УДК 004.8.032.26

Войцехович Л.Ю., Головко В.А., Рубанов В.С.

НЕКОТОРЫЕ АСПЕКТЫ ПРИМЕНЕНИЯ АЛГОРИТМОВ УСИЛЕНИЯ В МОДУЛЯРНЫХ НЕЙРОННЫХ СЕТЯХ ДЛЯ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

1. Введение. Оперативный обмен информацией становится неотъемлемым атрибутом успешной деятельности в любой сфере. В последнее время прорыв в этой области обеспечили компьютерные технологии: компьютерные сети, электронная коммерция, корпоративные web-сайты и др. Однако, наряду с необходимостью повышения надежности и скорости коммуникации, остро встал вопрос обеспечения защиты информационных ресурсов [1].

Для защиты компьютерных систем применяются различные подходы. Все подходы можно разбить на две основные категории: организационные и технические. В свою очередь технические подходы подразделяются на сетевые и хостовые. Далее в статье речь пойдет о сетевых средствах обеспечения безопасности, а именно, о системах обнаружения вторжений.

Задачей *Систем Обнаружения Вторжений (Intrusion Detection Systems - IDS)* является защита компьютерных сетей.

Наряду с правильной политикой безопасности, архитектурой межсетевых фильтров, антивирусным программным обеспечением и другими средствами IDS часто отводится роль основного элемента защиты. IDS используются в качестве средства раннего оповещения о сетевых проблемах. Это обусловлено размещением IDS в общей схеме обороны на сетевом уровне, на котором подозрительные действия могут быть обнаружены раньше, чем на более высоких уровнях. Кроме того, IDS способна предоставлять необходимые доказательства злоумышленных действий, а также выявлять скрытые тенденции, что становится возможным при анализе большого количества данных, обрабатываемых IDS.

К недостаткам существующих моделей IDS, в первую очередь, можно отнести уязвимость к новым атакам, низкую точность и скорость работы. Современные системы обнаружения вторжений плохо приспособлены к работе в реальном режиме времени, в то время как возможность обрабатывать большой объем данных в реальном времени – это определяющий фактор практического использования систем IDS. Указанные недостатки трудно устранить, используя только классические методы в области компьютерной безопасности. Поэтому в последнее время системы IDS активно изучаются.

Данная статья является продолжением серии статей [2, 3] посвященных разработке системы обнаружения атак на компьютерные сети, решающим элементом в которых является нейронная сеть. Основной задачей в области обнаружения вторжений является задача классификации. В данной статье рассмотрены некоторые статические структуры усиления, которые теоретически позволяют повышать точность и надежность распознавания.

Дополнительным аргументом в пользу использования ансамблей классификаторов в области обеспечения компьютерной безопасности является распространение в последнее время многоядерных процессоров, позволяющих добиться эффективности при параллельной обработке данных.

В разделе 2 вводится понятие ассоциативной машины и рассматриваются некоторые аспекты теории смешения мнений экспертов. В разделе 3 и 4 описываются соответственно алгоритмы усиления Boosting by Filtering и AdaBoost [4, 5]. В разделе 5 выполняется нейропостановка задачи классификации. Основные результаты экспериментов приведены в разделе 6. Выводы сделаны в последнем разделе статьи.

2. Понятие ассоциативной машины. Большинство сложных вычислительных задач можно представить в виде набора простых и небольших подзадач. Объединение решений этих подзадач будет равносильно решению исходной задачи. Решением подзадач занимаются *эксперты (EXP)*, каждый из которых специализируется в своей области, а комбинацию таких экспертов называют *Ассоциативной машиной (Committee Machine)*. Такая машина формирует обобщенное знание из знаний, которыми располагает каждый отдельный эксперт, при чем это интегрированное знание имеет наивысший приоритет.

Ассоциативные машины принято делить на две категории: статические и динамические. Динамические характеризуются тем, что данные на входе непосредственно оказывают влияние на формирование обобщенного решения модели.

В этой статье рассматриваются два метода статических алгоритмов для решения задачи классификации в области обнаружения вторжений. Первый алгоритм – *это алгоритм усиления за счет фильтрации (Boosting by Filtering)*. Второй – *алгоритм адаптивного усиления* (*AdaBoost)*, который является более совершенным алгоритмом усиления и получил свое название благодаря способности адаптивно подстраиваться к ошибкам отдельных экспертов. Оба алгоритма известны достаточно давно.

В целом методы усиления характеризуются использованием нескольких слабых алгоритмов (экспертов), которые обучаются на различных выборках, формируемых по определенным правилам. Для наблюдения феномена усиления предполагается, что точность эксперта должна хотя бы незначительно превышать 50% (т.е. быть немного выше точности угадывания).

Основная трудность заключается в том, чтобы найти оптимальное сочетание параметров отдельного классификатора, с одной стороны и всего ансамбля в целом с другой.

В следующих двух разделах более подробно остановимся на упомянутых выше алгоритмах усиления.

3. Описание алгоритма усиления Boosting by Filtering. В качестве входных данных в алгоритме усиления *Boosting by Filtering* используется обучающее множество (x_1, y_1), (x_2, y_2), …, (x_m, y_m), где x_i - отдельный образ (пример) из некоторого входного пространства образов X, а y_i – значение метки из пространства Y. В данном алгоритме учитывается мнение трех экспертов $t=1..3$.

Алгоритм обучения состоит из следующих шагов:

Алгоритм BOOSTING BY FILTERING:

1. Обучить первый классификатор на множестве *m* примеров.
2. Обучить второй классификатор также на множестве *m* примеров, причем множество примеров для этого этапа формируется в результате процедуры *"подбрасывания монетки"*.
3. Для обучения третьего классификатора используются те примеры, для которых мнение первого и второго классификатора расходятся.
4. Функция голосования для случая двух классов:

$$H(x) = sign\left(\sum_{t=1}^{3} h_t(x)\right). \qquad (1)$$

Первый эксперт обучается на исходном множестве примеров *m*. Обученный первый эксперт используется для фильтрации (filter) второго множества примеров. На этом этапе применяется процедура "подбрасывания монетки". Если выпала "решка", то выбирается следующий пример, который был неправильно распознан первым экспертом. Этот пример включается в обучающее множество второго эксперта. Если выпал "орел", то в обучающее множество для второго эксперта попадает правильно классифицированный на первом этапе пример.

Войцехович Леонид Юрьевич, *магистрант кафедры интеллектуальных информационных технологий Брестского государственного технического университета (БрГТУ).*

Головко Владимир Адамович, *д.т.н., заведующий кафедрой интеллектуальных информационных технологий БрГТУ.*

Рубанов Владимир Степанович*, к.ф.-м..н., доцент, декан факультета электронно-информационных систем БрГТУ.*

Беларусь, БрГТУ, 224017, Беларусь, г. Брест, ул. Московская, 267.

Процедура "подбрасывания монетки" повторяется, пока количество примеров во втором множестве не достигнет значения m.

Таким образом, в результате "подбрасывания монетки" для второго эксперта формируется обучающая выборка, половину которой составляют правильно классифицированные первым экспертом образы, а вторую – неправильно. Производится обучение второго эксперта.

Для третьего эксперта обучающая выборка включает только те примеры, на которых мнение первого и второго эксперта расходятся. Таким образом, третий эксперт позволяет разрешить ситуацию в случае возникновения неопределенности.

На последнем этапе алгоритма Boosting by Filtering обучаем третьего эксперта.

Основным недостатком алгоритма является то, что для его эффективного использования необходимо достаточно большое множество примеров.

4. Описание алгоритма усиления AdaBoost. Алгоритм AdaBoost был предложен Робертом Шепайре (Robert Schapire) и др. в 1995 году, и позволил преодолеть некоторые трудности, возникавшие при использовании более ранних версий алгоритмов усиления. Основными преимуществами AdaBoost (в том числе и по отношению к предыдущей модели усиления) являются:

- использование неограниченного числа экспертов;
- обучение можно производить на одной обучающей выборке, сформированной для всего ансамбля в целом.

В AdaBoost, как и в рассмотренном в разделе 2 алгоритме, используется обучающее множество $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, где x_i - отдельный образ из входного пространства X, а y_i – значение метки из пространства Y. AdaBoost обращается к заданному базовому алгоритму (слабая модель обучения) на каждой отдельной итерации $t=1, 2, \ldots, T$. Основная идея алгоритма заключается в том, чтобы сопоставить каждому образу x_i из обучающей выборки некоторое числовое значение – весовой коэффициент, т.е. задать распределение D на множестве примеров x_i. Весовой коэффициент для образа x_i на итерации t обозначается $D_t(i)$. На первом этапе алгоритма все веса принимают одинаковые значения, но на последующих этапах происходит их модификация, таким образом, что веса неверно классифицированных образов усиливаются. Это позволяет использовать для обучения следующего эксперта образы из обучающей выборки, с которыми у предыдущего эксперта возникали трудности при классификации.

Алгоритм ADABOOST:

1. Дан набор обучающих примеров $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ с метками y_i, $i=1..k$. Начальные веса примеров одинаковы: $D_1 = 1/m$. Шаги 2-4 повторяются для итераций $t = 1, 2, \ldots, T$.
2. Обучить классификатор на распределении D_t, получить классифицирующую гипотезу с ошибкой
$$e_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i). \qquad (2)$$
3. Вычислить
$$a_t = \frac{1}{2}\cdot \ln\left(\frac{1-e_t}{e_t}\right). \qquad (3)$$
4. Обновить распределение примеров
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t}\times\begin{cases} e^{-a_t} & if\,(h_t(x_i)=y_i) \\ e^{a_t} & if\,(h_t(x_i)\neq y_i)\end{cases}, \qquad (4)$$
где Z_t – нормализованная константа суммы D_{t+1} на единицу ($\sum_i D_{t+1}(i)=1$), так что получается нормированное весовое распределение.
5. Сформировать окончательную гипотезу-классификатор
$$H(x) = sign\left(\sum_{t=1}^{T} a_t h_t(x)\right) \qquad (5)$$
для случая двух классов, или
$$H(x) = \underset{y\in\{1,2,\ldots,k\}}{\arg\max}\left[\sum_{t:h_t(x)=y} a_t\right] \qquad (6)$$
для k классов.

Задачей слабой модели обучения является построение гипотезы h_t: $X \to Y$, соответствующей распределению D_t. Эффективность слабой гипотезы можно оценить используя параметр ошибки e_t:
$$e_t = \Pr_{i\sim D_t}[h_t(x_i)\neq y_i] = \sum_{i:h_t(x_i)\neq y_i} D_t(i). \qquad (7)$$

Из формулы видно, что при расчете ошибки учитывается распределение D_t, которое было получено для данной слабой модели обучения. На практике, алгоритм обучения следующей слабой модели может непосредственно использовать весовые коэффициенты образов D_t в процессе вычислений. Если же алгоритм обучения не позволяет этого сделать, то распределение D_t можно учитывать при генерации обучающей выборки для следующего эксперта (например, выбрать образы с максимальными значениями $D_t(i)$).

Когда сформирована слабая гипотеза h_t, в соответствии с алгоритмом AdaBoost рассчитывается параметр a_t. Значение a_t можно интерпретировать как "вес" гипотезы h_t, т.е. достоверность формируемого данным экспертом заключения. Причем $a_t>=0$, если $e_t<=1/2$, и a_t увеличивается по мере уменьшения значения e_t.

Далее пересчитываются значения распределения D_t, по формуле (4). Это позволяет увеличить весовые коэффициенты для тех примеров, которые были неправильно классифицированы гипотезой h_t, и уменьшить вес правильно распознанного примера. Таким образом, веса позволяют выявлять "проблемные" образы, чтобы при обучении следующего эксперта уделить им больше внимания.

Общее решение формируется в результате процедуры "голосования", причем формируется совместная гипотеза H всех T экспертов. В процедуре голосования учитывается вес a_t, присвоенный гипотезе h_t.

В случае алгоритма AdaBoost может наблюдаться эффект переобучения.

5. Реализация алгоритмов усиления на базе нейронной сети. В разделах 2 и 3 приведены общие алгоритмы усиления. Относительно экспертов, применяемых в этих алгоритмах, мы предполагаем только то, что они обеспечивают точность чуть выше точности угадывание (>50%). Более никаких существенных условий на экспертов не накладывается.

Выполним нейросетевую постановку задачи. В этом случае в качестве эксперта будет использоваться искусственная нейронная сеть. Архитектуры нейронных сетей, применяемые нами в области обнаружения вторжений рассмотрены в предыдущих работах. Далее в статье в качестве эксперта будем использовать следующую модель классификации (рис. 1).

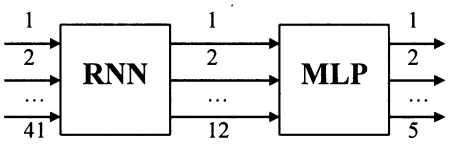

Рис. 1. *Базовая модель классификации (эксперт - EXP)*

Эта система обнаружения атак, состоит из рециркуляционной нейронной сети и многослойного персептрона, которые соединены последовательно. Каждый образ x_i, поступающий на вход модели представляет из себя 41-размерный вектор, который имеет соответствующую метку y_i. На основании значения y_i можно заключить, к какому из 5-ти классов сетевой активности относится входной вектор. В данной модели предопределены следующие классы:

- *Normal* – нормальное состояние сети, соответствует ситуации, когда в компьютерной сети не предпринимаются какие-либо действия, несущие угрозу безопасности.

- *DoS* – отказ в обслуживании, характеризуется генерацией большого объема трафика, что приводит к перегрузке и блокированию сервера.
- *U2R* – предполагает получение зарегистрированным пользователем привилегий локального суперпользователя (администратора).
- *R2L* – характеризуется получением доступа незарегистрированного пользователя к компьютеру со стороны удаленной машины.
- *Probe* – заключается в сканировании портов с целью получения конфиденциальной информации.

Задачей RNN является сжатие входного 41-размерного вектора в 12-размерный выходной вектор, который представляет из себя главные компоненты [6]. Многослойный персептрон осуществляет обработку сжатого пространства входных образов с целью распознавания класса атаки.

Такая модель обнаружения атак в рамках ансамбля классификаторов формирует слабую гипотезу h_t.

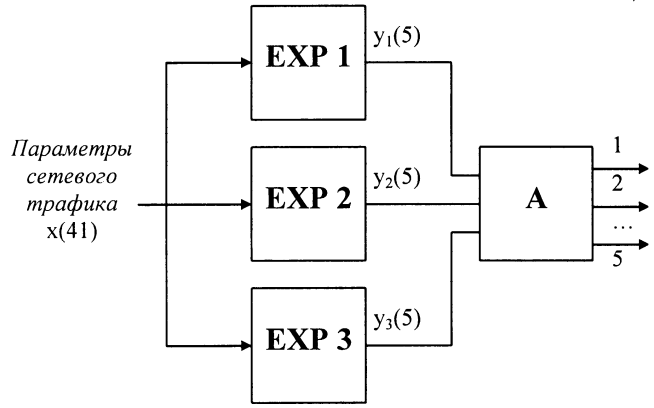

Рис. 2. *Модель алгоритма Boosting by Filtering*

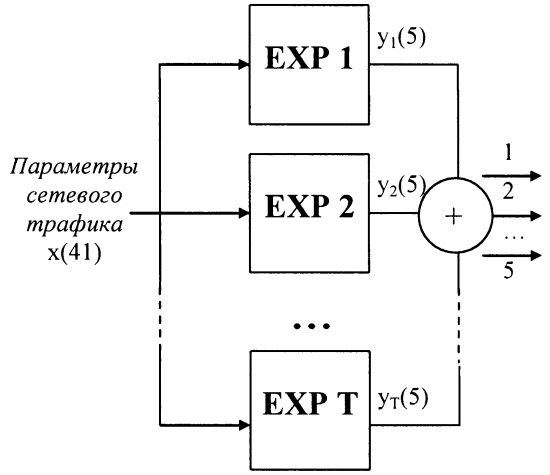

Рис. 3. *Модель алгоритма AdaBoost*

На рис. 2 приведена модель, соответствующая алгоритму усиления Boosting by Filtering. Арбитр осуществляет процедуру голосования и формирует совместное решение H всех трех экспертов. В качестве арбитра применяется двухслойная нейронная сеть.

В алгоритме AdaBoost (рис. 3) роль Арбитра выполняет Сумматор, который рассчитывает результат голосования суммируя частные решения с учетом весовых коэффициентов.

6. Результаты экспериментов. В данном разделе приведены результаты классификации образов сетевых атак с помощью рассматриваемых в статье подходов: i) отдельного эксперта, представляющего из себя последовательно соединенные RNN и MLP, ii) алгоритма Boosting by Filtering и iii) алгоритма AdaBoost.

Эксперименты проводились в соответствии с данными, приведенными в таблице 1.

Таблица 1. *Структура обучающей выборки и тестовых данных*

	DoS	U2R	R2L	Probe	Normal	Итого
обучающая выборка	3571	37	278	800	1500	6186
тестовая выборка	391458	52	1126	4107	97277	494020

Обучающая выборка использовалась для настройки нейронных сетей. После этапа обучения на каждую из предложенных моделей подавались образы из тестовой выборки и рассчитывались показатели эффективности (доля обнаруженных, доля распознанных атак и число ложных срабатываний системы). Результаты отображены в таблицах 2-4.

Таблица 2. *Результаты тестирования модели RNN+MLP*

класс	всего	обнаружено	распознано
DoS	391458	391441 (99.99%)	370741 (94.71%)
U2R	52	48 (92.31%)	42 (80.77%)
R2L	1126	1113 (98.85%)	658 (58.44%)
Probe	4107	4094 (99.68%)	4081 (99.37%)
normal	97277	---	50831 (52.25%)
Итого	494020	---	426353 **(86,30%)**

Таблица 3. *Результаты тестирования модели BOOSTING BY FILTERING*

класс	всего	обнаружено	распознано
DoS	391458	391443 (99.99%)	370663 (94.69%)
U2R	52	50 (96.15%)	42 (80.76%)
R2L	1126	1102 (97.87%)	1086 (96.45%)
Probe	4107	3954 (96.27%)	3939 (95.91%)
normal	97277	---	84728 (87.09%)
Итого	494020	---	460458 **(93,21%)**

При работе с ансамблями классификаторов появляются широкие возможности по настройке параметров нейронных сетей (таких как количество нейронов в скрытых слоях, количество циклов обучения и др.). С одной стороны, это позволяет добиться нужных результатов. Но с другой, значительно усложняет построение модели, поскольку требуется большое количество экспериментов для подбора оптимальной конфигурации.

Как видно из приведенных таблиц, методы усиления позволили уменьшить число ложных срабатываний системы обнаружения атак. Образы, принадлежащие классу normal, представляют достаточно широкий диапазон нормальных соединений в сети, поэтому возникают проблемы с их идентификацией. Кроме того, благодаря подавлению слабых гипотез, такие ассоциативные машины характеризуются большей устойчивостью результатов обучения.

В случае алгоритма AdaBoost рассматривались ситуации с разным количеством экспертов (3, 5, 7 и 9). Общая тенденция такова - в процессе обучения формируется один “ведущий” эксперт или группа таких экспертов, которые хорошо справляются с задачей классификации множества обучающих примеров. Такой эксперт(-ы) характе-

ризуется высоким значением параметра a_t, и, в соответствии с формулой (5, 6), оказывает значительное влияние на формирование общего решения всей модели в целом. Если такой эксперт плохо справляется с задачей обнаружения некоторого класса атак, то это отрицательно сказывается на результатах совместного решения.

Таблица 4 . *Результаты тестирования модели AdaBoost*

класс	всего	обнаружено	распознано
DoS	391458	389917 (99.61%)	369088 (94.29%)
U2R	52	51 (98.08%)	44 (84.62%)
R2L	1126	1119 (99.37%)	636 (56.48%)
Probe	4107	3908 (95.15%)	3668 (89.31%)
normal	97277	---	77212 (79.37%)
Итого	494020	---	450648 **(91,22%)**

Для устранения этого недостатка каждому нейрону выходного слоя эксперта ставилось в соответствие определенное значение q_{ty} из диапазона [0..1] (т.н. коэффициент "доверия"). Этот коэффициент рассчитывался для обученной нейронной сети с использованием примеров из обучающей выборки таким образом, чтобы:

- при $q = 1$, этот нейрон реагировал только на образы своего класса;
- при $q = 0$, на все образы из обучающей выборки.

Общее решение вычислялось по формуле (8), которая является модификацией формулы (6).

$$H(x) = \arg\max_{y \in \{1,2,\ldots,k\}} \left[\sum_{t:h_t(x)=y} a_t \cdot q_{ty} \right]. \qquad (8)$$

Такая методика применима только в случае ансамбля классификаторов.

Заключение. Как показали эксперименты, применение ансамблей классификаторов позволяет улучшить некоторые показатели эффективности модели. Однако это достигается за счет увеличения в разы количества вычислений. Кроме того, применение алгоритмов усиления не всегда приводит к улучшению распознавания определенных классов атак. Наряду с усилением наблюдаются процессы усреднения результата по ансамблю. Поэтому в каждом конкретном случае нужно решать отдельно, стоит ли применять алгоритм усиления.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Web Application Security Consortium. Классификация угроз [Электрон. ресурс]. - Режим доступа: www.webappsec.org.
2. V. Golovko and L. Vaitsekhovich. Neural Network Techniques for Intrusion Detection // In Proceedings of the International Conference on Neural Networks and Artificial Intelligence (ICNNAI-2006) / Brest State Technical University - Brest, 2006. - P. 65-69.
3. Головко В.А., Войцехович Л.Ю. и Шевеленков В.В. Нейросетевые принципы построения нейронных систем обнаружения атак на компьютерные сети // Вестник БрГТУ. Физика, математика, информатика. – 2006. - №5(41). – С. 14-19.
4. H.Drucker, R.Schapire and P.Simard. Improving performance in neural networks using a boosting algorithm // In S.J.Hanson, J.D.Cowan and C.L.Giles eds., Advanced in Neural Information Processing Systems 5, Denver, CO, Morgan Kaufmann, San Mateo, CA. - 1993. - P. 42-49.
5. Yoav Freund, Robert E. Schapire. A short introduction to boosting // Journal of Japanese Society for Artificial Intelligence. – 1999. - №14(5). – P. 771-780.
6. Головко В.А. Нейронные сети: обучение, организация и применение. Кн. 4: Учеб. пособие для вузов / Общая ред. А.И. Галушкина. – М.: ИПРЖР, 2001. – 256 с.

Материал поступил в редакцию 13.02.08

VAITSEKHOVICH L.U., GOLOVKO V.A., RUBANOV V.S. Some aspects of applying boosting algorithms in modular neural networks for intrusion detection

In this article the classification task in the domain of intrusion detection is considered. Often a chosen algorithm is not good enough for practical use. So the question arises how it is possible to improve the performance? In this case we can employ so-called Committee Machines that increase accuracy and reliability of the base classification model. These advantages are the result of dividing complex computational problems among several experts. The knowledge of each expert influences on the general conclusion of Committee Machine.

УДК 681.3 + 004.9

Палий И.О., Саченко А.А., Турченко В.А., Куриляк Ю.О., Капура В.А.

ОБНАРУЖЕНИЕ ЛИЦ С ПОМОЩЬЮ КОМБИНИРОВАННОГО КАСКАДА КЛАССИФИКАТОРОВ ДЛЯ ВИДЕОНАБЛЮДЕНИЯ

Введение. Обнаружение человеческих лиц (ОЛ) – это очень важная и быстроразвивающаяся область исследований, которая всегда является первым этапом к любой обработке лиц и имеет следующие приложения: распознавание лиц, видеоконференции, поиск изображений за содержанием, видеонаблюдение и др. ОЛ также является сложной задачей благодаря таким факторам, как масштаб, размещение, ориентация в пространстве, степень поворота и условия освещения лиц. Много разных подходов ОЛ [1] представлено в последние годы, которые базируются на знаниях, инвариантных признаках, сравнении с шаблоном и внешнем виде.

Сан и Поджио [2] разработали первый точный метод ОЛ, который базируется на внешнем виде. Они предложили несколько алгоритмов, которые потом были использованы во многих других системах ОЛ: нормализация входных изображений, генерация виртуаль-

Палий И.О., *научный сотрудник НИИ интеллектуальных компьютерных систем (НИИ ИКС) Тернопольского национального экономического университета (ТНЭУ), г. Тернополь, Украина.*

Саченко А.А., *доктор технических наук, профессор и зав. кафедрой информационно-вычислительных систем и управления факультета компьютерных информационных технологий и директор Американско-украинской программы по компьютерным наукам, Тернопольского национального экономического университета (ТНЭУ), г. Тернополь, Украина.*

Турченко В.А.*, кандидат технических наук, доцент кафедры информационно-вычислительных систем и управления факультета компьютерных информационных технологий ТНЭУ, глава группы нейронных сетей и параллельных вычислений в НИИ ИКС при ТНЭУ, г. Тернополь, Украина.*

Куриляк Ю.О., *младший научный сотрудник НИИ ИКС, ТНЭУ, г. Тернополь, Украина.*

Капура В.А., *младший научный сотрудник НИИ ИКС, ТНЭУ, г. Тернополь, Украина.*