УДК 004.056.5 **doi: 10.26906/SUNZ.2023.2.087**

С. Ю. Гавриленко[1], В. Д. Зозуля[1], В. В. Омельченко[2]

[1] Національний технічний університет "Харківський політехнічний інститут", Харків, Україна
[2] Харківський національний університет радіоелектроніки, Харків, Україна

ДОСЛІДЖЕННЯ МЕТОДІВ ПІДВИЩЕННЯ ЯКОСТІ КЛАСИФІКАЦІЇ НА НЕЗБАЛАНСОВАНИХ ДАНИХ

Анотація. **Предметом** дослідження є методи балансування вихідних даних. **Метою статті** є підвищення якості систем виявлення вторгнень у комп'ютерні мережі за рахунок використання методів балансування класів. Завдання: дослідити методи балансування класів та розробити метод класифікації на незбалансованих даних для підвищення рівня мережевої безпеки. **Використовуваними методами** є: методи штучного інтелекту, машинного навчання. Отримано такі результати: Досліджено методи балансування класів, які базуються на технології Undersampling, Oversampling та їх комбінації. Для подальшого дослідження обрано такі методи: SMOTEENN, SVMSMOTE, BorderlineSMOTE, ADASYN, SMOTE, KMeansSMOTE. У якості вихідних даних використано набір UNSW-NB 15, який містить інформацію про нормальне функціонування мережі та під час вторгнень. У якості базового класифікатора використано дерево рішень на основі CART (Classification And Regression Tree) алгоритму. За результатами досліджень отримано, що використання методу SMOTEENN надає можливість підвищити якість виявлення вторгнень у функціонування мережі. **Висновки**. Наукова новизна отриманих результатів полягає у комплексному використанні методів балансування даних та методу класифікації даних на основі дерева рішень для виявлення вторгнень у комп'ютерні мережі, що дозволило зменшити кількість помилок II роду.

Ключові слова: машинне навчання, мережева безпека, балансування даних, системи виявлення вторгнень, Undersampling, Oversampling, SMOTEENN, SVMSMOTE, BorderlineSMOTE, ADASYN, SMOTE, KMeansSMOTE.

Вступ

Системи виявлення вторгнень (СВВ) є одним із механізмів аналізу поведінки комп'ютерної мережі та інструментом моніторингу та спостереження підозрілої активності, аналізу ресурсів мережі, ідентифікації аномальних подій у мережі .

Функціонування мережі характеризується великим обсягом показників, що призводить до наявності труднощів з вибором найбільш інформативних показників та розробки моделей виявлення вторгнень.

Моделі виявлення вторгнень, зазвичай, базуються на процесі класифікації – групуванню об'єктів за певними ознаками та встановленні ієрархії між ними на основі спільних характеристик. Завдання класифікації найчастіше спирається та методи машинного навчання з учителем, та використовує розмічені дані для навчання. Це означає, що якість моделі багато в чому залежить від вихідних даних, на основі яких вона «навчається», в тому числі від збалансованості даних [1].

Проблема незбалансованих даних відноситься до ситуації, коли класи в цільовій змінній не однаково представлені в наборі даних і є однією з найскладніших проблем алгоритмів машинного навчання.

Дисбаланс може бути властивістю проблемної галузі коли присутність одного класу може домінувати над іншими класами. Це може бути пов'язано з тим, що часто неможливо або важко просто зібрати більше зразків міноритарного класу, щоб покращити розподіл класів. При цьому, потрібна модель, щоб класифікувати дані. Прикладами областей, які, зазвичай, оперують незбалансованими класами є: виявлення шахрайства, спаму, аномалій, викидів, відтоку клієнтів, виявлення рідкісних захворювань та ін. Ситуації з незбалансованими даними також досить поширені в кібербезпеці. Так, при виявленні мережних вторгнень кількість доброякісного трафіку на порядки перевищує кількість шкідливого. Іншим прикладом є боротьба з інсайдерськими загрозами, коли кількість даних про звичайну поведінку на порядок перевищує кількість даних про зловмисну поведінку.

Об'єктом дослідження є процес балансування вихідних даних з метою підвищення рівня мережевої безпеки. ***Предметом дослідження*** є методи балансування (семплінгу) вихідних даних.

Постановка проблеми та огляд наукових публікацій

Незбалансовані класифікації створюють проблеми для прогнозного моделювання тому, що більшість класифікаційних алгоритмів машинного навчання були розроблені на основі припущення про кількість прикладів для кожного класу, тобто вони не враховують класового розподілу/пропорції або балансу класів. Це призводить до побудови упереджених моделей із поганою прогностичною ефективністю, особливо для класу меншості. Модель повністю ігнорує клас меншості і маркує всі об'єкти класифікації мітками мажоритарного класу [2].

Крім того, як правило, клас меншості важливіший, і тому проблема більш чутлива до помилок класифікації для меншості, ніж для більшості. Це призводить до різної ціни помилок першого та другого роду при класифікації даних. Особливо це актуально для систем виявлення вторгнень, які, зазвичай, містять невелику кількість прикладів виявлення вторгнень і для яких помилково-негативна класифікація (помилка II роду) може мати фатальні наслідки [3].

Основна мета врівноваження класів полягає в тому, щоб збільшити частоту класу меншості, або зменшити частоту класу більшості. Це робиться для того, щоб отримати приблизно однакову кількість примірників для обох класів.

Один із підходів до вирішення зазначеної проблеми – застосування різних стратегій балансування класів. Виділяють наступні підходи балансування даних: Undersampling, Oversampling та їх поєднання (Over+Undersampling) [4].

У випадку Undersampling видаляють частину прикладів мажоритарного класу. При цьому зберігаються пропорції між класами, але зменшується кількість зразків у класі, який має більшість екземплярів. Використовують такі методи Undersampling: Random undersampling, Condensed nearest neighbors (CNN) undersampling, Tomek Links method, Edited nearest neighbors (ENN),One-sided selection (OSS, combines Tomeks Links and CNN), Neighborhood cleaning rule (NCR, combines CNN and ENN) [5,6]

Random undersampling це техніка, яка балансує дані шляхом випадкового видалення вибірок із основного класу. Недоліком даного методу є можливість втрати важливої інформації в процесі видалення вибірок.

Методи Condensed nearest neighbors (CNN) та Edited nearest neighbors (ENN) засновані на алгоритмі найближчого сусіда, який використовує відстань між точками для визначення того, які об'єкти повинні бути видалені. Алгоритм CNN ітераційно додає до нового набору даних ті об'єкти, які є найближчими сусідами до об’єкту іншого класу. ENN працює, видаляючи з набору даних будь-які зразки, які не узгоджуються з їхніми найближчими сусідами. Цей процес допомагає усунути шум і покращити якість даних шляхом видалення будь-яких викидів або неправильно позначених зразків. ENN часто використовується в поєднанні з іншими методами попередньої обробки даних для покращення загальної продуктивності алгоритмів машинного навчання [7]. Основна відмінність між CNN і ENN полягає в тому, що CNN працює шляхом конденсації класу більшості в репрезентативну підмножину даних, тоді як ENN видаляє приклади з класу більшості, які знаходяться поблизу межі прийняття рішення. Ще одна відмінність полягає в тому, що CNN аналізує дані один раз, тоді як ENN використовує кілька проходів.

Головним недоліком цих алгоритмів, як і більшості методів балансування, є можливість надмірної підгонки навчальних даних, що призводить до перенавчання моделі

Tomek Links (TL) method базується на пошуку пар екземплярів протилежних класів, які є своїми найближчими сусідами та видаляє їх. Це не тільки вирівнює баланс даних, а й робить межі класів більш чіткими і вираженими, що підвищує якість класифікації. Метод зв’язків Томека також використовується для видалення точок даних із шумом.

One-sided selection (OSS) поєднує метод CNN та метод Томека. На першому кроці застосовується правило Condensed nearest neighbors, а на другому видаляються всі мажоритарні спостереження, які беруть участь у зв'язках Томека. Таким чином, видаляються великі згустки мажоритарних спостережень, а потім область простору зі скупченням міноритарних очищається від мажоритарних, які створюють ефект шуму на межах класів і заважають їх розпізнаванню.

Neighborhood cleaning rule (NCR) класифікує усі спостереження за правилом трьох найближчих сусідів (3-NN). Потім видаляються такі приклади мажоритарного класу, які правильно розпізнані або є сусідами міноритарних прикладів або були неправильно класифіковані. Перевага даного підходу в тому, що збільшення області сусідства дає змогу краще «очищати» дані від шумів.

У випадку Oversampling збільшують число екземплярів міноритарного класу. Найбільш поширеними є такі методи: Random Oversampling, алгоритм SMOTE, ASMO, ADASYN[8].

Алгоритм Random Oversampling балансує дані за рахунок дублювання даних міноритарного класу. Це робить розподіл класів більш збалансованим, але все не відображає розподіл класів у реальному світі. Це може негативно вплинути на якість прогнозів моделі, особливо для рідкісних або важливих подій [9].

В основі алгоритму SMOTE лежить ідея генерації деякої кількості штучних спостережень, які були б схожі на спостереження, що є в міноритарному класі, але при цьому не дублювали їх. Для створення нових спостережень вибирають об’єкт міноритарного класу *b* та використовуючи метод найближчого сусіда знаходять *k* його найближчих сусідів *a*. Надалі алгоритм формує нові спостереження, вибираючи випадково місце на прямий між об'єктом *b* та кожним із його сусідів *a*. Завдяки такому підходу штучні спостереження завжди формуватимуться поблизу існуючих об'єктів, але не будуть збігатися з ними. Алгоритм SMOTE дозволяє задавати кількість спостережень, яку необхідно штучно згенерувати. При цьому ступінь подібності прикладів *a* і *b* можна регулювати шляхом зміни числа *k* його найближчих сусідів: чим менше *k*, тим вище буде ступінь подібності. Недоліком цього підходу є те, що алгоритм просто збільшує щільність спостережень у областях скупчення об’єктів. У випадку, коли приклади міноритарного класу розташовані рівномірно, це призводить до перемішування класів, що ускладнює класифікацію [10-12].

Алгоритм ASMO (Adaptive Synthetic Minority Oversampling) є модифікацією алгоритму SMOTE і надає можливість генерувати дані у всіх областях незалежно від скупченості об’єктів.

Ще одним недоліком алгоритму SMOTE є те, що він для кожного прикладу міноритарного класу створює одну і ту ж кількість штучних прикладів. Це не зовсім оптимально, оскільки не всі приклади однаково прості в навчанні. ADASYN є адаптивним алгоритмом, який створює синтетичні зразки для класів, які мають недостатньо представницьких зразків у навчальних даних. Як наслідок, підхід ADASYN покращує навчання щодо розподілу даних двома способами: зменшуючи зміщення, викликане дисбалансом класів, і адаптивно зміщуючи межу класифікаційних рішень у бік складних прикладів.

Алгоритм Borderline SMOTE (Synthetic Minority Oversampling Technique) є також модифікацією алгоритму SMOTE. Він фокусується саме на прикладах класу меншості, які розташовані на лінії розмежування класів. Для таких прикладів логічно генеру-

вати більше штучних спостережень, щоб зробити межу класу більш чіткою. BorderlineSMOTE генерує синтетичні приклади для цих граничних випадків замість надмірної вибірки всіх вибірок класу меншин, тим самим покращуючи продуктивність узагальнення класифікатора [13-14].

Алгоритм KMeansSMOTE спрямований на зменшення шумових синтетичних точок, які генерують інші методи балансування. Спочатку дані кластеризуються та визначаються кластери з високою часткою більше п'ятдесяти відсотків або визначену користувачем) вибірок міноритарного класу. Потім до кожного із обраних кластерів застосовується алгоритм SMOTE та генеруються нові об'єкти. Кількість цих згенерованих балів залежить від розрідженості класу меншості в кластері; чим більше розрідженість, тим більше нових об'єктів.

Алгоритм SVMSMOTE фокусується також на збільшенні об'єктів міноритарного класу вздовж межі розмежування класів. Він контрастує з методом K-Means, який ми бачили раніше, але узгоджується з варіантом Borderline). При цьому, для кожного з опорних векторів ми знаходимо його K-найближчих сусідів і створюємо вибірки вздовж лінії, що з'єднує опорний вектор і найближчих сусідів, використовуючи або інтерполяцію, або екстраполяцію. Якщо менше половини найближчих сусідів належать до класу більшості, ми виконуємо екстраполяцію. Це допомагає розширити територію класу меншості до території більшості. Якщо ні, виконуємо інтерполяцію. Ідея полягає в тому, що оскільки більшість сусідів належать до класу більшості, ми натомість консолідуємо поточну територію класу меншості.

SMOTEENN – це алгоритм, який поєднує в собі алгоритми SMOTE (Synthetic Minority Oversampling Technique) і Edited Nearest Neighbor (ENN) для генерації екземплярів міноритарного класу, а потім для видалення зашумлених даних.

Процес алгоритму SMOTEENN можна описати так:

1. Вибрати випадковим чином об'єкт x_j даних із класу меншості.

2. Обчислити відстань d_i між об'єктом x_j та його k найближчими сусідами.

3. Помножити отримане значення відстані на випадкове число $\alpha \in [0–1]$ та отримати координати нового об'єкту. Додати об'єкт до класу меншості як синтетично-згенерований екземпляр даних.

4. Повтори кроки 1–3, доки не буде досягнуто бажаної пропорції класу меншості.

5. Для видалення зашумлених даних, встановити параметр k – кількість найближчих сусідів (за замовченням k =3).

6. Для кожного об'єкту даних x_j, знайти k -найближчих сусідів серед інших спостережень у наборі даних.

7. Визначити клас більшості для знайдених сусідів m.

8. Якщо клас спостереження x_j та клас більшості m знайдених сусідів спостереження відрізняються, то спостереження x_j та його k -найближчих сусідів видаляються з набору даних.

9. Повторити кроки 6-8 доки не буде досягнуто бажаної пропорції кожного класу.

Перевагою даного алгоритму є можливість збалансувати розподіл класів, одночасно зменшуючи ризик перенавчання.

За результатами аналізу методів балансування класів для подальшого дослідження вибрано такі методи: SMOTEENN, SVMSMOTE, BorderlineSMOTE, ADASYN, SMOTE, KMeansSMOTE.

Дослідження методів балансування класів

У цій роботі для дослідження використано набір даних UNSW-NB 15 , який був створений інструментом IXIA PerfectStorm у лабораторії Cyber Range Австралійського центру кібербезпеки (ACCS) та містить інформацію про нормальне функціонування мережі та під час синтетичних вторгнень. Для захоплення 100 ГБ необробленого трафіку (наприклад, Pcap файлів). використовується інструмент Tcpdump [15].

Модифікований набір даних містить 39 атрибутів, які включають потік між хостами і перевірку мережевих пакетів для розрізнення наявності або відсутності дев'яти типів атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. Кількість записів у навчальному наборі складає 175 341 запис, а в тестовому наборі – 82 332 записи різних типів, атаки та нормального режиму [15].

Для дослідження ефективності використання обраних методів балансування класів, у середовищі Collab Python розроблено їх програмні моделі.

Для оцінки якості моделі до та після виконання балансування класів використано модель дерев рішень на основі CART (Classification And Regression Tree) алгоритму. Дерево рішень – це метод представлення вирішальних правил у визначеній ієрархії, що включає в себе елементи двох типів: вузли (node) і листя (leaf). Вузли містять в собі вирішальні правила і виконують перевірку прикладів відповідно до обраного атрибуту. Вибір атрибута відбувається на підставі критерію Джині:

$$I = \sum_{k=1}^{C} p_k (1 - p_k),$$

де p_k – це частка зразків, що належить до класу C для конкретного вузла.

Основний принцип методу полягає в послідовному, рекурсивному розбитті навчальної множини на підмножини із застосуванням вирішальних правил у вузлах. Для оцінки якості моделі було використано такі характеристики якості моделі: Точність (*Accuracy),* Влучність (*Precision*), повнота (*Recall*) та міра F_1 (F_1 *score*) [16].

Влучність і повнота характеризують різні сторони якості класифікатора:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN},$$

де TP – кількість вірно класифікованих позитивних подій, FP (помилка I роду, false alarm) – кількість невірно класифікованих позитивних подій, FN

(помилка II роду, miss target) – кількість невірно класифікованих негативних подій.

Чим вище влучність, тим менше помилкових спрацьовувань (помилка I роду). Чим вище повнота, тим менше помилкових пропусків (помилка II роду). Повнота демонструє здатність алгоритму виявляти клас взагалі, а Влучність – здатність відрізняти цей клас від інших класів.

Accuracy, на відміну від *Precision* і *Recall*, залежить від співвідношення класів і тому рідко застосовується за умови незбалансованих вибірок.

Міра F1 – поєднує Precision і Recall в агрегований критерій якості і є їх середнім гармонійним:

$$F_1 = \frac{2}{\frac{1}{\Pr ecision} + \frac{1}{\mathrm{Re}\, call}} = \frac{TP}{TP + 0.5(FP + FN)}$$

.

Оскільки нашим завданням є виявлення усіх позитивних прикладів (прикладів вторгнення в систему), то формальна постановка дослідження це максимізація характеристик якості моделі: *Recall* та F_1 *score*.

За результатами дослідження було проаналізовано показники якості моделі на основі дерев рішень тільки для класу шкідливого програмного забезпечення, яке моделює типи атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

Отримано середнє значення показників якості: *Recall* та F_1 *score* за умови використання різних методів балансування класів. Результати досліджень наведено на рис.1-2. Як видно із результатів *Recall* та F_1 *score* на не збалансованих даних є низькою і дорівнює приблизно 61%.

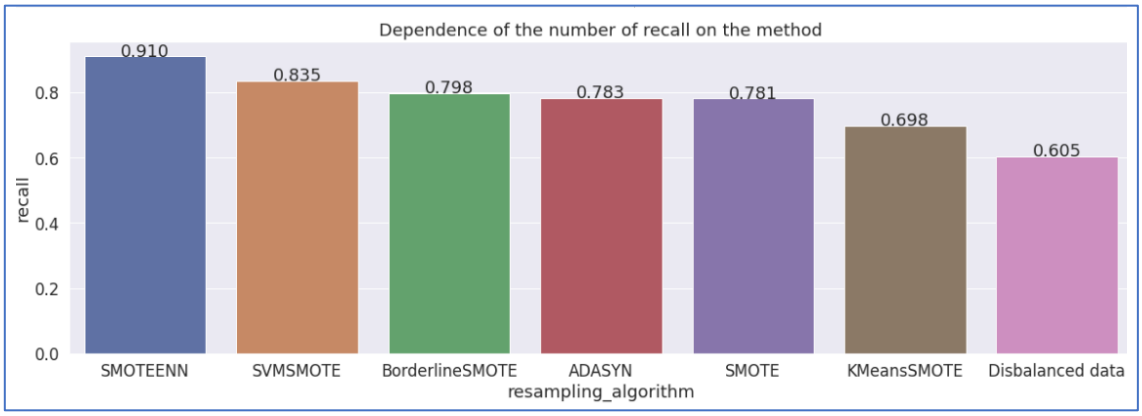

Рис. 1. Залежність повноти класифікації від методу балансування класів

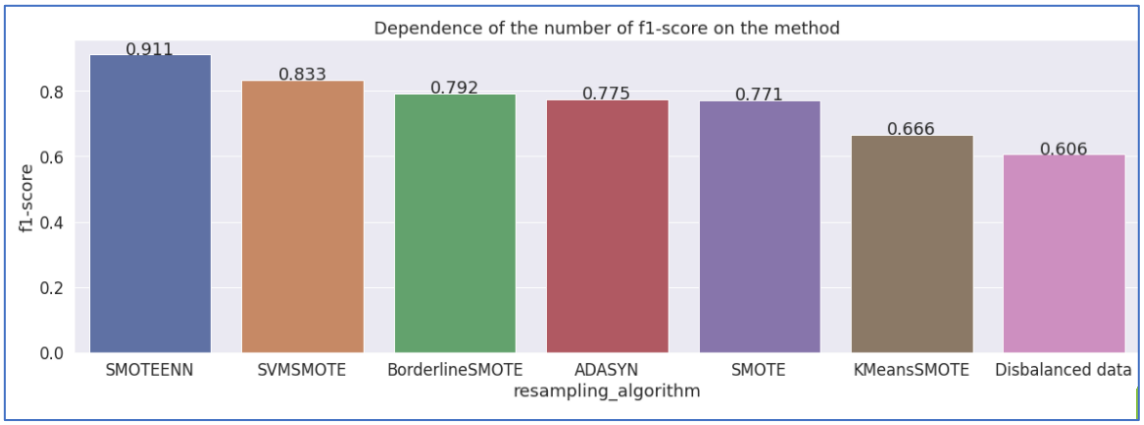

Рис. 2. Залежність міри F1 класифікації від методу балансування класів

У всіх випадках застосування методів балансування дозволило отримати більш високу якість класифікації, ніж на незбалансованих даних. Найкращі результати отримано для методу SMOTEENN. Використання цього методу на етапі попередньої обробки даних для балансування класів дозволило покращити якість на 30 % при вирішенні завдання ідентифікації вторгнень в функціонування комп'ютерних мереж .

Висновки

Таким чином, у даній статті вирішено завдання підвищення рівня мережевої безпеки за рахунок балансування вхідних даних моделі виявлення вторгнень.

Наукова новизна отриманих у даній статті результатів полягає у комплексному використанні методу балансування даних SMOTEENN та методу класифікації даних на основі дерев рішень з метою підвищення якості виявлення вторгнень у комп'ютерні мережі, що дозволило зменшити кількість помилок II роду.

Досліджено методи балансування класів, які базуються на технології Undersampling, Oversampling та їх комбінацій.

Для подальшого дослідження обрано такі методи: SMOTEENN, SVMSMOTE, BorderlineSMOTE, ADASYN, SMOTE, KMeans-SMOTE.

У якості вихідних даних використано набір UNSW-NB 15, який містить інформацію про нормальне функціонування комп'ютерної мережі та під час вторгнень.

Набір даних містить інформацію про потік між хостами і перевірку мережевих пакетів для розрізнення наявності або відсутності дев'яти типів атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

У якості базової моделі класифікації даних використано дерево рішень. Розроблено програмне забезпечення мовою Python у середовищі Collab.

Результати досліджень методів балансування класів показали, що найбільш якісним є використання методів SMOTEENN, SVMSMOTE.

За результатами досліджень отримано, що комплексне використання методу балансування даних SMOTEENN та методу класифікації даних на основі дерев рішень надає можливість підвищити на 30% кількість виявлених вторгнень у функціонування мережі.

СПИСОК ЛІТЕРАТУРИ

1. S. Gavrylenko, V. Chelak, S. Semenov. Development of Method for Identification the Computer System State based on the Decision Tree with Multi-Dimensional Nodes. *Radio Electronics, Computer Science, Control* (*RECSC*). 2022, V.4, pp.113-121.
2. Krawczyk, Bartosz. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 2016, V.5, pp.221-232.
3. C. Wheelus, E. Bou-Harb and X. Zhu. Tackling Class Imbalance in Cyber Security Datasets. *2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA*. 2018, pp.229-232.
4. Abdi L, Sattar H. To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Trans Knowl Data Eng*. 2016, V.28, pp.238–251.
5. Will Badr. Having an Imbalanced Dataset? Here Is How You Can Fix It. [Електронний ресурс] – Режим доступу: https://towardsdatascience. com/ having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb.
6. Jason Brownlee. Cost-Sensitive Learning for Imbalanced Classification. [Електронний ресурс] – Режим доступу: https://machinelearningmastery. com/cost-sensitive-learning-for-imbalanced-classification/.
7. D. L. Wilson. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man, and Cybernetics*. 1972, V.3, pp.408-421.
8. Luque A, Carrasco A, Martin A, Heras de las A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recogn*. 2019, pp.216–231.
9. Batista, Gustavo EAPA, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*. 2004,V.6, pp.20-29.
10. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over sampling technique. *J Artif Intellig Res*. 2002, pp.321–357.
11. Douzas G, Bacao F, Last F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Inf Sci*. 2018, V.465, pp.1–20.
12. Blagus R, Lusa L. SMOTE for High-dimensional class-imbalanced data. *BMC Bioinf*. 2013, V.14, pp.14-106.
13. Fu G.H., Xu F., Zhang B.Y., Yi L.Zh. Stable variable selection of class-imbalanced data with precision-recall criterion. *Chemometrics and Intelligent Laboratory Systems*. 2017, V.171, pp.241-250.
14. Haixiang G., Yijing L., Shang J., Mingyun G., Yuanyue H., Bing G. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications.* 2017, V.73, pp.220-239.
15. Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). *Military Communications and Information Systems Conference (MilCIS)*. 2015, pp.1-6.
16. Douzas Georgios, Fernando Bacao, and Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*. 2018, V. 465, pp.1-20.

Надійшла (received) 27.03.2023
Прийнято до друку (accepted for publication) 23.04.2023

Research of methods for improving the quality of classification on imbalanced data

Svitlana Gavrylenko, Vladislav Zozulia, Viktoriia Omelchenko

Abstract. The subject of the study is methods of balancing raw data. The purpose of the article is to improve the quality of intrusion detection in computer networks by using class balancing methods. Task: to investigate methods of balancing classes and to develop a classification method on imbalanced data to increase the level of network security. The methods used are: methods of artificial intelligence, machine learning. The following results were obtained: Class balancing methods based on Undersampling, Oversampling and their combinations were studied. The following methods were chosen for further research: SMOTEENN, SVMSMOTE, BorderlineSMOTE, ADASYN, SMOTE, KMeansSMOTE. The UNSW-NB 15 set was used as the source data, which contains information about the normal functioning of the network and during intrusions. A decision tree based on the CART (Classification And Regression Tree) algorithm was used as the basic classifier. According to the research results, it was found that the use of the SMOTEENN method provides an opportunity to improve the quality of detection of intrusions in the functioning of the network. Conclusions. The scientific novelty of the obtained results lies in the complex use of data balancing methods and the method of data classification based on decision trees to detect intrusions into computer networks, which made it possible to reduce the number of Type II errors.

Keywords: classification, imbalanced data, data balancing, Undersampling, Oversampling, SMOTEENN, SVMSMOTE, BorderlineSMOTE, ADASYN, SMOTE, KMeansSMOTE.