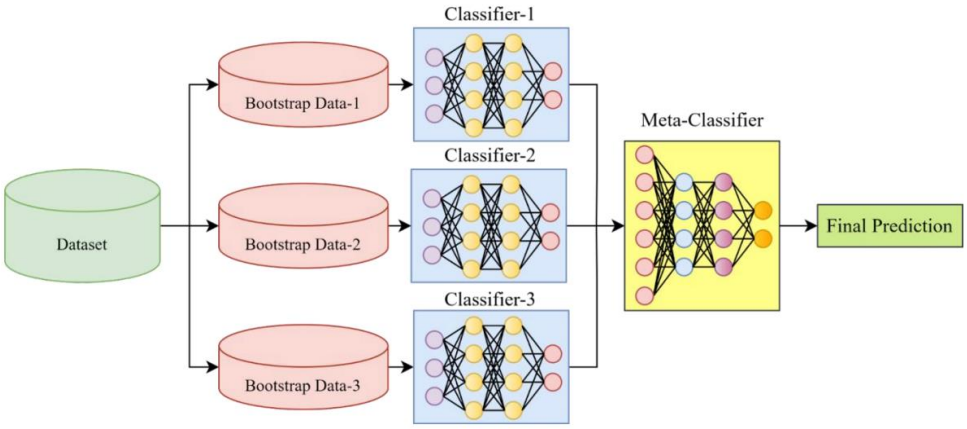

Рис. 4. Схема метода ансамбля Stacking

Среди основных преимуществ стекинга стоит выделить гибкость и разнообразие моделей, так как стекинг позволяет комбинировать модели с разными алгоритмическими подходами, что увеличивает разнообразие ансамбля. Это разнообразие помогает уловить сложные паттерны данных, которые могут быть не доступны отдельным моделям. Стекинг так же позволяет повысить точность модели за счет агрегации прогнозов через метамодель, поэтому стекинг часто превосходит по точности отдельные базовые модели и даже другие методы ансамблирования. Например, метамодель (обычно простая, как линейная или логистическая регрессия) может устранить смещения или ошибки, характерные для конкретных алгоритмов. Так же стоит отметить адаптивность метамодели. Мета-классификатор не ограничивается простым усреднением – он обучается находить оптимальные веса или нелинейные комбинации прогнозов, что особенно полезно в задачах с неочевидными взаимосвязями между предсказаниями базовых моделей.

Однако стоит выделить и основные недостатки стекинга, такие, как высокая вычислительная стоимость, ведь обучение множества моделей на нескольких уровнях требует значительных ресурсов и времени, что делает стекинг непрактичным для задач с жесткими временными ограничениями, а также риск переобучения, если метамодель слишком сложна или базовые модели имеют высокую корреляцию ошибок, ансамбль может переобучиться. Стоит отметить и сложность настройки, ведь подбор разнородных базовых моделей и метамодели требует глубокого понимания данных и тщательного экспериментирования.

3. Обзор существующих работ, связанных с исследованием ансамблевых методов машинного обучения. Ансамблевые методы машинного обучения активно исследуются в последние десятилетия, и значительное количество научных работ посвящено разработке и улучшению алгоритмов бэггинга, бустинга и стекинга, а также их комбинаций. Ниже представлен анализ некоторых из множества исследований, направленных на решение задачи повышения точности и надежности предсказаний с помощью ансамблевых подходов.

Статья “A survey of multiple classifier systems as hybrid systems”, опубликованная в журнале Information Fusion в 2014 году, написана авторами Michał Woźniak, Manuel Graña и Emilio Corchado [13]. Целью работы стало проведение систематического обзора так называемых многоуровневых классификаторов (multiple classifier systems – MCS), которые также могут рассматриваться, как гибридные системы машинного обучения. Авторы подробно рассматривают различные типы ансамблевых и гибридных моделей машинного

обучения, классифицируя их по архитектуре, способу комбинирования и используемым алгоритмам. На основе проведенного анализа авторы делают следующие важные выводы:

- Ансамблевые методы значительно превосходят отдельные модели по метрикам точности и устойчивости к переобучению.
- Разнообразие моделей в ансамбле играет ключевую роль: чем выше диверсификация ошибок, тем выше вероятность корректного коллективного решения.
- Современные гибридные подходы позволяют объединять преимущества разных алгоритмов, что особенно важно при работе со сложными или зашумленными данными.
- Вычислительная сложность остаётся одним из главных ограничений ансамблевых систем, особенно при использовании больших моделей и глубоких сетей.
- Практическая применимость ансамблевых методов продемонстрирована в таких областях, как медицина, финансовый анализ, компьютерное зрение и распознавание речи.

В работе «CatBoost: unbiased boosting with categorical features» авторов Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., Gulin, A. (NeurIPS 2018) [14] представлен алгоритм CatBoost – реализация градиентного бустинга, разработанная специально для эффективной обработки категориальных признаков без предварительного кодирования. Авторы предлагают метод, позволяющий напрямую использовать категориальные переменные в процессе обучения деревьев решений, что делает CatBoost особенно полезным в задачах, где данные содержат большое количество нечисловых факторов. Статья включает описание архитектуры CatBoost, сравнение производительности с другими популярными реализациями бустинга (XGBoost, LightGBM), а также серию экспериментов на реальных и синтетических данных. CatBoost показал лучшие результаты по точности по сравнению с XGBoost и LightGBM и меньшую склонность к переобучению. Особенно заметное преимущество наблюдалось на наборах с большим количеством категориальных признаков. На наборах без категориальных признаков CatBoost показал сравнимую или немного худшую скорость обучения, чем LightGBM, но качество предсказания оставалось конкурентоспособным.

В работе Sagi & Rokach (2018) [15] рассматриваются различные стратегии построения ансамблей, включая стекинг, и выделяются ключевые факторы, влияющие на их эффективность: разнообразие базовых моделей, корректность их прогнозов и способ комбинирования. Авторы также акцентируют внимание на том, что успех стекинга напрямую зависит от качества мета-обучения, которое может быть выполнено с использованием линейной регрессии, SVM, деревьев решений и других алгоритмов.

4. Описание эксперимента и пример практической реализации ансамблевого подхода. Был разработан ансамблевый подход, используя метод бэггинга, на основе сверточных нейронных сетей (CNN) и применен для классификации дефектов микроструктуры металла. Поскольку метод Boosting из-за последовательного обучения на ошибках, допущенных предыдущей моделью, не предоставляет возможность запуска параллельного обучения моделей, а метод Stacking требует времени и большой вычислительной мощности для обучения моделей на нескольких уровнях, то для реализации ансамбля нейросетей был выбран метод Bagging. Данный метод позволит реализовать параллельное обучение моделей и не потребует такого количества ресурсов, как Stacking.

В рамках исследования использовался специализированный набор данных, который включает изображения микроструктуры металла с пятью классами, соответствующими степени разрушения (0 – исправное состояние, 1 – разрушение). Исходные данные содержат

100 изображений размером 1600×1200 пикселей. При количестве данных для обучения в размере сотни или нескольких сотен из каждого класса, который необходимо распознать, будет велика вероятность переобучения модели. Такого количества данных может оказаться мало для такой не простой задачи, как классификация. В этом случае модель может показывать высокий процент точности на примерах из обучающих данных и низкий на этапе тестирования на новых данных.

Это сложная, но реалистичная проблема в области машинного обучения, поскольку сбор даже небольшого количества данных может оказаться чрезвычайно дорогим или трудоемким, как, например, в случае данной исследовательской работы – это невозможность автоматизации сбора данных. Специалист вручную производит срез металла и делает фотографию на оборудовании, повторяя процедуру для создания новых образцов. Для извлечения максимальной пользы из небольшого количества данных реализуется механизм дискретизации каждого такого образца размером 1600×1200 пикселей на 9 частей с последующим рядом случайных преобразований полученных «частей» исходного изображения. Подобные преобразования позволят дополнить выборку обучающих примеров так, что модель никогда не увидит дважды один и тот же образец, а также расширить набор обучающих примеров.

Для реализации этих и дальнейших возможностей использовался Keras – это высокоуровневый API, который может использовать функции библиотеки TensorFlow. Класс ImageDataGenerator() позволил создать наборы изображений с искажениями в зависимости от настроенных параметров, а также выполнять нормализацию изображений. ImageDataGenerator предоставляет различные способы искажений изображения – случайные повороты в определенном диапазоне градусов, сдвиг изображения (вправо, влево, вверх и вниз), наклоны изображения и увеличение – эти методы искажений требуют заполнения пикселями образовавшегося после перемещения изображения пустого пространства.

Из полученного датасета тестовая выборка составляет 10% от общего объёма. Классы имеют значения: 0, 0.27, 0.55, 0.83, 1. Образцы изображений каждого класса представлены на рисунке 5.

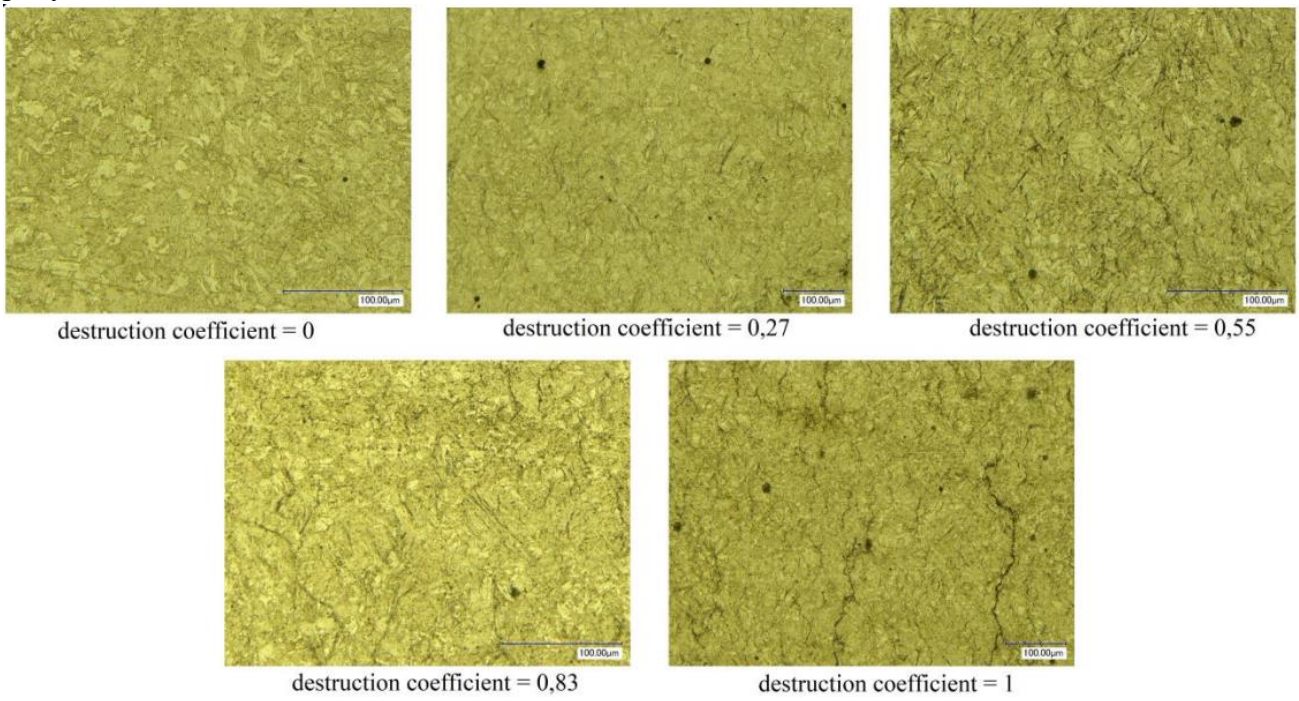

Рис. 5. Образцы изображений микроструктуры образца металла с различной степенью разрушения

В качестве базовой модели ансамбля была реализована сверточная нейронная сеть. Сверточная нейронная сеть была выбрана, как наиболее подходящий инструмент для анализа изображений микроструктур металла благодаря их способности автоматически извлекать пространственные признаки, устойчивости к вариациям данных и возможности эффективного масштабирования через ансамбли.

Структура сверточной нейронной сети представлена на рисунке 6.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 128, 128, 16)      448

 max_pooling2d (MaxPooling2D  (None, 64, 64, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 64, 64, 16)        2320

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 16)       0
 2D)

 conv2d_2 (Conv2D)           (None, 32, 32, 32)        4640

 max_pooling2d_2 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 16, 16, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 8, 8, 64)         0
 2D)

 flatten (Flatten)           (None, 4096)              0

 dense (Dense)               (None, 1024)              4195328

 dense_1 (Dense)             (None, 5)                 5125

=================================================================
Total params: 4,226,357
Trainable params: 4,226,357
Non-trainable params: 0
_________________________________________________________________
```

Рис. 6. Структура сверточной нейронной сети

Размер одного пакета представляет собой количество обучающих примеров для одной итерации обучения нейронной сети. Чем больше размер пакета, тем больше памяти потребуется для вычислений. Эмпирически было подтверждено, что использование меньших размеров партий обеспечивает более быструю сходимость к успешным решениям. Выбор размера пакета со степенью двойки помогает увеличить эффективность и производительность, т.к. пакеты будут обработаны процессорами параллельно. Таким образом, был выбран размер пакета или batch size равным 32. В качестве функции потерь была выбрана categorical_crossentropy, так как этот тип функции используется при многоклассовой классификации. Для выбора гиперпараметров модели воспользуемся HyperOpt и Hyperas. Это библиотеки Python для оптимизации гиперпараметров модели. Hyperopt реализует несколько алгоритмов для настройки параметров, которые позволяют получить наилучшие параметры для данной модели: метод на основе байесовской оптимизации Tree of Parzen Estimators (TPE), Random Search и Simulated Annealing. Варианты структуры модели представлены на рисунке 7.

```
model = Sequential()
model_choice = {{choice(['one', 'two'])}}
if model_choice == 'one':
    model.add(Conv2D(16, (3,3), padding = 'same', activation ='relu', input_shape = (128,128,3)))
    model.add(MaxPool2D((2,2)))
    model.add(Conv2D(16, (3,3), padding = 'same', activation ='relu'))
    model.add(MaxPool2D((2,2)))
    model.add(Conv2D(32, (3,3), padding = 'same', activation ='relu'))
    model.add(MaxPool2D((2,2)))
    model.add(Conv2D(64, (3,3), padding = 'same', activation ='relu'))
    model.add(MaxPool2D((2,2)))
elif model_choice == 'two':
    model.add(Conv2D(32, (3,3), padding = 'same', activation ='relu', input_shape = (128,128,3)))
    model.add(MaxPool2D((2,2)))
    model.add(Conv2D(32, (3,3), padding = 'same', activation ='relu'))
    model.add(MaxPool2D((2,2)))
    model.add(Conv2D(64, (3,3), padding = 'same', activation ='relu'))
    model.add(MaxPool2D((2,2)))
    model.add(Conv2D(128, (3,3), padding = 'same', activation ='relu'))
    model.add(MaxPool2D((2,2)))

model.add(Flatten())
model.add(Dense({{choice([256, 512, 1024])}}, activation='relu'))
model.add(Dense(5, activation = 'softmax'))

model.compile(optimizer = {{choice(['rmsprop', 'adam', 'sgd'])}},
              loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

Рис. 7. Вариации гиперпараметров архитектуры модели для оптимизации

По результатам работы метода Hyperas были получены в качестве оптимальных значений настраиваемых параметров следующие:

- модель CNN номер один;
- значение нейронов в полносвязном слое 1024;
- функция оптимизации обучения модели adam.

В качестве эксперимента было проведено сравнение работы одной сверточной нейронной сети и ансамбля, состоящего из 6 сверточных нейронных сетей, для классификации изображений для 3, 4 и 5 классов.

Результаты эксперимента приведены в таблице 1.

Таблица 1. Точность классификации тестовых объектов

Количество исследуемых классов	Точность базовой модели (сверточная нейронная сеть)	Точность ансамбля моделей
3 класса	98%	98,5%
4 класса	94%	95,5%
5 классов	88%	91%

Приведенные в таблице 1 результаты демонстрируют, что ансамблевый подход на основе бэггинга обеспечивает более высокую точность классификации по сравнению с одиночной CNN, причем значимость улучшения точности зависит от сложности задачи и становится более заметной с увеличением числа классов. При классификации 3-х классов точность ансамбля 98,5% лишь незначительно превышает точность базовой модели 98%. Улучшение точности 0,5%. Это может быть связано с тем, что при малом количестве классов задача классификации относительно проста, и даже одиночная CNN способна достичь почти максимальной точности. В таком случае ансамблирование дает лишь небольшой прирост за счет усреднения случайных ошибок. При классификации 4-х классов разница в точности увеличивается от 94% для базовой модели до 95,5% для ансамбля. Улучшение точности 1,5%. Это говорит о том, что с усложнением задачи ансамбль начинает демонстрировать преимущество за счет комбинирования предсказаний нескольких моделей, снижая влияние

переобучения отдельных CNN. При классификации 5-и классов преимущество ансамбля становится наиболее выраженным, 88% для базовой модели и 91% для ансамбля. Улучшение точности 3%. Увеличение количества классов усложняет классификацию, и одиночная модель начинает чаще ошибаться из-за недостаточной обобщающей способности. Ансамбль же компенсирует это за счет агрегации предсказаний, что снижает дисперсию ошибок.

Заключение. Современные промышленные системы, требующие точной классификации состояний технических объектов, сталкиваются с вызовами, связанными с зашумленностью данных, нелинейностью зависимостей и высокой размерностью признаков. В таких условиях ансамблевые методы машинного обучения демонстрируют превосходство над традиционными подходами.

Проведенное исследование подтвердило, что рассмотренные методы ансамблирования – бэггинг, бустинг и стекинг – могут эффективно решать задачи диагностики за счет:

- снижения дисперсии ошибок (бэггинг),
- коррекции смещения (бустинг),
- гибкой агрегации разнородных моделей (стекинг).

Практический эксперимент с использованием метода бэггинга на основе сверточных нейронных сетей показал увеличение точности классификации дефектов микроструктуры металла по сравнению с одиночной моделью. Это подтверждает, что агрегация предсказаний нескольких моделей позволяет минимизировать влияние случайных ошибок отдельных алгоритмов.

Однако применимость данного подхода в различных условиях ограниченна природой исходных данных и имеет ряд ограничений:

- вычислительная затратность – обучение нескольких параллельно работающих CNN требует больших вычислительных ресурсов и затрат оперативной памяти, что может быть критично в реальных условиях;
- зависимость от аугментации – качество генерации дополнительных данных критически влияет на результат модели;
- незначительный прирост точности для простых задач – практический эксперимент показал, что при малом числе классов преимущество ансамбля может не оправдывать вычислительных затрат.

Несмотря на ограничения, внедрение предложенного подхода способно повысить надежность и безопасность эксплуатации сложных технических систем, что подтверждает актуальность и практическую ценность проведенного исследования.

Список источников

1. Кашницкий Ю.С. История развития ансамблевых методов классификации в машинном обучении. Июнь 2015. – DOI:10.13140/RG.2.1.3933.2007.
2. Городецкий В.И. Методы и алгоритмы коллективного распознавания / В.И. Городецкий, С.В. Серебряков // Труды СПИИРАН, 2006. – №3 (1). – С. 139-171.
3. Краковский Ю.М. Исследование современных методов построения прогнозирующих ансамблей применительно к задаче интервального прогнозирования / Ю.М. Краковский, А.Н. Лузгин // Современные технологии. Системный анализ. Моделирование, 2017. – №3 (55). – С. 94-101.
4. Hoeffding W. Probability inequalities for sums of bounded random variables. Journal of the American statistical association, 1963, vol. 58, no. 301, pp. 13-30, DOI:10.2307/2282952.
5. Шитиков В.К. Классификация, регрессия, алгоритмы Data Mining с использованием R. / В.К. Шитиков, С.Э. Мастицкий, 2017. – URL: https://ranalytics.github.io/data-mining/044-Ensembles.html.
6. Breiman L. Pasting small votes for classification in large databases and on-line. Machine Learning, 1999, vol. 36, no. 1, pp. 85-103.
7. Ho T. The random subspace method for constructing decision forests. Pattern analysis and machine intelligence, 1998, vol. 20, no. 8, pp. 832-844.

8. Louppe G., Geurts P. Ensembles on random patches. Machine learning and knowledge discovery in databases, 2012, pp. 346-361.
9. Кугаевских А.В. Классические методы машинного обучения: учебное пособие / А.В. Кугаевских, Д.И. Муромцев, О.В. Кирсанова, 2022. – С. 43-46.
10. Салахутдинова К.И. Алгоритм градиентного бустинга деревьев решений в задаче идентификации программного обеспечения / К.И. Салахутдинова, И.С. Лебедев, И.Е. Кривцова // Научно-технический вестник информационных технологий, механики и оптики, 2018. – Том 18. – № 6. – С. 1016-1022.
11. Zhu J., Zou H., Rosset S., et al. Multi-class AdaBoost. Statistics and its interface, 2009, vol. 2, pp. 349-360.
12. Снопкова И.А. Исследование эффективности процедуры стекинга при решении задач классификации / И.А. Снопкова, Л.В. Липинский // Актуальные проблемы авиации и космонавтики, 2020. – Том 2. – С. 89-90.
13. Woźniak M., Graña M., Corchado E. A survey of multiple classifier systems as hybrid systems. Information fusion, 2014, vol. 18, pp. 3-15, DOI:10.1016/j.inffus.2013.04.006.
14. Prokhorenkova L., Gusev G., Vorobev A., et al. CatBoost: unbiased boosting with categorical features. Advances in Neural Information Processing Systems (NeurIPS), 2018, vol. 31, pp. 1-9, DOI:10.48550/arXiv.1706.09516.
15. Sagi O., Rokach L. Ensemble learning: a survey. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2018, vol. 8, no. 4, p. 1249, DOI:10.1002/widm.1249.

Ломакина Любовь Сергеевна. *Доктор технических наук, профессор, профессор кафедры «Вычислительные системы и технологии» НГТУ им. Р.Е. Алексеева. AuthorID: 8366, SPIN: 8087-4043, ORCID: 0000-0002-5598-6079, llomakina@list.ru.*

Двитовская Алиса Николаевна. *Аспирант кафедры «Вычислительные системы и технологии» НГТУ им. Р.Е. Алексеева. AuthorID: 1313137, SPIN: 5923-2349, ORCID: 0009-0007-3250-5520, avd.alisa@gmail.com.*

Корелин Кирилл Александрович. *Магистрант кафедры «Вычислительные системы и технологии» НГТУ им. Р.Е. Алексеева. AuthorID: 1314707, SPIN: 5952-2314, ORCID: 0009-0009-4838-1838, kirill-korelin2@mail.ru.*

UDC 004.8
DOI:10.25729/ESI.2025.40.4.003

Diagnostics and forecasting of technical and technological object states based on ensemble machine learning technologies

Lyubov S. Lomakina, Alisa N. Dvitovskaya, Kirill A. Korelin

Nizhny Novgorod state technical university named after R.E. Alekseev,
Russia, Nizhny Novgorod, *llomakina@list.ru*

Abstract. The article investigates the application of ensemble machine learning methods for diagnosing and forecasting the states of technical and technological objects under conditions of noisy data, nonlinear dependencies, and high-dimensional feature spaces. The relevance of the work is driven by the need to enhance the reliability of industrial systems through minimizing accident risks and optimizing operational processes. Traditional approaches demonstrate insufficient accuracy in complex scenarios, motivating the use of ensemble technologies that combine predictions from multiple models to achieve robust results. The primary focus is on Bagging, Boosting, and Stacking methods, their mathematical foundations, and practical implementation. An experiment was conducted using an ensemble of convolutional neural networks (CNNs) for classifying defects in metal microstructure. The results showed an increase in prediction accuracy with an increasing number of classes compared to a single model, confirming the effectiveness of ensembles in reducing error variance and correcting model bias. The proposed approach demonstrates potential for integration into industrial systems, enhancing diagnostic reliability and operational safety of complex technical systems.

Keywords: technical object diagnostics, state forecasting, ensemble methods, bagging, boosting, stacking, convolutional neural networks

References

1. Kashnitsky Y.S. Istoriya razvitiya ansamblevykh metodov klassifikatsii v mashinnom obuchenii [History of the development of ensemble classification methods in machine learning]. June 2015. DOI:10.13140/RG.2.1.3933.2007.
2. Gorodetsky V.I., Serebryakov S.V. Metody i algoritmy kollektivnogo raspoznavaniya [Methods and algorithms of collective recognition]. Trudy SPIIRAN [Proceedings of SPIIRAS], 2006, no. 3 (1), pp. 139-171.
3. Kravkovsky Y.M., Luzgin A.N. Issledovanie sovremennykh metodov postroeniya prognoziruyushchikh ansambley primenitel'no k zadache interval'nogo prognozirovaniya [Research of modern methods for constructing predictive ensembles for interval forecasting]. Sovremennye tekhnologii. Sistemnyy analiz. Modelirovanie [Modern Technologies. System Analysis. Modeling], 2017, no. 3 (55), pp. 94-101.
4. Hoeffding W. Probability inequalities for sums of bounded random variables. Journal of the American statistical association, 1963, vol. 58, no. 301, pp. 13-30, DOI:10.2307/2282952.
5. Shitikov V.K., Mastitsky S.E. Klassifikatsiya, regressiya, algoritmy Data Mining s ispol'zovaniem R [Classification, regression, Data Mining algorithms using R], 2017, available at: https://ranalytics.github.io/data-mining/044-Ensembles.html.
6. Breiman L. Pasting small votes for classification in large databases and on-line. Machine Learning, 1999, vol. 36, no. 1, pp. 85-103.
7. Ho T. The random subspace method for constructing decision forests. Pattern analysis and machine intelligence, 1998, vol. 20, no. 8, pp. 832-844.
8. Louppe G., Geurts P. Ensembles on random patches. Machine learning and knowledge discovery in databases, 2012, pp. 346-361.
9. Kugaevskikh A.V., Muromtsev D.I., Kirsanova O.V. Klassicheskie metody mashinnogo obucheniya: uchebnoe posobie [Classical machine learning methods: textbook], 2022, pp. 43-46.
10. Salakhutdinova K.I., Lebedev I.S., Krivtsova I.E. Algoritm gradientnogo bustinga derev'ev resheniy v zadache identifikatsii programmnogo obespecheniya [Gradient boosting algorithm of decision trees in the problem of software identification]. Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki [Scientific and technical journal of information technologies, mechanics and optics], 2018, vol. 18, no. 6, pp. 1016-1022.
11. Zhu J., Zou H., Rosset S., et al. Multi-class AdaBoost. Statistics and its interface, 2009, vol. 2, pp. 349-360.
12. Snopkova I.A., Lipinsky L.V. Issledovanie effektivnosti protsedury stekinga pri reshenii zadach klassifikatsii [Study of the effectiveness of the stacking procedure in solving classification problems]. Aktual'nye problemy aviatsii i kosmonavtiki [Actual problems of aviation and cosmonautics], 2020, vol. 2, pp. 89-90.
13. Woźniak M., Graña M., Corchado E. A survey of multiple classifier systems as hybrid systems. Information fusion, 2014, vol. 18, pp. 3-15, DOI:10.1016/j.inffus.2013.04.006.
14. Prokhorenkova L., Gusev G., Vorobev A., et al. CatBoost: unbiased boosting with categorical features. Advances in Neural Information Processing Systems (NeurIPS), 2018, vol. 31, pp. 1-9, DOI:10.48550/arXiv.1706.09516.
15. Sagi O., Rokach L. Ensemble learning: a survey. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2018, vol. 8, no. 4, p. 1249, DOI:10.1002/widm.1249.

Lomakina Lyubov Sergeevna. *Doctor of Technical Science, Professor, Professor of the Department of Computing Systems and Technologies at the NNSTU named after R.E. Alekseev. AuthorID: 8366, SPIN: 8087-4043, ORCID: 0000-0002-5598-6079, llomakina@list.ru.*

Dvitovskaya Alisa Nikolaevna. *Postgraduate student of "Computational Systems and Technologies," NSTU named after R.E. Alekseev. AuthorID: 1313137, SPIN: 5923-2349, ORCID: 0009-0007-3250-5520, avd.alisa@gmail.com*

Korelin Kirill Aleksandrovich. *Master's Student, Department of "Computational Systems and Technologies," NSTU named after R.E. Alekseev. AuthorID: 1314707, SPIN: 5952-2314, ORCID: 0009-0009-4838-1838, kirill-korelin2@mail.ru*

Статья поступила в редакцию 17.04.2025; одобрена после рецензирования 27.05.2025; принята к публикации 16.10.2025.

The article was submitted 04/17/2025; approved after reviewing 05/27/2025; accepted for publication 10/16/2025.