Выбор способа фильтрации диагностических данных в системах непрерывного мониторинга объектов транспортной инфраструктуры

Д. В. Ефанов,
д-р техн. наук, доцент, профессор кафедры «Автоматика, телемеханика и связь на железнодорожном транспорте» Российского университета транспорта (РУТ (МИИТ))

В. Н. Мячин,
д -р техн. наук, профессор, генеральный директор ООО НТЦ «Комплексные системы мониторинга»

Г. В. Осадчий,
технический директор ООО НТЦ «Комплексные системы мониторинга»

М. В. Зуева,
инженер-проектировщик ООО НТЦ «Комплексные системы мониторинга»

В статье приведены результаты экспериментов по выбору способа фильтрации первичных диагностических данных, который должен повысить достоверность ретроспективного анализа состояния объектов транспортной инфраструктуры в системах непрерывного мониторинга. Определен наиболее эффективный способ реализации процедуры фильтрации в этом случае.

Развитие науки и техники в первой четверти XXI в. позволило перейти к реализации технически сложных объектов транспортной инфраструктуры во всем мире. В крупнейших промышленных центрах мира возводятся различные по сложности искусственные сооружения, пролегающие в регионах с различным ландшафтом, пути сообщения строятся на различных уровнях, отсчитывая от поверхности земли, реализуются крупные хабы, которые объединяют разнородные транспортные системы, и т. д.

Важнейший компонент в процессе эксплуатации подвижных единиц и объектов транспортной инфраструктуры — техническое обслуживание и их ремонт. Использование современных технических средств автоматизации позволяет оснащать технически сложные объекты специализированными датчиками получения значений различных физических величин, объединять их в сети, централизовывать данные и строить развитые системы непрерывного мониторинга ответственных параметров диагностируемых объектов **[1]**. Наличие системы непрерывного мониторинга помогает получать объективные данные о техническом состоянии объектов, обрабатывать их и на стадиях зарождения выявлять потенциально опасные дефекты. Это касается объектов как с медленнотекущими процессами развития дефектов, так и с быстротекущими. Техническими средствами мониторинга в настоящее время охватывается большое число объектов транспортной инфраструктуры **[2–7]**, а сами диагностические данные весьма разнообразны: от набора электрических и механических параметров до видеоряда с объектами мониторинга.

Вне зависимости от конкретного приложения, будь то мониторинг параметров системы управления движением поездов в метрополитенах или на магистральных железных дорогах, либо же мониторинг мостовых конструкций, системы мониторинга обладают похожими структурами, а также имеют общие положительные и отрицательные качества. Обратим внимание читателя на проблемы обработки диагностической информации.

Современные системы мониторинга позволяют получать громадные объемы диагностических данных даже по одному датчику, не говоря уже о целой системе датчиков. Например, при анализе динамических параметров объектов мониторинга требуется частота опроса датчиков до 50 Гц, что за час по одному датчику дает до 180 тыс. измерений (4,32 млн измерений в сутки). Общие объемы поступающих данных достигают десятков гигабайтов. Во многих системах мониторинга результаты измерений накапливаются во временных хранилищах, откуда впоследствии извлекаются и заносятся в общую базу данных, в результате чего возникают проблемы синхронизации диагностической информа-

Рис. 1. Объект мониторинга

ции. Даже при решении этой проблемы неизбежно требуется анализ качества диагностической информации. В диагностических данных нередки следующие виды ошибок: остановка передачи данных с измерительного устройства, нетипичные выбросы данных, отказы измерительного оборудования, накопление ошибок от помех, разрывы данных и т. д. В итоге диагностические данные оказываются «грязными» и неполными, а достоверность результатов мониторинга, как следствие, невысокой.

Еще одна проблема мониторинга — необходимость обработки данных по сравнению с пороговыми значениями (нормалями, уставками и т. п.), установленными в нормативно-технической и справочной документации, что само по себе исключает качественный анализ диагностической информации. Для решения многих задач прогнозирования требуются достаточно продолжительные временны́е ряды данных, достигающие порой десятков лет, что помогает значительно улучшить результаты мониторинга. Хранилища, позволяющие накапливать данные на такой продолжительный срок, должны обладать высокой надежностью.

Из вышеприведенного описания непосредственно следует необходимость первоначальной синхронизации и фильтрации диагностической информации для ее качественного последующего анализа с помощью средств искусственного интеллекта **[8–13]**. Рассмотрим задачу фильтрации информации на примере обработки массивов диагностических данных от технических средств мониторинга известного искусственного сооружения — моста на остров Русский во Владивостоке. Следует отдельно отметить, что в области систем диагностирования и мониторинга во всем мире упомянутая задача весьма актуальна, а готовых решений для нее, несмотря на широкое распространение методов мониторинга, до сих пор не имеется.

Краткая характеристика объекта мониторинга

Объект мониторинга представляет собой самый длинный в мире вантовый мост, построенный во Владивостоке к открытию саммита АТЭС в 2012 г. (*рис. 1*). Мост пролегает через пролив Босфор Восточный, соединяет полуостров Назимова с мысом Новосильского на острове Русском и имеет самый большой в мире пролет среди вантовых мостов — длиной 1104 м. Высота моста составляет 324 м, что является второй величиной в мире. Мост на остров Русский как национальное достояние изображен на банкноте номиналом 2 тыс. руб.

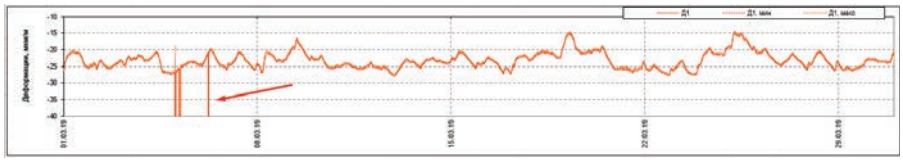

Рис. 2. «Сырые» данные с аномалиями

В **[14]** описаны особенности контроля геометрии рассматриваемого моста на стадии его сооружения. Для мониторинга технического состояния моста в процессе эксплуатации развернута структурированная система, включающая в себя 50 датчиков деформации, 88 датчиков деформации и температуры, 16 датчиков температуры, 21 акселерометр, 12 двухкоординатных инклинометров, 2 датчика перемещения и 6 датчиков давления. Для сбора диагностической информации использовано 295 измерительных каналов.

Данные от измерительных датчиков хранятся на сервере в системе управления базами данных InterBase XE. Диагностические данные в базе представлены в таблицах и записаны с частотой одно измерение в пять секунд и одно измерение в десять минут. Диагностические данные поступают непрерывно следующими друг за другом блоками, содержащими последовательности заданной продолжительности отсчетов группы каналов. Разбиение непрерывного потока данных на блоки (или объединение мгновенных отсчетов в блоки) позволяет организовать цикл обработки данных, осуществляемый параллельно их сбору — с небольшой задержкой на выполнение алгоритмов обработки. В то же время непрерывное следование блоков данных друг за другом обеспечивает сохранение всей измерительной информации без потерь.

Основная проблема получаемых массивов данных состоит в том, что в них присутствуют выбросы, причины которых установить не удается. Наличие таких выбросов мешает анализу диагностической информации. Кроме того, в массивах данных часто отсутствуют данные за определенный период ввиду отключения системы мониторинга по различным причинам (запланированные отключения, «падение» базы данных и т. п.). Именно поэтому возникает задача фильтрации диагностической информации с удалением выбросов и аномальных значений. Получение же «чистых» (отфильтрованных) данных позволяет производить дальнейший качественный анализ диагностической информации.

Эксперименты по фильтрации диагностических данных

Чтобы получать диагностические данные, пригодные к обработке с при-

менением искусственного интеллекта, требуется найти алгоритм или комбинацию алгоритмов для поиска и очистки выбросов (аномалий) в «сырых» данных. Например, на *рис. 2* представлены данные, в которых аномалии имеются (указаны красной стрелкой).

В поиске подходящего метода фильтрации «сырых» диагностических данных применялись различные методы: статистические, машинного обучения, обработки временны́х рядов. Рассмотрим особенности их использования на примере фильтрации реальных диагностических данных с объекта мониторинга во Владивостоке.

Среди статистических методов были опробованы два: «трех сигм» и «99 перцентиль» **[15]**. К сожалению, статистические методы показали себя хорошо не на всех диагностических данных. Например, на *рис. 3* продемонстрированы результаты фильтрации данных от датчика температуры RBG4DP (данные получены за март 2019 г.). Из верхнего графика на *рис. 3* следует, что графики имеют большое количество аномалий. Метод «трех сигм» обеспечивает некоторые улучшения (см. средний график), однако работает не столь «чисто», оставляя «хвосты». Метод «99 перцентиль» для представленных данных сработал существенно лучше, однако при этом привел к удалению лишних данных для экстремумов. Для других массивов диагностических данных наблюдались и противоположные результаты: где-то лучше оказывался первый метод, где-то — второй. Но общим является то, что их прямое использование не приводит к качественному решению задачи фильтрации диагностических данных.

Статистические методы работают хорошо на данных, которые имеют немногочисленные аномалии и подходящее (в лучшем случае, близкое к нормальному) распределение. Но не для всех массивов диагностических данных представленные методы обеспечивают качественный результат. Другими словами, статистические методы не универсальны, поскольку нет ни одного метода, каковой можно было бы эффективно использовать для всех массивов диагностических данных.

Методы машинного обучения показали себя на некоторых данных лучше статистических методов, а на некоторых — хуже. Рассмотрим, например, применение одноклассовой модели

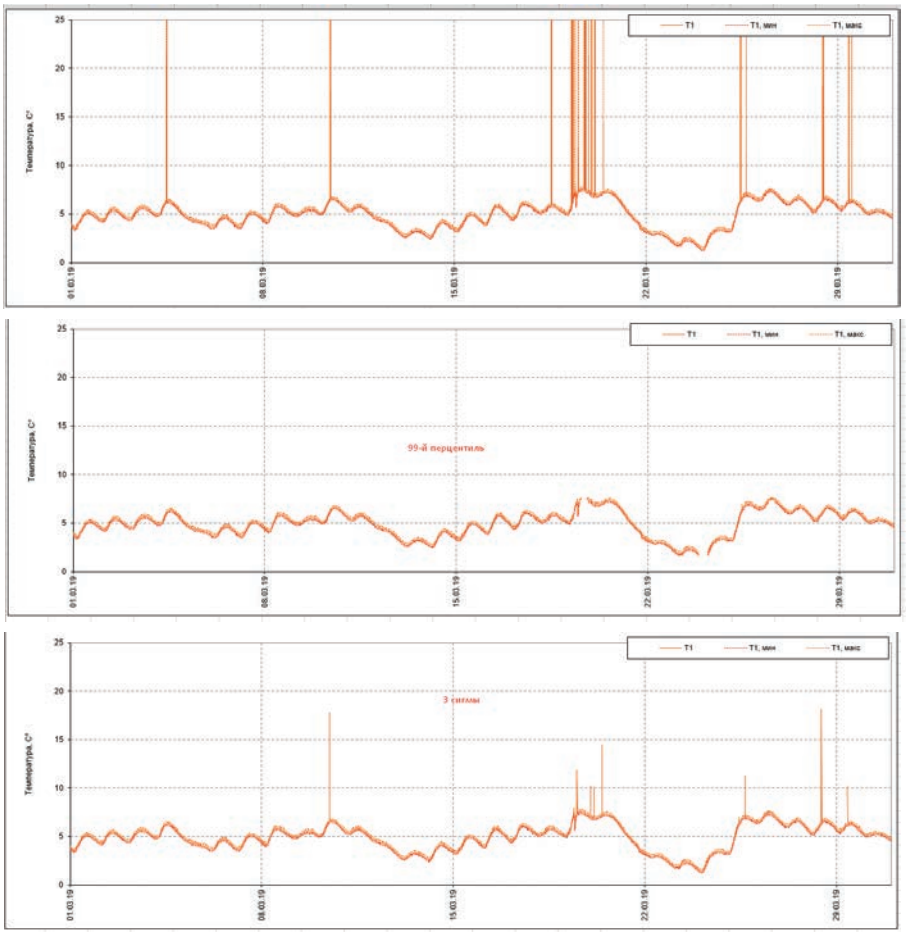

Рис. 3. Результаты использования фильтрации статистическими методами: верхний график — «сырые» данные с датчика; средний график — данные, отфильтрованные методом «трех сигм»; нижний график — данные, отфильтрованные методом «99 перцентиль»

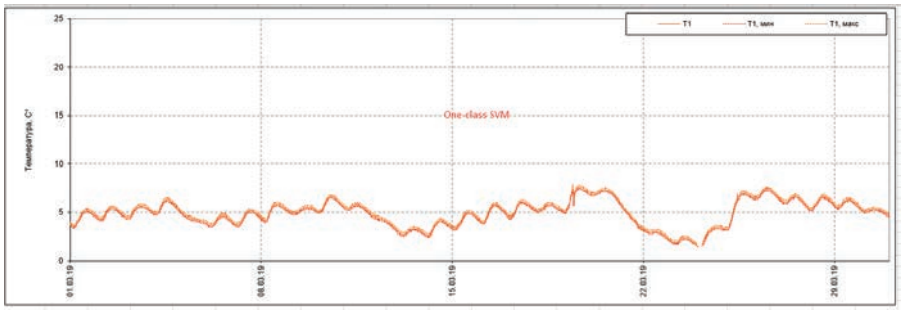

Рис. 4. Результаты использования фильтрации методом One-class SVM со стандартными параметрами библиотеки Python для данных с датчика RBG4DP

машины опорных векторов (One-class SVM) **[16]** для данных с того же датчика температуры RBG4DP (см. верхний график на *рис. 3*).

Метод One-class SVM, при использовании которого применялись стандартные параметры библиотеки Python — Scikit-learn (sklearn.svm.OneClassSVM) **[17]**, показал себя лучше на рассматриваемых данных (см. *рис. 4*).

Отметим, что при реализации sklearn.svm.OneClassSVM важны следующие параметры:

• kernel — ядро (линейное — linear; полиномиальное — poly; радиальные базисные функции — rbf; сигмоидальное — sigmoid; свое заданное);

• nu — верхняя граница на процент ошибок и нижняя на процент опорных векторов (0,5 по умолчанию);

• degree — степень для полиномиального ядра;

• gamma — коэффициент для функции ядра (1/n_features по умолчанию);

• coef0 — параметр в функции полиномиального или сигмоидального ядра (взято из **[18]**).

Применение sklearn.svm.OneClassSVM с указанными параметрами для ряда графиков приводило к тому, что некоторые

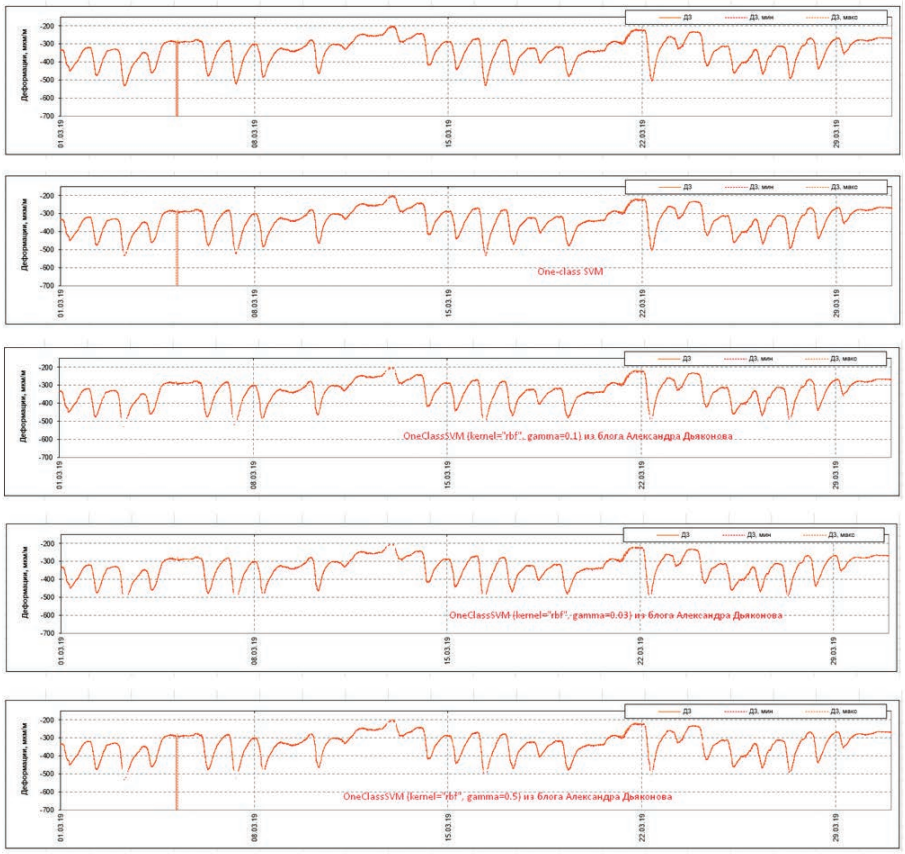

Рис. 5. Результаты использования фильтрации машинными методами: верхний график — «сырые» данные с датчика; второй сверху график — данные, отфильтрованные методом One-class SVM со стандартными параметрами библиотеки Python; третий-пятый графики сверху — применение параметров из блога А. Дьяконова с различными значениями gamma (0,1, 0,3 и 0,5 соответственно)

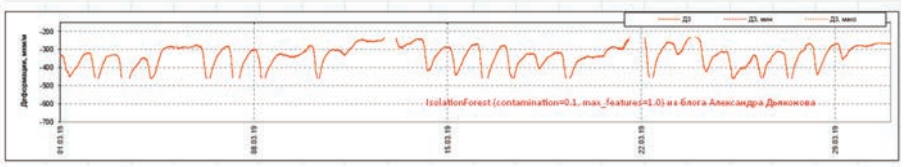

Рис. 6. Результаты использования фильтрации методом IsolationForest

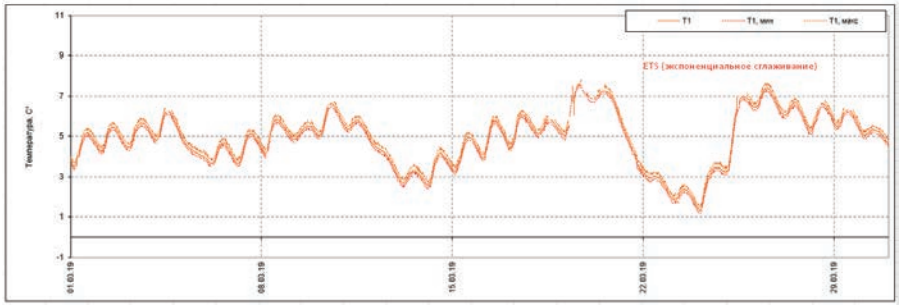

Рис. 7. Результаты использования фильтрации данных с датчика RBG4DP методом ETS

однократные выбросы оставались незамеченными. Например, упомянутый результат получен при обработке данных с датчика деформации M6L2DP, установленного на левой стойке пилона М6 на высоте 67,9 м (см. *рис. 5*). Поэтому было решено подобрать параметры так, чтобы улучшить результаты. Для выполнения указанной задачи были использованы материалы блога А. Дьяконова **[18]**. Результаты применения метода с различными параметрами для данных с датчика M6L2DP приведены на *рис. 5* (нижние три графика). Программный код для данных с датчика M6L2DP явно сработал лучше, чем код по предыдущему методу, причем с параметром gamma = 0,1 фильтр удалил меньше «нормальных» значений, чем с параметром gamma = 0,3. Увеличение значения параметра gamma приводит к ухудшению результатов фильтрации (см., например, нижний график на *рис. 5* при gamma = 0,5). Очевидно, что упомянутый метод при gamma = 0,1 продемонстрировал наилучший результат. Притом этот результат оказался даже хуже, чем после применения метода «трех сигм» для представленных «сырых» данных. В ходе эксперимента были использованы и другие методы машинного обучения.

Например, метод IsolationForest из **[18]** (одна из вариаций метода случайного леса **[19]**). Описание IsolationForest дано в **[20]**. Метод подразумевает построение деревьев случайного леса до исчерпания выборки данных. При построении ветвления в дереве выбираются случайный признак и случайное расщепление (порог разбиения). Для каждого измеренного значения определяется мера его нормальности — среднее арифметическое глубин листьев деревьев, в которые оно попало (изолировалось).

Важные параметры реализации sklearn.ensemble.IsolationForest:

- n_estimatorsint — количество базовых оценок в ансамбле;
- max_samplesint — число случайных выборок, на которое будет производиться разделение;
- contamination — доля выбросов в наборе данных;
- max_featuresint — число признаков, по которым ищется разбиение (в нашем случае признак один).

Был использован метод IsolationForest с параметрами contamination = 0,1 и max_features = 1,0 из **[18]**.

Метод IsolationForest «снимает» много лишнего, что означает много удалений полезных данных. Изменения параметров программной модели не принесли значительного результата.

Среди методов обработки временны́х рядов лучше всего показали себя модели ETS (экспоненциальное сглаживание) и ARIMA. Основные идеи были взяты из **[21, 22]**.

Для фильтрации данных с датчика RBG4DP (верхний график на *рис. 3*) неплохо показала себя модель ETS (*рис. 7*). В основе этой модели лежит экспоненциальное сглаживание (метод прогнозирования, при котором значения переменной за все предыдущие периоды входят в прогноз, экспоненциально теряя свой вес со временем **[22]**). Модель реализована с помощью функции ets() из библиотеки fpp2 на языке R.

Для данных с датчика деформации M6L2DP лучше показала себя модель ARIMA (*рис. 8*).

Модели ARIMA — это интегрированные модели авторегрессии (скользящего среднего). Модели ARIMA, являющиеся расширением модели ARMA, применяют ее не сразу к заданным временны́м рядам, а после ее предварительного дифференцирования, которое представляет собой временной ряд, полученный путем вычисления разницы между последовательными значениями исходного временно́го ряда.

Поясним процесс фильтрации с помощью моделей ETS и ARIMA. На *рис. 9* представлена последовательность поиска и удаления аномалий с использованием данных моделей (верхние графики соответствуют фильтрации одного из рядов данных с датчика RBG4DP с использованием модели ETS, нижние — фильтрации данных с датчика M6L2DP с использованием модели ARIMA. Слева представлены исходные графики с аномалиями, в средней части — графики с выделенными аномалиями, а в правой части — «очищенные» графики. Отметим, что при использовании модели ETS к данным со второго датчика один из выбросов остается.

Модели для временны́х рядов показали себя неплохо в решении задачи фильтрации «сырых» диагностических данных. Однако при использовании подобных моделей требуется встроить автоматическую проверку на то, какую из них необходимо применить для определенного ряда, а также реализовать проверку того, нужно ли вообще искать выбросы в указанном ряду, чтобы не применять к нему лишних функций (это необходимо потому, что в данных всегда удаляется много полезной информации). Для решения такой задачи может быть использована проверка гипотезы о наличии аномальных наблюдений, например методом Ирвина **[23]**. Однако метод Ирвина имеет ряд ограничений, которые не позволяют напрямую очищать диагностические данные, а потому нуждается в доработке.

Из приведенных примеров и из большого числа экспериментов с «сырыми» данными становится ясным, что единой функции для всех массивов диагностической информации нет (либо нам не удалось ее найти). Следующим шагом стала проверка готовых библиотек для выявления аномалий в данных на R и Python.

Из всех попробованных библиотек для выявления аномалий данных на языках R и Python лучше всего показала себя

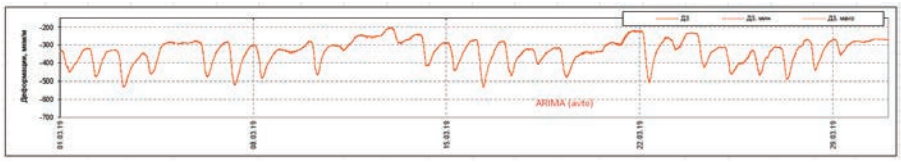

Рис. 8. Результаты использования фильтрации данных с датчика M6L2DP методом ARIMA

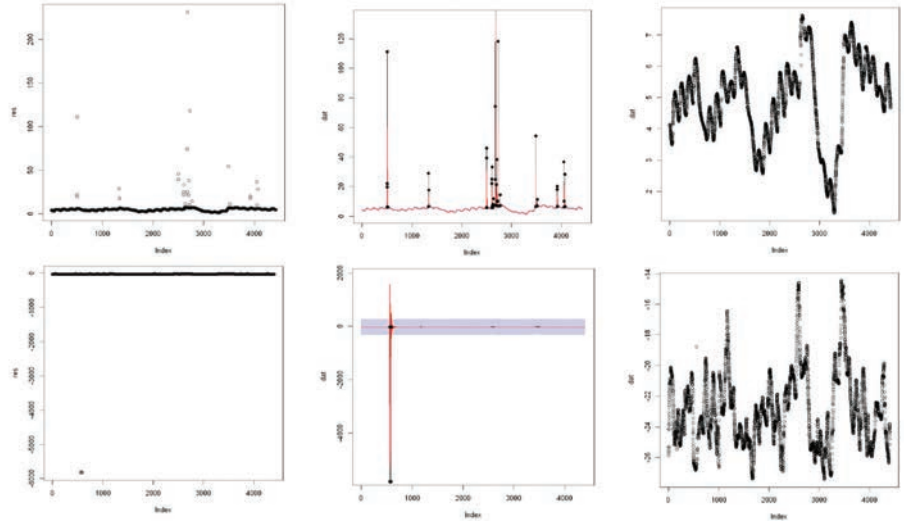

Рис. 9. Этапы фильтрации диагностических данных с датчиков RBG4DP (ETS, графики сверху) и M6L2DP (ARIMA, графики снизу): слева направо исходный график, график с выделенными аномалиями, «очищенный» график

библиотека Forecast на языке R, в частности функция tsoutliers() из указанной библиотеки (именно она сейчас используется нами для удаления выбросов). На *рис. 10* и *11* приведены результаты использования этой функции при фильтрации данных от рассмотренных выше датчиков (сравните график на *рис. 10* с верхним графиком на *рис. 3* и график на *рис. 11* с верхним графиком на *рис. 5*).

Из экспериментов с различными подходами к фильтрации диагностических данных следует, что наиболее универсален последний из рассмотренных методов, он также дает приемлемый результат фильтрации аномалий. Если статистическими методами удавалось удачно отфильтровать «сырые» данные в 50 % случаев (более 500 графиков «сырых» диагностических данных), методами машинного обучения — примерно в 60 %, методами обработки временны́х рядов — примерно в 70 %, то фильтрация, имеющаяся в библиотеке на R, срабатывала примерно в 98 % случаев (в остальных были незначительные огрехи).

Необходимо отметить, что для решения поставленной задачи могут использоваться автоэнкодеры (авто-

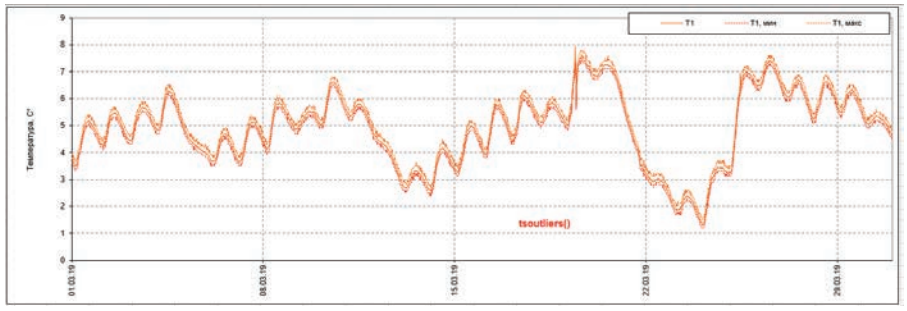

Рис. 10. Результаты использования функции tsoutliers() для датчика RBG4DP

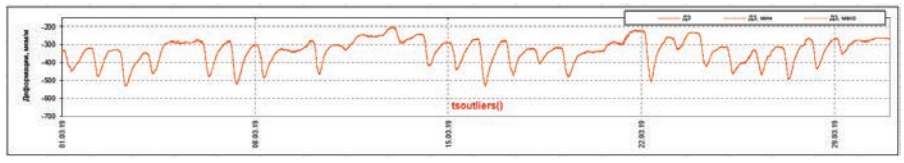

Рис. 11. Результаты использования функции tsoutliers() для датчика M6L2DP

кодировщики) — специальные искусственные нейронные сети, позволяющие применять обучение без учителя при использовании метода обратного распространения ошибки **[24]**. Эти методы, однако, учитывать нужно в последнюю очередь, поскольку они требуют трудоемкой настройки под каждый конкретный случай либо автоматизации подбора параметров для каждого датчика.

Добавим также, что в настоящей статье речь идет об обработке уже имеющихся диагностических данных (ретроспективная обработка, или постобработка), что свойственно системам с большим количеством часто опрашиваемых датчиков и, соответственно, дающим большие объемы информации. Для обработки данных от устройств в режиме реального времени должны применяться другие подходы, например изначальная фильтрация входящих данных путем отсеивания заведомо некорректных измерений (решение этой задачи требует особенной организации архитектуры обработки данных, позволяющей не удалять полезные данные до их поступления в хранилища диагностической информации).

Современные системы мониторинга, по сути, являются «системами ради системы», накапливают большое количество «сырых» диагностических данных для последующего анализа экспертом-оператором и какого-либо простейшего автоматизированного анализа путем сравнения с эталонными значениями. Сегодня такие системы — скорее хранилища внушительных объемов информации. К сожалению, указанные особенности присущи подавляющему большинству систем мониторинга во многих отраслях науки и техники, включая транспорт. Однако технологии непрерывно совершенствуются: как элементная база компонентов систем мониторинга, так и методы обработки диагностической информации. Фильтрация «сырых» диагностических данных — первый шаг к получению «чистых» данных для последующей обработки с применением искусственного интеллекта и к выходу на достоверное решение ключевых, по нашему мнению, задач мониторинга, выражаемых цепочкой терминов «диагноз – прогноз – остаточный ресурс». Т

Литература

1. Ефанов, Д. В. Функциональный контроль и мониторинг устройств железнодорожной автоматики и телемеханики. — Санкт-Петербург : ПГУПС, 2016.
2. Belyi, A. A., Karapetov, E. S., Efimenko Yu. I. Structural Health and Geotechnical Monitoring During Transport Objects Construction and Maintenance (Saint-Petersburg Example) // Procedia Engineering. — 2017. — Vol. 189. — P. 145–151.
3. Smarsly, K., Hartmann, D. Autonomous Monitoring of Masonry Dams Based on Multi-agent Technology // 4th Congress on Dams. Struga, 2017. P. 1–10.
4. Belyi, A., Osadchy, G., Dolinskiy, K. Practical Recommendations for Controlling of Angular Displacements of High-Rise and Large Span Elements of Civil Structures // Proceedings of 16th IEEE East-West Design & Test Symposium (EWDTS'2018). Kazan, 2018. P. 176–183.
5. Belyi, A., Shestovitskii, D., Myachin, V., Sedykh, D. Development of Automation Systems at Transport Objects of MegaCity // Proceedings of 17th IEEE East-West Design & Test Symposium (EWDTS'2019). Batumi, 2019. P. 201–206.
6. Belyi, A., Shestovitskii, D., Karapetov, E., Sedykh, D., Linkov, V. Main Solutions of Structural Health Monitoring in Managing the Technical Condition of Transport Objects // Ibid. P. 213–218.
7. Efanov, D. V., Osadchy, G. V., Barch, D. V., Belyi A. Permanent Monitoring Systems of the Contact-Wire of Railroad Catenary: The Main Tasks of Implementation // Ibid. P. 484–487.
8. Smarsly, K., Lehner, K., Hartmann, D. Structural Health Monitoring Based on Artificial Intelligence Techniques // Computing in Civil Engineering. Pittsburgh, 2007. P. 111–118.
9. Böhm, T. Remaining Useful Life Prediction for Railway Switch Engines Using Artificial Neural Networks and Support Vector Machines // International Journal of Prognostics and Health Management. — 2017. — Vol. 8 (Special Issue on Railways & Mass Transportation). — P. 1–15.
10. Heidmann, L. Smart Point Machines: Paving the Way for Predictive Maintenance // Signal+Draht. — 2018. — Bd. 110. — Heft 9. — S. 70–75.
11. Neumann, T., Guzmán, D. N., Groos, J. C. Transparent Failure Diagnostics for Railway Switches Using Bayesian Networks // Ibid. — 2019. — Bd. 111. — Heft 12. — S. 23–31.
12. Busse, R. Performance Monitoring for Level Crossing Protection Systems // Ibid. — 2020. — Bd. 112. — Heft 1+2. — S. 46–50.
13. Luckey, D., Fritz, H., Legatiuk, D., Dragos, K., Smarsly, K. Artificial Intelligence Techniques for Smart City Applications // Proceedings of the International ICCCBE and CIB W78 Joint Conference on Computing in Civil and Building Engineering 2020. São Paolo, 2020. P. 1–14.
14. Курепин, В. М. Комплексный метод контроля геометрии вантовых мостов / В. М. Курепин, С. В. Задворнов, Р. С. Кузнецов [и др.] // Дороги. — 2011. — № 10. — С. 32–35.
15. Козлов, М. В. Введение в математическую статистику / М. В. Козлов, А. В. Прохоров. — Москва : Издательство МГУ, 1987.
16. Cristianini, N., Shawe-Taylor, J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge : Cambridge University Press, 2000.
17. Sklearn.svm.OneClassSVM. — URL: scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html (дата обращения: 26.04.2020).
18. Поиск аномалий (Anomaly Detection). — URL: dyakonov.org/2017/04/19/%D0%BF%D0%BE%D0%B8%D1%81%D0%BA-%D0%B0%D0%BD%D0%BE%D0%BC%D0%B0%D0%BB%D0%B8%D0%B9-anomaly-detection (дата обращения: 26.04.2020).
19. Breiman, L. Random Forests // Machine Learning. — 2001. — Vol. 45. — Iss. 1. — P. 5–32.
20. Sklearn.svm.IsolationForest. — URL: scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html (дата обращения: 26.04.2020).
21. Соболев, К. В. Автоматический поиск аномалий во временны́х рядах : магистерская диссертация. — Москва : МФТИ, 2018.
22. Forecasting: Principles & Practice. — URL: robjhyndman.com/uwafiles/fpp-notes.pdf (дата обращения: 26.04.2020).
23. Трофименко, С. В. Модификация метода выявления аномальных уровней временных рядов : методика и численные эксперименты / С. В. Трофименко, А. Я. Маршалов, Н. Н. Гриб [и др.] // Современные проблемы науки и образования : [сайт]. — URL: science-education.ru/ru/article/view?id=15130 (дата обращения: 26.04.2020).
24. Autoencoders. — URL: ufldl.stanford.edu/tutorial/unsupervised/Autoencoders (дата обращения: 26.04.2020).