РАЗРАБОТКА НЕЙРОСЕТЕВОЙ И ИММУНОКЛЕТОЧНОЙ СИСТЕМЫ ОБНАРУЖЕНИЯ СЕТЕВЫХ ВТОРЖЕНИЙ

Д. С. Улезло
Факультет математики и информатики, Гродненский государственный университет имени Янки Купалы
Гродно, Республика Беларусь
E-mail: dulezla@gmail.com

В данной статье рассмотрены подходы к решению задач классификации сетевых вторжений путем реализации некоторых методов машинного обучения, связанных с нейронными сетями и именноклеточными системами. Анализируется возможность применения в данной предметной области инструментов и библиотек, используемых на языке программирования Python. Описываются варианты оценки эффективности работы системы.

Введение

С развитием информационных технологий возросла роль информационных систем в таких областях человеческой деятельности, как экономика, медицина, транспорт, государственное управление. Растет и необходимость обеспечения безопасности цифровых данных. Из-за непрерывного развития информационной области и постоянных изменений способов и методов несанкционированного доступа к данным, стандартные технические и программные решения не всегда способны предотвратить или своевременно реагировать на вторжения в систему.

Различные методы машинного обучения, применение искусственного интеллекта могут способствовать повышению качества обнаружения сетевых угроз на ранних стадиях [1].

I. Использование нейронных сетей

Искусственная нейронная сеть представляет из себя набор аналитических методов и математических моделей, взаимодействие которых способно решать конкретные задачи машинного обучения. Нейронные сети – мощный метод моделирования, позволяющий воспроизводить чрезвычайно сложные зависимости. Они нелинейны по своей природе и легко справляются с «проклятием размерности». Нейронные сети учатся на примерах, что позволяет улучшать качество работы со временем [1].

Существует множество готовых решений, реализующих те или иные аспекты машинного обучения и нейронных сетей. В описываемой системе предпочтение отдается языку программирования Python, что позволяет с легкостью манипулировать многомерными сложными данными.

Изначально обучающую выборку необходимо визуализировать, проанализировать и привести к виду, пригодному для обучения нейронной сети: некоторые категориальные признаки заменить набором бинарных, отсеять шумы и малозначительные признаки.

Библиотека scikit-learn используется на Python и позволяет сокращать размерности данных, решает задачи кластеризации, построения регрессии. Начиная с версии 0.18 есть встроенная поддержка нейронных сетей. В пакет включены эффективные методы классификации данных, такие как SVM, метод k-ближайших соседей, лес принятия решений. Сетевые процессы поддаются тщательному анализу и классификации при использовании вышеупомянутых инструментов [2].

II. Использование иммуноклеточных систем

Искусственная иммунная система представляет из себя автоматизированную вычислительную систему, моделирующую упрощенную работу иммунной системы позвоночных. В своей работе использует теории отрицательного отбора, клональной селекции, дендритного алгоритма и иммунных сетей. Реагирует на данные, не имеющие схожие прецеденты в обучающих выборках. Позволяет классифицировать сложные объекты при ограниченном размере обучащей выборки [3].

Клональный алгоритм отбора – класс алгоритмов, основанных на теории клоновой селекции приобретённого иммунитета, которая объясняет, как Б и Т лимфоциты улучшают их реакцию на антигены с течением времени, что называется affinity mutation. Эти алгоритмы сосредоточены на атрибутах теории Дарвина, где выбор основан на близости взаимодействия антигенов и антител и репродукции на принципе деления клеток и и вариациях на основе соматических гипермутаций [4].

Библиотека SOMPY (Self Organizing Map in Python) может быть использована для построение самоорганизующихся карт Кохонена на языке программирования Python. Структура SOM схожа с somtoolbox в Matlab. Для ускорения процессов используется распараллеливание обучения. Использование данного пакета улучшает реакцию системы на новые типы сетевых угроз [5].

III. Оценка эффективности работы системы

Для обучения и проверки качества работы системы используются тестовые данные ресурса KDD 99 Classifier Learning Contest. Для оценки используется средний процент ошибки. Для того, чтобы у разных ошибок были разные веса используются матрицы весов. Искусственно в обучающие и тестовые выборки добавлены шумы.

Существуют и другие различные критерии, позволяющие оценить качество работы алгоритма. Ниже перечислены наиболее важные из них:

- Accuracy – в простейшем случае такой метрикой может быть доля объектов, для которых алгоритм принял правильное решение. Важным недостатком данного подхода к оценке является одинаковый вес для всех объектов. Такой способ оценки может быть некорректен в случае, если распределение объектов в обучающей выборке сильно смещено в сторого нескольких конкретных классов;
- Точность (precision) и полнота (recall) являются метриками, которые наиболее часто встречаются при оценке алгоритмов классификации информации. Точность системы в пределах класс – доля объектов действительно принадлежащих данному классу относительно все данных, которые система отнесла к этому классу. Полнота системы – это доля найденных классификатором объектов, принадлежащих классу относительно всех данных этого класса в тестовой выборке.
- Матрица неточностей (confusion matrix) – матрица размера N на N, где N – количество классов. Столбцы таблицы резервируются за экспертными решениями, а строки за решениями классификатора. Когда мы классифицируем объект из тестовой выборки, мы инкрементируем число стоящее на пересечени строки класса который вернул классификатов и столбца класса к которому действительно относится документ. Позволяет эффективно и удобно рассчитать полноту и точность на практике. При небольшом количестве различных классов, этот подход позволяет наглядно представить результаты работы классификатора [6].

IV. Заключение

Обучение с учителем позволяет эффективно устранять шумы и классифицирует различные типы угроз, но требует большой обучающей выборки и плохо справляется с новыми видами вторжений. Обучение без учителя с меньшей точностью классифицирует объекты, но качественно справляется с новыми типами угроз.

Синтез методологий может быть реализован при помощи инструментов и библиотек языка программирования Python, что позволяет создать качественную системы обнаружения сетевых вторжений.

1. Кадан, А. М., Улезло Д. С. Методы машинного обучения в решении задач информационной безопасности: статья / Кадан, А. М., Улезло Д. С. // Уфа, Труды III Международной конференции «Интеллектуальные технологии обработки информации и управления» – 2015 – С. 41-44.
2. Vollmer, T., Manic M. Manic M. Computationally efficient neural network intrusion detection security awareness. // 2nd International Symposium on Resilient Control Systems – 2009 – P. 25–30.
3. Кадан, А. М., Улезло Д. С. Анализ структуры самоорганизующихся карт Кохонена: статья / Кадан, А. М., Улезло Д. С. // Воронеж, Сборник научных трудов по материалам международной заочной научно-практической конференции «Актуальные направления научных исследований XXI века: теория и практика» 2015 – Т. 3, Ч. 4 — С. 243-246. – ISSN 2308-8877
4. Kephart J. O. Artificial Immune Systems: A New Computational Intelligence Approach // Springer – 2002 – P. 57–58. – ISBN 1852335947
5. Komar M., Golovko V., Sachenko A., Bezobrazov S. Development of neural network immune detectors for computer attacks recognition and classification // IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS) – 2013 – Vol. 2. P. 665–668.
6. Manning C., Raghavan P., Schütze H. An Introduction to Information Retrieval // Cambridge UP – 2009 – P. 581.