

МЕТОДЫ И СРЕДСТВА ОБНАРУЖЕНИЯ НЕШТАТНЫХ СИТУАЦИЙ, ВОЗНИКАЮЩИХ НА ЭЛЕМЕНТАХ ЖИЛИЩНО-КОММУНАЛЬНОЙ ИНФРАСТРУКТУРЫ*

И. А. Шанин¹, С. А. Ступников², В. Н. Захаров³

Аннотация: Работа относится к области разработки специализированных информационных систем на основе технологии Интернета вещей. Рассматривается подход к программной реализации компонентов модуля обнаружения нештатных ситуаций как составной части информационной системы, обеспечивающей поддержку технического обслуживания элементов жилищной инфраструктуры в целях оперативного контроля их состояния, предиктивного ремонта и оповещения о возникающих нештатных ситуациях и регулярных событиях. Описаны алгоритмы и особенности реализации компонентов, осуществляющих построение моделей функционирования элементов жилищной инфраструктуры и обнаружение нештатных ситуаций. Приводятся экспериментальные результаты применения подхода для обнаружения нештатных ситуаций на модельных наборах данных.

Ключевые слова: Интернет вещей; анализ данных; обнаружение нештатных ситуаций; жилищно-коммунальная инфраструктура

DOI: 10.14357/19922264180310

1 Введение

Активное развитие технологии Интернета вещей, предполагающей, в частности, оснащение физических предметов средствами взаимодействия друг с другом или с внешней средой, открывает перспективы появления и совершенствования новых классов информационных систем для решения насущных задач различного рода. Например, в области жилищно-коммунальной инфраструктуры оснащение элементов инфраструктуры (таких как трансформаторы, насосы, кондиционеры и др.) разнообразными датчиками — температуры, тока, напряжения, звука — позволяет собирать с них данные и осуществлять их анализ с целью поддержки технического обслуживания.

Данная работа проводится в рамках проекта, нацеленного на создание комплекса программно-технических решений по созданию информационной системы, обеспечивающей поддержку технического обслуживания элементов жилищной инфраструктуры в целях оперативного контроля их состояния, предиктивного ремонта и оповещения о возникающих нештатных ситуациях и ре-

гулярных событиях. К нештатным ситуациям относятся, в частности, отказ элементов жилищной инфраструктуры в эксплуатационный период, заявленный производителем; к регулярным событиям — техническое обслуживание и замена элементов по истечении эксплуатационного периода. В рамках проекта рассматриваются такие элементы жилищной инфраструктуры, как трансформаторы, насосы, вентиляционные установки и кондиционеры, газовые котлы, трубчатые электронагреватели (ТЭНы), электрические лампы.

В работе [1] авторами была предложена архитектура информационной системы, включающая модули сбора данных, хранения данных, обнаружения и предсказания нештатных ситуаций, информирования пользователя о произошедших и потенциально возможных нештатных ситуациях. Одной из основных составляющих архитектуры является модуль обнаружения и предсказания нештатных ситуаций, включающий, в частности, компоненты построения моделей функционирования элементов жилищной инфраструктуры и обнаружения нештатных ситуаций [1]. В данной работе рассматривается подход к программной реализации упо-

*Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации (уникальный идентификатор проекта RFMEFI60717X0176).

¹Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, v08shinin@gmail.com

²Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, sstupnikov@ipiran.ru

³Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, vazakharov@ipiran.ru

мянутых компонентов модуля. Описаны алгоритмы функционирования и особенности реализации компонентов модуля (разд. 2). Приводятся экспериментальные результаты применения подхода для обнаружения нештатных ситуаций на модельных наборах данных (разд. 3).

2 Алгоритмы функционирования компонентов модуля обнаружения нештатных ситуаций

2.1 Построение моделей функционирования элементов жилищной инфраструктуры

Компонент реализует следующие подходы к анализу временных рядов: сезонная интегрированная модель авторегрессии — скользящего среднего (SARIMA — seasonal autoregressive integrated moving average) и скрытые марковские модели (HMM — hidden Markov models).

Модель SARIMA. Являясь обобщением классической модели авторегрессии — скользящего среднего (ARMA — autoregressive moving average) [2], данная модель находит массу применений в анализе временных рядов, в частности в задаче прогнозирования и интерполяции значений. Модель использует понятие тренд-сезонного разложения временного ряда [3], представляя наблюдаемые значения в виде суммы (или произведения) трех компонент: *тренда*, *сезонной составляющей* и *остатков*. Трендовая составляющая отражает долгосрочное изменение показателей, сезонная составляющая отражает периодические колебания временного ряда. Остатки не имеют ни сезонных, ни трендовых особенностей. Для обучения модели необходимо оценить гиперпараметры p, d, q, P, D, Q и s :

- целочисленные параметры p и q соответствуют порядкам моделей авторегрессии (AR — autoregression) и скользящего среднего (MA — moving average), их значение отображает число элементов временного ряда, участвующих в построении линейных моделей AR и MA;
- параметр d соответствует порядку дифференцирования, необходимому для удаления несезонных трендов и приведения ряда к стационарному виду;
- целочисленный параметр s соответствует периоду сезонности — периодических колебаний временного ряда;

- параметры P, D и Q имеют тот же смысл, что и p, d и q , но относятся к сезонной составляющей ряда.

Параметры p, d, q, P, D и Q принимают значения из множества $\{0, 1, 2, 3\}$. Для выбранных наборов значений гиперпараметров оцениваются коэффициенты моделей AR и MA.

Входные данные (clean_data) при построении модели представляют собой набор кортежей вида $\langle \text{timestamp: time, value: real} \rangle$ — временной ряд значений некоторого датчика, установленного на элементе жилищной инфраструктуры, соответствующий штатному режиму работы элемента. Построение модели включает следующие шаги:

- вывести на экран графики функций автокорреляции (ACF — autocorrelation function) и частичной автокорреляции (PACF — partial ACF) данных. По свойствам этих графиков можно сделать вывод о возможных значениях гиперпараметров модели, тем самым существенно ускорив время обучения;
- считать введенные пользователем множества возможных значений гиперпараметров $p_range, d_range, q_range, P_range, D_range$ и Q_range , являющиеся подмножествами множества $\{0, 1, 2, 3\}$ и значение s^* параметра s ;
- для всех возможных наборов значений гиперпараметров p, d, q, P, D и Q и входных данных clean_data вычисляется значение информационного критерия Акаике (AIC — Akaike information criterion), выбираются конкретные значения гиперпараметров $p^*, d^*, q^*, P^*, D^*, Q^*$ и s^* , соответствующие минимальному значению AIC;
- обучить модель SARIMA на основании clean_data и значений $p^*, d^*, q^*, P^*, D^*, Q^*$ и s^* , при этом определить AR^* и MA^* — кортежи вещественных коэффициентов линейных моделей AR и MA;
- вычислить остатки построенной модели на входных данных clean_data;
- проверить статистическую гипотезу о нестационарности остатков временного ряда с использованием расширенного теста Дики–Фуллера (ADF — augmented Dickey–Fuller test). В случае если p -значение теста больше 0,05, вернуть предупреждение о неприменимости SARIMA, иначе вернуть набор параметров модели в виде кортежа $\langle p^*, d^*, q^*, P^*, D^*, Q^*, AR^*, MA^* \rangle$.

Для реализации указанных шагов использовались функции пакета p^* , statsmodels¹ языка Python.

Модель НММ. Данная модель [4] основывается на гипотезе о том, что у анализируемой системы имеется конечное множество «скрытых» состояний. Таким образом, помимо последовательности значений анализируемого временного ряда существует скрытая последовательность состояний (система может принимать ровно одно состояние в каждый момент времени). Скрытая марковская модель характеризуется набором параметров $\langle T, N, P, M, A \rangle$, где T — множество возможных скрытых состояний; N — множество возможных наблюдаемых значений; P — множество условных вероятностных распределений $p(x|t_i)$ для каждого скрытого состояния t_i ; M — матрица размера $|T| \times |T|$, элементы M_{ij} которой являются вероятностями перехода системы из состояния t_i в состояние t_j ; A — вероятностное распределение первого состояния скрытой марковской модели (априорное распределение).

В данном подходе метки классов неисправностей в каждый момент времени будут напрямую соответствовать скрытым состояниям модели. Используются два принципиально разных способа применения скрытых марковских моделей: в первом случае активно используется обучающая выборка, в которой сигнал уже разделен на сегменты, соответствующие исправным и неисправным режимам работы; разметка используется на этапе обучения параметров модели по методу максимума правдоподобия. Второй способ не предполагает наличие размеченной выборки, в этом случае для обучения модели применяется алгоритм Баума–Уэлша (частный случай EM (expectation-maximization) алгоритма) [5]. Главным ограничением такого подхода является необходимость заранее оценить число типов различных нештатных ситуаций, а следовательно, применимость НММ заметно снижается в случае отсутствия размеченной обучающей выборки. На практике также оказывается эффективным смешанный подход: проводить инициализацию параметров модели с помощью обучающей выборки по методу максимума правдоподобия, чтобы потом улучшить параметры с помощью алгоритма Баума–Уэлша.

Входные данные (training_data) при построении модели представляют собой набор кортежей вида $\langle \text{value: real, hidden_state: int} \rangle$, где value — элемент последовательности значений анализируемого временного ряда, а hidden_state — разметка элементов последовательности по классам

аномалий. При реализации используется класс HiddenMarkovModel пакета pomegranate² языка Python; для обучения на неразмеченных данных используется вызов метода fit на входных данных с параметром algorithm = 'baum-welch'. Для обучения на размеченных данных используется вызов метода fit на входных данных с параметрами labels = training_data.hidden_state, algorithm = labeled'; а затем — вызов метода fit на входных данных с параметром algorithm = 'baum-welch'.

В результате обучения формируется модель, содержащая конкретные значения вышеупомянутых параметров $\langle T, N, P, M, A \rangle$.

2.2 Обнаружение нештатных ситуаций

Компонент получает на вход параметры построенных моделей SARIMA и НММ и анализируемые временные ряды показаний датчиков, прошедшие предобработку. Цель работы данного компонента — бинарная классификация показаний датчиков на показания, соответствующие штатной работе исправного оборудования, и показания, соответствующие различным нештатным ситуациям (аномалиям во временном ряду).

Обнаружение нештатных ситуаций при помощи модели SARIMA. Модель SARIMA позволяет прогнозировать следующее значение временного ряда, зная несколько предыдущих значений. Таким образом, для обнаружения аномалий требуется сравнить реальные значения временного ряда с границами доверительного интервала SARIMA. Входными данными при обнаружении аномалий являются кортеж $\text{model} = \langle p, d, q, P, D, Q, \text{AR}, \text{MA} \rangle$ параметров модели и набор data кортежей вида $\langle \text{timestamp: time, value: real} \rangle$. В качестве выходных данных выступает бинарный массив labels, по числу элементов совпадающий с data, элементы которого принимают значения из множества $\{0, 1\}$, где 0 означает штатность соответствующего элемента data, а 1 — его аномальность. Процедура обнаружения аномалий включает следующие шаги:

- инициализировать переменную filtered_data значением data;
- для каждого i начиная с $\max(p, q)$ до $|\text{data}| - 1$ на основании первых i элементов временного ряда filtered_data предсказать $(i + 1)$ -й элемент ряда с использованием модели model, вычислить предсказанное значение forecast_value и доверительный интервал confidence_interval;
- проверить вхождение значения $\text{data}[i].\text{value}$ в интервал confidence_interval. В случае вхож-

¹<http://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html>.

²<https://pomegranate.readthedocs.io/en/latest/>.

дения присвоить 0 элементу $labels[i]$, иначе присвоить 1 элементу $labels[i]$ и присвоить значение $\langle data[i].timestamp, forecast_value \rangle$ элементу $filtered_data[i]$ для обеспечения возможности дальнейшего предсказания: фактически, при этом во временном ряду аномальное значение заменяется на штатное, предсказанное моделью.

Для реализации указанных шагов использовались функции пакета `statsmodels` языка Python.

Обнаружение нештатных ситуаций при помощи модели НММ. Здесь задача детектирования нештатных режимов работы оборудования рассматривается как задача сегментации сигнала: по известному набору параметров НММ требуется восстановить наиболее вероятную последовательность скрытых состояний системы. Данная задача решается при помощи алгоритма Витерби — классического алгоритма динамического программирования.

На входе процедура обнаружения аномалий получает обученную скрытую марковскую модель `model` и анализируемые данные `data` — набор кортежей вида $\langle timestamp: time, value: real \rangle$, соответствующие реальной (возможно, нештатной) работе оборудования — источника данных. При реализации используется вызов метода `predict` класса `HiddenMarkovModel` из пакета `pomegranate` языка Python на наборе `data` с параметром `algorithm = 'viterbi'`. Результатом работы является целочисленный массив `labels`, по числу элементов совпадающий с `data`, значения которого соответствуют скрытым состояниям (режимам работы) системы. Значение 0 соответствует штатному режиму работы, а все остальные — аномальным режимам.

3 Экспериментальные результаты

В рамках проекта RFMEFI60717X0176 для проведения экспериментов создан исследовательский стенд, развернутый на нескольких объектах жилищной инфраструктуры, включающий аппаратно-программные контроллеры и датчики (расходомеры, датчики входного/выходного тока/напряжения, температуры, звука, освещенности, давления), установленные на трансформаторах, водяных насосах, вентиляционных установках, кондиционерах, газовых котлах, ТЭНах, электрических лампах. В настоящее время продолжается сбор данных с компонентов исследовательского стенда для их последующего анализа. В рамках же данной статьи эксперименты проводятся на двух готовых наборах данных: Intel Lab Data и LUCE. Краткое описание наборов данных приведено ниже.

Данные исследовательской лаборатории Intel в Беркли. Набор данных состоит из показаний бытовых домашних метеостанций, размещенных в помещении лаборатории: измерений датчиков освещенности, влажности, температуры и напряжения [6]. Измерения снимались раз в 31 секунду на протяжении 36 дней, таким образом было собрано 2,3 млн записей. Данные представлены в виде набора кортежей $\langle date: yyyy-mm-dd, time: hh:mm:ss.xxx, temperature: real, light: real \rangle$. Для анализа использовались данные по температуре и освещенности. Одной из важных особенностей этого набора данных является ярко выраженная сезонность измерений (периодические колебания) с периодом 24 ч, связанная с суточным изменением температуры и освещенности.

Данные эксперимента LUCE проекта SensorScope. Данный набор данных также представляет собой показания метеостанций, но датчики расположены не в помещении, а на улице (использовались уличные метеостанции) [7]. Измерения считывались каждые 120 с, в набор данных вошли измерения за 44 дня. Среди измеренных значений присутствуют данные об окружающей среде, такие как температура, относительная влажность, солнечная радиация, скорость ветра и др. Для анализа была выбрана температурная составляющая измерений. Таким образом, кортеж анализируемых данных имеет вид $\langle date: yyyy-mm-dd, time: hh:mm:ss.xxx, temperature: real \rangle$. В данных также ярко выражена сезонная составляющая с периодом 24 ч.

Классификация возможных аномалий в наборах данных Intel Lab и SensorScope предложена в [8]. Позднее в [9] было сформулировано следующее определение. Пусть r_i — значение в момент времени i , предсказываемое моделью, а $r_i + eps_i$ — наблюдаемое значение. Тогда предлагается рассматривать следующие типы аномалий:

- *случайные аномалии*: одиночно встречающиеся значения-выбросы;
- *неисправности*: частое наличие в данных неверных значений таких, что $eps_i > t$ (где t — некоторое пороговое значение); в последовательности неверных значений не предполагается наличие закономерности.

В работе [10] разработаны алгоритмы моделирования данных классов аномалий, в частности построен контрольный набор данных, в который включена размеченная выборка, включающая в себя смоделированные аномалии. Результаты генерации неисправностей опубликованы в виде контрольного набора данных [11], в котором содержатся данные Intel Lab и SensorScope, включающие синтезированные неисправности. Кроме то-

го, для каждого измерения добавлено контрольное значение, соответствующее классу нештатной ситуации либо ее отсутствию в конкретный момент. Классы размечены следующим образом: «0» — нет аномалий; «1» — случайные аномалии; «2» — неисправности; «4» — смещение; «8», «16», «32» — полиномиальные искажения, сгенерированы разными алгоритмами. Эти данные были выбраны для контроля качества детектирования аномалий разрабатываемой системы.

На рассматриваемых данных были применены методы обнаружения аномалий, основанные на моделях SARIMA и HMM. Из набора данных Intel Lab Data были использованы данные по датчикам света, из набора данных LUCE — данные по датчикам температуры.

В результате настройки моделей SARIMA наиболее часто встречались следующие наборы гиперпараметров $\langle p, d, q, P, D, Q \rangle$: $\langle 0, 1, 1, 1, 1, 1 \rangle$ и $\langle 0, 1, 1, 0, 1, 1 \rangle$, при сезонном параметре $s = 24$ ч (сутки). Эксперименты показали, что модели SARIMA лучше подходят для обнаружения краткосрочных случайных аномалий, так как в случае долгосрочных аномалий доверительные интервалы ее предсказаний становятся шире, а следовательно, слабее.

Метод HMM показал себя более успешным на продолжительных аномалиях (неисправностях), при этом заметно уступая методу SARIMA в краткосрочных аномалиях (случайных выбросах). В работе [12] также для долгосрочных аномалий отдается предпочтение скрытым марковским моделям.

Так, на наборе Intel Lab Data средняя доля обнаруженных случайных аномалий составила 92,6% для SARIMA и 29,3% для HMM, а доля обнаруженных неисправностей составила 43,2% для SARIMA и 69,1% для HMM. На наборе LUCE средняя доля обнаруженных случайных аномалий составила 89,2% для SARIMA и 32,7% для HMM, а доля обнаруженных неисправностей составила 40,1% для SARIMA и 71,9% для HMM. Доля ложных срабатываний при этом не превышает 2% от общего числа измерений.

В работе [8] метод SARIMA был применен к тем же наборам данных, но моделирование аномалий проводилось иначе. Кроме того, параметры модели SARIMA использовались фиксированные: $\langle 0, 1, 1, 0, 1, 1 \rangle$. Результаты на случайных аномалиях оказались схожими — более 96% ARIMA (других сравнимых типов аномалий в этой работе не было).

Литература

1. Kovalev D., Shanin I., Stupnikov S., Zakharov V. Data mining methods and techniques for fault detection and predictive maintenance in housing and utility infrastructure // Conference (International) on Engineering Technologies and Computer Science. — IEEE, 2018. doi: 10.1109/EnT.2018.00016.
2. Box G. E. P., Jenkins G. M., Reinsel G. C., Ljung G. M. Time series analysis: Forecasting and control. — 5th ed. — Wiley, 2015. 712 p.
3. Cleveland R. B., Cleveland W. S., McRae J. E., Terpenning I. STL: A seasonal-trend decomposition // J. Off. Stat., 1990. Vol. 6. Iss. 1. P. 3–73.
4. Bengio Y., Frasconi P. An input output HMM architecture // NIPS Proceedings. — MIT Press, 1995. P. 427–434.
5. Rabiner L. R. A tutorial on hidden Markov models and selected applications in speech recognition // Readings in speech recognition / Eds. A. Waibel, K.-F. Lee. — San Francisco, CA, USA: Morgan Kaufmann, 1990. P. 267–296.
6. Bodik P., Hong W., Guestrin C., Madden S., Paskin M., Thibaux R. Intel Lab Data. Intel Berkeley Research lab, 2004. <http://db.csail.mit.edu/labdata/labdata.html>.
7. Sensorscope: Sensor networks for environmental monitoring. Lausanne Urban Canopy Experiment (LUCE). — EPFL, 2006. <https://lcav.epfl.ch/page-145180-en.html>.
8. Sharma A. B., Golubchik L., Govindan R. Sensor faults: Detection methods and prevalence in real-world datasets // ACM Trans. Sens. Netw., 2010. Vol. 6. Iss. 3. P. 23.
9. Baljak V., Tei K., Honiden S. Fault classification and model learning from sensory Readings — Framework for fault tolerance in wireless sensor networks // 8th Conference (International) on Intelligent Sensors, Sensor Networks and Information Processing. — IEEE, 2013. P. 408–413.
10. De Bruijn B., Nguyen T. A., Bucur D., Tei K. Benchmark datasets for fault detection and classification in sensor data // 5th Conference (International) on Sensor Networks Proceedings. — SCITEPRESS, 2016. P. 185–195.
11. De Bruijn B., Nguyen T. A., Bucur D., Tei K. Benchmark datasets for fault detection and classification in sensor data, 2015. <http://tuananh.io/datasets>.
12. Warriach E. U., Aiello M., Tei K. A machine learning approach for identifying and classifying faults in wireless sensor network // 15th Conference (International) on Computational Science and Engineering. — IEEE, 2012. P. 618–625.

Поступила в редакцию 15.07.18

METHODS AND TOOLS FOR FAULT DETECTION ON ELEMENTS OF HOUSING AND UTILITY INFRASTRUCTURE

I. A. Shanin, S. A. Stupnikov, and V. N. Zakharov

Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The work belongs to the area of development of specific information systems based on the Internet of Things technology. An approach for program implementation of a module intended for detection of faults on elements of housing and utility infrastructure is proposed. The module is considered as a part of an information system aimed at technical maintenance of mentioned elements: condition monitoring, predictive maintenance, fault detection, and reporting. Operation algorithms of module components are described: building of operation models for housing and utility infrastructure elements and fault detection. The approach is applied on a couple of datasets for fault detection, experimental results are presented.

Keywords: Internet of Things; data analysis; fault detection; housing and utility infrastructure

DOI: 10.14357/19922264180310

Acknowledgments

The research is supported by Ministry of Education and Science of the Russian Federation (project’s unique identifier RFMEFI60717X0176).

References

1. Kovalev, D., I. Shanin, S. Stupnikov, and V. Zakharov. 2018. Data mining methods and techniques for fault detection and predictive maintenance in housing and utility infrastructure. *Conference (International) on Engineering Technologies and Computer Science*. IEEE. doi: 10.1109/EnT.2018.00016.
2. Box, G. E. P., G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. 2015. *Time series analysis: Forecasting and control*. 5th ed. Wiley. 712 p.
3. Cleveland, R. B., W. S. Cleveland, J. E. McRae, and I. Terpenning. 1990. STL: A seasonal-trend decomposition. *J. Off. Stat.* 6(1):3–73.
4. Bengio, Y., and P. Frasconi. 1995. An input output HMM architecture. *NIPS Proceedings*. MIT Press. 427–434.
5. Rabiner, L. R. 2004. A tutorial on hidden Markov models and selected applications in speech recognition. *Readings in speech recognition*. Eds. A. Waibel and K.-F. Lee. San Francisco, CA: Morgan Kaufmann. 267–296.
6. Bodik, P., W. Hong, C. Guestrin, S. Madden, M. Paskin, and R. Thibaux. 2004. Intel Lab Data. Intel Berkeley Research lab. Available at: <http://db.csail.mit.edu/labdata/labdata.html> (accessed July 16, 2018).
7. Sensorscope. 2006. Sensorscope: Sensor networks for environmental monitoring. Lausanne Urban Canopy Experiment (LUCE). EPFL. Available at: <https://lcav.epfl.ch/page-145180-en.html> (accessed July 16, 2018).
8. Sharma, A. B., L. Golubchik, and R. Govindan. 2010. Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Trans. Sens. Netw.* 6(3):23.
9. Baljak, V., K. Tei, and S. Honiden. 2013. Fault classification and model learning from sensory Readings — Framework for fault tolerance in wireless sensor networks. *8th Conference (International) on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE. 408–413.
10. De Bruijn, B., T. A. Nguyen, D. Bucur, and K. Tei. 2016. Benchmark datasets for fault detection and classification in sensor data. *5th Conference (International) on Sensor Networks Proceedings*. SCITEPRESS. 185–195.
11. De Bruijn, B., T. A. Nguyen, D. Bucur, and K. Tei. 2015. Benchmark datasets for fault detection and classification in sensor data. Available at: <http://tuananh.io/datasets/> (accessed July 16, 2018).
12. Warriach, E. U., M. Aiello, and K. Tei. 2012. A machine learning approach for identifying and classifying faults in wireless sensor network. *15th Conference (International) on Computational Science and Engineering*. IEEE. 618–625.

Received July 15, 2018

Contributors

Shanin Ivan A. (b. 1991) — junior scientist, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; v08shanin@gmail.com

Stupnikov Sergey A. (b. 1978) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; sstupnikov@ipiran.ru

Zakharov Victor N. (b. 1948) — Doctor of Science in technology, associate professor; Scientific Secretary, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; vzakharov@ipiran.ru