УДК 004.056.53

Informatics and Mathematical Methods in Simulation
Vol. 4 (2014), No. 2, pp. 149-157

# ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ФОРМИРОВАНИЯ СЦЕНАРИЕВ ПОВЕДЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

**В.М. Рувинская, А.В. Молдавская**

Одесский национальный политехнический университет
просп. Шевченко, 1, Одесса, 65044, Украина; e-mail: amme4od@mail.ru

Статья посвящена проблеме автоматизированного формирования сценариев, описывающих поведение вредоносных программ. Приведены пути её поэтапного решения с помощью методов машинного обучения, обоснован выбор конкретных методов. Предложено представление сценариев в виде байесовских сетей. Применимость этого решения продемонстрирована на примере. Также описаны требования к обучающей выборке, выполнение которых на этапе сбора информации позволит увеличить точность работы методов машинного обучения. Проведены эксперименты, демонстрирующие повышение качества результата при повышении качества выборки.

**Ключевые слова:** вирусология, антивирус, сценарии, машинное обучение, ядерные функции, байесовские сети, структурное обучение.

## Введение

Воздействие вредоносных программ на сегодняшний день является одной из основных угроз для компьютерных систем. Появление и распространение новых угроз происходит крайне быстро [1]. Вследствие этого быстрое обнаружение и идентификация новых вредоносных программ является основной задачей разработчиков антивирусного программного обеспечения. Быстрое и эффективное решение этой задачи с использованием исключительно статических методов, таких как ручной или автоматизированный сигнатурный анализ, сегодня невозможно по причине активного использования авторами вирусов техник, затрудняющих этот анализ (обфускация, полиморфизм, шифрование кода и т.д.). Однако одновременно с этим, поведение многих новых вредоносных программ остаётся типичным: отправка данных пользователя на сторонние адреса, генерация трафика, саморассылка и самокопирование [2]. В связи с этим растёт интерес к динамическим методам анализа, в частности, к методам поведенческого анализа вредоносных программ. Использование их совместно со статическими позволяет более полно и глубоко исследовать образцы угроз. Благодаря этому разработчики антивирусного программного обеспечения могут своевременно обнаруживать новые угрозы и вносить соответствующие изменения в антивирусные базы и каталоги вредоносных программ, предотвращая дальнейшее распространение этих новых угроз.

Работа посвящена вопросам автоматизированного формирования сценариев поведения вредоносных программ на основе обучения по примерам. Применение таких сценариев для обнаружения угроз было впервые предложено в [3]. Там же приведены примеры сценариев и предложен аппарат для их описания на основе регулярных выражений. Согласно [3], сценарий представляет собой модель представления знаний, описывающую обобщенное поведение как последовательность действий с учетом

альтернатив, циклов, а также иерархий с использованием регулярных выражений. Задание сценариев поведения вредоносных программ вручную – очень трудоёмкая задача. Поэтому *целью* нашего исследования является выбор методов машинного обучения, которые подходили бы для формирования описаний поведения вредоносных программ по примерам. Для этого следует решить такие *задачи*:

- изучить и проанализировать существующие методы машинного обучения, выбрать наиболее подходящие;
- описать требования к входным данным, выдвигаемые этими методами, в контексте наблюдения вредоносных программ;
- опробовать выбранные методы, оценить результат.

## Обзор методов машинного обучения, подходящих для формирования сценариев

Задачу формирования сценария с помощью машинного обучения можно условно разделить на два этапа:

- получение списка событий сценария;
- получение связей между событиями.

Метод получения списка событий должен позволять обобщать действия различных вредоносных программ, носящие сходный характер. Очевидно, что перед нами, по сути, стоит задача классификации либо кластеризации. То есть методом получения списка событий может стать любой метод классификации большого количества данных. Однако нужно учитывать, что мы будем иметь дело с достаточно большим объёмом данных, которые также могут быть зашумленными вследствие наличия не совсем типичных вредоносных программ. Для этого придётся применять достаточно новые и развитые методы, а именно методы машинного обучения на ядерных функциях, адаптированные для классификации именно по большим объёмам прецедентов (так называемые ядерные методы) [4]. Они специально предназначены для классификации сильно перемешанных данных и хорошо справляются с зашумленными данными.

## Ядерные методы обучения

Методы обучения на ядрах, также известные как ядерные машины (kernel machines), применяются в решении задач классификации и кластеризации. Идея методов состоит в том, чтобы представить пространство классифицируемых данных $X$, для которых не существует линейного разделителя, в пространстве более высокой мерности $Y$, где такой разделитель можно будет построить. Для осуществления этого необходимо вычислить для каждого вектора данных функцию нелинейного отображения $\Phi : x \to y$, где $x$ – элемент пространства исходных данных $X$ (более низкой размерности), а $y$ – его проекция на пространство $Y$ (более высокой размерности). Сложность такого вычисления возрастает пропорционально увеличению мерности пространства. Кроме того, мерность пространства $Y$ может быть достаточно велика, и в таком случае $\Phi(x)$ вычислить невозможно. Решить данную проблему можно с помощью так называемого ядерного трюка. Для вычисления $\Phi$ используется скалярное произведение двух векторов, для которого требуется учитывать мерность пространств. Но возможно также заменить его специальной ядерной функцией $K(x, y)$, которая обладает свойствами скалярного произведения. В результате сама функция $\Phi(x)$ и, соответственно, мерность пространства перестают играть роль в вычислениях: теперь достаточно вычислить $K(x, y)$. То есть, использование ядерных функций позволяет использовать в пространстве $Y$ те же алгоритмы, что для пространства $X$

без необходимости учитывать мерность $Y$. Алгоритмы, использующие этот приём, а вместе с ним и функцию $\Phi(x)$, и называют ядерными.

Для построения ядерной функции $\Phi(x)$ в общем виде достаточно следовать теореме Мерсера, приведенной в [5], которая определяет необходимые свойства функции. Там же приведены правила порождения функции. Также новые функции получают, осуществляя линейные комбинации и композиции существующих. При этом каждая конкретная функция оптимально подходит только для определённого ряда задач. Это связано с разнотипностью разброса данных в исходном пространстве в различных задачах, что нужно учитывать при переносе данных в пространство более высокой мерности.

После выбора ядра требуется решить задачу подбора числовых параметров ядерной функции. Существуют методы автоматического подбора параметров. Примерами наиболее точных являются довольно затратный метод поиска по сетке (grid search) и симплекс-метод Нелдера-Мида.

Важно различать ядерные функции и методы классификации или кластеризации с их использованием (ядерные машины): в один и тот же метод может быть внедрена произвольная ядерная функция. Достаточно, чтобы к этим методам был применим ядерный трюк.

Наиболее типовым и широко применяемым примером ядерных машин, используемых для классификации, являются машины опорных векторов (supporting vector machines, SVM). Метод опорных векторов осуществляет классификацию через отделение классов друг от друга гиперплоскостями, или линейными разделителями. По обеим сторонам линейного разделителя, или разделяющей гиперплоскости, строится ещё две гиперплоскости таким образом, чтобы полученные в результате классы находились вне отделённой ими области, или зазора. Качество классификации зависит от ширины образовавшегося между классами зазора. Примером метода кластеризации служит ядерный вариант алгоритма k-means. Метод k-means, или метод $k$-средних, находит кластеры, обнаруживая наиболее подходящие их центры, для чего нужно заранее знать количество кластеров. Ядерные функции позволяют применять эти методы в случаях с трудноразделимыми сильно перемешанными данными [6]. Их параметры настраивают таким образом, чтобы оптимизировать точность и гибкость работы метода. Например, константа $c$ в методе SVM влияет на ширину зазора.

## Методы структурного обучения. Применение байесовских сетей для описания сценариев

Получив список событий, следует определить связи между ними, чтобы получить сценарии. Обобщённое описание формата сценария приведено в [3]. Уточним, что каждое событие в этом формате называется целью либо подцелью вредоносной программы. Очевидно, следует построить сеть, связи в которой способны соответствовать необходимым регулярным выражениям. Кроме того, связи должны быть направленными, чтобы отражать причинно-следственные связи между событиями. Наконец, такая сеть должна быть вероятностной, чтобы обеспечить гибкость в принятии решений, которая наиболее важна в обнаружении новых модификаций известных угроз. Из всех применяемых сегодня вероятностных направленных графов, с описаниями некоторых из которых можно ознакомиться в [7], для нас наиболее подходящими являются байесовские сети. Это обусловлено тем, что байесовская вероятность является более гибкой по сравнению с частотной. Она способна обновляться по мере получения новой информации. Как частотная, так и байесовская вероятности оперируют с данными статистики. Однако частотная вероятность использует её лишь как описание параметров, а байесовская – для получения

вероятности того, что определённые параметры (выражаемые случайными переменными) достигнут определённого целевого состояния, причём при условии достижения другими связанными с ними параметрами определённых состояний. Итак, байесовская вероятность выражает причинно-следственные связи, демонстрируя каким образом (с какой вероятностью и при каких своих значениях) причины влияют на следствия [8]. Благодаря этому, её можно успешно использовать в задачах принятия решений (к каковым относится и наша задача). Что касается использования регулярных выражений в сценариях, предлагаемого в [3], то оно окажется заменено отношениями условной вероятности и условной независимости, как будет показано ниже. Условная вероятность – это вероятность выполнения одного события, если было выполнено другое. Условная независимость – это отношение между тремя переменными, в котором одна переменная передаёт информацию о зависимости между двумя другими [9].

Процесс построения байесовской сети с помощью некоторого алгоритма называется структурным обучением. По своему основанию и принципу работы алгоритмы обучения структуре делятся на классы, представленные ниже.

1. Основанные на ограничениях. Основной идеей алгоритмов, использующий данный метод, является проведение анализа условной независимости. В результате определяется, какие узлы связаны, а какие – нет. После этого строится «скелет» сети, то есть определяется структура графа без направлений. На последнем этапе определяются направления дуг. Возможная ошибка алгоритмов этого класса: на выходе могут быть получены неориентированные рёбра вследствие невозможности принятия однозначного решения.

Рассмотрим простейший алгоритм этой категории – алгоритм Петера-Кларка (PC) [10]. Он принимает за основу «наихудший» случай, в котором все переменные связаны со всеми (а потому структура образуется наиболее сложная из возможных). Далее ведётся поиск условно независимых друг от друга переменных, то есть таких, которые связаны между собой другими переменными. Раз связь между ними обеспечена промежуточными узлами, то непосредственная связь должна быть удалена. Благодаря этому будет видно, как первая переменная влияет на вторую с учётом влияния промежуточных переменных. Если же оставить непосредственную связь, эта информация окажется потеряна.

Другими, более сложными, но и более перспективными алгоритмами являются, например, американский SLA (Simple Learning Algorithm) [11] и корейский SCD (Sequential Causal Discover) [12]. Оба они используют свойства и особенности байесовских сети как вероятностного направленного графа.

2. Метрические. Основаны на мере качества, т.е. достоверности сети. Такие алгоритмы находят наиболее вероятную структуру сети, причём вероятность вычисляется на основе специальной метрической функции (иначе называемой метрикой либо функционалом качества). Возможная ошибка алгоритмов данного класса: установление ошибочных связей.

Одной из наиболее широко используемых сегодня метрик является K2 [14, 15]. Она образована из формулы Байеса, в которую в качестве аргументов подставлены исходная база и искомая структура. Суть работы алгоритма на основе данной метрики заключается в следующем. В ходе работы алгоритма для каждой случайной переменной $x_i$ ($i \in [1,n]$, где $n$ – количество переменных) производится оценка всех возможных её родителей и их комбинаций с помощью метрической функции. Наибольшее вычисленное значение функции соответствует наиболее вероятному набору родителей. Для вычисления используется подсчёт в обучающей выборке количества комбинаций всех возможных значений родителей и самой переменной $x_i$. То есть, вероятность того, что переменная $x_j$ ( $j \in [1,n]$ , где $n$ – количество

переменных) является родителем $x_i$, зависит от того, сколько раз встречается одна и та же комбинация их возможных значений в базе. Это объясняется тем, что часто встречающиеся комбинации выражают собой некоторую закономерность, существующую между случайными переменными. Эта закономерность и является искомой связью, то есть частью искомой структуры.

Существуют также метрики, основанные на теории информации. По своему принципу работы они достаточно сильно отличаются от класса байесовых. В них существуют механизмы для поиска наиболее компактной структуры сети, однако вместе с этим данные алгоритмы могут страдать от переобучения, или оверфиттинга (ошибочного обнаружения закономерностей в обучающей выборке, которых в реальности не существует) либо излишней вычислительной сложности. Известные примеры метрик на основе теории информации – информационный критерий Акаике (AIC) и родственный ему информационный критерий Байеса (BIC).

Эксперименты показали [14], что метрики байесовой группы на больших выборках работают успешнее информационных, но проигрывают им на небольших. Алгоритмы же на основе ограничений могут строить сеть не полностью, сталкиваясь с неопределённостями [10], чего с метрическими алгоритмами не происходит. Исходя из сказанного, в нашей задаче предпочтительнее использование метрик на основе теоремы Байеса.

Структурное обучение – ресурсоёмкая задача, поскольку зачастую не удаётся избежать использования так называемого «жадного поиска» нужного графа в пространстве графов либо других вычислительно сложных элементов алгоритма. Сократить объёмы вычислений позволяет предварительное упорядочивание переменных, что указано в спецификациях ряда алгоритмов [11–13]. В результате, поиск родителей и комбинаций родителей для каждой переменной ведётся только среди предваряющих её переменных. Во многих задачах такое требование к исходным данным является неудобным ограничением, требующим вмешательства эксперта, и от него стремятся избавиться. Однако в задаче построения сценария оно превращается в преимущество: поскольку мы наблюдаем следующие друг за другом события, то переменные окажутся естественным образом упорядочены ещё на этапе построения списка переменных (то есть на этапе классификации).

## Требования к качеству обучающей выборки

Успешность построения сети напрямую зависит от качества обучающей выборки, а именно от репрезентативности и зашумленности входных данных. В том случае, когда от зашумленности не удаётся избавиться, можно снизить её влияние, дополнив выборку большим количеством лишённых шума данных. Более сложной задачей является обеспечение достаточной репрезентативности. В общем виде оно достигается тем, что в процессе эксперимента следует добиваться того, чтобы каждая рассматриваемая случайная переменная принимала все свои возможные значения, после чего отследить, как это отразилось на состояниях всех прочих переменных. То есть, следует обеспечить разнообразие результатов. Благодаря тому, что в наблюдаемом поведении или модели существуют скрытые связи (которые и должен выявить алгоритм структурного обучения), это разнообразие позволит прояснить закономерности. Обучающая выборка должна демонстрировать функции вредоносной программы так, чтобы передавать их специфичность и взаимосвязь друг с другом. То есть, при проведении экспериментов следует охватить как можно больше вариантов поведения каждой вредоносной программы. Для этого недостаточно только наблюдения за поведением вредоносной программы в нормальных условиях, где она будет успешно выполнена от начала до конца. Необходимо также обеспечить условия,

в которых активность любой функции вредоносной программы можно прервать, не затрагивая прочих её функций. Тогда, если некоторые другие функции зависят от прерванной, они также не выполнятся, что и будет зафиксировано в результатах эксперимента. Следовательно, будет обнаружена условная независимость между прерванной функцией и всеми прочими.

**Опробование метода построения структуры сценария на основе байесовской сети**

Рассмотрим на тестовом примере, как качество входных данных влияет на точность построения сети. Возьмём выборку по поведению некоторого количества почтовых червей из энциклопедии SecureList от Лаборатории Касперского [15]. Она приведена в таблице 1. В данной выборке значение 1 соответствует выполнению события, значение 0 – невыполнению.

Составленный Лабораторией Касперского [16] на основе жизненного цикла червя сценарий поведения приведен на рис.1.

**Таблица 1.**

Выборка с данными по поведению типовых почтовых червей

| Наименование | Создание копии на диске | Изменение реестра | Сканирование адресной книги | Сканирование файлов | Саморассылка |
|---|---|---|---|---|---|
| Email-Worm.VBS.Challenge | 1 | 1 | 1 | 0 | 1 |
| Email-Worm.Win32.Silver | 1 | 1 | 1 | 1 | 1 |
| Email-Worm.VBS.KakWorm | 1 | 1 | 1 | 0 | 1 |
| Email-Worm.Win32.Mintal.003 | 1 | 1 | 1 | 0 | 1 |
| Email-Worm.Win32.Sober.a | 1 | 1 | 0 | 1 | 1 |
| Email-Worm.JS.Nevezed | 1 | 0 | 1 | 0 | 1 |
| Email-Worm.Win32.Dumaru.b | 1 | 1 | 0 | 1 | 1 |
| Email-Worm.Win32.Gismor | 1 | 1 | 1 | 0 | 1 |
| Email-Worm.JS.Sigbug | 1 | 1 | 1 | 0 | 1 |
| Email-Worm.Win32.Hermes.a | 0 | 0 | 1 | 0 | 1 |
| Email-Worm.Win32.Pepex.a | 1 | 1 | 0 | 1 | 1 |
| Email-Worm.Win32.SysClock | 1 | 1 | 1 | 1 | 1 |
| Email-Worm.Win32.Scorpion | 1 | 1 | 1 | 0 | 1 |

В выборке зафиксировано нормальное поведение 13 наименований червей в условиях, где они не встретили никаких препятствий для своей работы. Попробуем построить байесовскую сеть на основе таких данных с помощью алгоритма структурного обучения. Воспользуемся для этого метрическим алгоритмом К2, так как его результаты работы легче для понимания и демонстрации, чем результаты работы метрических алгоритмов на основе теории информации и алгоритмов на ограничениях. Алгоритм на основе теоремы Байеса К2 вычисляет вероятности всех возможных вариантов структуры сети, что может пригодиться для оценки того, где и насколько ошибся алгоритм в процессе обучения и соответствующим образом скорректировать структуру. Результат работы алгоритма (наиболее вероятная для данных из табл. 1 структура графа) приведен на рис. 2.

Из этого результата следует, что для рассылки копий на другие электронные адреса с зараженного компьютера программе-червю достаточно создать свою копию на жёстком диске этого компьютера. От сканирования же адресной книги или файлов на

диске (откуда на самом деле червь и извлекает адреса для рассылки) это действие якобы не зависит. Очевидно, результат в этом месте содержит ошибку.

Дополним теперь исходную выборку таким образом, чтобы она содержала не только данные о том, как ведёт себя червь в норме, но и о том, что будет происходить с его действиями при невыполнении одного из них. Эти дополнительные данные представлены в таблице 2.

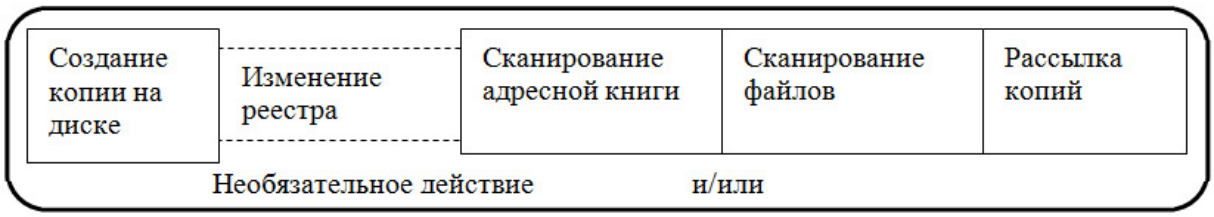


**Рис. 1.** Типичный сценарий поведения почтового червя

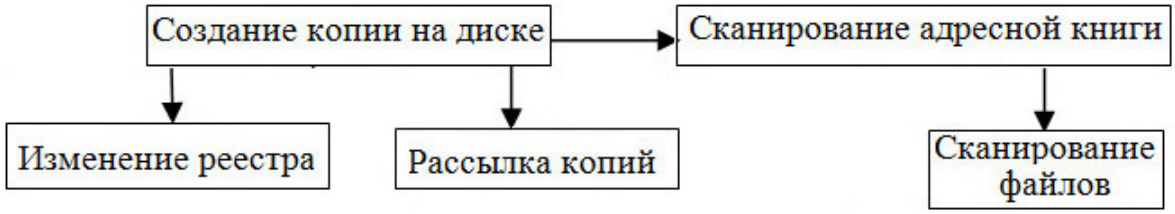


**Рис. 2.** Результат структурного обучения сети на основе первого варианта выборки

**Таблица 2.**

Дополнительные данные о поведении червя с условием прерывания некоторых событий

| Наименование | Создание копии на диске | Изменение реестра | Сканирование адресной книги | Сканирование файлов | Саморассылка |
|---|---|---|---|---|---|
| Email-Worm.VBS.Challenge | 1 | 1 | 1 | 0 | 1 |
| Email-Worm.VBS.Challenge | 1 | 0 | 0 | 0 | 0 |
| Email-Worm.VBS.Challenge | 1 | 1 | 0 | 0 | 0 |

Поясним эти данные. В первой строке мы для наглядности оставили нормальное поведение червя. Во второй предположили, что изменение реестра (а именно, запись червя в автозапуск) не произошло. Поскольку работа червя построена на активации в виде программы автозапуска, невыполнение данного события означает невозможность выполнения всех остальных функций червя. Поэтому все соответствующие события принимают значение 0. В третьей строке предполагаем, что не произошло сканирования адресной книги. Учитывая, что сканирование файлов данный червь не способен выполнить, саморассылка оказывается невозможной и тоже приобретает значение 0.

Таким образом, выборка приобрела больше репрезентативности, то есть более полно описывает предметную область. Результат работы алгоритма К2 с новой выборкой приведен на рис. 3.

Поясним этот результат. Тот факт, что из изменения реестра не следует ни один из следующих узлов, говорит о том, что изменение реестра является необязательным событием в сценарии, что соответствует экспертной версии. Сканирование файлов действительно следует из сканирования адресной книги: если червь не сканировал адресную книгу, вероятность того, что он будет сканировать файлы, повышается, так как из файлов тоже можно извлекать адреса электронной почты. Рассылка копий следует как из сканирования адресной книги, так и из сканирования файлов по той же причине: адреса можно получить из обоих источников. Сходящаяся связь аналогична отношению «и/или». Таким образом, сеть построена верно и полностью соответствует сценарию.


**Рис. 3.** Результат структурного обучения сети на основе второго варианта выборки

## Выводы

Сделан вывод, что использование механизмов машинного обучения для формирования сценариев возможно. Преимущества этого подхода по сравнению с ручным составлением сценариев заключаются в минимизации нагрузки на эксперта при составлении базы знаний для экспертной системы, использующей сценарии. Были описаны особенности работы с ядерными машинами, применяемыми в классификации и кластеризации. Сделан вывод, что они подходят для формирования обобщённых классов поведения вредоносных программ. Рассмотрена возможность построения сценариев с помощью методов машинного обучения, применяемых в обучении байесовских сетей. Представление в виде байесовских сетей позволяет выразить причинно-следственные связи между событиями, а также ветвление, которое часто встречается в сценариях. Было продемонстрировано, как сетевая структура отображает обязательные и необязательные подцели, регулярное выражение «и/или». Кроме того, были разработаны требования к обучающей выборке, после чего было экспериментально подтверждено их влияние на точность результата при выполнении структурного обучения.

В последующих исследованиях может быть рассмотрена возможность представления циклов в сценариях, построенных на ациклических байесовских сетях. Дальнейшим направлением работы станет также углубление описания сценария как структуры данных.

## Список литературы

1. Mansfield-Devine, S. Significant rise in cybercrime against public sector organisations. / Steve Mansfield-Devine. – Computer Fraud & Security, Issue 1. – 2012. – p.1-3.
2. Honig, A. Practical Malware Analysis. / A. Honig. – No Starch Press, 2012. – 800 p.
3. Рувинская, В. М. Эвристические методы детектирования вредоносных программ на основе сценариев / В. М. Рувинская, Е. Л. Беркович, А. А. Лотоцкий // Искусственный интеллект. – 2008. – № 3. – С. 197-207.

4. Gammerman, A. Hedging Predictions in Machine Learning. / A. Gammerman, V.Vovk // Comput. J. 50, 2. – 2007. – p. 51-163.
5. Wang R. Introduction to Orthogonal Transforms. / Ruye Wang //Cambridge University Press. – 2012. – 590 p.
6. Grigorios F. Tzortzis. The global kernel k-means algorithm for clustering in feature space./ G. F. Tzortzis , A. C. Likas // IEEE Transactions on Neural Networks, v.20 – 2009. – n.7. – p.1181-1194.
7. Толпин, Д.А. Вероятностные сети для описания знаний /Д.А. Толпин // Информационные процессы. – М.: Русинвест, 2007. – т. 7, № 1. – с.93-103.
8. Ullman, D.G. Making Robust Decisions. / D.G. Ullman // Trafford Publishing. – 2006. – 348 p.
9. Тулупьев, А. Л. Байесовские сети: логико-вероятностный подход. / А.Л. Тулупьев, С.И. Николенко, А.В. Сироткин. – СПб. : Наука, 2006. – 608 с.
10. Spirtes, P. Causation, Prediction, and Search. / P. Spirtes, C. Glymour & R. Scheines // MIT Press, Adaptive Computation and Machine Learning, second edition. – 2006. – 549 p.
11. Cheng, J. Learning Bayesian Networks from Data: An Information-Theory Based Approach. / J. Cheng, R. Greiner, J. Kelly. //University of Ulster. – 2001. – 74 p.
12. Lee, S. A New Polynomial Time Algorithm for Bayesian Network Structure Learning. / S. Lee, J. Yang, S. Park. // Berlin, ADMA. – 2006. – p.501-508.
13. Cooper, F.G. A bayesian method for the induction of probabilistic networks from data. / F. G. Cooper, E. Herkovits // Stanford University School of Medicine, Stanford – 1993. – 39 p.
14. Carvalho, A.M. Scoring functions for learning Bayesian networks. - Tec. Rep. / Alexandra M. Carvalho // INESC-ID 54/2009. – 2009. – 48p.
15. SecureList - Новые описания детектируемых объектов. [Электронный ресурс] // Режим доступа: http://www.securelist.com/ru/descriptions
16. Вирусы и средства борьбы с ними. Учебный курс. [Электронный ресурс] // Режим доступа: http://www.csid.omsu.omskreg.ru/docs/docs/viruses.pdf

## ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ФОРМУВАННЯ СЦЕНАРІЇВ ПОВЕДІНКИ ЗЛОВМИСНИХ ПРОГРАМНИХ ЗАСОБІВ

О.В. Молдавська, В.М. Рувінська

Одеський національний політехнічний університет,
просп. Шевченка, 1, Одеса, 65044, Україна; e-mail: amme4od@mail.ru

Стаття присвячена проблемі автоматизованого формування сценаріїв, що описують поведінку зловмисних програм. Приведено шляхи її поетапного розв'зання за допомогою сценаріїв, обгрунтовано вибір конкретних методів. Запропоновано представлення сценаріїв за допомогою баєсівських мереж, продемонстровано результати експериментів на вибірках даних різного рівня якості.
**Ключові слова:** зловмисне програмне забезпечення, сценарії, машинне навчання, ядерні функції, баєсівські мережі, структурне навчання

## USING MACHINE LEARNING METHODS TO FORM THE SCENARIOS OF BEHAVIOR OF MALICIOUS SOFTWARE

A.V. Moldavskaya, V.M. Ruvinskaya

Odessa National Polytechnic University,
1 Shevchenko Ave., Odessa, 65044, Ukraine; e-mail: amme4od@mail.ru

The paper is devoted to the problem of computer-aided forming of scenarios, with the latter describing the behavior of malicious software. The means of step by step problem solution using machine learning methods were discussed, and the choice of specific methods was substantiated. Presenting the possible scenarios in the form of Bayesian networks was proposed. Applicability of this approach was exemplified. Moreover, there were described the requirements to learning sample, which, if met at the data acquisition stage, would increase the accuracy of operation of machine learning methods. The experiments were conducted to demonstrate the improvement of the result quality with improvement of sample quality.
**Keywords:** virusology, antivirus, scenario, machine learning, nuclear functions, Bayesian networks, structural learning.