УДК 004.732.056 **doi: 10.26906/SUNZ.2020.3.075**

С. Ю. Гавриленко, І. В. Шевердін

Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна

ІДЕНТИФІКАЦІЯ СТАНУ КОМП’ЮТЕРНОЇ СИСТЕМИ НА ОСНОВІ АНСАМБЛЕВОГО МЕТОДУ КЛАСИФІКАЦІЇ

Анотація. Предметом статті є дослідження методів ідентифікації стану комп’ютерної системи. **Метою** статті є розробка методу ідентифікації аномального стану комп’ютерної системи на основі ансамблевих методів. **Завдання.** Дослідити та виділити події операційних системах сімейства Windows, розробити програмне забезпечення для виділення подій; дослідити використання ансамблевих класифікаторів на основі беггінгу та бустінгу та дерев рішень для ідентифікації стану комп’ютерної системи. Використовуваними **методами** є: методи машинного навчання та ансамблеві методи класифікації. Отримано такі **результати**: У якості вихідних даних виділено наступні класи подій операційних системах сімейства Windows: події міжпроцесної комунікації, події взаємодії з файловою системою, події інтернет-з’єднання, події взаємодії з реєстром. Досліджено методи ідентифікації аномального стану комп’ютерних систем на базі ансамблевих методів, а саме беггінгу, бустінгу та дерева рішень *J48* для ідентифікації стану комп’ютерної системи. Виконано оцінку ефективності розроблених ансамблевих класифікаторів. За результатами досліджень для ідентифікації стану комп’ютерної системи запропоновано ансамблевий метод класифікації на основі беггінгу та дерева рішень J48. **Висновки.** Наукова новизна отриманих результатів полягає у виділенні процесів функціонування комп’ютерної системи та створенні ансамблевого методу для на основі беггінгу та дерева рішень J48, що надає можливість підвищити точність ідентифікації стану комп’ютерної системи.

Ключові слова: комп’ютерна система, події операційної системи, машинне навчання**,** аномальний стан, дерева рішень, ансамблеві методи класифікації, бустинг, беггінг.

Вступ

Активне впровадження комп’ютерних систем в усі види діяльності суспільства, постійне зростання їх обчислювальної потужності, використання комп’ютерних мереж різного масштабу привели до того, що загрози втрати конфіденційної інформації в системах обробки даних стали невід’ємною частиною практично будь-якої діяльності і потребують захисту. Безпечне функціонування таких систем насьогодні є пріоритетним напрямком і визначає роль державі на міжнародній арені та є актуальним завданням. На сьогодні комп’ютерна система характеризується великим обсягом показників її функціонування. Одним із найбільш поширеним методів аналізу великих обсягів даних (*data mining*) є методи машинного навчання (*machine learning*) які збудовані таким чином, щоб безпосередньо працювати з величезними масивами інформації.

Постановка проблеми та огляд наукових публікацій. В основі машинного навчання лежить інтелектуальний аналіз даних (*Data mining*) – багатоетапний автоматизований ітеративний процес виявлення знань в базах даних, заснований на аналізі величезних масивів інформації з метою виявлення прихованих моделей [1-3]. Для аналізу даних і класифікації використовуються складні математичні алгоритми, що базуються на машинних методах навчання: класичні методи [4], методи навчання з підкріпленням [5], дерева рішень і ансамблеві методи [6, 7], нейромережі і глибоке навчання [8] та ін. Найбільш популярні алгоритми машинного навчання наведено в [9]. Серед них – ансамблеві методи, які базуються на об’єднанні базових класифікаторів [10].

В [11] виконано порівняльне дослідження різних методів побудови ансамблів. Однак в даних роботах не наведено дослідження ефективності використання різних методів побудови дерев рішень у сукупності з різними ансамблевими методами прийняття рішень.

Крім того, наявність різних типів даних, що характеризують стан функціонування комп’ютерної системи потребує їх аналізу у сукупності з дослідженням методів побудови класифікаторів для вирішення завдань ідентифікації стану комп’ютерної системи. **Постановка завдання.** Метою статті є дослідження та розробка методу ідентифікації стану комп’ютерної системи на основі ансамблевих класифікаторів.

Формування вихідних даних

Аналіз подій операційної системи надає можливість оцінити стан комп’ютерної системи. В операційних системах сімейства *Windows* всі події можна поділити на 4 основні типи: події міжпроцесної комунікації, події взаємодії з файловою системою, події інтернет-з’єднання, події взаємодії з реєстром операційної системи [12]. Для збору подій комп’ютерної системи використано програмний додаток “Process Monitor”. Зібрано статистику процесів для нормального та аномального режимів (табл. 1)

Кожний із процесів характеризується наступними атрибутами:

1. Process Name – ім’я процесу;
2. Operation – тип операції (наприклад: (RegOpenKey, CloseFile and etc.);
3. Image Path – шлях до реєстру, наприклад: C:\Users\VirtualUser\Desktop\ZipFileSystemZipper.CSV або HKCU\Software\Classes\CLSID\{56AD4C5D-B908-4F85-8FF1-7940C29B3BCF}\Instance;
4. Result – результат виконання операції, наприклад: SUCCESS, REPARSE, NAME NOT FOUND, BUFFER OVERFLOW та ін.;
5. Event Class – тип події (реєстр, міжпроцесна комунікація, інтернет комунікація, виведення на носії, наприклад: File System, Registry та ін.);

6. Image Path – шлях до виконуваного файлу, який ініціював подію (наприклад: C:\Windows\Explorer.EXE);

7. Company – розробник програмного продукту та процесу, який ініціював цю подію, наприклад: Microsoft Corporation;

8. Description – опис програмного компоненту, наприклад: Windows Explorer;

9. User – имя користувача, який ініціалізував процес, наприклад: DESKTOP-159T3OE\VirtualUser;

10. Command Line – параметри командного рядка, наприклад: C:\ Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p;

11. Integrity – пріоритет і важливість виконуваної події, наприклад: System, Medium, High, Low;

12. Category – тип операції, наприклад: read, write, read metadata, write metadata та ін.;

13. Authentication ID – ID користувача з метою виявлення перейменування користувачів і груп, наприклад: 00000000: 000270cb;

Таблиця 1 – **Статистика процесів**

№	Назва процесу	Зміст процесу
Нормальний режим		
0	SystemProcesses_0	події КС, зібрані в режимі очікування;
1	SystemProcesses_1	події КС, зібрані в режимі очікування після активного використання;
2	ZipFileSystemZipper	події КС, зібрані при архівації вбудованим компонентом в провідник ;
3	OpenFoldersAndFiles	події КС, зібрані при відкритті директорій і файлів;
4	ExtractFilesSystemZipper	події КС, зібрані при розархівуванні вбудованим компонентом в провідник;
5	EditingTxtFile	події КС, зібрані при відкритті та редагуванні файлу;
6	DeleteToRecycleBin	події КС, зібрані при видаленні файлів з директорії в кошик;
7	DeleteFilesPerm	події КС, зібрані при видаленні файлів без приміщення в кошик;
8	CopyFiles71	події КС, зібрані при копіюванні 71 файлу;
9	CopyFiles	події КС, зібрані при копіюванні 10 файлів.
10	ZipFile7Zip	події, КС зібрані при архівування програмою 7Zip;
11	ExtractFiles7Zip	події КС, зібрані при розархівуванні програмою 7Zip.
Аномальний режим		
12	VirusPetya	системні і шкідливі події на прикладі роботи вірусу Petya;

Використовуючи параметри Process Name і Image Path можна з точністю ідентифікувати ініціатора події, наприклад, конкретний виконуваний файл і його процес.

Розробка ансамблевого методу класифікації

У якості інструменту для аналізу стану КС та оцінки ансамблевих класифікаторів було обрано ПЗ “Weka (Waikato Environment for Knowledge Analysis)” [13], яке містить набір засобів віртуалізації і компонентів для інтелектуального аналізу даних та вирішення завдань прогнозування.

Функціонал програми дозволяє виконати завдання аналізу даних, кластеризації, регресійний аналіз та ін.

За результатами попередніх досліджень, у якості класифікатора використано метод на основі дерева рішень *J*48.

Алгоритм *J*48 є аналогом алгоритму C4.5, який реалізовано на мові *Java* у додатку *Weka*. Алгоритм C4.5 є вдосконаленою версією алгоритму *ID*3 до якого була додана процедура відсікання гілок (*Pruning*), можливість роботи з числовими атрибутами, а також можливість побудови дерева з неповною навчальною вибіркою, в якій відсутні значення деяких атрибутів.

Алгоритм C4.5 вибирає атрибут на підставі нормалізованого приросту інформації або інформаційної ентропії (*Gain Ratio*):

$$H = -\sum_{i=1}^{n} \left(N_i / N \right) \cdot \log \left(N_i / N \right)$$

де n – число класів у вихідній підмножині, N_i – число прикладів i-го класу, N – загальна кількість прикладів в підмножині.

Для підвищення ефективності ідентифікації стану КС досліджено ансамблеві класифікатори на основі беггінгу та бустінгу.

Ефективність роботи класифікаторів була оцінена такими показниками:

1. Час навчання (*Learning time*);
2. Час тестування (*Testing time*);
3. Кількість визначених та невизначених (*Known and Unknown Instances*) об’єктів в абсолютному значенні;
4. Кількість правильно класифікованих об’єктів в абсолютному та процентному значенні (*Correctly Classified Instances*);
5. Кількість неправильно класифікованих об’єктів в абсолютному та процентному значенні (*Incorrectly Classified Instances*);
6. Абсолютна помилка класифікації (*Relative absolute error, RAE*);

7. Відносна квадратична помилка класифікації (*Root relative squared error, RRSE*);

Спочатку, у якості *навчальної* вибірки для класифікатора використано статистику процесів System Processes_0, зібрану в режимі очікування для нормального стану функціонування КС. Результати моделювання дозволили отримати два класифікатора на основі беггінгу та бустінгу, кожний із яких є ансамблем дерев рішень J48.

Отримані класифікатори надалі були використані для оцінки стану КС.

У якості *тестової* вибірки використано статистику процесів для нормального та аномального режимів, наведену в табл. 1 (рядки 1-12).

Результати тестування КС на основі бустингу наведено в табл. 2. У якості алгоритму прийняття рішень ансамблевого класифікатора на основі бустингу використано алгоритм *AdaBoostM1*. Вибірка для тренування містить 23145 подій нормальної та аномальної роботи системи. Коренем дерев є назва процесу, що надає змогу виявити відхилення у роботі системних процесів.

Як видно із табл.2, має місце значне збільшення кількості нерозпізнаних подій (*Unknown Instances*) особливо для аномальних процесів. Так для вірусного процессу *VirusPetya* кількість невизначених подій складає 203843. Крім того процес характеризується: наявністю неправильно класифікованих об'єктів (0,1451%), наявністю абсолютних значень некоректних подій (574 події), абсолютної помилки класифікації (0,1573), відносної квадратичної помилки класифікації (4,0078).

Дані показники є індикаторами зміни роботи системи.

Таблиця 2 – **Результати тестування КС на основі бустингу назви процесу**

Назва процесу	**Learning time, с**	**Testing Time, с**	**Total Instances**	**Unknown Instances**	**Correctly**	**Correctly Classified Instances, %**	**Incorrectly**	**Incorrectly Classified Instances, %**	**RAE**	**RRSE**
SystemProcesses	0.05	0.01	1476	179	1297	100	0	0	0	0
ZipFile7Zip	-	7.38	87702	13834	73868	100	0	0	0	0
ZipFileSystemZipper	-	10.72	83788	6096	77692	100	0	0	0	0
OpenFoldersAndFiles	-	31.64	242901	11065	231836	100	0	0	0	0
ExtractFilesSystemZipper	-	30.63	275014	73	274941	100	0	0	0	0
ExtractFiles7Zip	-	3.25	45654	16346	29308	100	0	0	0	0
EditingTxtFile	-	7.9	85464	18469	66995	100	0	0	0	0
DeleteToRecycleBin	-	1.86	55894	79	55815	100	0	0	0	0
DeleteFilesPerm	-	0.53	36969	28	36941	100	0	0	0	0
CopyFiles71	-	2.54	89807	3092	86715	100	0	0	0	0
CopyFiles	-	0.4	25731	172	25559	100	0	0	0	0
VirusPetya	-	435.03	599514	203843	395097	99.8549	574	0.1451	0.1573	4.0078

Таким чином, класифікатор на основі бустінгу та дерева рішень J48 дозволив виявити тільки невелику кількість аномальних подій вірусу Petya. Однак, події інших програм, наприклад 7Zip, були не виділені, хоча вони не брали участі в навчальній вибірці. Ознакою наявності нового процесу було тільки зростання кількості невизначених подій, а саме – 16346. Всі події крім *VirusPetya* не були розпізнані, хоча не входили до навчальної вибірки. Таким чином, чутливість класифікатора є невеликою. Алгоритм лише частково впорався з поставленим завданням ідентифікації аномального стану, вплив на систему додаткових процесів не був виявленим.

Результати тестування КС на основі беггінгу показали наступну статистику (табл. 3).

Як видно із табл. 3 також має місце значна кількість нерозпізнаних подій (*Unknown Instances*), особливо для аномальних процесів. Так для вірусного процессу *VirusPetya* кількість невизначених подій складає 599514. Крім того зафіксована аномальна поведінка КС, породжена програмою *7Zip*. Ідентифікуючі параметри мають високі значення, а саме значення помилок класифікації *RAE* і *RRSE*, які описують відносну помилку розпізнавання навчальної і тестової вибірки. *RAE* та *RRSE* можуть мати значення більше за нуль при повному визначенні подій у виборці. Це пояснюється тим, що вони описують відносне значення простого предиктору, нормалізуючи його шляхом ділення на загальну квадратну помилку простого предиктору. Взявши квадратний корінь з відносної квадратної помилки, можна зменшити помилку до тих розмірів, що і передбачувана величина. Цей результат характеризує точність класифікатора.

Таблиця 3 – **Результати тестування КС на основі беггінгу**

Назва процесу	Learning time, c	Testing Time, c	Total Instances	Unknown Instances	Correctly	Correctly Classified Instances, %	Incorrectly	Incorrectly Classified Instances, %	RAE	RRSE
SystemProcesses	0.1	0.05	1476	179	1297	99.845	2	0.154	4.062	7.909
ZipFile7Zip	-	8.54	87702	13834	73868	100	0	0	14.616	22.102
ZipFileSystemZipper	-	11.73	83788	6096	77545	99.810	147	0.189	2.285	9.127
OpenFoldersAndFiles	-	32.85	242901	11065	231354	99.792	482	0.207	0.777	5.912
ExtractFilesSystemZipper	-	30.39	275014	73	274941	100	0	0	0.144	2.238
ExtractFiles7Zip	-	4.1	45654	16346	29308	100	0	0	3.256	12.004
EditingTxtFile	-	9.41	85464	18469	66978	99.974	17	0.025	1.498	7.919
DeleteToRecycleBin	-	2.21	55894	79	55815	100	0	0	0.061	1.608
DeleteFilesPerm	-	0.69	36969	28	36941	100	0	0	0.024	1.028
CopyFiles71	-	3.16	89807	3092	86714	99.998	1	0.001	0.743	5.329
CopyFiles	-	0.56	25731	172	25559	100	0	0	0.111	2.295
VirusPetya	-	469.32	599514	203843	395097	99.854	574	0.145	4.538	12.546

Таким чином, ансамблеві класифікатори для яких у якості навчальної вибірки використано лише статистику *SystemProcesses_0,* зібрану в режимі очікування при нормальному стані функціонування КС потребують донавчання з використанням більшої кількості процесів навчальної вибірки.

Такі класифікатори можуть бути використаними тільки як експрес-класифікатори. Ознакою аномальності функціонування таких класифікаторів є збільшення кількості нерозпізнаних подій.

Для подальшого дослідження використано класифікатор на основі беггінгу.

Донавчання класифікатора на основі беггінгу було виконано з використанням усіх процесів, наведених в табл. 1, окрім процесу *VirusPetya*.

Вибірка для тренування містила 871497 подій, котрі об’єднують усі події для нормального режиму роботи.

Результати ідентифікації КС на онові отриманого класифікатора наведено в табл. 4.

Таблиця 4 – **Результати тестування КС на основі беггінгу**

Назва процесу	Learning time, c	Testing Time, c	Total Instances	Unknown Instances	Correctly	Correctly Classified Instances, %	Incorrectly	Incorrectly Classified Instances, %	RAE	RRSE
SystemProcesses	92,64	0,11	1476	0	1476	100	0	0	0,0073	0,2749
ZipFile7Zip	91,56	8,57	87702	0	87702	100	0	0	0,0295	0,4563
ZipFileSystemZipper	89,12	5,34	83788	0	83788	100	0	0	0,0057	0,3478
OpenFoldersAndFiles	92,86	9,29	242901	0	242901	100	0	0	0,0031	0,2901
ExtractFilesSystemZipper	90,43	12,84	275014	0	275014	100	0	0	0,0095	0,3215
ExtractFiles7Zip	88,39	6,56	45654	0	45654	100	0	0	0,0198	0,8174
EditingTxtFile	93	3,97	85464	0	85464	100	0	0	0,0043	0,1325
DeleteToRecycleBin	89,15	3,74	55894	0	55894	100	0	0	0,0074	0,7319
DeleteFilesPerm	91,98	4,83	36969	0	36969	100	0	0	0,0069	0,1345
CopyFiles71	92,77	4,38	89807	0	89807	100	0	0	0,0074	0,4357
CopyFiles	90	1,71	25731	0	25731	100	0	0	0,0056	0,3509
VirusPetya	-	876,44	599514	131997	431246	92,2418	36271	7,7582	12,4715	40,7097

Класифікатор побудовано на основі назви процесу. Як видно із табл. 4, усі події що приймали участь у навчанні були розпізнані як коректні. Якщо проаналізувати процес, породжений *VirusPetya*, то 131997 подій не ідентифіковано (що є ознакою аномальності). Із 467517 ідентифікованих подій *Virus Petya* 7,76% подій мають інші атрибути, що дозволяє їх охарактеризувати як аномальні. Наявність таких подій призводить до збільшення помилок класифікації *RAE* та *RRSE*. Процес ідентифікації стану КС та виявлення шкідливого програмного забезпечення та прикладі *VirusPetya* тривав близько 8 хвилин.

Висновки

В даній роботі досліджено події функціонування операційних системах сімейства Windows та виділено чотири їх основні типи: події міжпроцесної комунікації, події взаємодії з файловою системою, події інтернет-з'єднання, події взаємодії з реєстром операційної системи. Проаналізовано та виділено атрибути подій, що надає можливість ідентифікувати ініціатора події (конкретний виконуваний файл та процес). Розроблено програмне забезпечення для виділення подій, що дозволило сформувати вихідні дані для оцінки стану комп'ютерної системи. Розглянуто ансамблеві класифікаторів на основі беггінгу, бустінгу та дерева рішень J48 для ідентифікації стану комп'ютерної системи. У якості інструменту для аналізу стану КС та оцінки ансамблевих класифікаторів було обрано ПЗ Weka. Виконано оцінку ефективності розроблених ансамблевих класифікаторів. За результатами досліджень для ідентифікації стану комп'ютерної системи запропоновано ансамблевий метод класифікації на основі беггінгу та дерева рішень J48, що надає можливість підвищити точність ідентифікації стану комп'ютерної системи.

СПИСОК ЛІТЕРАТУРИ

1. Алпайдин Э. Машинное обучение: новый искусственный интеллект /Э. Алпайдин – М.: Изд.гр. Точка, 2017. – 208 с.
2. Вьюгин В.В. Математические основы машинного обучения и прогнозирования. / В.В. Вьюгин // – Москва: МЦНМО, 2013. – 304 с.
3. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных /П. Флах.//–Москва: ДМКПресс, 2015. – 400 с.
4. Марманис Х. Алгоритмы интеллектуального интернета. Передовые методики сбора, анализа и обработки данных. / Х.Марманис, Д.Бабенко .– Сб-П, М: Символ, 2011 . – 468 с.
5. Саттон Ричард С., Барто Эндрю Г. Обучение с подкреплением = Reinforcement Learning. – 2-е издание. – М.: ДМК пресс, 2020. – 552 с.
6. Кафтанников И. Л., Парасич А. В. Особенности применения деревьев решений в задачах классификации // Вестн. ЮУрГУ. Сер. «Компьютерные технологии, управление, радиоэлектроника». 2015, Т. 15. № 3. с. 26–32.
7. Cha Zhang. Ensemble Machine Learning. Methods and Applications / Cha Zhang, Yunqian Ma .— New York Dordrecht Heidelberg London: Springer, 2012 .– 329 p.
8. Тархов Д. А. Нейросетевые модели и алгоритмы / Д. А. Тархов.– Москва: Радиотехника, 2014. – 352 с.
9. Vipin Kumar. The Top Ten Algorithms in DataMining– Taylor & Francis Group, LLC, 2009 .– 2006 p.
10. Joseph Rocca, Baptiste Rocca. "Ensemble methods: bagging, boosting and stacking". [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.
11. Kristína Machová, Miroslav Puszta, František Barčák, and Peter Bednár, , " A Comparison of the Bagging and the Boosting Methods Using the Decision Trees Classifiers", Computer Science and Information Systems, 3(2), 2006, pp.57-72, DOI: 10.2298/CSIS0602057M.
12. Гавриленко С.Ю. Розробка методу оцінки стану комп'ютера на базі аналізу системних подій / С.Ю. Гавриленко, І.В. Шевердін //Науковий вісник Івано-Франківського національного технічного університету нафти і газу – Івано-Франківськ, 2018, №1(40), сс.108-114
13. "WEKA. The workbench for machine learning". [Електронний ресурс]. – Режим доступу: https://www.cs.waikato.ac.nz/ml/weka/.

Received (Надійшла) 27.06.2020
Accepted for publication (Прийнята до друку) 22.07.2020

A computer system state identification based on the ensemble classification method

S. Gavrilenko, I. Sheverdin

Annotation. The subject of f this article is the study of methods of identifying a computer system state. **The purpose** of the article is development of a method for identifying computer system abnormal state based on ensemble methods. **Objective: i**nvestigate and distinguish events in Windows operating systems, develop software for collecting events; investigate the use of ensemble classifiers based on begging, busting and decision trees for identifying the state of a computer system. **The methods used are**: machine learning methods and ensemble classification methods. The following results were obtained: The following events classes in Windows operating systems were selected as source data: process communication events, file system interaction events, internet connection events, and registry interaction events. Identification methods of abnormal computer system state were studied based on ensemble methods such as begging, boosting, and J48 decision tree for identifying the state of a computer system. The effectiveness of developed ensemble classifiers was evaluated. Based on the research results, the begging ensemble classification method and the J48 decision tree is proposed for identifying the computer system state. **Conclusions.** The scientific novelty of the obtained results consists in selecting the computer system functioning processes and creating an ensemble method for identifying the computer system state based on begging and the J48 decision tree, which makes it possible to increase the identification accuracy.

Keywords: computer system, operating system events, machine learning, MS Windows abnormal state, decision trees, ensemble classification methods, boosting, begging.