https://doi.org/10.31891/2307-5732-2025-357-29
УДК 004.056:004.852

КОНОТОПЕЦЬ МИКОЛА
Інститут спеціального зв'язку та захисту інформації Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
https://orcid.org/0000-0002-6963-1877
e-mail: nikolyalux@gmail.com
ТУРОВСЬКИЙ ОЛЕКСАНДР
Державний університет інформаційно-комунікаційних технологій
https://orcid.org/0000-0002-4961-0876
e-mail: s19641011@ukr.net
АКСАМИТНИЙ ОЛЕКСАНДР
Інститут спеціального зв'язку та захисту інформації Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
https://orcid.org/0009-0004-4439-6286
e-mail: alexaksamutnuy@gmail.com
МАТІЙКО АЛЕКСАНДРА
Інститут спеціального зв'язку та захисту інформації Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
https://0000-0002-6947-5958
e-mail: alexm1710@ukr.net

ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ RANDOM FOREST ТА XGBOOST У ЗАДАЧІ КЛАСИФІКАЦІЇ ІНЦИДЕНТІВ БЕЗПЕКИ

У статті представлено порівняльне дослідження ефективності моделей машинного навчання Random Forest та XGBoost у задачі багатокласової класифікації інцидентів безпеки в інформаційних системах.

У процесі дослідження побудовано дві моделі класифікації інцидентів на основі алгоритмів Random Forest та XGBoost. Окрім емпіричної оцінки якості моделей, у роботі описано їх принципи роботи та подано математичне обґрунтування. Для Random Forest сформульовано ймовірнісну модель ансамблю дерев, використано індикаторні функції, проаналізовано функцію відступу та узагальнену помилку моделі. Для XGBoost детально розглянуто процедуру побудови дерева на кожній ітерації, оптимізацію функціоналу втрат з регуляризаційним компонентом, використання градієнтів другого порядку та критерій приросту. Така формалізація забезпечує глибше теоретичне розуміння механізмів роботи алгоритмів і пояснює їх поведінку в реальних умовах.

Проведено порівняльний аналіз ефективності моделей за ключовими метриками класифікації: точністю (accuracy), повнотою (recall), точністю позитивних прогнозів (precision) та F1-мірою. Визначено, що модель Random Forest показала дещо вищу загальну точність (91.97%) та кращу здатність виявляти хибнопозитивні інциденти, що є важливою перевагою в умовах перевантаження SOC великою кількістю сигналів. У свою чергу, модель XGBoost продемонструвала стабільну класифікацію справжніх загроз (TruePositive), що є критичним для оперативного реагування в системах інформаційної безпеки.

Результати дослідження можуть бути ефективно використані в інтеграції розроблених моделей у системи SIEM, SOAR та інші платформи інформаційної безпеки для автоматизованої попередньої класифікації подій.

Ключові слова: *класифікація інцидентів, інформаційна безпека, машинне навчання, Random Forest, XGBoost, хибнопозитивні спрацювання.*

KONOTOPETS MYKOLA, AKSAMYTNYI OLEKSNDR, MATIYKO ALEXANDRA
Institute of Special Communications and Information ProtectionNational Technical University of Ukraine"Igor Sikorsky Kyiv Polytechnic Institute"
TUROVSKY OLEKSANDR
State University of Information and Communication Technologies

COMPARATIVE ANALYSIS OF RANDOM FOREST AND XGBOOST MACHINE LEARNING MODELS IN THE TASK OF SECURITY INCIDENT CLASSIFICATION

The article presents a comparative study of the effectiveness of the Random Forest and XGBoost machine learning models in the problem of multi-class classification of security incidents in information systems.

In the process of the study, two incident classification models based on the Random Forest and XGBoost algorithms were built. In addition to the empirical assessment of the quality of the models, the paper describes their principles of operation and provides a mathematical justification. For Random Forest, a probabilistic model of an ensemble of trees was formulated, indicator functions were used, the regression function and the generalized error of the model were analyzed. For XGBoost, the procedure for building a tree at each iteration, optimization of the loss functional with a regularization component, the use of second-order gradients and the growth criterion were considered in detail. Such a formalization provides a deeper theoretical understanding of the mechanisms of operation of the algorithms and explains their behavior in real conditions.

A comparative analysis of the effectiveness of the models was conducted on key classification metrics: accuracy, recall, precision, and F1-measure. It was determined that the Random Forest model showed a slightly higher overall accuracy (91.97%) and a better ability to detect false positive incidents, which is an important advantage in conditions of SOC overload with a large number of signals. In turn, the XGBoost model demonstrated stable classification of true threats (TruePositive), which is critical for rapid response in information security systems.

The results of the study can be effectively used in the integration of the developed models into SIEM, SOAR, and other information security platforms for automated preliminary classification of events.

Keywords: *incident classification, information security, machine learning, Random Forest, XGBoost, false positives.*
Стаття надійшла до редакції / Received 21.07.2025
Прийнята до друку / Accepted 15.08.2025

Вступ і формулювання актуальності проблеми.

Одним із ключових викликів у сучасних системах виявлення інцидентів безпеки є значна кількість хибно позитивних спрацювань (false positives), які ускладнюють оперативне реагування та збільшують навантаження на аналітиків центрів обробки інцидентів (SOC). Особливо це актуально для екосистеми Microsoft, де захисні механізми генерують велику кількість сигналів, що не завжди свідчать про реальні загрози. Така ситуація призводить до інформаційного шуму, втрати довіри до систем попередження, а в окремих випадках – до ігнорування справжніх інцидентів.

Постановка проблеми

З метою покращення точності класифікації інцидентів та зменшення кількості помилкових спрацювань все частіше застосовуються алгоритми машинного навчання, здатні виявляти приховані закономірності у великих обсягах даних. Серед таких алгоритмів особливе місце займають Random Forest та XGBoost, які добре зарекомендували себе у задачах класифікації. Проте, кожен із цих підходів має свої особливості щодо інтерпретації, чутливості до параметрів, стійкості до надлишкових ознак і обробки незбалансованих вибірок.

Незважаючи на широку популярність обох алгоритмів, відсутній однозначний висновок щодо того, який із них є більш ефективним для задач класифікації інцидентів безпеки. Враховуючи критичну важливість точності класифікації у сфері кібербезпеки, постає проблема проведення емпіричного порівняння моделей Random Forest та XGBoost на реальних даних, що відображають інциденти безпеки.

Аналіз літературних джерел та постановка завдання дослідження

Актуальним напрямом досліджень у сфері інформаційної безпеки є розробка методів виявлення аномалій, які можуть свідчити про потенційні загрози чи порушення. У наукових роботах [1,2] виявлення аномалій трактується як задача класифікації або кластеризації відхилень від нормальної поведінки мережі, користувача чи системного середовища. Це завдання покладається на системи виявлення інцидентів безпеки, які мають забезпечувати своєчасне та точне виявлення відхилень, що можуть свідчити про потенційні кібератаки або інші загрози. Водночас суттєвою проблемою сучасних систем виявлення інцидентів безпеки є велика кількість хибнопозитивних спрацювань (false positives), що безпосередньо впливає на ефективність функціонування центрів реагування на інциденти безпеки (SOC). Дослідження [3-7] підкреслюють, що надмірний обсяг FP-інцидентів спричиняє інформаційне перевантаження аналітиків, ускладнює процес пріоритезації подій, викликає втому персоналу та значно знижує швидкість і точність реагування на реальні загрози. У результаті це не лише впливає на якість оперативного моніторингу, а й створює умови для потенційного пропуску критичних атак.

Значну увагу в науковій літературі приділено математичному обґрунтуванню принципів роботи алгоритмів Random Forest та XGBoost. У роботах [8-11] Random Forest інтерпретується як ймовірнісна модель ансамблю дерев, де використання індикаторних функцій та поняття функції відступу дозволяє оцінити здатність моделі до узагальнення. Також розглядаються властивості узагальненої похибки у контексті теореми Чебишова та імовірнісних границь. У свою чергу, алгоритм XGBoost аналізується як оптимізаційна модель, що мінімізує регуляризований функціонал втрат шляхом послідовного додавання дерев рішень. У працях [12-14] описано використання градієнтів та гессіанів у процесі обчислення приросту інформації для вибору розбиттів у вузлах дерева, що забезпечує високу ефективність і контроль над перенавчанням. Таке математичне трактування дозволяє не лише пояснити емпіричну ефективність моделей, а й підвищити їх прозорість, що має важливе значення для застосування в критично важливих сферах, таких як інформаційна безпека.

Разом з тим, питання застосування методів машинного навчання для виявлення та класифікації інцидентів безпеки загалом є об'єктом активного наукового інтересу [15–17]. У роботах [18,19] аналізується ефективність алгоритмів Random Forest та XGBoost у задачах багатокласової класифікації, зокрема в умовах обробки великих обсягів даних з дисбалансом класів. Автори зазначають, що обидва алгоритми добре масштабуються та демонструють стабільні результати в умовах наявності шуму в даних.

Незважаючи на широке висвітлення окремих аспектів теми, існує обмежена кількість комплексних робіт, у яких одночасно поєднано формальний математичний аналіз моделей, глибокий емпіричний експеримент на реальному масштабному наборі даних та практичну оцінку ефективності моделей. Це й визначає наукову новизну даного дослідження, в якому представлено як теоретичне обґрунтування обраних моделей, так і практичну реалізацію класифікації інцидентів на основі набору даних Microsoft Security Incident Prediction.

Мета та завдання дослідження

Метою статті є проведення порівняльного аналізу ефективності моделей машинного навчання Random Forest та XGBoost у задачі класифікації інцидентів безпеки.

Основна частина

Для досягнення мети та виконання завдання, доцільним є дослідження особливостей роботи моделей машинного навчання Random Forest та XGBoost. Дослідимо принцип роботи Random Forest, що викладений в роботах [8-11].

Метод Random Forest базується на побудові ансамблю класифікаторів, кожен з яких є деревом рішень, що тренується на випадковій підмножині навчальних даних та ознак. Завдяки такій стохастичній побудові кожного дерева забезпечується різноманітність моделей усередині ансамблю. Результати окремих дерев об'єднуються шляхом голосування для отримання кінцевого прогнозу, що дозволяє досягти вищої точності, стійкості до перенавчання та стабільності у порівнянні з використанням окремого дерева рішень. Формалізація роботи Random Forest потребує чіткого математичного опису ймовірнісної моделі, яка охоплює як механізм побудови окремих дерев, так і поведінку ансамблю загалом.

На початку розглянемо ймовірнісні основи моделі. Нехай X — випадкова величина з математичним сподіванням μ та стандартним відхиленням σ. Застосуємо нерівність Чебишова, що дозволяє оцінити ймовірність відхилення від середнього значення:

$$P(|X-\mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2},$$

де $\epsilon > 0$. Ця нерівність забезпечує верхню оцінку ймовірності появи великих відхилень в окремих прогнозах дерев ансамблю.

Уведемо індикаторну функцію для будь-якої події A у просторі елементарних подій:

$$I_A(x) = f(x) = \begin{cases} 1, & x \in A, \\ 0, & \text{інакше.} \end{cases}$$

Індикаторна функція дозволяє формалізувати механізм прийняття рішень окремими деревами в ансамблі.

Опишемо також теорему про обмежену збіжність, яка забезпечує перестановку границі та інтегрування для обмежених функцій. Нехай послідовність $h_k(x)$ задовольняє нерівність $h_k(x) \leq M$, де $M > 0$, тоді:

$$\lim_{k\to\infty} \int_S h_k(x)dx = \int_S \lim_{k\to\infty} h_k(x)dx.$$

Дана теорема використовується при аналізі асимптотичної поведінки ансамблю Random Forest із зростанням кількості дерев.

Після розгляду ймовірнісних основ ансамблевого методу перейдемо до опису базового елемента моделі — дерева класифікації.

Нехай маємо n спостережень, для яких задано вектори ознак $\{x_i\}_3^2$ та відповідні вихідні значення y_i. Повна вибірка даних записується у вигляді:

$$D = \{(x_1 y_1), \dots, (x_n y_n)\}.$$

Кожен вектор ознак має вигляд:

$$x_k = (x_{k1}, \dots, x_{kd}).$$

Дамо формальне означення дерева класифікації.

Дерево класифікації — це дерево рішень, у якому в кожній вершині приймається бінарне рішення на основі перевірки умови $x_i < a$, де a — певне порогове значення, що може залежати від конкретної вершини.

У кореневій вершині містяться всі приклади $(x_k y_k)$, які далі розподіляються між дочірніми вузлами відповідно до прийнятого на вершині рішення. Розбиття триває доти, доки всі елементи у листових вузлах належатимуть до одного класу.

На кожному вузлі вибираються ознака x_i та поріг a таким чином, щоб мінімізувати різноманітність у дочірніх вузлах. Як міра різноманітності часто використовується критерій Джині.

Розбиття триває доти, доки кожен листовий вузол містить об'єкти лише одного класу. Нехай визначено два класи: C_1 — відхилення, C_2 — норма. Розглянемо вибірку S у поточному вузлі. Для побудови дочірніх вузлів виконується розбиття:

$$S = S_1 \cup S_2.$$

Кожна підмножина S_j містить об'єкти, розподілені за класами C_1, C_2. Позначимо:

$$\hat{P}(S_j) = \frac{|S_j|}{|S|},$$

де $|S|$ — кількість об'єктів у вибірці S, а $\hat{P}(S_j)$ — частка елементів у C_j.

Частку елементів класу C_i у множині S_j визначимо як:

$$\hat{P}(C_i|S_j) = \frac{|S_j \cap C_i|}{|S_j|}.$$

Варіація у вузлі S_j обчислюється за формулою:

$$g(S_i) = \sum_{i=1}^{2} \hat{P}(C_i|S_j)\,(1 - \hat{P}(C_i|S_j)).$$

Максимальне значення $g(S_j)$ досягається при рівномірному розподілі між класами. Мінімальне — коли всі елементи належать одному класу.

Індекс Джині для вузла визначається як зважена сума варіацій:

$$G = \hat{P}(S_1)g(S_1) + \hat{P}(S_2)g(S_2).$$

Індекс Джині використовується для оптимізації розбиття на кожному кроці побудови дерева. Для подальшої формалізації задачі класифікації дослідимо поняття випадкового вектора.

Нехай $X = (X_1, \dots, X_d)$ — випадковий вектор, компоненти якого є випадковими величинами, визначеними на спільному ймовірнісному просторі. Спільний розподіл цього вектора визначається мірою μ на просторі R^d наступним чином:

$$\mu(A) = P(x \in A),$$

де $A \subset R^d$ — вимірна множина.

Для задач класифікації розглядається пара (X, Y), де Y — випадкова величина, що приймає значення у скінченій множині класів. Вектор (X, Y) описує спільний розподіл ознак та міток класів у моделі. Реалізації вибірки розглядаються як незалежні реалізації з даного розподілу.

Сформулюємо формальне визначення моделі Random Forest.

Нехай задано навчальну вибірку:

$$D = \{(x_1, y_1), \dots, (x_n, y_n)\},$$

що є незалежними реалізаціями випадкового вектора $((X, Y)$, тобто $(x_i, y_i) \sim (X, Y)$.

Метою є побудова класифікатора, що на основі вхідного вектора ознак x передбачає мітку y, використовуючи дані вибірки D.

Розглядається ансамбль класифікаторів:

$$h = \{hh_1(x), \dots, h_K(x)\}.$$

Кожен класифікатор $h_k(x)$ є деревом рішень, повністю визначеним набором параметрів:

$$\Theta_k = (\Theta_{k1}, \Theta_{k2}, \dots, \Theta_{kp}),$$

що включає вибір ознак для розбиття, їх порогові значення, а також структуру дерева. Таким чином, класифікатор записується у параметричній формі:

$$h_k(x) = h(x \mid \Theta_k).$$

Параметри Θ_k формуються випадково з деякого розподілу параметрів Θ, що забезпечує стохастичність побудови окремих дерев у ансамблі.

Фінальна модель Random Forest визначається як сукупність класифікаторів виду:

$$h(x \mid \Theta_1), \dots, h(x \mid \Theta_k).$$

Для отримання остаточного рішення використовується агрегація голосів окремих дерев. Для кожного вхідного вектора x кожне дерево формує свій прогноз, а підсумковий клас визначається за правилом більшості:

Після формального визначення моделі дослідимо основні властивості ансамблевих методів, зокрема поняття відступу та узагальненої помилки.

Розглянемо фіксований ансамбль класифікаторів:

$$h = \{h_1(x), \dots, h_K(x)\},$$

що працює з випадковим вектором даних (x, y).

Для довільної події A, що стосується результатів роботи класифікаторів $h_k(x)$, визначимо емпіричну ймовірність:

$$\hat{P}(A) = \frac{1}{K} \sum_{k=1}^{K} I_A h_K((x)),$$

де I_A — індикатор події A, тобто $\hat{P}(A)$ є часткою класифікаторів, для яких подія A реалізується.

Введемо поняття емпіричної функції відступу:

$$\hat{m}(x, y) = \hat{P}_k(h_k(x) = y) - \max_{j \neq y} \hat{P}_k(h_k(x) = j),$$

яка характеризує різницю між середньою кількістю голосів за правильний клас та середньою кількістю голосів за найближчий інший клас.

Функція $\hat{m}(x, y)$ відображає впевненість ансамблю у правильності класифікації прикладу (x, y).

Визначимо узагальнену помилку ансамблю як імовірність того, що відступ негативний:

$$e = P_{x,y}(\hat{m}(x, y) < 0,$$

де під $P_{x,y}$ розуміється ймовірність за спільним розподілом (X, Y).

З наближенням кількості дерев K до нескінченності, узагальнена помилка прагне до:

$$e \xrightarrow[K \to \infty]{} P_{x,y}(P_\Theta(h(x, \Theta) = y) - \max_{j \neq y} P_\Theta(h(x, \Theta) = j) < 0),$$

де P_Θ — ймовірність за випадковими параметрами дерева Θ.

Далі формалізуємо модель Random Forest як ансамбль випадкових класифікаторів та розглянемо оцінку її узагальненої помилки.

Замість фіксованого ансамблю класифікаторів $\{h_k(x)\}_{k=1}^{K}$ розглянемо стохастичну модель Random Forest.

У цій моделі кожен класифікатор визначається параметрами Θ за допомогою функції:

$$h\{x|\Theta\},$$

де параметри Θ обираються з відомого розподілу, що визначає різноманітність дерев у ансамблі.

Визначимо функцію відступу ансамблю Random Forest:

$$m(x,y) = P_\theta(h(x|\Theta) = y) - \max_{j \neq y} P_\theta(h(x|\Theta) = j),$$

яка описує різницю між ймовірністю коректної класифікації та максимальною ймовірністю некоректної класифікації серед усіх інших класів.

Сила ансамблю визначається як математичне сподівання функції відступу:

$$s = E_{x,y}(x,y).$$

Для узагальненої помилки ансамблю виконується наступна оцінка, що базується на нерівності Чебишова:

$$e = P_{x,y}(m(x,y) < 0) \leq P_{x,y}(|m(x,y) - s| \geq s) \leq \frac{V(m)}{s^2},$$

де $V(m)$ — дисперсія функції відступу $m(x,y)$. Отримане співвідношення задає верхню межу узагальненої помилки ансамблю.

Було формалізовано Random Forest як ансамбль стохастично побудованих дерев рішень, розкрито математичний механізм його генералізації через відступ та встановлено оцінки узагальненої похибки за допомогою ймовірнісних нерівностей.

Дослідимо принцип роботи XGBoost, що викладений в роботах [12-14].

Метод XGBoost заснований на ітеративному побудові ансамблю дерев рішень за принципом градієнтного бустингу. Кожне наступне дерево створюється таким чином, щоб мінімізувати залишкову похибку попереднього ансамблю шляхом наближення до антиградієнта функції втрат. Для підвищення стабільності моделі вводяться регуляризаційні члени, що контролюють складність дерев і запобігають перенавчанню. На відміну від Random Forest, де окремі моделі будуються незалежно, у XGBoost дерева формуються послідовно, кожне з урахуванням попередніх помилок. Формалізація роботи XGBoost ґрунтується на строгому математичному описі оптимізації функціоналу втрат із використанням методів функціонального градієнта, другого порядку апроксимації та регуляризації параметрів моделі.

Алгоритм XGBoost застосовується для задач навчання з учителем, у яких маємо навчальну вибірку, що складається з векторів ознак x_i та відповідних цільових значень y_i. Завданням є побудова моделі, яка на основі нових вхідних даних x передбачає відповідне значення y.

У загальній постановці модель у навчанні з учителем визначає функціональну залежність між вхідними ознаками та прогнозованим значенням. У найпростішому випадку розглядається лінійна модель, в якій прогнозоване значення має вигляд:

$$\hat{y}_i = \sum_j \theta_j x_{ij},$$

де θ_j — параметри моделі, які необхідно визначити за даними. В залежності від конкретної задачі, така модель може застосовуватися як для регресії, так і для класифікації. Наприклад, у задачі логістичної регресії до результату застосовується логістичне перетворення для обчислення ймовірності належності до позитивного класу.

Параметри θ є невідомими величинами, які підлягають навчанню на основі наявної навчальної вибірки. У контексті лінійних моделей ці параметри представляють ваги ознак.

Навчання моделі передбачає знаходження таких параметрів θ, які мінімізують невідповідність між прогнозами моделі та фактичними значеннями цільової змінної. Для цього вводиться цільова функція (objective function), що складається з двох складових:

$$obj(\theta) = L(\theta) + \Omega(\theta),$$

де $L(\theta)$ — функція втрат (помилка навчання), а $\Omega(\theta)$ — регуляризаційний доданок.

Функція втрат вимірює точність моделі відносно навчальних даних. Наприклад, при використанні середньоквадратичної помилки вона має вигляд:

$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2.$$

У задачах класифікації часто застосовується логістична функція втрат:

$$L(\theta) = \sum_i [y_i \ln\left(1 + e^{-\hat{y}_i}\right) + (1 - y_i)\ln(1 + e^{\hat{y}_i})].$$

Регуляризаційний доданок контролює складність моделі та запобігає перенавчанню. Його включення дозволяє балансувати між точністю моделі на навчальних даних та її узагальнюючою здатністю. Такий баланс між складністю та узагальнюючою здатністю моделі в теорії машинного навчання відомий як компроміс між зміщенням і дисперсією (bias-variance tradeoff).

Раніше описані основи навчання з учителем складають фундамент для побудови більш складних моделей ансамблевого типу, зокрема — для градієнтного бустингу. XGBoost реалізує ансамбль дерев рішень, що будується послідовно з урахуванням помилок попередніх моделей.

У якості базових моделей XGBoost використовує дерева типу CART (Classification and Regression Trees). На відміну від класичних дерев рішень, у яких листові вузли містять лише кінцеві класи, в CART кожному листу відповідає числове значення, що дозволяє використовувати єдину модель як для класифікації, так і для регресії.

Замість одного дерева використовується ансамбль з K дерев, прогноз якого визначається як сума відповідей окремих дерев:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),$$

де кожне дерево f_k є функцією, що належить до простору всіх можливих дерев F.

Цільова функція, яка мінімізується під час навчання, має вигляд:

$$obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k),$$

де $l(y_i, \hat{y}_i)$ — функція втрат, що відображає точність прогнозу, а $\omega(f_k)$ — регуляризаційний член, який контролює складність дерева f_k.

На відміну від методів випадкових лісів, у яких дерева будуються незалежно, у XGBoost побудова кожного наступного дерева враховує залишкову помилку попередньої композиції. Таким чином, модель поступово вдосконалює свої прогнози, компенсуючи похибки попередніх дерев.

Після визначення моделі переходимо до процедури навчання, яка полягає в послідовній побудові дерев з метою мінімізації цільової функції. Як і у всіх задачах навчання з учителем, процес навчання формулюється як задача мінімізації функціоналу втрат з регуляризацією:

$$obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \omega(f_k),$$

де t — кількість дерев, побудованих на поточній ітерації.

В основі XGBoost лежить ідея послідовного додавання нових дерев до моделі в рамках адитивної схеми:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i),$$

де кожне нове дерево f_t додається до композиції, доповнюючи попередню суму прогнозів.

На кожній ітерації завдання зводиться до знаходження такого нового дерева f_t, яке мінімізує оновлений функціонал втрат:

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \omega(f_t).$$

У випадку середньоквадратичної функції втрат (MSE) функціонал приймає вигляд:

$$obj^{(t)} = \sum_{i=1}^{n} [(y_i - \hat{y}_i^{(t-1)} - f_t(x_i))^2] + \omega(f_t).$$

Для узагальнення на довільні функції втрат l використовується розклад Тейлора другого порядку навколо прогнозу попередньої ітерації:

$$obj^{(t)} \approx \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \omega(f_t),$$

де:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}, \; h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)^2}}.$$

Відкинувши сталі члени, отримуємо цільову функцію, яка оптимізується на поточній ітерації:

$$\sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \omega(f_t).$$

Таким чином, оптимізація кожного нового дерева зводиться до розв'язання задачі, що залежить лише від обчислених градієнтів g_i та гессіанів h_i, що дозволяє узагальнювати метод на широкий спектр функцій втрат.

Ключовою особливістю моделі XGBoost є введення формального регуляризаційного члена $\omega(f)$, що контролює складність дерева та запобігає перенавчанню. Для формалізації складності кожного дерева $f_t(x)$ уточнюємо його подання:

$$f_t(x) = \omega_{q(x)}, w \in R^T, q: R^d \to \{1,2, \dots, T\},$$

де:

T — кількість листів у дереві,

$q(x)$ — функція, що відображає кожний об'єкт x до відповідного листа дерева,

w_j — прогнозоване значення для листа j.

Складність дерева визначається як:

$$\omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2,$$

де параметри γ та λ контролюють відповідно: штраф за кількість листів та штраф за величини значень у листах.

З урахуванням такої регуляризації функціонал на t-ій ітерації набуває вигляду:

$$obj^{(t)} \approx \sum_{j=1}^{T} [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2.$$

Групуючи за листами дерева:

$$obj^t = \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T,$$

де $I_j = \{i | q(x_i) = j\}$ — множина індексів об'єктів, що потрапляють до листа j.

Визначивши агреговані суми:

$$G_j = \sum_{i \in I_j} g_i, H_j = \sum_{i \in I_j} h_i,$$

одержуємо остаточний вираз функціоналу:

$$obj^{(t)} = \sum_{j=1}^{T} [G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + \gamma T.$$

Оскільки функціонал є квадратичним відносно w_j, його мінімум досягається при:

$$w_j^* = -\frac{G_j}{H_j + \lambda}.$$

Підставляючи це значення у функціонал, отримуємо оптимальне значення цільової функції:

$$obj^* = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T.$$

Отриманий вираз дозволяє кількісно оцінити якість побудови певної структури дерева $q(x)$, що стає критерієм для вибору оптимального розбиття при побудові дерева на кожній ітерації.

Після формалізації функціоналу якості дерева виникає задача знаходження оптимальної структури дерева. Оскільки повний перебір усіх можливих дерев є обчислювально неможливим, XGBoost застосовує локально оптимальну стратегію, поступово оптимізуючи розбиття на кожному кроці побудови дерева.

На кожному етапі оптимізації розглядається розбиття існуючого листа на два підлисти — лівий та правий. Для оцінки доцільності такого розбиття вводиться функція приросту (gain), яка обчислюється за формулою:

$$Gain = \frac{1}{2}\left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} + \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right) - \gamma,$$

де:

G_L, G_R — суми градієнтів по кожній з двох груп після розбиття,

H_L, H_R — суми гессіанів по кожній з груп,

λ — коефіцієнт регуляризації величин листів,

γ — штраф за створення додаткового листа.

Формула Gain дозволяє кількісно оцінити користь від кожного можливого розбиття. Якщо приріст виявляється меншим за параметр γ, новий поділ не здійснюється, оскільки додавання нового листа не приносить істотного покращення моделі — таким чином реалізується механізм автоматичної обрізки гілок.

Щоб ефективно знаходити найкраще розбиття, для кожної ознаки дані сортуються у порядку зростання, після чого здійснюється послідовний перегляд потенційних точок розбиття. Такий підхід забезпечує обчислювальну ефективність навіть для великих вибірок.

Варто зазначити, що дана локально оптимальна стратегія пошуку розбиттів працює добре на практиці, хоча не гарантує знаходження глобально оптимальної структури дерева. Особливо це стосується випадків, коли важлива взаємодія кількох ознак — оскільки оптимізація проводиться лише по одній ознаці на кожному кроці.

Було формалізовано XGBoost як регуляризований градієнтний бустинг дерев рішень, розкрито його математичний апарат через функціональну оптимізацію другого порядку, описано механізм побудови структури дерева за допомогою критерію приросту та встановлено строгий контроль узагальнюючої похибки завдяки введенню регуляризаційних параметрів.

Опишемо обраний набір даних для навчання моделей.

Дослідження ґрунтується на датасеті Microsoft Security Incident Prediction [20], який містить дві підвибірки: навчальну вибірку розміром 9 516 837 записів (69.7%) та тестову вибірку обсягом 4 147 992 записи (30.3%). Сукупний обсяг даних становить 13 664 829 записів. Кожен запис включає 45 ознак, що описують різноманітні параметри інцидентів безпеки, зокрема категорію загрози (Category), техніку атаки згідно з класифікацією MITRE (MitreTechniques), типи залучених сутностей (EntityType), рівень підозри (SuspicionLevel), фінальний вердикт системи (LastVerdict), а також допоміжні ідентифікатори й технічні атрибути. Цільова змінна IncidentGrade має три категорії: TruePositive (істинно позитивний інцидент), FalsePositive (хибнопозитивний інцидент) та BenignPositive (доброякісний позитивний інцидент). Розподіл класів у навчальній вибірці демонструє наступні пропорції (див. рис. 1): BenignPositive — 43.4%, TruePositive — 35.1% та FalsePositive — 21.5%.

На етапі попередньої обробки даних було виявлено значну кількість пропусків у низці ключових ознак, таких як MitreTechniques, ThreatFamily, SuspicionLevel та LastVerdict, де частка відсутніх значень перевищувала 80-90%. З метою спрощення початкової фази аналізу було прийнято рішення виключити ознаки з критично високим рівнем пропусків, залишивши стабільні та репрезентативні змінні. Для побудови моделі було обрано три категоріальні ознаки: Category, EntityType та EvidenceRole. Ідентифікаційні поля, зокрема Id, OrgId, DeviceId, ApplicationId та інші числові ID-поля, вилучено через відсутність у них релевантного семантичного навантаження. Текстові ознаки було перетворено у числовий формат шляхом застосування методу Label Encoding, що забезпечило коректну обробку даних алгоритмами машинного навчання.

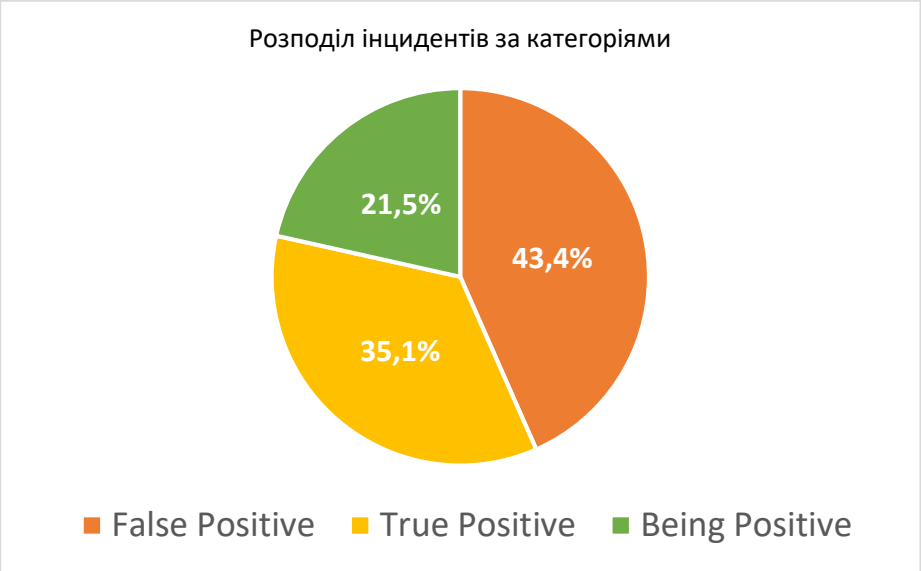

Рис. 1. Розподіл інцидентів за категоріями

Для оцінки ефективності моделей використано метрики оцінки класифікації, що визначені в роботі [7]. Вони мають наступне математичне визначення. Метрика точності позитивних прогнозів (precision) характеризує частку правильно класифікованих позитивних випадків серед усіх випадків, які модель віднесла до відповідної категорії. Її обчислення здійснюється за формулою:

$$Precision = \frac{TP}{TP+FP},$$

де TP (True Positives) – кількість коректно класифікованих позитивних прикладів, а FP (False Positives) – кількість помилкових позитивних передбачень.

Метрика повноти (recall) відображає частку правильно виявлених позитивних випадків серед усіх фактичних позитивних прикладів та обчислюється за формулою:

$$Recall = \frac{TP}{TP+FN},$$

де FN (False Negatives) — кількість пропущених позитивних прикладів.

Метрика F1-score являє собою гармонійне середнє між precision та recall і визначається за формулою:

$$F1score = 2 * \frac{Precision*Recall}{Precision+Recall}.$$

Застосування F1-score є особливо доцільним у випадках із дисбалансом класів, оскільки вона забезпечує узгоджену оцінку точності та повноти класифікації.

Модель класифікації інцидентів безпеки була побудована на основі методу контрольованого навчання (supervised learning), в рамках якого було розглянуто два підходи ансамблевого моделювання – Random Forest та XGBoost, що відрізняються за принципами багатокласової оптимізації.

Побудуємо модель класифікації із застосуванням алгоритму Random Forest. У процесі навчання модель використовувала вхідний вектор ознак, сформований на основі відібраних параметрів Category, EntityType та EvidenceRole. Цільова змінна IncidentGrade подавалася у вигляді числових класів після попереднього кодування категоріальних ознак за допомогою Label Encoding. Модель Random Forest будувалася шляхом формування ансамблю незалежних дерев рішень, де кожне дерево навчалося на випадкових підмножинах спостережень та ознак, а підсумкове рішення приймалося шляхом агрегування прогнозів окремих дерев методом більшості голосів. Враховуючи стійкість Random Forest до перенавчання, у даній реалізації не здійснювалося спеціальне налаштування глибини дерев, швидкості навчання або параметрів регуляризації. Для забезпечення відтворюваності експерименту було застосовано фіксований генератор псевдовипадкових чисел.

Реалізуємо модель із застосуванням алгоритму градієнтного бустингу на основі дерев рішень XGBoost. У процесі навчання модель використовувала той самий вхідний вектор ознак, сформований на основі параметрів Category, EntityType та EvidenceRole.

Цільова змінна Incident Grade подавалася у вигляді числових класів. Параметри моделі включали використання функції втрат multi: softmax для мультикласової класифікації з трьома класами, максимальну глибину дерев $max\_depth = 5$, швидкість навчання $eta = 0.1$, частковий відбір ознак $colsample\_bytree = 0.3$, параметр регуляризації $alpha = 10$, а також метод побудови дерев hist з обчисленнями на GPU.

Навчання здійснювалося з максимальною кількістю ітерацій до 10 000 кроків із застосуванням механізму ранньої зупинки після 250 ітерацій без покращення показників.

Якість побудованих моделей Random Forest та XGBoost, що представлено в таблиці 1, оцінювалася за допомогою стандартних метрик класифікації: точності (accuracy), точності позитивних прогнозів (precision), повноти (recall) та інтегральної метрики F1-score.

За результатами валідації отримано загальну точність класифікації 91.97% для моделі Random Forest та 91.8% для моделі XGBoost відповідно. Значення precision, recall та F1-score у середньозваженому обчисленні також склали 91.97% та 91.8%, що свідчить про стабільну збалансовану ефективність моделі по всіх класах. Додатковий аналіз якості класифікації здійснено із застосуванням нормалізованої матриці помилок, що відображає відносний розподіл правильних і помилкових передбачень у відсотковому вираженні по кожному класу. Отримані нормалізовані матриці помилок (див. рис. 2) підтверджують задовільний рівень класифікації, при цьому частина помилкових позитивних передбачень залишається пріоритетною зоною для вдосконалення.

Таблиця 1

Класифікаційний звіт моделі Random Forest та XGBoost

	Random Forest			XGBoost			
Клас	Precision	Recall	F1-score	Precision	Recall	F1-score	К-ть подій
0	0.91	0.94	0.92	0.92	0.93	0.93	17 529 40
1	0.90	0.86	0.88	0.89	0.86	0.87	902 698
2	0.95	0.93	0.94	0.94	0.93	0.94	1 492 354
Accuracy			0.92			0.92	4 147 992
Сер. арифм.	0.92	0.91	0.91	0.91	0.91	0.91	

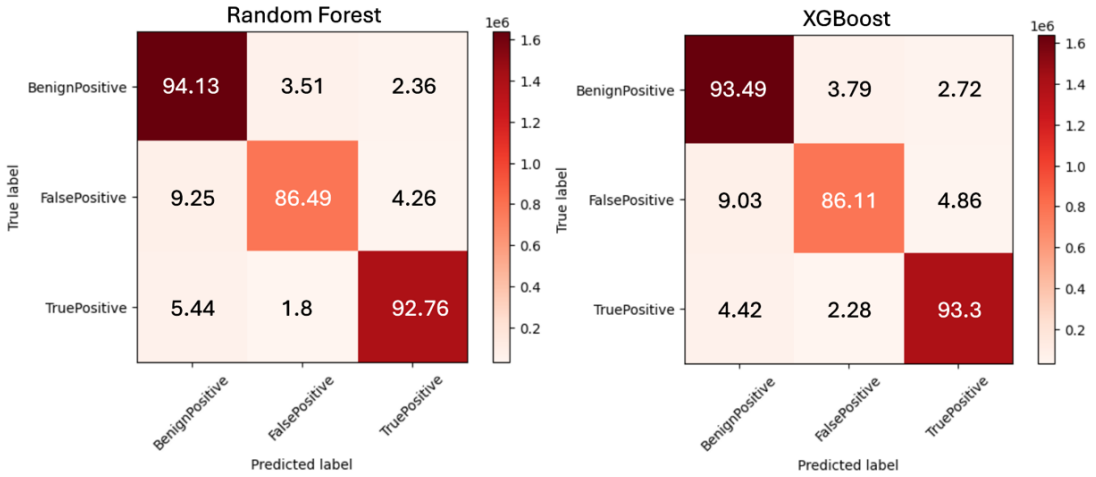

Рис.2. Нормалізовані матриці помилок моделей Random Forest та XGBoost

Висновки з даного дослідження і перспективи подальших розвідок у даному напряму

У роботі було здійснено порівняльне дослідження ефективності моделей Random Forest та XGBoost у задачі багатокласової класифікації інцидентів безпеки навчених на наборі даних Microsoft Security Incident Prediction. Обидва підходи ансамблевого навчання продемонстрували високі результати, хоча відмінності у стратегіях оптимізації обумовлюють певні локальні переваги кожного з методів при класифікації різних підкласів інцидентів. Модель Random Forest продемонструвала дещо кращу загальну точність (91.97% проти 91.8%) і вищі значення precision та F1-score для більшості класів, що свідчить про її незначну перевагу в рамках поставленого завдання. Основною перевагою Random Forest виявилась краща здатність класифікувати хибнопозитивні інциденти, що є важливим у контексті скорочення хибних спрацьовувань у системах виявлення загроз. У свою чергу, модель XGBoost виявила стабільнішу поведінку у класифікації класів TruePositive та BenignPositive, демонструючи кращу здатність розрізняти справжні загрози та безпечні події. Отримано стабільні середньозважені значення precision, recall та F1-score на рівні понад 91.8%, що вказує на збалансовану роботу обох моделей по всіх класах. Незначна різниця в підсумкових показниках свідчить про приблизно однаковий загальний рівень ефективності моделей, проте їх структурні особливості відкривають перспективи для майбутнього комбінування у гібридних рішеннях.

Для покращення результатів дослідження в майбутньому доцільно зберегти більшу кількість інформативних ознак, зокрема MitreTechniques, ThreatFamily, SuspicionLevel та LastVerdict, реалізуючи обробку пропусків шляхом введення категорії "Unknown" замість повного виключення ознак. Це дозволить моделі навчатися на повнішому представленні ситуацій та не втрачати потенційно значущу інформацію через просте вилучення колонок. Окрім цього, рекомендовано здійснити інженерію ознак з використанням поля Timestamp: витяг таких характеристик, як година доби, день тижня або сезон, може суттєво покращити сприйняття часових закономірностей у виникненні інцидентів. Текстові поля,

як-от AlertTitle та ApplicationName, варто піддавати попередній обробці за допомогою embedding-методів, зокрема TF-IDF або BERT, що дозволить отримати узагальнені векторні представлення складних семантичних ознак. Особливу увагу слід приділити питанням розширення навчальної вибірки як за обсягом, так і за варіативністю. Збільшення кількості інцидентів у навчальному наборі, особливо тих, що стосуються рідкісних класів або складних для розпізнавання шаблонів, здатне підвищити стійкість і здатність моделі до генералізації. Розширення різноманітності включених даних, зокрема з різних джерел, типів організацій або часових періодів, дозволить моделі краще вловлювати контекстуальні відмінності, адаптуватися до нових сценаріїв і знижувати ймовірність помилкової класифікації в нових умовах.

Практичне застосування запропонованих моделей передбачає їх інтеграцію у роботу центрів реагування на інциденти безпеки (Security Operation Centers, SOC). Автоматизована класифікація інцидентів дозволяє зменшити навантаження на аналітиків, пріоритизувати розслідування найнебезпечніших подій, зменшити кількість хибних спрацьовувань та пришвидшити реакцію на реальні загрози. Такі моделі можуть бути вбудовані в системи SIEM (Security Information and Event Management), SOAR (Security Orchestration, Automation and Response), платформи виявлення аномалій та моніторингу мережевої активності. Їх використання особливо ефективне в середовищах з великою кількістю подій безпеки, де необхідне масштабоване, адаптивне та прозоре рішення для попереднього сортування та ранжування інцидентів.

Література

1. Azmi, M. F., Karim, H. A., & AlDahoul, N. (2023). Anomaly detection for network security. *International Journal of Membrane Science and Technology, 10*(1), 299–316. https://doi.org/10.15379/ijmst.v10i1.1808
2. Bialas, A., Michalak, M., & Flisiuk, B. (2019). Anomaly detection in network traffic security assurance. In *Advances in Intelligent Systems and Computing* (pp. 46–56). Springer. https://doi.org/10.1007/978-3-030-19501-4_5
3. Bitdefender. (n.d.). Every hour SOCs run, 15 minutes are wasted on false positives. *Bitdefender Blog.* https://www.bitdefender.com/en-us/blog/businessinsights/every-hour-socs-run-15-minutes-are-wasted-on-false-positives (Accessed June 2, 2025)
4. Help Net Security. (2019, August 29). SOCs still overwhelmed by alert overload, struggle with false-positives. *Help Net Security.* https://www.helpnetsecurity.com/2019/08/29/soc-alert-overload/ (Accessed June 2, 2025)
5. Layman, L., & Roden, W. A. (2023). A controlled experiment on the impact of intrusion detection false alarm rate on analyst performance. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. https://doi.org/10.1177/21695067231192573
6. Alahmadi, B. A., Axon, L., & Martinovic, I. (2022). 99 % false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *31st USENIX Security Symposium*. https://www.usenix.org/system/files/sec22-alahmadi.pdf
7. Armendariz, X.-I. D. (2025). *Managing false positives in SOC operations: Solutions and best practices* (Master's thesis). https://upcommons.upc.edu/bitstream/handle/2117/424635/tfg-memoria-final.pdf?sequence=2 (Accessed June 6, 2025)
8. Zivkovic, S. (2022, August 30). #012 Machine learning – Introduction to random forest – Master Data Science. *Master Data Science.* https://datahacker.rs/012-machine-learning-introduction-to-random-forest/ (Accessed June 7, 2025)
9. Attanasi, E. D., & Coburn, T. C. (2023). Random forest. In *Encyclopedia of Mathematical Geosciences* (pp. 1182–1185). Springer. https://doi.org/10.1007/978-3-030-85040-1_265
10. Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random forest algorithm overview. *Babylonian Journal of Machine Learning, 2024*, 69–79. https://doi.org/10.58496/bjml/2024/007
11. [Author unknown]. (n.d.). Mathematics of random forests. https://math.bu.edu/people/mkon/MA751/L19RandomForestMath.pdf (Accessed June 12, 2025)
12. [Author unknown]. (n.d.). Introduction to Boosted Trees – XGBoost 3.0.2 documentation. *XGBoost Documentation.* https://xgboost.readthedocs.io/en/stable/tutorials/model.html (Accessed June 13, 2025)
13. [Author unknown]. (2018, August). Unveiling mathematics behind XGBoost. *KDnuggets.* https://www.kdnuggets.com/2018/08/unveiling-mathematics-behind-xgboost.html (Accessed June 13, 2025)
14. Chen, T., & Guestrin, C. (2016). XGBoost. In *KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. …). ACM. https://doi.org/10.1145/2939672.2939785
15. Zehra, S., et al. (2023). Machine learning-based anomaly detection in NFV: A comprehensive survey. *Sensors, 23*(11), 5340. https://doi.org/10.3390/s23115340

16. Ajila, S. A., Lung, C.-H., & Das, A. (2021). Analysis of error-based machine learning algorithms in network anomaly detection and categorization. *Annals of Telecommunications.* https://doi.org/10.1007/s12243-021-00836-0

17. Labonne, M. (2020). *Anomaly-based network intrusion detection using machine learning* (Electronic thesis/dissertation). http://www.theses.fr/2020IPPAS011

18. Esnaashari, M., & Moradi, N. (2025). Predicting vulnerability to malware using machine learning models: A study on Microsoft Windows machines. *arXiv.* https://doi.org/10.48550/arXiv.2501.02493

19. Arizal, A. F. A., et al. (2024). Performance comparative study on zero day malware detection using XGBoost and random forest classifiers. *International Journal of Innovative Computing, 14*(2), 33–40. https://doi.org/10.11113/ijic.v14n2.449

20. Kaggle. (n.d.). Microsoft security incident prediction [Dataset]. *Kaggle.* https://www.kaggle.com/datasets/Microsoft/microsoft-security-incident-prediction/data (Accessed June 19, 2025)

References

1. Azmi, M. F., Karim, H. A., & AlDahoul, N. (2023). Anomaly detection for network security. *International Journal of Membrane Science and Technology, 10*(1), 299–316. https://doi.org/10.15379/ijmst.v10i1.1808

2. Bialas, A., Michalak, M., & Flisiuk, B. (2019). Anomaly detection in network traffic security assurance. In *Advances in Intelligent Systems and Computing* (pp. 46–56). Springer. https://doi.org/10.1007/978-3-030-19501-4_5

3. Bitdefender. (n.d.). Every hour SOCs run, 15 minutes are wasted on false positives. *Bitdefender Blog.* https://www.bitdefender.com/en-us/blog/businessinsights/every-hour-socs-run-15-minutes-are-wasted-on-false-positives (Accessed June 2, 2025)

4. Help Net Security. (2019, August 29). SOCs still overwhelmed by alert overload, struggle with false-positives. *Help Net Security.* https://www.helpnetsecurity.com/2019/08/29/soc-alert-overload/ (Accessed June 2, 2025)

5. Layman, L., & Roden, W. A. (2023). A controlled experiment on the impact of intrusion detection false alarm rate on analyst performance. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. https://doi.org/10.1177/21695067231192573

6. Alahmadi, B. A., Axon, L., & Martinovic, I. (2022). 99 % false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *31st USENIX Security Symposium*. https://www.usenix.org/system/files/sec22-alahmadi.pdf

7. Armendariz, X.-I. D. (2025). *Managing false positives in SOC operations: Solutions and best practices* (Master's thesis). https://upcommons.upc.edu/bitstream/handle/2117/424635/tfg-memoria-final.pdf?sequence=2 (Accessed June 6, 2025)

8. Zivkovic, S. (2022, August 30). #012 Machine learning – Introduction to random forest – Master Data Science. *Master Data Science.* https://datahacker.rs/012-machine-learning-introduction-to-random-forest/ (Accessed June 7, 2025)

9. Attanasi, E. D., & Coburn, T. C. (2023). Random forest. In *Encyclopedia of Mathematical Geosciences* (pp. 1182–1185). Springer. https://doi.org/10.1007/978-3-030-85040-1_265

10. Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random forest algorithm overview. *Babylonian Journal of Machine Learning, 2024*, 69–79. https://doi.org/10.58496/bjml/2024/007

11. [Author unknown]. (n.d.). Mathematics of random forests. https://math.bu.edu/people/mkon/MA751/L19RandomForestMath.pdf (Accessed June 12, 2025)

12. [Author unknown]. (n.d.). Introduction to Boosted Trees – XGBoost 3.0.2 documentation. *XGBoost Documentation.* https://xgboost.readthedocs.io/en/stable/tutorials/model.html (Accessed June 13, 2025)

13. [Author unknown]. (2018, August). Unveiling mathematics behind XGBoost. *KDnuggets.* https://www.kdnuggets.com/2018/08/unveiling-mathematics-behind-xgboost.html (Accessed June 13, 2025)

14. Chen, T., & Guestrin, C. (2016). XGBoost. In *KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. …). ACM. https://doi.org/10.1145/2939672.2939785

15. Zehra, S., et al. (2023). Machine learning-based anomaly detection in NFV: A comprehensive survey. *Sensors, 23*(11), 5340. https://doi.org/10.3390/s23115340

16. Ajila, S. A., Lung, C.-H., & Das, A. (2021). Analysis of error-based machine learning algorithms in network anomaly detection and categorization. *Annals of Telecommunications.* https://doi.org/10.1007/s12243-021-00836-0

17. Labonne, M. (2020). *Anomaly-based network intrusion detection using machine learning* (Electronic thesis/dissertation). http://www.theses.fr/2020IPPAS011

18. Esnaashari, M., & Moradi, N. (2025). Predicting vulnerability to malware using machine learning models: A study on Microsoft Windows machines. *arXiv.* https://doi.org/10.48550/arXiv.2501.02493

19. Arizal, A. F. A., et al. (2024). Performance comparative study on zero day malware detection using XGBoost and random forest classifiers. *International Journal of Innovative Computing, 14*(2), 33–40. https://doi.org/10.11113/ijic.v14n2.449

20. Kaggle. (n.d.). Microsoft security incident prediction [Dataset]. *Kaggle.* https://www.kaggle.com/datasets/Microsoft/microsoft-security-incident-prediction/data (Accessed June 19, 2025)