УДК 004.056.5:004.72
DOI: 10.31673/2412-9070.2022.0502124

В. О. СОСНОВИЙ, аспірант;
І. В. ЗАМРІЙ, канд. фіз.-мат. наук, доцент,
Державний університет телекомунікацій, Київ

БЕЗПЕКА МЕРЕЖІ З ВИКОРИСТАННЯМ РЕКУРЕНТНОЇ НЕЙРОМЕРЕЖІ

Зростання кількості кібератак та шкідливих програм, що спостерігається останнім часом, яскраво свідчить про те, що наявних контрзаходів проти цього явища все ще недостатньо. Хакери стають дедалі обережнішими у своїх підходах передусім завдяки розробленню все якіснішого програмного забезпечення, насамперед — аби уникнути виявлення.

Відтак, дедалі очевиднішою стає потреба в ефективному автоматизованому вирішенні кібербезпеки, якого можна досягти за допомогою глибинних нейронних мереж.

У статті досліджено ефективність повторюваних нейронних мереж (Recurrent Neural Networcs, RNN) для боротьби в кіберпросторі. Проведений експеримент показує, що RNN з довготривалою короткочасною пам'яттю (Long Short-Term Memory, LSTM) працює набагато краще, ніж класичні алгоритми машинного навчання (SVM і Random Forest) з точністю відповідно 99,70, 98,55 та 99,42%. Це можливо, оскільки RNN мають вбудовану пам'ять, яка може запам'ятати кілька попередніх станів і неявно виокремити характерні риси, сховану складну структуру та комплекс послідовного зв'язку в даних, який допоможе досягти кращої точності.

Ключові слова: кібербезпека; глибинне навчання; рекурентні нейронні мережі (RNN); LSTM; машинне навчання; виявлення шкідливих програм; SVM.

Вступ

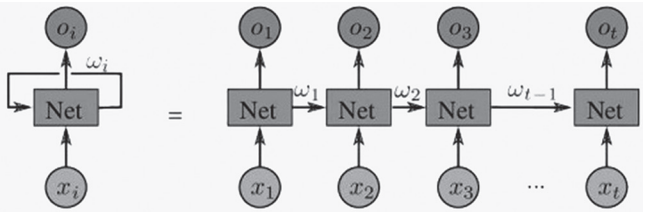
Постановка проблеми. Завдяки прогресу в розробленні алгоритмів машинного навчання підходи глибокого навчання на основі нейронної мережі можна застосувати до безпеки мережі передусім для виявлення нових варіантів шкідливого програмного забезпечення та раніше невідомих атак нульового дня. Глибинне навчання є підмодулем машинного навчання, і його також називають глибокими нейронними мережами (Deep Neural Network, DNN) [1]. Запровадження глибокого навчання в кібербезпеці безперечно допоможе у співвіднесенні подій, визначенні шаблонів і виявленні раніше невідомих атак і аномальної поведінки, щоб зміцнити безпеку/перспективу будь-якої оборонної програми та зменшити рівень невідомих атак. На щастя, сучасне глибоке навчання продемонструвало надзвичайну ефективність у багатьох давніх проблемах штучного інтелекту (ШІ), зокрема оброблення природної мови, комп'ютерному баченні, розпізнаванні мови [2]. Нині підходи до глибокого навчання застосовувалися до різноманітних випадків використання кібербезпеки, починаючи від виявлення вторгнень, аналізу трафіку, аналізу зловмисного програмного забезпечення для Android і мережного шкідливого програмного забезпечення. Завдяки таким підходам є можливість виявляти кібератаки та загрози, вивчаючи складну помітну структуру, приховані послідовні зв'язки та ієрархічні подання функцій із великого набору даних безпеки, передаючи інформацію більш ніж на один прихований рівень [3]. Отже, кібербезпека має здатність застосовувати переваги машинної орієнтації/поглибленого навчання проти збільшення кібератак/загроз і для підвищення рівня виявлення зловмисного програмного забезпечення, сортування подій, підтвердження порушень і попередження організацій про проблеми безпеки.

Аналіз останніх досліджень і публікацій. Останнім часом атаки на мережі у кіберпросторі зростають зі швидкістю, яка випереджає здатність захисників писати та розгортати нові підписи для виявлення цих нових атак, загроз і шкідливих програм. І тому потрібні завчасні заходи безпеки, щоб скоротити або уникнути зростання кількості кібератак і загроз [4]. Хоча є перелік інструментів, зокрема брандмауери, антивірусне програмне забезпечення, системи виявлення вторгнень (Intrusion Detection System, IDS) і системи захисту від вторгнень (Intrusion Prevention System, IPS), які працюють у стеках для захисту від атак і виявлення порушень [5]. Однак зловмисники все ще залишаються небезпечними, оскільки їм достатньо знайти лише одну лазівку в системі, яка потребує захисту. Також через збільшення кількості підімкнених систем до інтернету поверхня атаки теж розширюється, що призводить до зростання ризику атаки. Крім того, зловмисники вдосконалюються, розробляючи методи нульового дня і варіанти зловмисного програмного забезпечення, які обходять заходи безпеки, даючи їм змогу залишатись довше не поміченими [6]. Численні атаки, зокрема атаки D-Dos, Man in the center, втеча інформації, PROBE, User-To-Root, Remote-To-Local тощо використовуються хакерами або противниками для отримання незаконного/несанкціонованого доступу до будь-яких вебсайтів, непублічних мереж і інформації в наших персональних комп'ютерах. Отже, щоб уникнути загроз і атак, розумним та ефективним вирішенням для кіберзахисту буде використання машинного/глибинного навчання.

Формулювання мети статті. Мета написання статті — оцінити ефективність RNN для одного випадку виявлення домену зловмисного програмного забезпечення, створеного з алгоритму генерації домену (Domen Generation Algorithm, DGA — підпрограми, яка активує зловмисне програмне забезпечення з новими доменами на вимогу або на льоту), і порівняти з класичними алгоритмами машинного навчання.

Основна частина

RNN належать до сімейства нейронних мереж, які працюють на послідовних даних. Класична нейронна мережа передбачає, що всі входи та виходи не залежать один від одного. Переважно вхідні дані надходять із двох джерел: одні із сьогодення, а інші з минулого. Попередня інформація зберігається в самоповторюваному циклі, який зазвичай називають рекурентним (рисунок).



Повторювана структура нейронної мережі

Ліворуч на рисунку зображено класичну структуру RNN. Праворуч — розгорнуту версію, де інформація з минулого переноситься на більш пізній часовий крок [7]. Зважаючи на вхідну послідовність $X = (X_1, X_2, ..., X_t)$, функцію переходу для моделі RNN математично можна виразити так:

$$h_t = g_n(w_{xh}X_t + w_{hh}h_{t-1} + b_h), \quad (1)$$

$$o_t = g_n(w_{oh}h_t + b_o), \quad (2)$$

де x_t — вхідний вектор; g_n — нелінійна функція активації; h_t — прихований вектор стану; o_t — вихідний вектор; члени w і b — відповідно ваги та зміщення.

RNN призводить до зникнення градієнта помилок, коли він запам'ятовується для виклику інформації протягом більших часових кроків [8]. Щоб мінімізувати проблему зникнення градієнта, було введено відсікання градієнта. Пізніше було запропоновано LSTM [9]. Він має блок пам'яті, а не простий блок у RNN, який допомагає зберігати інформацію. Блок пам'яті сформовано з комірки пам'яті, яка містить вхідні, вихідні та пропускні елементи. І ворота, і стан клітини забезпечують взаємодію. Основна функція воріт полягає в тому, щоб контролювати інформацію в комірці пам'яті. Ці ворота допомагають мережі LSTM зберігати та запам'ятовувати інформацію довше, ніж RNN.

З огляду на вхідну послідовність $X = (x_1, x_2, ..., x_t)$ функцію переходу для моделі LSTM математично можна подати в такий спосіб:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (3)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (4)$$

$$c_t = t_{an}h(W_c[h_{t-1}, x_t] + b_c), \quad (5)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (6)$$

$$h_t = o_t * t_{an}h(c_t), \quad (7)$$

де x_t — вхідний вектор; h_t — прихований вектор стану; c_t — вектор стану комірки; o_t — вихідний вектор; f_t — забутий вектор стану; члени w і b — відповідно ваги та зміщення.

Зазвичай LSTM є складною мережею. Нещодавно було репрезентовано мінімізовану версію LSTM під назвою Gated Recurrent Unit (GRU). GRU подібний до LSTM, але більш ефективний з погляду обчислень і легший для навчання, ніж LSTM. У GRU функціональність шлюзів забуття та введення, які є в LSTM, поєднується, щоб сформувати шлюз оновлення. Шлюз оновлення зазначає обсяг минулої пам'яті, який має зберігатися в GRU. Також було запропоновано варіант RNN, ідентифікаційний RNN, який ініціалізує відповідну вагову матрицю RNN із застосуванням матриці ідентичності або її масштабованої версії, і використовує Rectified Linear Unit (ReLU) як нелінійну функцію активації для протидії проблемі зникнення вибухового градієнта.

Для класифікації зловмисного програмного забезпечення домену було реалізовано п'ять різних моделей (SVM, Random forest, RNN і RNN із шаровою та пакетною нормалізацією) для порівняння. Набір даних було взято з DGA-Domains-from-data-drivensecurity.info із загальним розміром 1,6 ГБ. В експерименті використовувалася 10-разова перехресна перевірка. Набір даних було розділено на 70/30 відповідно для навчання та тестування. Дані формуються з чотирьох стовпців (хост, домен, клас і підклас). Для цього дослідження використовувалися стовпці хост і клас. Хост містить усі домени, включно з таким доменом верхнього рівня, як .com, .org тощо, тому замість стовпця домену, який не має домен верхнього рівня, використовувався хост. Стовпці класу вважаються цільовими. Він охоплює два класи «dga» і «legit». Перший клас «dga» — це домен зловмисного програмного забезпечення, а другий «legit» — це легальний домен, їх утворено відповідно з 81261 і 52665 параметрів. Для моделей SVM і Random Forest було зроблено виокремлення функцій із даних хосту. Для виокремлення ознак використовувався TF-IDF (частота термінів — зворотна частота документа). Отже, SVM і випадковий ліс отримують витягнуту функцію з TF-IDF як вхідні дані. Форма виокремленої ознаки з кожного зразка дорівнює (39×1). Тож, вхід SVM і Random Forest є (39×1), а вихід — двійковим (0 і 1), тобто 1 для «dga» або домену зловмисного програмного забезпечення та 0 для «законного» домену.

Застосовувалися різні методи підготовки даних для моделей RNN із токенізацією вхідних даних. Дані хосту було доповнено для введення виправлень для моделі та збільшено після доповнення, якщо довжина символів була менша за 55. Максимальна довжина кожної послідовності була 55. Нормалізація шару та пакета використовувалася між повністю підімкненими шарами, а регуляризацію для пришвидшення навчання моделі RNN і випадання (0,001) було взято, щоб уникнути переобладнання. Вхідні дані моделей RNN є послідовністю форм (55×1), а вихідні дані моделей RNN містяться в діапазоні (0, 1), оскільки функція активації є сигмоподібною. Для здобуття остаточних класів використовували такі результати: більш ніж 0,5 як 1 і менш ніж 0,5 як 0.

Метрику оцінки точності було застосовано для оцінювання запропонованої моделі. Це частка загальної кількості прогнозів, які були правильно класифіковані, тобто відношення правильно прогнозованого спостереження до загального спостереження, де частота істинно позитивних (TP) і істинно негативних (TN) правильно класифіковані, тоді як хибно-позитивні (FP) і хибно-негативні (FN) неправильно класифіковані. Досліджувану модель (рекурентні нейронні мережі) було оцінено за класичними класифікаторами машинного навчання, а випадок використання кібербезпеки SVM і випадкового лісу — на основі алгоритму, згенерованого доменом (DGA). Три моделі було навчено та протестовано на одному наборі даних (зловмисне програмне забезпечення домену). Детальні результати запропонованих моделей RNN та інших моделей машинного навчання для сценарію використання подано в таблиці.

Результати тесту

Алгоритми	Task Name	Accuracy, %	Precision	Recall	F-score
SVM	Domain malware classification	0,985	0,699	0,489	0,389
Random forest	Domain malware classification	0,994	0,761	0,992	0,95
RNN with LSTM	Domain malware classification	0,997	1,00	1,00	1,00
RNN (Layer Normalization)	Domain malware classification	0,994	0,98	0,88	0,99
RNN (Batch Normalization)	Domain malware classification	0,946	0,95	0,94	0,94

Висновки

У цій статті описано дослідження моделі рекурентних нейронних мереж (RNN) для безпеки мережі з використанням виявлення зловмисного програмного забезпечення домену як сфери застосування. Продуктивність RNN та інших класичних класифікаторів машинного навчання вивчається й оцінюється для класифікації зловмисного програмного забезпечення в домені використання кібербезпеки та порівнянь. З цього дослідження випливає, що RNN має кращу точність, ніж класичні класифікатори машинного навчання (SVM і Random forest). Це можливо, оскільки RNN мають вбудовану здатність пам'яті, яка може зберігати та відтворювати кілька попередніх станів, а також неявно виокремлювати основні особливості, приховану чи основну складну структуру та складні послідовні зв'язки в даних, які допомагають досягти кращої точності. Отже, це буде актуально для створення програм у режимі реального часу для аналізу шкідливих дій у мережі.

Список використаної літератури

1. ***Le Cun Y., Bengio Y., Hinton G.*** *Deep learning. 2015. Nature 521(7553). 436 p.*

2. ***Vinayakumar R., Soman KP., Prabaharan Poornachandran.*** *A Comparative Analysis of Deep learning Approaches for Network Intrusion Detection Systems (N-IDSs) // International Journal of Digital Crime and Forensics. July 2019.*

3. ***Application*** *of Deep Learning Architectures for Cyber Security / R. Vinayakumar, K. P. Soman, Prabaharan Poornachandran, S. Akarsh // Advanced Sciences and Technologies for Security Applications, 2019. URL:*

https://doi.org/10.1007/978-3-030-16837-7_7

4. ***Devakunchari R., Sourabh, Prakhar Malik.*** *A Study of Cyber Security using Machine Learning Techniques // International Journal of Innovative Technology and Exploring Engineering (IJITEE).*

5. ***A Survey*** *of Deep Learning Methods for Cyber Security Corbett Information / Daniel S. Berman, Anna L. Buczak, Jeffrey S. Chavis, Cherita L. 2019. 10, 122.*

6. ***Mohammed Harun Babu R, Vinayakumar R, Soman KP.*** *RNNSecureNet: Recurrent neural networks for Cyber security use-cases.*

7. ***Lei Tai, Ming Liu.*** *Deep learning in Mobile Robotics- from perception to control systems: ASurvey on Why and Why not // Journal of Latex Class File. August 2015. Vol. 14, No. 8.*

8. ***Hochreiter S., Schmidhuber J.*** *Long short-term memory // Neural Comput. 1997. 9. P. 1735–1780.*

9. ***Sak H. Senior A. W.*** *Processing acoustic sequences using long short-term memory (LSTM) neural networks that include recurrent projection layers. U.S. Patent No. 9,620,108. 11 Apr. 2017.*

V. O. Sosnovyi, I. V. Zamrii

RECURRENT NEURAL NETWORKS IN CYBER SECURITY

The recent increase in cyber-attacks and malware clearly demonstrates that current countermeasures do not seem to be enough to protect against it, as hackers become more cautious in their approach with the cunning of developing systems that automatically rewrite and reorder their malicious software to avoid detection. Typical machine learning approaches that learn a classifier based on a manually created feature vector are not robust enough to such reordering. Hence, the need for an effective automated cyber security solution using deep neural networks. In this article, we demonstrate research on the effectiveness of recurrent neural networks (RNNs) for combat in cyberspace. The conducted experiment shows that RNN with Long Short Term Memory (LSTM) performs much better than classical machine learning algorithms (SVM and Random Forest) with accuracy of 99.70%, 98.55% and 99.42%, respectively. This is possible because RNNs have a built-in memory that can remember multiple previous states and implicitly extract distinctive features, hidden complex structure, and complex sequential relationships in the data, which helps achieve better accuracy. This paper describes an investigation of a recurrent neural network (RNN) model for cyber security using domain malware detection as an application area. The performance of RNN and other classical machine learning classifiers is studied and evaluated for malware classification in the cyber security usage domain and compared. From this study, it can be seen that RNN has better accuracy than classical machine learning classifiers (SVM and Random forest). This is possible because RNNs have a built-in memory capability that can store and replay multiple previous states, and implicitly extract salient features, hidden/underlying complex structure, and complex sequential relationships in the data, which help achieve better accuracy. Thus, it will be useful for creating a real-time application for analyzing malicious activities on the network.

Keywords: cyber security; deep learning; Recurrent Neural Networks (RNN); LSTM; Machine Learning; malware detection; SVM.