УДК 004.855.5 DOI: 10.14529/mmp160408

МОДИФИКАЦИЯ АЛГОРИТМА СЛУЧАЙНОГО ЛЕСА ДЛЯ КЛАССИФИКАЦИИ НЕСТАЦИОНАРНЫХ ПОТОКОВЫХ ДАННЫХ

А.В. Жуков, Д.Н. Сидоров

Предложен метод классификации нестационарных потоковых данных. К таким данным относятся характеристики поведения сложных систем, процессы, обладающие высокой степенью стохастичности, такие как скорость ветра. В данной работе предложена эффективная модификация алгоритма случайного леса, позволяющая повысить точность классификации состояния путем взвешивания ответов отдельных классификаторов композиции. Опираясь на метод Accuracy Weighted Ensemble (AWE), взвешивание производится в соответствии с оценкой ошибки каждого классификатора на новых данных. Такая оценка производится с использованием метода k ближайших соседей и внутренней структуры случайного леса. В качестве стратегии обновления композиции используется замена классификаторов с низкой точностью на новых данных. Приводятся результаты тестирования предложенного метода и сравнение с другими современными методами.

Ключевые слова: классификация; смещение концепта; случайный лес; решающие деревья; композиции.

Введение

Анализ потоковых данных, характеризующих поведение нестационарных динамических систем, является важной задачей, возникающей в различных областях науки и производства. Например, одной из таких задач является идентификация состояния энергосистемы, так как при анализе электроэнергетических сетей большое влияние на поведение системы оказывают различные социально-экономические факторы, характеризуемые высоким уровнем стохастичности. Идентификация состояния, как и многие другие задачи электроэнеретики (см. библиографию в [1]), сводится к задаче классификации поступающих потоковых данных.

Одним из наиболее распространенных подходов к решению задачи классификации является использование композиционных методов. Композиционные методы (или ансамбли) классификации формируют набор различных моделей классификации для достижения лучшей точности, чем у каждой модели в отдельности. Композиции широко используются в самых различных исследованиях. Одним из самых эффективных композиционных алгоритмов общего назначения является метод построения решающих деревьев «случайный лес», RandomForest [2]. Этот алгоритм использует бэггинг [3] и метод случайных подпространств [4] для создания композиции высоко декоррелированых деревьев решений, что позволяет достигать достаточно высокой точности и устойчивости к шуму в данных.

Во многих работах, посвященных исследованию методов классификации с помощью деревьев решений и их композиций, рассматривается лишь стационарный случай. При работе с нестационарными потоковыми данными мы имеем дело с изменением природы данных во времени. Такое изменение в данных называют сменой

концепта, то есть изменением вероятностных закономерностей данных. По характеру изменений можно провести (см., например, [5]) следующую классификацию смены концепта: резкое изменение (sudden drift), постепенное изменение (gradual drift), последовательное изменение (incremental drift), случайно повторяющееся (reoccuring contexts). В данном исследовании мы будем говорить только о постепенном и последовательном изменении.

Говоря о потоке мы имеем в виду данные, поступающие последовательно по одному примеру или целыми блоками с одинаковыми, либо различными временными интервалами. Нужно также отметить, что работа с потоковыми данными накладывает ограничения на используемую память, время работы и на количество проходов по данным при обучении, что усложняет процесс создания новых эффективных решений в этой области.

Существует много различных подходов к работе со сменой концепта, в том числе, с использованием композиционного подхода [6]. Однако, в данной работе мы ограничимся рассмотрением композиционных методов работы с нестационарными потоковыми данными, основанными на решающих деревьях. Для более детального рассмотрения читатель может обратиться к монографии [7] и обзору [8].

В работе [9], посвященной композиционному методу Accuracy Weighted Ensemble (AWE), представлен подход, позволяющий повысить точность классификации потоковых данных за счет взвешивания ответов отдельных классификаторов композиции. Вес каждого рассчитывается с помощью оценки ошибки на вновь поступивших данных. Чем больше ошибка, тем меньший вес присваивается алгоритму. Такая оценка формируется благодаря использованию буфера с ретроспективными данными.

С момента изобретения оригинального Random Forest было предложено несколько способов адаптации к нестационарным потоковым данным различной природы. Метод Incremental Extremely Random Forest [10] специализируется на потоках с малым количеством измерений. В качестве базовых классификаторов в таком ансамбле выступают полностью рандомизированные деревья (Extremely Randomized Trees) [11] с критерием качества разбиения, основанном на индексе Джини.

Online Random Forest [12] опирается на идеи онлайн бэггинга [13] и полностью рандомизированных деревьев [11]. Таким образом, удается получить достаточно быстрый онлайн алгоритм, позволяющий использовать его для задачи отслеживания объектов в видеопотоке.

Метод Streaming Random Forest [14] пользуется оценкой Хефдинга при построении решающих деревьев и обрезкой ветвей как стратегией забывания.

Иной современный метод Mondrian Forests [15] вводит так называемые Мондриановские деревья, которые включают зависимость от времени в каждое разбиение (ветвь дерева).

1. Постановка задачи

В задаче классификации мы имеем X – множество описаний объектов, Y – конечное множество номеров меток классов. Существует неизвестная целевая зависимость – отображение $y^* : X \to Y$, значения которой известны только на объектах конечной обучающей выборки $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Требуется построить алгоритм $a : X \to Y$, способный классифицировать произвольный объект $x \in X$. Как

критерий качества алгоритма может быть использована точность на тренировочной выборке $X^l = \{(\check{x}_1, \check{y}_1), \ldots, (\check{x}_l, \check{y}_l)\}$, равная

$$\mathcal{A}(a, X^l) = \frac{1}{l}\sum_{i=1}^{l}[a(\check{x}_i) = \check{y}_i].$$

В случае нестационарных потоковых данных мы также должны учитывать необходимость в адаптации модели.

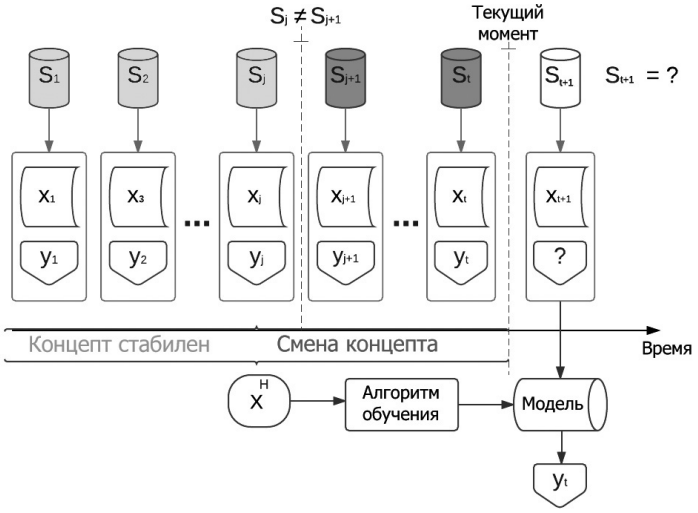

Рис. 1. Схема работы с потоковыми данными со сменой концепта

В рамках так называемой теории последовательного обучения [10], как показано на рис. 1, мы имеем упорядоченный по времени набор измерений $\{X\}_1^t$, $x_t \in X$ – измерение полученное в момент времени t из источника S_t, $y_t \in Y$ – метка класса соответствующая x_t, $X^H = \{x_1, \ldots, x_t\}$ – исторические (ретроспективные) данные, $x_{t+1} \in X$ – текущее измерение. Задача состоит в том, чтобы предсказать метку класса y_{t+1} для x_{t+1}, настоящее значение которой будет получено на следующем шаге. В качестве критерия качества будем использовать среднюю точность по всем тестовым примерам или блокам примеров $\bar{\mathcal{A}}$. Как показано в [16], смена концепта может быть представлена в виде изменений в вероятностях классов $P(c_i)$, $p(x|c_i)$ или $p(x)$, где

$$p(c_i|x) = \frac{P(c_i)p(x|c_i)}{p(x)}. \tag{1}$$

Таким образом, концепт (или источник данных) может быть определен как набор априорных вероятностей классов и условных плотностей вероятности:

$$S_t = \{(P(c_1), p(x|c_1)), \ldots, (P(c_k), p(x|c_k))\}.$$

Для работы с нестационарными данными мы должны построить предположение о S_{t+1}.

2. Метод PDSRF

Для создания нового метода классификации, способного эффективно работать в условиях смены концепта, мы должны ответить на следующие вопросы:

1) как адаптировать Random Forest для работы с потоковыми данными,

2) как определить стратегию обновления (забывания) модели и базовое предположение относительно S_{t+1}.

Методологически композиционный подход позволяет адапитроваться к изменениям в данных следующими способами: адаптировать базовый классификатор, произвести манипуляции с обучающей выборкой, адаптировать агрегирующую функцию (корректирующую операцию) или изменить структуру композиции. В этой работе мы используем комбинацию представленных подходов. В противоположность оригинальному Random Forest в качестве агрегирующей операции используется взвешенное голосование. При этом реализуется подход, подобный предложенному в [9], так, каждому базовому алгоритму ставится в соответствие весовая функция $\omega_t(x) \in [0, 1]$, которая, тем меньше, чем больше предполагаемая ошибка алгоритма $E_t(x)$ в точке x. Эта функция должна быть приближена на основе ретроспективных данных. При этом мы предполагаем, что характер изменения $\omega_t(x)$ по времени достаточно медленный, то есть используем базовое предположение $S_t = S_{t+1}$.

Для вычисления весов алгоритмов ансамбля необходимо хранить некоторый набор измерений. Для этих целей мы используем скользящее окно постоянной длины по времени (как предложено в периодически обновляемом случайном лесе [17]). Длина окна оценивается из эмпирических соображений и может быть получена с помощью перекрестной проверки. Так как мы имеем дело с потоковыми алгоритмами, то на размер окна также должны быть наложены ограничения по используемой памяти и времени работы, которые определяются условиями конкретной задачи.

В целях сокращения затрат по памяти и времени выполнения, в качестве базовых алгоритмов используются полностью рандомизированные деревья. При этом каждое из деревьев в отдельности используется в режиме оффлайн, без каких-либо модификаций.

Таким образом, в данной статье предлагается оригинальный подход, называемый Proximity Driven Streaming Random Forest (PDSRF), реализущий принципы как алгоритма Random Forest, так и метода AWE. В качестве базовых алгоритмов композиции используется Extremely Randomized Tree [11], что позволяет ускорить процесс обучения за счет использования только отдельных деревьев в режиме «офф-лайн» и обновления композиции простой заменой деревьев с низкой точностью. При этом замена происходит только в том случае, если средняя точность всего ансамбля меньше заданного порога, выбираемого из эмпирических соображений. Также ограничивается количество замен с целью лимитировать вычислительные затраты и не допустить переобучения ансамбля.

Оценка ошибки конкретного классификатора на новых примерах выполняется из следующих предположений:

1) классификаторы имеют близкие ошибки на «похожих» примерах,

2) функция ошибки для конкретного классификатора медленно меняется со временем $S_t = S_{t+1}$.

Это позволяет приблизить функцию ошибки с помощью взвешенного метода k ближайших соседей. В качестве обучающей выборки извлекаются примеры из буфера, а в качестве значений отклика – ошибки классификаторов на примерах буфера. Таким образом, для того, чтобы найти веса для классификации конкретного примера, первоначально происходит поиск похожих примеров в буфере, затем усреднение

и полученное значение определяется следующим образом:

$$\omega_i = \frac{1}{E_i^2(x) + \Delta}, \tag{2}$$

где $E_i(x)$ – предполагаемая ошибка i-го классификатора ансамбля на примере x, а Δ – малый положительный параметр, который задает максимальный возможный вес.

Приближение будем строить с помощью взвешенного метода k ближайших соседей. Путь мы имеем буфер скользящего окна $B^q = (\acute{x}_1, \acute{y}_1), \ldots, (\acute{x}_q, \acute{y}_q)$ с размером q. Задана функция расстояния $\rho(x, \acute{x})$. Для примера $x \in X$ упорядочим все примеры буфера в порядке возрастания расстояния $\rho(x, \acute{x}_{1;x}) \leq \rho(x, \acute{x}_{2;x}) \leq \cdots \leq \rho(x, \acute{x}_{q;x})$, тогда искомая функция может быть оценена как

$$\tilde{E}_m^{t+1}(x) = \frac{\sum_{i=1}^k E_m^t(\acute{x}_{x,i})\rho(x, \acute{x}_{x,i})}{\sum_{i=1}^k \rho(x, \acute{x}_{x,i})}, \tag{3}$$

где k – количество ближайших соседей, m – номер алгоритма ансамбля.

Функция расстояния для метода k ближайших соседей может быть определена разными способами. Однако, с целью сокращения вычислительной сложности, целесообразно использовать внутреннюю функцию близости Random Forest Proximity. Эта функция использует структуру деревьев для получения значения близости следующим образом: если два примера попадают в одну и ту же ячейку пространства признаков, соответствующую терминальному узлу дерева, то значение увеличивается на единицу. По завершении процесса итоговая функция определяется по количеству деревьев.

Формально, такая метрика может быть определена по аналогии с KeRF [18]. Пусть для ансамбля из M деревьев, мы имеем обучающую выборку $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ и набор независимых случайных величин $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \cdots, \boldsymbol{\Theta}_M\}$, которые не зависят от $\mathcal{D}_n$ и используются в процедурах рандомизации, для выбора случайного разбиения и построения случайных подвыборок (создание бутстрэпа). Тогда определим $A_n(x, \Theta_j)$ как ячейку пространства признаков, соответствующую терминальной ветви дерева, в которую попадает пример x, если дерево построено при использовании Θ_j по выборке $\mathcal{D}_n$. В этих терминах метрика близости может быть выражена как

$$Prox(x, u) = \frac{1}{M} \sum_{j=1}^{M} [x \in A_n(u, \Theta_j)]. \tag{4}$$

Таким образом, мы имеем два механизма адаптации к изменениям в анализируемых данных: замена неэффективных базовых алгоритмов и адаптивное правило объединения ответов алгоритмов композиции.

3. Тестирование

Для тестирования был выбран набор данных CoverType [19]. Этот набор данных широко используется для тестирования методов, работающих в условиях смещения концепта. Набор содержит данные о изменении лесного покрова и состоит из 581 012 примеров и 54 атрибутов.

Реализация предложенного алгоритма осуществлялась средствами языка C++. Эффективность работы алгоритмов оценивалась для каждого нового блока в отдель-

ности, затем происходило усреднение. После тестирования новый блок попадал в ретроспективную выборку. Таким образом, каждый новый блок сначала используется для тестирования, а затем для дообучения модели [20].

Предлагаемый алгоритм протестирован с различными размерами блока, скользящего окна, количеством ближайших соседей и размером ансамбля (табл. 1). Чтобы сделать результаты более интерпретируемыми наряду с средней точностью $\bar{A}$ предложенного алгоритма, представлена точность ансамбля без взвешивания $\bar{A}^*$. Так мы можем наглядно показать вклад операции в итоговую точность работы алгоритма. В табл. 2 показано, что предложенный алгоритм превосходит другие представленные алгоритмы по точности. Результаты для алгоритмов HOT, AUE2, AUE1, Lev, DWM, Oza, AWE, Win получены с использованием настроек описанных в работе [21].

Таблица 1

Средняя точность предложенного алгоритма при различных параметрах, где k – количество ближайших соседей

размер блока	размер окна	k	$\bar{A}^*$	$\bar{A}$
300	1000	5	77,65	81,15
300	1000	10	77,69	81,05
300	1000	20	77,69	80,52
300	1500	5	77,67	81,11
300	1500	10	77,89	81,21
300	1500	20	77,8	80,83
500	500	5	82,76	86,38
500	500	10	82,73	86,16
500	500	20	82,68	86,02
500	1000	5	82,76	86,45
500	1000	10	82,87	86,27
500	1000	20	82,75	86,04
500	1500	5	82,75	86,49
500	1500	10	82,74	86,29
500	1500	20	82,7	85,96

Таблица 2

Сравнение результатов тестирования различных алгоритмов на наборе данных

Метод	PDSRF	HOT	AUE2	PDSRF (без взвешивания)	AUE1	Lev	DWM	Oza	AWE	Win
$\bar{A}$	87,42	86,48	85,20	82,79	81,24	81,04	80,84	80,40	79,34	77,19

Как показано на рис. 2, предложенное взвешивание позволяет значительно повысить точность работы. И хотя такая мера сопряжена с дополнительными вычислениями, их количество удается сократить благодаря использованию внутренней функции близости. При этом, такое приближение не уступает в точности другим функциям,

таким как эвклидово расстояние (рис. 3). Нужно также заметить, что при достаточно больших размерах скользящего окна оптимальное (в смысле максимума точности) количество ближайших соседей стремится к единице. Сравнение результатов работы алгоритма с различным количеством ближайших соседей приведено в табл. 3.

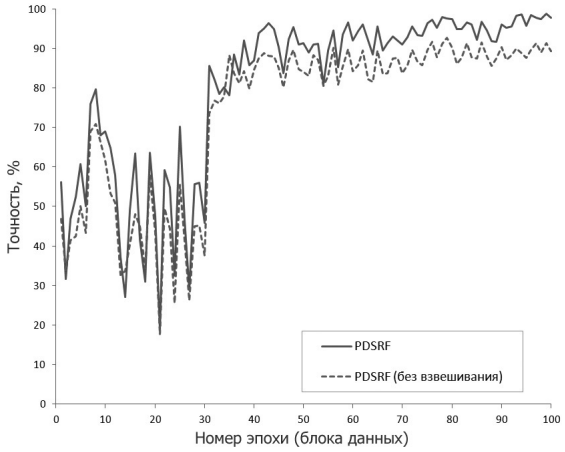

Рис. 2. Точность (accuracy) работы предложенного алгоритма с взвешиванием и без него на первых 100 блоках набора данных CoverType

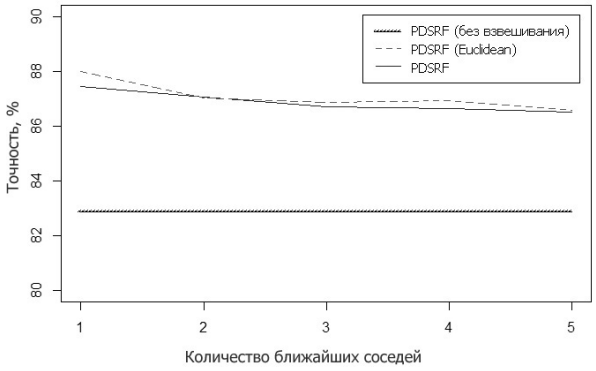

Рис. 3. Средняя точность предложенного алгоритма для ансамбля без взвешивания, а также с использованием внутренней функции близости случайного леса и эвклидового расстояния для размера блока равного 500 и размера окна 1500 примеров

Заключение

В сравнении с современными алгоритмами для работы со сменой концепта, такими как Online Random Forest и AWE, предложенный метод требует больших вычислительных ресурсов как на стадии обучения, так и на стадии предсказания, однако, он обладает наибольшей точностью среди известных нам подходов. Предлагаемый подход (как и классический Random Forest) остается хорошо распараллеливаемым и может быть эффективно реализован с использованием технологии GPGPU.

Таблица 3

Средняя точность алгоритма на первых 100 блоках

размер блока	размер окна	k	количество деревьев	$\mathcal{A}^*$	$\mathcal{A}$
500	1000	1	5	75,03	79,25
500	1000	1	7	74,37	79,42
500	1000	1	10	74,34	78,76
500	1000	1	13	74,22	79,56
500	1000	1	15	74,00	79,72
500	1000	1	17	74,18	80,09
500	1000	1	20	74,23	79,90
500	1000	2	5	74,96	78,77
500	1000	2	7	74,36	79,33
500	1000	2	10	74,14	79,51
500	1000	2	13	74,03	79,02
500	1000	2	15	74,02	79,51
500	1000	2	17	74,35	79,68
500	1000	2	20	74,09	79,85
500	1000	3	5	74,83	77,78
500	1000	3	7	74,66	79,59
500	1000	3	10	74,34	79,14
500	1000	3	13	74,35	79,17
500	1000	3	15	74,09	79,32
500	1000	3	17	74,09	79,68
500	1000	3	20	74,13	80,22
500	1000	5	5	74,59	78,70
500	1000	5	7	74,29	78,76
500	1000	5	10	74,26	78,66
500	1000	5	13	73,94	79,52
500	1000	5	15	74,37	80,16
500	1000	5	17	74,09	79,41
500	1000	5	20	74,01	80,16

Как показано в табл. 3, высокая точность может быть достигнута при достаточно малом количестве базовых алгоритмов ансамбля.

Нужно заметить, что предложенный подход может быть эффективно использован для работы с постепенными и последовательными изменениями концепта. Алгоритм чувствителен к изменению всех параметров и, соответственно, каждый параметр должен быть тщательно настроен.

Так как предлагаемый в работе метод наследует многие свойства случайного леса, то он также имеет потенциальную возможность для работы в режиме без учителя. Структура леса может быть использована для заполнения пропусков в данных, что крайне важно для работы с приложениями, в которых необходимо предусмотреть возможность потерь данных.

Планируется применение описанного в этой работе метода для решения задачи идентификации состояний электроэонергетической системы с целью ее мониторинга.

Данная работа поддержана грантом Российского научного фонда 14-19-00054.

Литература / References

1. Tomin N., Zhukov A., Sidorov D., Kurbatsky V., Panasetsky D., Spiryaev V. Random Forest Based Model for Preventing Large-Scale Emergencies in Power Systems. *International Journal of Artificial Intelligence*, 2015, vol. 13, no. 1, pp. 221–228.
2. Breiman L. Random Forests. *Machine Learning*, 2001, vol. 45, no. 1, pp. 5–32. DOI: 10.1023/A:1010933404324
3. Breiman L. Bagging Predictors. *Machine Learning.* 1996, vol. 24, no. 2, pp. 123–140. DOI: 10.1023/A:1018054314350.
4. Ho Tin Kam. The Random Subspace Method for Constructing Decision Forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 1998, vol. 20, no. 8, pp. 832–844. DOI: 10.1109/34.709601
5. Žliobaitė Indrė. *Learning under Concept Drift: an Overview.* arXiv preprint arXiv:1010.4784. 2010.
6. Haixun Wang, Wei Fan, Yu P.S., Han J. Mining Concept-Drifting Data Streams Using Ensemble Classifiers. *Proceedings of SIGKDD, August 24–27, 2003*, Washington, DC, 2003, pp. 226–235.
7. Gama J. *Knowledge Discovery from Data Streams.* Singapore, CRC Press Publ., 2010. DOI: 10.1201/EBK1439826119
8. Kuncheva L. Classifier Ensembles for Changing Environment. *Multiple Classifier Systems, 2004 5th Intl. Workshop*, Springer-Verlag, 2004, pp. 1–15. DOI: 10.1007/978-3-540-25966-4_1
9. Haixun Wang, Wei Fan, Yu P.S., Han J. Mining Concept-Drifting Data Streams Using Ensemble Classifiers. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2003, pp. 226–235. DOI: 10.1145/956750.956778
10. Aiping Wang, Guowei Wan, Zhiquan Cheng, Sikun Li. An Incremental Extremely Random Forest Classifier for Online Learning and Tracking. *Image Processing (ICIP), 2009 16th IEEE International Conference.* IEEE, 2009, pp. 1449–1452.
11. Geurts P., Ernst D., Wehenkel L. Extremely Randomized Trees. *Machine Learning*, 2006, vol. 63, no. 1, pp. 3–42. DOI: 10.1007/s10994-006-6226-1
12. Santner J., Saffari A., Leistneret C. et al. On-Line Random Forests. *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference.* IEEE, 2009, pp. 1393–1400.
13. Oza N.C. Online Bagging and Boosting. *Systems, Man and Cybernetics, 2005 IEEE International Conference.* IEEE, vol. 3, 2005, pp. 2340–2345. DOI: 10.1109/icsmc.2005.1571498
14. Abdulsalam H., Skillicorn D.B., Martin P. Classification Using Streaming Random Forests. *Knowledge and Data Engineering, IEEE Transactions.* 2011, vol. 23, no. 1, pp. 22–36.
15. Lakshminarayanan B., Roy D.M., Teh Yee Whye. Mondrian Forests: Efficient Online Random Forests. *Advances in Neural Information Processing Systems*, 2014, pp. 3140–3148.

16. Kelly M.G., Hand D.J., Adams N.M. The Impact of Changing Populations on Classifier Performance. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 1999, pp. 367–371. DOI: 10.1145/312129.312285
17. Zhukov A., Kurbatsky V., Tomin N. et al. Random Forest Based Model for Emergency State Monitoring in Power Systems. *Mathematical Method for Pattern Recognition: Book of Abstract of the 17th All-Russian Conference with Interneational Participation.* Svetlogorsk, TORUS PRESS, 2015, pp. 274.
18. Scornet E. Random Forests and Kernel Methods. *IEEE Transactions on Information Theory*, 2016, vol. 62, no. 3, pp. 1485–1500. DOI: 10.1109/TIT.2016.2514489
19. Blake C.L., Merz C.J. *UCI Repository of Machine Learning Databases.* 1998.
20. Brzezinski D. *Mining Data Streams with Concept Drift. Diss. MS Thesis. Dept. of Computing Science and Management.* Poznan University of Technology, 2010.
21. Brzezinski D., Stefanowski J. Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm. *Neural Networks and Learning Systems, IEEE Transactions*, 2014, vol. 25, no. 1, pp. 81–94.

Алексей Витальевич Жуков, аспирант кафедры «Информационные технологии», Институт математики, экономики и информатики, Иркутский национальный исследовательский технический университет (г. Иркутск, Российская Федерация), zhukovalex13@gmail.com.

Денис Николаевич Сидоров, доктор физико-математических наук, ведущий научный сотрудник, Институт систем энергетики им. Л.А. Мелентьева СО РАН; профессор, кафедра «Информационные технологии», Институт математики, экономики и информатики Иркутского государственного университета; Иркутский национальный исследовательский технический университет (г. Иркутск, Российская Федерация), contact.dns@gmail.com.

Поступила в редакцию 27 мая 2016 г.

MSC 68T05 **DOI: 10.14529/mmp160408**

MODIFICATION OF RANDOM FOREST BASED APPROACH FOR STREAMING DATA WITH CONCEPT DRIFT

A.V. Zhukov, Institute of Mathematisc, Economics and Computer Science, Irkutsk State University, Irkutsk, Russian Federation, zhukovalex13@gmail.com,
D.N. Sidorov, Melentiev Energy Systems Institute, Siberian Branch of Russian Academy of Sciences, Irkutsk State University, Irkutsk National Research Technical University, Irkutsk, Russian Federation, dsidorov@isem.irk.ru

In this paper concept drift classification method was presented. Concept drift methods have potential in complex systems analysis and other processes which have stochastic nature like wind power. We present decision tree ensemble classification method based on the Random Forest algorithm for concept drift. Inspired by Accuracy Weighted Ensemble (AWE) method the weighted majority voting ensemble aggregation rule is employed. Base learner weight in our case is computed for each sample evaluation using base learners accuracy and intrinsic proximity measure of Random Forest. Our algorithm exploits ensemble pruning as a forgetting strategy. We present results of empirical comparison of our method and other state-of-the-art concept drift classifiers.

Keywords: decision tree; concept drift; ensemble learning; classification; random forest.

Received May 27, 2016