Прогнозирование метрики Facebook на основе машинного обучения

Emmanuel Sam
Faculty of Computing and Information Technology
Wisconsin International University College, Accra, Ghana
elsam@wiuc-ghana.edu.gh

Sebastian Basterrech
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University, Prague, Czech Republic
Sebastian.Basterrech@fel.cvut.cz

С. А. Ярушев
Кафедра Информатики, РЭУ им. Г.В. Плеханова
Sergey.Yarushev@icloud.com

А. Н. Аверкин
Вычислительный центр им. А.А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук
Averkin2003@inbox.ru

***Аннотация*. В данной статье мы оцениваем эффективность трех известных методов машинного обучения для прогнозирования публикации в Facebook. Социальные медиа оказывают огромное влияние на социальное поведение. Поэтому разработать автоматическую модель для прогнозирования влияния должностей в социальных медиа может быть полезно обществу. В этой статье, мы анализируем эффективность прогнозирования импакта публикаций на основе трех популярных методов: поддержки векторной регрессии (SVR), сети состояний эха (ESN) и адаптивной нейро-нечеткой логической системы (ANFIS). Оценка проводилась по общедоступному и известному базовому набору данных.**

Ключевые слова: прогнозирование; машинное обучение; нейро-нечеткие сети; нейронные сети; социальные сети

I. Введение

В настоящее время социальные медиа влияют на коллективное поведение, и они играют очень важную роль в распространении информации. По этой причине автоматический метод прогнозирования метрики в социальных медиа может быть полезен в нескольких областях, таких как маркетинг, образовательные системы, безопасность и т. д.

В данном исследовании мы анализируем альтернативные инструменты машинного обучения для прогнозирования группы показателей Facebook. Цель состоит в том, чтобы построить систему автоматического прогнозирования влияния публикаций в Facebook. Предыдущее исследование в [1] продемонстрировало возможность прогнозирования некоторых метрик Facebook с поддержкой метода опорной регрессии (SVR). Немногие из показателей дали хорошие результаты, когда оценка производится с использованием средней абсолютной процентной погрешности. В этом исследовании мы прогнозируем 3 метрики: комментарии, репосты и лайки. Эти показатели являются взяты из результатов работы [1], где была определена некая мера: *Всего взаимодействий сообщения*, которая определяется как сумма таких показателей, как комментарии, репосты и лайки. В этой статье мы представляем результаты, полученные тремя популярными методами. Метод опорных регрессий (SVR), которая базируется в ядрах. Echo State Network (ESN), которая является техникой, основанной на производительности повторяющихся нейронных сетей и линейных регрессий. Кроме того, мы оцениваем Adaptive Neuro-Fuzzy Inject System (ANFIS), которая представляет собой распределенную параллельную систему с нечеткими правилами. Производительность моделей ANFIS в задачах прогнозирования показана в этой работе [2].

Остальная часть этой статьи организована следующим образом. Раздел 2 рассматривает, как методы машинного обучения, которые должны быть изучены в этом исследовании, были применены в других исследованиях, связанных с метриками в социальных сетях. Раздел 2 содержит описание методов и их свойств, а также методологию этого исследования. Описание данных, результат анализа и соответствующие обсуждения представлены в разделе 3. Выводы и рекомендации для будущей работы приведены в разделе 4.

II. Методы машинного обучения

A. Постановка задачи

Пусть $x(t)$ – p-мерные входные данные, а $y(t)$ – одномерная выходная переменная. Учитывая учебный набор данных, составленный T реальными парами ввода-вывода $(x(t), y(t))$. Цель состоит в том, чтобы определить модель $\varphi(\cdot)$. Для прогнозирования переменной результата на основе набора функций ввода. Заметим, что у нас есть несколько выходных переменных, и мы моделируем каждый из них независимо. Модель оценивается с использованием количественной меры, называемой функцией стоимости, которая измеряет качество модели обучения. В этой статье мы используем самую

популярную метрику, когда выходная переменная является реальным значением, Mean Squared Error (E_{MSE}):

$$E_{MSE} = \frac{1}{T}(\hat{y}(t) - y(t))^2, \qquad (1)$$

где $\hat{y}(t)$обозначает предсказание для входа *x(t).*

B. Нейро-Нечеткая Сеть ANFIS

ANFIS – это аббревиатура Adaptive-Network-Based Fuzzy Inference System – адаптивная сеть нечеткого вывода. Она была предложена Янгом (Jang) в начале девяностых [3]. ANFIS является одним из первых вариантов гибридных нейро-нечетких сетей – нейронной сети прямого распространения сигнала особого типа. Архитектура нейро-нечеткой сети изоморфна нечеткой базе знаний. В нейро-нечетких сетях используются дифференцируемые реализации треугольных норм (умножение и вероятностное ИЛИ), а также гладкие функции принадлежности. Это позволяет применять для настройки нейро-нечетких сетей быстрые алгоритмы обучения нейронных сетей, основанные на методе обратного распространения ошибки. Ниже описываются архитектура и правила функционирования каждого слоя ANFIS-сети. Материал базируется на книге [4].

Входы сети в отдельный слой не выделяются. На рис. 1 изображена ANFIS-сеть с двумя входными переменными (x_1 и x_2) и четырьмя нечеткими правилами. Для лингвистической оценки входной переменной x_1 используется 3 терма, для переменной x_2 – 2 терма.

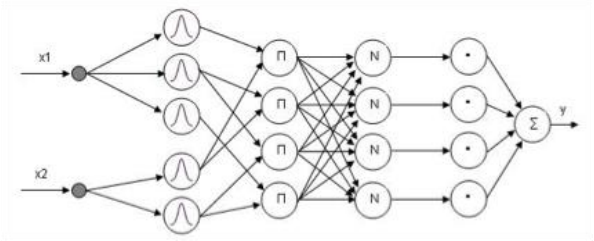

Рис. 1. Пример ANFIS-сети

Введем следующие обозначения, необходимые для дальнейшего изложения:

$x_1, \ldots, x_n$– входы сети;

y – выход сети;

R_r: Если $x_1 = a_{1,r}$и … и $x_n = a_{n,r}$, то $y = b_{0,r} + b_{1,r}x_1 + \cdots + b_{n,r}x_n$- нечеткое правило с порядковым номером r;

m – количество правил , $r = \overline{1,m}$;

$a_{i,r}$– нечеткий терм с функцией принадлежности $\mu_r(x_i)$, применяемый для лингвистической оценки переменной x_i в r-ом правиле ($r = \overline{1,m}$, $i = \overline{1,n}$);

$b_{q,r}$– действительные числа в заключении r-го правила ($r = \overline{1,m}$,$\tau_i^* = \frac{\tau_r}{\sum_{j=\overline{1,m}}\tau i}$).

ANFIS-сеть функционирует следующим образом [5]:

Слой 1. Каждый узел первого слоя представляет один терм с колоколообразной функцией принадлежности. Входы сети x_1, x_2,…, x_n соединены только со своими термами. Количество узлов первого слоя равно сумме мощностей терм-множеств входных переменных. Выходом узла являются степень принадлежности значения входной переменной соответствующему нечеткому терму:

$$\mu_r(x_i) = \frac{1}{1+\left|\frac{x_i - c}{a}\right|^{2b}}, \qquad (2)$$

где a, b и c – настраиваемые параметры функции принадлежности.

Слой 2. Количество узлов второго слоя равно m. Каждый узел этого слоя соответствует одному нечеткому правилу. Узел второго слоя соединен с теми узлами первого слоя, которые формируют антецеденты соответствующего правила. Следовательно, каждый узел второго слоя может принимать от 1 до n входных сигналов. Выходом узла является степень выполнения правила, которая рассчитывается как произведение входных сигналов. Обозначим выходы узлов этого слоя через τ_r, r=(1,m)¯.

Слой 3. Количество узлов третьего слоя также равно m. Каждый узел этого слоя рассчитывает относительную степень выполнения нечеткого правила:

$$\tau_r^* = \frac{\tau_r}{\sum_{j=\overline{1,m}}\tau_j} \qquad (3)$$

Слой 4. Количество узлов четвертого слоя также равно m. Каждый узел соединен с одним узлом третьего слоя, а также со всеми входами сети (на рис. 1 связи с входами не показаны). Узел четвертого слоя рассчитывает вклад одного нечеткого правила в выход сети:

$$y_r = \tau_r^* \cdot \left(b_{0,r} + b_{1,r}x_1 + \cdots + b_{n,r}x_n\right). \qquad (4)$$

Слой 5. Единственный узел этого слоя суммирует вклады всех правил:

$$y = y_1 + \cdots y_r \ldots + y_m. \qquad (5)$$

Типовые процедуры обучения нейронных сетей могут быть применены для настройки ANFIS-сети так как, в ней использует только дифференцируемые функции. Каждая итерация процедуры настройки выполняется в два этапа. На первом этапе на входы подается обучающая выборка, и по невязке между желаемым и действительным поведением сети итерационным методом наименьших квадратов находятся оптимальные параметры узлов четвертого слоя. На втором этапе остаточная невязка передается с выхода сети на входы, и методом обратного распространения ошибки модифицируются параметры узлов первого слоя. При этом найденные на первом этапе коэффициенты заключений правил не изменяются. Итерационная процедура настройки продолжается пока

невязка превышает заранее установленное значение. Для настройки функций принадлежностей кроме метода обратного распространения ошибки могут использоваться и другие алгоритмы оптимизации, например, метод Левенберга-Марквардта.

C. Метод опорных регрессий

Поддержка векторной регрессии (SVR) – это версия известнейшего метода опорных векторов (SVM) [6]. Модель SVR была предложена в [7], это метод, применяемый к случаю регрессии. Подобно SVM, алгоритм SVR использует нелинейные отображения, называемые ядрами, для преобразования входного пространства в пространство с высокой размерностью. Он создает регрессионную модель с использованием подмножества обучающих примеров, называемых вспомогательными векторами [8]. В методе используется глобальный параметр «который должен изучить функцию f (x), которая не превышает «отклонения от цели путем определения полосы вокруг функции регрессии ». Другой глобальный параметр, обозначаемый C, управляет компромиссом между ошибкой предсказания и плоскостностью полосы вокруг f (x (t)). Наконец, тестовый экземпляр x (t) можно предсказать, используя следующее уравнение:

$$f\big(x(t)\big) = \beta_0 + \sum_i^q \beta_i K\big(x(i), x(t)\big), \text{для } i = 1,2,\dots,q \qquad (6)$$

где: x_i – опорные векторы (точки, которые выходят за пределы или на границе трубки), *x(t)* является тестовым экземпляром, который должен быть предсказан β, является вектором параметров, определяемым SVR, а $K(.,.)$ – это функция ядра, используемая для преобразования входных данных в пространство с более высокой размерностью [6]. SVM применяется в анализе социальных сетей для классификации китайских пользователей Facebook в интровертах и экстравертах на основе их стеновых сообщений на Facebook [9]. SVM также превзошел другие классификации в нескольких сравнительных анализах в контексте социальных сетей [10]. SVR применяется в [1] для прогнозирования показателей производительности социальных сетей [1].

D. Echo state network

Начиная с начала 2000-х годов, Reservoir Computing (RC) приобрела известность в сообществе нейронных сетей [11]. Модель RC имеет динамическую систему, называемую резервуаром, которая расширяет входные данные в высокоразмерное пространство. Затем модель использует контролируемый инструмент обучения для прогнозирования результатов модели. Чаще всего модель RC использует простую линейную регрессию из карты функций и выходного пространства. Модели RC широко используются в таких областях, как: классификация образцов [12], распознавание речи [13], [14], качество речи [15] и предсказание временных рядов [11], [12], [16], [17], [18], Прогнозирование интернет-трафика [19] и т.д. Наиболее формальным является резервуар – функция временного расширения из входного пространства $\mathbb{R}^p$в большее пространство $\mathbb{R}^d$ с $p \ll d$. Обозначим через *x (t)*$\in \mathbb{R}^p$ входную модель в любое время t и *y(t)* - цель модели в любое время t. Рекурсии моделируются вектором состояния $s(t) \in \mathbb{R}^d$

$$s(t) = \varphi(s(t-1), x(t)) \qquad (7)$$

где $\varphi(.)$ – функция разложения.

Обозначим модельные связи с помощью w^{in} и w^r, которые являются матрицами размеров $d \times p$ (для входных весов) и $d \times d$ (для весов резервуара). Характерной чертой модели является то, что эти весовые матрицы фиксированы во время алгоритма обучения [11]. Они случайным образом инициализируются и сохраняются. Чтобы вычислить вывод ESN, соответствующий новому входу *x(t)* (вектор-столбец), модель сначала вычисляет новое состояние резервуара$s(t) = (s_1(t), \dots, S_d(t))^t$, вычисленное по формуле

$$s(t) = \tanh(w^r s(t-a) + w^{in} x(t)) \qquad (8)$$

Вектор w^{out} является единственным регулируемым параметром в модели ESN, который обычно оценивается с использованием регрессии гребня между вектором *[1, s]* и цель.

III. Результаты эксперимента

A. Выборка данных

Набор данных содержит входных 7 функций и 3 выходных переменных, необходимых для определения импакта. Выходными переменными являются: *комментарии, репосты и лайки*. В переменной *комментарии* учитывается количество комментариев, вызвавших конкретную запись. Переменные *репосты* относятся к числу раз, когда сообщение было передано другим пользователям. Переменная *лайки* подсчитывает количество понравившихся, вызванных сообщением.

ТАБЛИЦА I Метрика публикаций в Facebook

Переменная	**Значимость**	**Медиана**	**Режим**	**Ср. Отклонение**	**Максимум**	**Минимум**
Число комментариев	7	3	0	21	372	0
Число лайков	178	101	98	323	5172	0
Число репостов	27	19	13	43	790	0

B. Результаты эксперимента

Обучение ESN проводилось с размером резервуара 25 нейронов и спектральным радиусом 0.5 (оба параметра важны при проектировании модели [10]). Аналогичное число скрытых нейронов имеет модель ELM. Алгоритм SVR, применяемый в этом исследовании, использует функцию ядра Gaussian radial basis (RBF). Ширина

гауссовского ядра была установлена равной 0.1. Полоса, определенная вокруг функции регрессии, была установлена равной 0.1, а C параметр, который контролирует компромисс между ошибкой предсказания и плоскостностью полосы вокруг функции регрессии, был установлен в 1000. Выбор этих значений для параметров была основана на [20]. Обучение ANFIS проводилось с 400 парами данных. ANFIS имеет 7 входных уровней, 1 выходной уровень (для каждой из 3 переменных). В качестве входного элемента функции-члена был выбран метод Гаусса, и мы принимаем 3 члена функции принадлежности для каждого входного слоя. В выходном слое мы выбираем постоянный тип MF. Для тренировочной сети мы используем гибридный метод оптимизации. Этот гибридный метод объединяет метод обратного распространения ошибки с методом наименьших квадратов. Для подготовки модельных параметров потребовалось 2 эпохи обучения. В таблице 1 представлена полученная MSE из трех методов обучения, примененных в этом исследовании. Новые предлагаемые методы в этой статье (ESN и ANFIS) дают лучшие результаты, чем SVR.

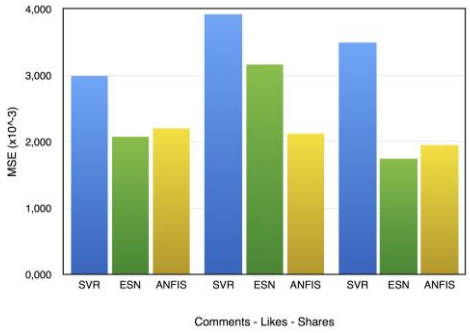

Рис. 2. Средняя квадратичная ошибка прогнозирования

Несмотря на то, что ANFIS, по-видимому, лучше работает для прогнозирования количества лайков, модель ESN имеет лучшую точность в двух других случаях. На рис. 2 представлена иллюстрация наших результатов, где легко сравнивать различную точность среди моделей.

ТАБЛИЦА II РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ

Метод	***Комментарии***	***Лайки***	***Репосты***
SVR	0.002998	0.003917	0.003496
ESN	**0.002068**	0.0033163	**0.001740**
ANFIS	0.0022092	**0.002121**	0.001957

IV. ЗАКЛЮЧЕНИЕ

В этой работе мы показываем точность предсказания трех моделей: SVR, ESN и ANFIS. Мы прогнозировали влияние публикации в социальной сети Facebook. Набор данных содержит 7 входов и 3 выхода, которые используются для пост-воздействия. Выходные переменные: комментарии, репосты и лайки. Новые предлагаемые методы в этой статье (ESN и ANFIS) дают лучшие результаты, чем SVR. В будущей работе мы продолжим эксперименты с использованием модели ANFIS для других наборов данных, чтобы сравнить ее с другими вторыми методами. Мы также планируем протестировать в этой задаче глубинные нейронные сети.

СПИСОК ЛИТЕРАТУРЫ

[1] S. Moro, P. Rita, and B. Vala, “Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach,” Journal of Business Research, vol. 69, no. 9, pp. 3341–3351, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jbusres.2016.02.010

[2] N. Averkin and S. Yarushev, “Hybrid approach for time series forecasting based on anfis and fuzzy cognitive maps,” in Soft Computing and Measurements (SCM), 2017 XX IEEE International Conference on. IEEE, 2017, pp. 379–381.

[3] J.-S. Jang, “Anfis: adaptive-network-based fuzzy inference system,”IEEE transactions on systems, man, and cybernetics, vol. 23, no. 3, pp. 665–685, 1993.

[4] D. Nauck, F. Klawonn, and R. Kruse, Foundations of neuro-fuzzy systems. John Wiley & Sons, Inc., 1997.

[5] A. Averkin, S. Yarushev, I. Dolgy, and A. Sukhanov, “Time series forecasting based on hybrid neural networks and multiple regression.” Springer, 2016, pp. 111–121.

[6] C. Cortes and V. Vapnik, “Support-Vector Networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: http://dx.doi.org/10.1023/A:1022627411411

[7] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support Vector Regression Machines,” Neural Information Processing Systems, vol. 1, pp. 155–161, 1996.

[8] T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Data Mining,” The Mathematical Intelligencer, vol. 27, no. 2, pp. 83–85, 2009. [Online]. Available: http://www.springerlink.com/index/D7X7KX6772HQ2135.pdf

[9] K. H. Peng, L. H. Liou, C. S. Chang, and D. S. Lee, “Predicting personality traits of Chinese users based on Facebook wall posts,” in 2015 24th Wireless and Optical Communication Conference, WOCC 2015, 2015, pp. 9–14.

[10] R. Joshi and R. Tekchandani, “Comparative analysis of twitter data using supervised classifiers,” in 2016 International Conference on Inventive Computation Technologies (ICICT), vol. 3, Aug 2016, pp. 1–6.

[11] M. Luko˘sevi˘cius and H. Jaeger, “Reservoir Computing Approaches to Recurrent Neural Network Training,” Computer Science Review, vol. 3, pp. 127–149, 2009.

[12] H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks,” German National Research Center for Information Technology, Tech. Rep. 148, 2001.

[13] D. Verstraeten, B. Schrauwen, M. D’Haene, and D. Stroobandt, “An experimental unification of reservoir computing methods,” Neural Networks, vol. 20, no. 3, pp. 287–289, 2007.

[14] W. Maass, T. Natschläger, and H. Markram, “Computational models for generic cortical microcircuits,” in Neuroscience Databases. A Practical Guide. Boston, Usa: Kluwer Academic Publishers, June 2003, pp. 121–136.

[15] S. Basterrech and G. Rubino, “Real-time Estimation of Speech Quality Through the Internet using Echo State Networks,” Journal of Advanced in Computer Networks (JACN), vol. 1, no. 3, september 2013.

[16] J. J. Steil, “Backpropagation-Decorrelation: online recurrent learning with O(N) complexity,” In Proceedings of IJCNN’04, vol. 1, 2004.

[17] B. Schrauwen, M. Wardermann, D. Verstraeten, J. J. Steil, and D. Stroobandt, “Improving Reservoirs using Intrinsic Plasticity,” Neurocomputing, vol. 71, pp. 1159–1171, March 2007.

[18] S. Basterrech, C. Fyfe, and G. Rubino, “Self-organizing Maps and Scale-invariant Maps in Echo State Networks,” in 11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Córdoba, Spain, November 22-24, 2011, November 2011, pp. 94–99. [Online]. Available: http://dx.doi.org/10.1109/ISDA.2011.6121637

[19] S. Basterrech and G. Rubino, “Echo State Queueing Network: A new Reservoir Computing Learning Tool,” in 10th IEEE Consumer Communications and Networking Conference, CCNC 2013, Las Vegas, NV, USA, January 11-14, 2013, 2013, pp. 118–123. [Online]. Available: http://dx.doi.org/10.1109/CCNC.2013.6488435

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.