УДК 519.6

Вестник СибГАУ
2014. № 5(57). С. 115–121

РЕШЕНИЕ ЗАДАЧ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ ПРИ ПОМОЩИ АВТОМАТИЧЕСКИ ГЕНЕРИРУЕМЫХ АНСАМБЛЕЙ НЕЙРОННЫХ СЕТЕЙ

М. Е. Семенкина, Е. А. Попов

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: semenkina88@mail.ru, epopov@bmail.ru

При современном уровне развития компьютерных систем и их взаимосвязей задачи обеспечения информационной безопасности становятся все более актуальными. Автоматизация проектирования детекторов спама, атак на компьютерные сети способна повысить скорость реагирования на вновь возникающие угрозы. Нейронные сети являются одним из наиболее часто применяемых для этих целей подходов, однако их создание является сложной интеллектуальной процедурой. Качество получаемых решений может быть повышено за счет создания ансамблей нейронных сетей. Поэтому автоматизация проектирования ансамблей нейронных сетей при помощи эволюционных алгоритмов способна освободить экспертов в области компьютерной безопасности от необходимости разработки алгоритмического ядра и избежать высоких требований к квалификации конечных пользователей. Однако эффективность применения эволюционных алгоритмов существенно зависит от выбора их настроек, что является сложной задачей даже для специалистов в области эволюционного моделирования. Поэтому для автоматизации настройки эволюционных методов используется самоконфигурация в ходе работы алгоритма. Предлагается использовать самоконфигурируемый алгоритм генетического программирования для создания символьного выражения, учитывающего решения отдельных нейронных сетей из предварительного пула, содержащего 20 нейронных сетей, заранее автоматически сгенерированных при помощи самоконфигурируемого алгоритма генетического программирования. Тестирование предлагаемых алгоритмов выполнялось на репрезентативном множестве тестовых задач и показало высокую эффективность автоматического генерирования ансамблей нейронных сетей на основе самоконфигурируемых эволюционных алгоритмов. Эффективность разработанного подхода была оценена на двух задачах из области компьютерной безопасности, таких как обнаружение спама и выявление PROBE-атак. Проведенное сравнение с альтернативными подходами показало, что рассматриваемый в данной статье метод способен эффективно решать задачи, стоящие перед экспертами в области безопасности компьютерных систем.

Ключевые слова: эволюционные алгоритмы, самоконфигурирование, искусственные нейронные сети, ансамбль, автоматическое генерирование, обнаружение спама, выявление PROBE-атак.

Vestnik SibGAU
2014, No. 5(57), P. 115–121

COMPUTER SECURITY PROBLEMS SOLVING BY AUTOMATICALLY DESIGNED NEURAL NETWORK ENSEMBLES

M. E. Semenkina, E. A. Popov

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: semenkina88@mail.ru, epopov@bmail.ru

Today, computers are becoming more powerful and interconnected that makes their security one of the most important concerns. Conventional security software requires a lot of human effort to identify and work out threats. This human labor intensive process can be more efficient by applying machine learning algorithms. Artificial neural networks are one of the most widely used data mining techniques here. The highly increasing computing power and technology made possible the use of more complex intelligent architectures, taking advantage of more than one intelligent system in a collaborative way. This is an effective combination of intelligent techniques that outperforms or competes to simple standard intelligent techniques. One of the hybridization forms, the ensemble technique, has been applied in many real world problems. In this paper, artificial neural networks based ensembles are used for solving the computer security problems. We apply the self-configuring genetic programming technique to construct symbolic regression formula that shows how to compute an ensemble decision using the component ANN decisions. The algorithm involves different operations and math functions and uses the models providing the diversity among the ensemble members. Namely, we use

neural networks, automatically designed with our GP algorithm, as the ensemble members. The algorithm automatically chooses component ANNs which are important for obtaining an efficient solution and doesn't use the others. Performance of the approach is demonstrated with test problems and then applied to two real world problems from the field of computer security – intrusion and spam detection. The proposed approach demonstrates results competitive to known techniques. With the approach developed an end user has no necessity to be an expert in the computational intelligence area but can implement the reliable and effective data mining tool.

Keywords: evolutionary algorithms, self-configuration, artificial neural networks, ensemble, automated design, spam and intrusion detection.

Введение. В современном мире компьютерные системы становятся все более мощными и вместе с тем все более взаимосвязанными, что делает обеспечение компьютерной безопасности одной из наиболее важных задач. Все более разнообразные атаки несут все больше опасности, начиная от нежелательных сообщений по электронной почте, которые могут обмануть пользователей и тем самым предоставить доступ к компьютеру опасным вирусам, которые могут уничтожить данные или нарушить функционирование компьютерных систем. Обычно для обеспечения безопасности прилагается большое количество ресурсов для выявления угроз и выработки методов противодействия. Этот процесс, требующий привлечения экспертов, может быть более эффективным в случае применения алгоритмов автоматического обучения [1]. Существует множество исследований, связанных с применением методов интеллектуального анализа данных для решения конкретных задач компьютерной безопасности, например, обнаружение вторжения [2–5] или обнаружение спама [6]. Искусственные нейронные сети (ИНС) являются одним из наиболее широко используемых методов интеллектуального анализа данных ([6–10]) в системах компьютерной безопасности.

Увеличение вычислительных мощностей и развитие технологий сделали возможным использование более сложных архитектур интеллектуальных информационных систем, использующих более одной интеллектуальной технологии в процессе решения задачи. При эффективном сочетании интеллектуальных технологий результаты превосходят или сопоставимы с результатами отдельных интеллектуальных информационных технологий.

Один из видов гибридизации – ансамбль – применялся при решении многих реальных задач. Было отмечено, что разнообразие членов, составляющих ансамбль, играет важную роль в данном подходе [11]. Для поддержания разнообразия среди членов ансамбля были предложены различные методы, такие как алгоритмы, работающие на различных наборах функций [12], или обучение на различных подмножествах (например, бэггинг [13] и бустинг [14]). Некоторые методы, такие как нейронные сети, могут быть обучены на одних и тех же наборах данных, однако иметь разнообразные структуры [15]. Простое усреднение, взвешенное усреднение, принятие решений большинством голосов и ранжирование являются общими методами, обычно применяемыми для расчета выхода ансамбля.

Йоханссон и др. [16] использовали алгоритм генетического программирования (ГП) [17] для построения ансамбля из предопределенного числа ИНС, где функциональное множество состояло из операций усреднения и умножения и в терминальное множество вошли модели (т. е. ИНС) и константы. В работе [18] был предложен подобный подход, в котором сначала генерируется определенное количество нейронных сетей, а затем применяется алгоритм ГП, создающий символьное выражение для учета мнений отдельных членов ансамбля.

В этой статье применяется самоконфигурируемый алгоритм генетического программирования для построения формулы, которая показывает, как вычислить решение ансамбля с использованием решений отдельных нейронных сетей. Алгоритм включает в себя различные операции и математические функции и использует различные модели для поддержания разнообразия среди участников ансамбля. А именно, в качестве участников ансамбля используются нейронные сети, автоматически сгенерированные с помощью алгоритма генетического программирования. Алгоритм автоматически выбирает ИНС, которые являются важными для получения эффективного решения и не использует другие.

При использовании разработанного алгоритма конечные пользователи не должны быть экспертами в области интеллектуального анализа данных, но смогут получать надежные и эффективные решения. Это делает подход очень полезным для специалистов в области компьютерной безопасности, так как освобождает их от дополнительных усилий при применении интеллектуальных информационных технологий (ИИТ) и позволяет им сосредоточить свое внимание в области их экспертизы, т. е. компьютерной безопасности как таковой.

Самоконфигурируемый алгоритм генетического программирования. Прежде чем предложить использовать алгоритм генетического программирования конечным пользователям, например специалистам по компьютерной безопасности, для разработки инструментов анализа, необходимо освободить их настройки эволюционных алгоритмов, что является проблемой даже для экспертов в эволюционных вычислениях. Выбор параметров и настроек алгоритма генетического программирования является действительно трудоемким процессом, поэтому необходимо предложить способ для снятия этой проблемы.

Применяемый в данной работе алгоритм использует динамическую адаптацию вероятностей применения операторов на уровне популяции и централизованную методику управления [19; 20]. Чтобы избежать сложностей при настройке вещественных параметров, используется дискретное множество вариантов

настроек, а именно, типы селекций, скрещивания и уровней мутации (средний, низкий, высокий). Каждый из типов операторов имеет свое собственное распределение вероятностей. Например, есть 5 видов селекции: пропорциональная, ранговая, турнирная с тремя размерами турнира. Во время инициализации все вероятности равны 0,2, а в ходе выполнения алгоритма они будут меняться в соответствии с особым правилом таким образом, что сумма вероятностей будет всегда равна 1 и ни одна вероятность не сможет быть меньше, чем заранее определенное значение.

Когда алгоритм создает следующее поколение, сначала необходимо сформировать список операторов с помощью распределения вероятностей применения операторов. Затем алгоритм выбирает родителей при помощи выбранного оператора селекции, генерирует потомков при помощи выбранного оператора скрещивания, мутирует с выбранным уровнем мутаций и сохраняет потомков в промежуточной популяции. Когда промежуточная популяция заполняется, вычисляется пригодность и обновляется распределение вероятностей оператора в соответствии с эффективностью оператора. Эффективность оператора есть отношение средней пригодности потомков, полученных при помощи этого оператора, и средней пригодности всей популяции потомков. Победивший оператор увеличивает вероятность своего применения за счет остальных операторов. После этого формируется родительская популяция. Алгоритм останавливается при выполнении критерия остановки. Данный алгоритм будем называть самоконфигурируемым алгоритмом генетического программирования (SelfCGP).

Так как общепринятый набор тестовых задач для алгоритмов генетического программирования является «открытым вопросом» [21], были использованы задачи символьной регрессии с 17 тестовыми функциями из [22] для предварительной оценки. Результаты тестирования [23] продемонстрировали, что в среднем надежность SelfCGP по 17 тестовым функциям выше, чем усредненная лучшая надежность стандартного алгоритма генетического программирования. Кроме того, SelfCGP превосходит стандартный алгоритм генетического программирования и по расходу вычислительных ресурсов. Основное преимущество самоконфигурируемого алгоритма генетического программирования заключается в отсутствии необходимости выбора параметров алгоритма без каких-либо потерь в производительности, что делает данный алгоритм удобным для применения конечными пользователями, не являющимися экспертами в области эволюционного моделирования.

Автоматическое генерирование искусственных нейронных сетей при помощи самоконфигурируемого алгоритма генетического программирования. Обычно алгоритмы генетического программирования работают с представлением хромосомы в виде дерева, определяемого функциональным и терминальным множествами, и используют специальные операции преобразования решений (селекция, скрещивание, мутация и др.) до тех пор, пока не будет выполнено условие критерия остановки [17].

Для автоматического создания искусственных нейронных сетей в функциональное множество были включены 16 функций активации, таких как биполярная сигмоида, однополярная сигмоида, линейная функция, пороговая функция, гауссиан и др. Функциональное множество включает в себя специальные операции для постановки нейронов и групп нейронов в различные слои и создания связей между ними.

Алгоритм генетического программирования формирует деревья, из которых получаются структуры нейронных сетей. Обучение нейронных сетей проводят для оценки их пригодности, которая зависит от точности аппроксимации или количества неправильно классифицированных случаев. Все весовые коэффициенты нейронной сети настраиваются с помощью самоконфигурируемого генетического алгоритма (SelfCGA) [24], работающего аналогично SelfCGP, не требующего усилий конечных пользователей для настройки алгоритма. Лучшее решение, найденное алгоритмом генетического программирования, дает структуру нейронной сети, которая дополнительно настраивается при помощи самоконфигурируемого генетического алгоритма, гибридизированого с локальным спуском.

Сравнение эффективности нейронных сетей, созданных при помощи самоконфигурируемого алгоритма генетического программирования, выполнялось с альтернативными методами на множестве задач из [25]. Материалы для сравнения были получены из [26], где вместе с результатами авторского алгоритма (CROANN) были приведены результаты 15 других подходов при решении трех задач классификации (ирисы, рак, диабет) из [25].

Анализируя результаты сравнения, можно заметить, что эффективность рассмотренного подхода достаточно высока по сравнению с альтернативными алгоритмами (1-е, 3-е и 4-е места из 15 соответственно). Однако главное преимущество предложенного подхода заключается в отсутствии необходимости для конечного пользователя в экспертных знаниях о нейросетевом моделировании и эволюционных методах. Дополнительным преимуществом является размер получаемых нейронных сетей, которые содержат немного нейронов на скрытом слое и связей между ними, а кроме того, используют не все доступные входы задачи, осуществляя таким образом автоматический отбор наиболее информативных признаков.

Можно заключить, что самоконфигурируемый алгоритм генетического программирования является подходящим инструментом для автоматического создания нейронных сетей, который можно использовать для генерирования ансамблей нейронных сетей.

Объединение ансамблей нейронных сетей и самоконфигурируемого алгоритма генетического программирования. Имея подходящий инструмент для автоматизированного проектирования ИНС, который не требует усилий пользователя для своей настройки, можно применить алгоритм SelfCGP для построения формулы вычисления решения ансамбля с использованием решений отдельных нейронных сетей. Алгоритм генетического программирования для символьной регрессии (построения аналитических

выражений) включает в себя различные операции и математические функции и использует нейросетевые модели, обеспечивающие разнообразие среди участников ансамбля. В численных экспериментах в качестве участников ансамбля использовались нейронные сети, автоматически разработанные алгоритмом SelfCGP. Алгоритм из заданного набора автоматически выбирает ИНС, которые являются важными для получения эффективного решения, и не использует другие. Члены будущего ансамбля выбираются из предварительного пула, который включает в себя 20 ИНС, созданных заранее с помощью SelfCGP. Для проектирования каждой ИНС наборы данных были случайным образом разделены на три части, а именно, обучающая выборка для отдельных ИНС (60 %), проверочная выборка для отдельных ИНС (20 %) и тестовая выборка для ансамблей ИНС (20 %).

Сначала было выполнено сравнение эффективности метода построения ансамбля на основе SelfCGP с альтернативными подходами. Для этого были использованы те же три задачи из [25] и одна реальная задача анализа данных о прогнозировании деградации солнечных батарей (SAD) из [27]. Усредненные по 20 прогонам результаты представлены в табл. 1. Вторая строка табл. 1 содержит эффективность отдельной нейронной сети, сгенерированной при помощи SelfCGP. Числа в первых трех столбцах представляют собой ошибку классификации, вычисленную по формулам из [26]. В последнем столбце для задачи прогнозирования деградации солнечных батарей космического аппарата приведено относительное отклонение от истинных значений.

Результаты в табл. 1 показывают, что метод построения ансамбля ИНС на основе SelfCGP превосходит как стандартные методы построения ансамблей, так и отдельные нейронные сети, сгенерированные с помощью SelfCGP.

Генерирование ансамблей нейронных сетей для решения задач компьютерной безопасности. Убедившись в обоснованности выбранного подхода, проверим его эффективность при решении сложных задач из области компьютерной безопасности.

Первой задачей является обнаружение PROBE-атак. Соответствующее множество данных “KDD’99 Cup” взято из репозитория машинного обучения [25]. Для оценки эффективности подхода все примеры, относящиеся к PROBE-атакам, были помечены как принадлежащие к первому классу, остальные принадлежат ко второму классу. В ходе экспериментов использовались только следующие атрибуты: 1, 3, 5, 8, 33, 35, 37, 40. Выбор этих атрибутов был выполнен эмпирически на основе анализа литературы, их описание можно найти в [28]. Результаты сравнивались с альтернативными подходами из [29]. Результаты сравнения представлены в табл. 2 ниже.

Из табл. 2 можно заключить, что детектор атак, автоматически сгенерированный самоконфигурируемым алгоритмом генетического программирования для создания ансамблей ИНС, демонстрирует высокую эффективность, сравнимую с лучшими известными результатами (PSO-RF и RF). Классификатор на основе отдельных ИНС, сгенерированных с помощью SelfCGP, также демонстрирует вполне конкурентоспособную эффективность.

Второй задачей является обнаружение спама в электронной почте. Соответствующий набор данных был также взят из [25]. Этот набор данных включает в себя 4600 примеров сообщений, как являющихся спамом, так и не являющихся. Данные содержат 57 атрибутов, два из которых являются целыми числами, а остальные – вещественными. Детектор должен отделить спам от не спама.

Результаты альтернативных подходов для сравнения были взяты из [30], где было выполнено сравнение эффективности многоуровневых персептронов (MLP) и метода создания ансамбля на основе многоуровневых персептронов AdaBoost (Boost). Результаты нашего подхода были усреднены по 20 прогонам. Сравнение результатов приведено в табл. 3.

По табл. 3 можно увидеть, что ансамбли ИНС, автоматически сгенерированные при помощи SelfCGP, превосходят все другие подходы, второе место занимает ансамбль принятия решения большинством голосов на основе нейронных сетей, сгенерированных при помощи SelfCGP. Более того, отдельные нейронные сети, сгенерированные при помощи SelfCGP, превосходят альтернативные подходы (Boost, RL).

Таблица 1

Сравнение методов ансамблирования

Классификаторы	Ирисы (% ошибок)	Рак (% ошибок)	Диабет (% ошибок)	SAD (относительное отклонение)
SelfCGP+ANN+Ensemble	**0**	**0**	**17,18**	**0,0418**
SelfCGP+ANN	0,0133	1,05	19,69	0,0543
Ансамбль ИНС с взвешенным усреднением	0,0267	1,03	19,03	0,0503
Ансамбль ИНС с простым усреднением	0,0267	1,09	19,75	0,0542

Таблица 2

Сравнение эффективности алгоритмов при решении обнаружения PROBE-атак

Классификаторы	Доля обнаружения, %	Доля ложноположительных результатов, %
PSO-RF	**99,92**	**0,029**
SelfCGP+ANN+Ensemble	**99,79**	**0,027**
Random Forest	**99,80**	0,100
SelfCGP+ANN	98,78	0,097
Bagging	99,60	0,100

Окончание табл. 2

Классификаторы	Доля обнаружения, %	Доля ложноположительных результатов, %
PART (C4.5)	99,60	0,100
NBTree	99,60	0,100
Jrip	99,50	0,100
Ансамбль с голосованием большинством голосов	99,41	0,043
Ансамбль с взвешенным усреднением	99,17	0,078
Ансамбль с простым усреднением	99,18	0,122
BayesNet	98,50	1,000
SMO (SVM)	84,30	3,800
Logistic	84,30	3,400

Таблица 3

Сравнение эффективности детекторов спама

Детекторы	Ошибка, %
SelfCGP+ANN+Ensemble	5,04
Ансамбль с принятием решения большинством голосов	5,23
Ансамбль с взвешенным усреднением	5,33
Ансамбль с простым усреднением	5,43
SelfCGP+ANN	5,43
Boost	6,48
RL	7,41
MLP	8,33

Заключение. Автоматическое проектирование ансамблей ИНС на основе SelfCGP позволяет повысить эффективность анализа данных. Полученные результаты подтверждены при решении двух реальных задач из области компьютерной безопасности.

Вычислительные усилия для реализации описанного подхода и сложность получаемой модели выше по сравнению с какой-либо отдельной интеллектуальной информационной технологией. Тем не менее это обычный недостаток любой техники ансамблирования, что компенсируется более высокой эффективностью. Действительно дополнительный расход вычислительных ресурсов вызван необходимостью запуска алгоритма генетического программирования, генерирующего формулу учета мнений отдельных ИНС. Однако этот дополнительный расход значительно меньше, чем требуется для генерирования даже одной ИНС, т. е. не может рассматриваться как серьезный недостаток в условиях применения (т. е. проектирования) десятков ИНС. В то же время, эксперименты показывают, что SelfCGP никогда не включает в ансамбль все отдельные ИНС, содержащиеся в исходном наборе, а выбирает лишь несколько из них. Так как большая часть вычислительной сложности ансамбля состоит из вычислительных усилий, необходимых для расчета выхода для каждой отдельной ИНС, то предложенный подход имеет преимущество по сравнению с обычными методами ансамблирования, которые применяют все доступные отдельные ИНС для последующего усреднения или голосования. Напомним также, что конечный пользователь не должен быть экспертом в области вычислительного интеллекта, но при этом может использовать надежный и эффективный инструмент интеллектуального анализа данных. Все это позволяет сделать вывод, что разработанный в данном исследовании инструмент очень полезен для специалистов по компьютерной безопасности, так как освобождает их от дополнительных усилий по разработке и реализации алгоритмического ядра для создания интеллектуальных информационных технологий.

Дальнейшее развитие этого подхода направлено на расширение его функциональности путем включения других видов ИИТ (систем на нечеткой логике, деревьев решений, нейро-нечетких систем, других видов ИНС и т. д.).

Библиографические ссылки

1. Machine Learning and Data Mining for Computer Security / M. Maloof (ed.). Springer, 2006.

2. Victoire T. A., Sakthivel M. A. Refined Differential Evolution Algorithm Based Fuzzy Classifier for Intrusion Detection // European Journal of Scientific Research. 2011. Vol. 65. No. 2. P. 246–259.

3. Bloedorn E. E., Talbot L. M., DeBarr D. D. Data Mining Applied to Intrusion Detection: MITRE Experiences // Machine Learning and Data Mining for Computer Security: Methods and Applications. London : Springer, 2006.

4. Julisch K. Intrusion Detection Alarm Clustering // Machine Learning and Data Mining for Computer Security Methods and Applications. London : Springer, 2006.

5. Patcha A., Park J.-M. An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends // Computer Networks. 2007.

6. Özgür L., Güngör T., Gürgen F. Spam Mail Detection Using Artificial Neural Network and Bayesian Filter // Intelligent Data Engineering and Automated Learning – IDEAL 2004 : Lecture Notes in Computer Science. 2004. Vol. 3177. P. 505–510.

7. An intrusion detection system based on neural network / C. Han [et al.] // Proceedings of Mechatronic Sci-

ence, Electric Engineering and Computer (MEC). 2011. P. 2018–2021.

8. Analysis of ANN-based Echo State Network Intrusion Detection in Computer Networks / S. Saravanakumar [et al.] // International Journal of Computer Science and Telecommunications. 2012. Vol. 3, No. 4. P. 8–13.

9. Network intrusion detection system: A machine learning approach / M. Panda [et al.] // Intelligent Decision Technologies. 2011. Vol. 5(4). P. 347–356.

10. A Comparative Analysis of Artificial Neural Network Technologies in Intrusion Detection Systems / S. Pervez [et al.] // Proceedings of the 6th WSEAS International Conference on Multimedia, Internet & Video Technologies. 2006. P. 84–89.

11. Dietterich T. G. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization // Machine Learning. 2000. Vol. 40, No. 2. P. 139–158.

12. Ho T. K., Hull J. J., Srihari S. N. Decision combination in multiple classifier systems // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1994. Vol. 16, No. 1. P. 66–75.

13. Breiman L. Bagging predictors // Machine Learning. 1996. Vol. 24 (2). P. 123–140.

14. Friedman J. H., Hastie T., Tibshirani R. Additive logistic regression: a statistical view of boosting // Annals of Statistics. 2000. Vol. 28, No. 2. P. 337–374.

15. A learning algorithm for neural network ensembles / Navone H. D. [et al.] // Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. 2001. No. 12. P. 70–74.

16. Building Neural Network Ensembles using Genetic Programming / U. Johansson [et al.] // International Joint Conference on Neural Networks. 2006.

17. Poli R., Langdon W. B., McPhee N. F. A Field Guide to Genetic Programming [Электронный ресурс]. URL: http://www.gp-field-guide.org.uk.

18. Bukhtoyarov V., Semenkina O. Comprehensive evolutionary approach for neural network ensemble automatic design // Proceedings of the IEEE World Congress on Computational Intelligence. 2010. P. 1640–1645.

19. Gomez J. Self-Adaptation of Operator Rates in Evolutionary Algorithms // GECCO 2004, LNCS. 2004. Vol. 3102. P. 1162–1173.

20. Meyer-Nieberg S., Beyer H.-G. Self-Adaptation in Evolutionary Algorithms // Parameter Setting in Evolutionary Algorithm. 2007. P. 47–75.

21. Open issues in genetic programming / M. O'Neill [et al.] // Genetic Programming and Evolvable Machines. 2010. Vol. 11. P. 339–363.

22. Real-parameter black-box optimization benchmarking 2009 / S. Finck [et al.] // Presentation of the noiseless functions. Technical Report Researh Center PPE. 2009.

23. Semenkin E., Semenkina M. Self-configuring genetic programming algorithm with modified uniform crossover // IEEE Congress on Evolutionary Computation (CEC'2012). 2012. P. 1918–1923.

24. Semenkin E., Semenkina M. Self-Configuring Genetic Algorithm with Modified Uniform Crossover Operator // ICSI 2012. LNCS. 2012. Vol. 7331, Part 1. P. 414–421.

25. Frank A., Asuncion A. UCI Machine Learning Repository [Электронный ресурс]. Irvine, CA : University of California, School of Information and Computer Science, 2010. URL: http://archive.ics.uci. edu/ml.

26. Yu J. J. Q., Lam A. Y. S., Li V. O. K. Evolutionary Artificial Neural Network Based on Chemical Reaction Optimization // IEEE Congress on Evolutionary Computation (CEC'2011). 2011.

27. Bukhtoyarov V., Semenkin E., Shabalov A. Neural Networks Ensembles Approach for Simulation of Solar Arrays Degradation Process // Hybrid Artificial Intelligent Systems : Lecture Notes in Computer Science. 2012. Vol. 7208. P. 186–195.

28. Cost-based Modelling for Fraud and Intrusion Detection: Results from the JAM Project / S. Stolfo [et al.] // Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX '00). 2000.

29. Malik A. J., Shahzad W., Khan F. A. Binary PSO and random forests algorithm for PROBE attacks detection in a network // IEEE Congress on Evolutionary Computation. 2011. P. 662–668.

30. Dimitrakakis C., Bengio S. Online Policy Adaptation for Ensemble Classifiers // IDIAP Research Report 03–69. 2006.

References

1. Maloof M. (ed.) Machine Learning and Data Mining for Computer Security. Springer. 2006.

2. Victoire T. A., Sakthivel M. A Refined Differential Evolution Algorithm Based Fuzzy Classifier for Intrusion Detection. *European Journal of Scientific Research*, 2011, vol. 65, no. 2, p. 246–259.

3. Bloedorn E. E., Talbot L. M., DeBarr D. D. Data Mining Applied to Intrusion Detection: MITRE Experiences. *Machine Learning and Data Mining for Computer Security: Methods and Applications.* London: Springer, 2006

4. Julisch K. Intrusion Detection Alarm Clustering. *Machine Learning and Data Mining for Computer Security Methods and Applications.* London, Springer, 2006.

5. Patcha A., Park J.-M. An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. *Computer Networks*, 2007.

6. Özgür L., Güngör T., Gürgen F. Spam Mail Detection Using Artificial Neural Network and Bayesian Filter. *Intelligent Data Engineering and Automated Learning – IDEAL 2004. Lecture Notes in Computer Science*, 2004, vol. 3177, p. 505–510.

7. Han C., Li Y., Yang D., Hao Y. An intrusion detection system based on neural network*Proceedings of Mechatronic Science, Electric Engineering and Computer (MEC)*, 2011, p. 2018–2021.

8. Saravanakumar S., Mohanaprakash T. A., Dharani R., Kumar C. J. Analysis of ANN-based Echo State Network Intrusion Detection in Computer Networks. *International Journal of Computer Science and Telecommunications*, 2012, vol. 3, no. 4, p. 8–13.

9. Panda M., Abraham A., Das S., Patra M. R. Network intrusion detection system: A machine learning approach. *Intelligent Decision Technologies*, 2011, vol. 5(4), p. 347–356.

10. Pervez S., Ahmad I., Akram A., Swati S. U. A Comparative Analysis of Artificial Neural Network Technologies in Intrusion Detection Systems. *Proceedings of the 6th WSEAS International Conference on Multimedia, Internet & Video Technologies*, 2006, p. 84–89.

11. Dietterich T. G. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 2000, vol. 40, no. 2, p. 139–158.

12. Ho T. K., Hull J. J., Srihari S. N. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994, vol. 16, no. 1, p. 66–75.

13. Breiman L. Bagging predictors. *Machine Learning*, 1996, vol. 24 (2), p. 123–140.

14. Friedman J. H., Hastie T., Tibshirani R. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 2000, vol. 28, no. 2, p. 337–374.

15. Navone H. D., Granitto P. M., Verdes P. F., Ceccatto H.A. A learning algorithm for neural network ensembles. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 2001, no. 12, p. 70–74.

16. Johansson U., Lofstrom T., Konig R., Niklasson L. Building Neural Network Ensembles using Genetic Programming. *International Joint Conference on Neural Networks*. 2006.

17. Poli R., Langdon W. B., McPhee N. F. A Field Guide to Genetic Programming. Available at: http://www.gp-field-guide.org.uk.

18. Bukhtoyarov V., Semenkina O. Comprehensive evolutionary approach for neural network ensemble automatic design. *Proceedings of the IEEE World Congress on Computational Intelligence*, 2010, p. 1640–1645.

19. Gomez J. Self-Adaptation of Operator Rates in Evolutionary Algorithms. *GECCO 2004, LNCS*, 2004, vol. 3102, p. 1162–1173.

20. Meyer-Nieberg S., Beyer H.-G. Self-Adaptation in Evolutionary Algorithms. *Parameter Setting in Evolutionary Algorithm*, 2007, p. 47–75.

21. O'Neill M., Vanneschi L., Gustafson S., Banzhaf W. Open issues in genetic programming. *Genetic Programming and Evolvable Machines*, 2010, vol. 11, p. 339–363.

22. Finck S., et al. Real-parameter black-box optimization benchmarking 2009. *Presentation of the noiseless functions. Technical Report Researh Center PPE*. 2009.

23. Semenkin E., Semenkina M. Self-configuring genetic programming algorithm with modified uniform crossover. *IEEE Congress on Evolutionary Computation (CEC'2012)*, 2012, p. 1918–1923.

24. Semenkin E., Semenkina M. Self-Configuring Genetic Algorithm with Modified Uniform Crossover Operator. *ICSI 2012. LNCS*, 2012, vol. 7331, part 1, p. 414–421.

25. Frank A., Asuncion A. UCI Machine Learning Repository. Available at: http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science, 2010.

26. Yu J. J. Q., Lam A. Y. S., Li V. O. K. Evolutionary Artificial Neural Network Based on Chemical Reaction Optimization. *IEEE Congress on Evolutionary Computation (CEC'2011)*. 2011.

27. Bukhtoyarov V., Semenkin E., Shabalov A. Neural Networks Ensembles Approach for Simulation of Solar Arrays Degradation Process. *Hybrid Artificial Intelligent Systems. Lecture Notes in Computer Science*, 2012, vol. 7208, p. 186–195.

28. Stolfo S., Fan W., Lee W., Prodromidis A., Chan P. Cost-based Modelling for Fraud and Intrusion Detection: Results from the JAM Project. *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX '00)*. 2000.

29. Malik A. J., Shahzad W., Khan F. A.: Binary PSO and random forests algorithm for PROBE attacks detection in a network. *IEEE Congress on Evolutionary Computation*, 2011, p. 662–668.

30. Dimitrakakis C., Bengio S. Online Policy Adaptation for Ensemble Classifiers. *IDIAP Research Report 03-69*. 2006.