**УДК 004.9**

# ЗАХИСТ CRM-МОДЕЛЕЙ ВІД ЗАРАЖЕННЯ ДАНИХ

О. В. Тимченко, В. О. Чорняк

*Національний Університет «Львівська Політехніка»,*
*вул. Степана Бандери, 12, Львів, 79013, Україна*

*У статті розглянуто проблему коли до даних для CRM інколи по потрапляє невелика домішка записів, які виглядають правдоподібно, проходять звичайні перевірки, але зсувають рішення моделей. У підсумку система частіше помиляється з тим, кому писати або телефонувати, яку знижку давати, як розподіляти бюджет. Це веде до зайвих витрат, гіршого утримання клієнтів і нижчої точності прогнозів.*

*В статті розглядається практичну систему CRMPGuard. Вона працює поверх існуючих процесів і робить три речі. Перше. Перевіряє походження і правдоподібність даних та відправляє підозрілі партії даних карантин для перевірки. Друге. Шукає нетипові скупчення і окремі записи, що надто впливають на навчання, та зменшує їхній внесок. Третє. Оновлює модель на очищених підмножинах у безпечному контурі і лише потім повертає рішення у роботу. Усі кроки фіксуються для аудиту і відповідності вимогам до захисту персональних даних. Якщо домішка невелика, помилка моделі збільшується приблизно у тій самій пропорції. Після відсікання підозрілих прикладів похибка зменшується. Тобто навіть за наявності домішок якість рішень тримається під контролем. Це означає менше хибних контактів і знижок не тим клієнтам, стабільнішу роботу кампаній і швидше відновлення після інцидентів.*

***Ключові слова:*** *CRM моделі, зараження даних, перевірка походження, виявлення відхилень, навчання з обмеженням впливу підозрілих записів, якість прогнозів.*

**Постановка проблеми.** Моделі CRM (керування взаєминами з клієнтами) визначають щоденні рішення про контакт, персональні пропозиції, бюджет кампаній і превенцію відтоку. Їхня точність залежить від даних. Навіть мала частка правдоподібних, але спотворених записів непомітно зміщує параметри, псує прогнози і веде до зайвих витрат та помилкових керованих дій. Це явище ми називаємо зараженням даних і відрізняємо його від розподільного дрейфу, який є природною зміною поведінки клієнтів.

Ризики з’являються уздовж усього CRM-конвеєра. Джерела включають підробні ліди у формах і мобільних застосунках, бот трафік і маніпулятивні відгуки, похибки під час збагачення і злиття профілів, вплив похідних ознак у сховищі, помилки розмітки та ефекти зворотного зв’язку під час розгортання. Узагальнену схему поверхонь ризику наведено на рис. 1.

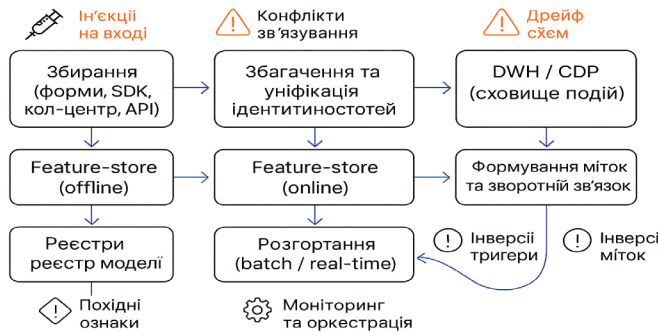


Рис. 1. Узагальнена схема CRM-конвеєра з позначеними поверхнями атаки

Рис. 1. Узагальнена схема CRM-конвеєра з позначеними поверхнями атаки

**Аналіз останніх досліджень та публікацій.** Зараження даних у CRM-системах це коли у масив чистих записів потрапляє невелика, але особлива домішка, яка виглядає правдоподібно, проходить звичайні перевірки, проте непомітно псує прогнози моделі. Це не те саме, що природні зміни у поведінці клієнтів. Там дані змінюються з часом, а тут хтось або щось підмішує записи так, щоб зсунути рішення у хибний бік [1-5].

Зазвичай захист будується на трьох рівнях. Рівень даних це перевірка джерел, змісту і часу подій, пошук дублювань, карантин підозрілих партій. Рівень навчання це навчання так, щоб окремі підозрілі записи впливали менше, а після оновлення робимо додаткове калібрування й перевірку [6].

Відоме перенавчання як тренування нової версії моделі окремо від робочої, з подальшим порівнянням результатів [7]. Використовують стійкі способи оцінювати помилку, щоб одиничні зіпсовані точки не змінювали модель. Оцінюють внесок кожного запису, щоб підозрілі впливали менше. [8-10].

**Метою роботи** є розроблення методики захисту CRMPGuard шляхом запобігання зараження CRM моделі незначними домішками, розробка метрики виявлення зараження основі збуреного емпіричного ризику та розбіжностей між чистим і зараженим розподілом даних.

**Виклад основного матеріалу дослідження.**

**Модель загроз і формальне означення «зараження даних».**

Рамка захисту CRMPGuard поєднує перевірку походження і правдоподібності даних з карантином підозрілих партій, багатоканальне виявлення відхилень на рівні ознак і записів та стійке навчання із зменшенням внеску підозрілих прикладів. Інтеграція сумісна з процесами MLOps і вимогами захисту даних [11].

Маємо три сторони для формування CRMPGuard. Перша це CRM-система з етапами збору подій, збагачення, сховищем ознак, навчанням і розгортанням. Друга це той хто підмішує домішку. Третя це постачальники даних через яких це може статися. Точки появи домішки прості – форми на сайті або мобільному застосунку, етап збагачення і злиття профілів, сховище ознак з потоковою та пакетною гілками,

формування міток і вибір батчів для навчання, повернення впливу моделі назад у дані під час експлуатації.

Формально ми вважаємо що до чистого розподілу даних $P$ потрапляє невелика домішка $Q$ з часткою $\varepsilon$. Тоді модель бачить суміш $P_\varepsilon = (1-\varepsilon)\ P + \varepsilon Q$ і її помилка рахується на цій суміші:

$$R_\varepsilon(f) = \mathrm{E}_{(x,y)\sim P_\varepsilon}\left[\ell(f(x),y)\right] = (1-\varepsilon)\mathrm{E}_P\left[\ell(f(x),y)\right] + \varepsilon\mathrm{E}_Q\left[\ell(f(x),y)\right]. \quad (1)$$

Тобто загальна помилка це майже помилка на чистих даних плюс невелика частина помилки на домішці. Якщо ε = 0.02 то на кожні сто записів два можуть бути домішкою і їхній внесок у середню помилку становить приблизно два відсотки суміші.

Щоб говорити про правдоподібні домішки ми не дозволяємо їм різко відрізнятися від нормальних даних. Усі допустимі $Q$ мають лежати всередині кулі радіуса ρ:

$$\begin{aligned} \mathrm{B}_f(P,\rho) &= \left\{Q : D_f(Q\|P) \le \rho\right\}, \\ D_f(P_\varepsilon\|P) &= \mathrm{E}_P\left[f\left(\frac{dP_\varepsilon}{dP}\right)\right] \le \rho. \end{aligned} \quad (2)$$

Тут: $P$ – чисті дані, $Q$ – домішка, $\varepsilon$ – частка домішки, $\ell$ – помилка на одному прикладі, $R_\varepsilon(f)$ – середня помилка на суміші, $f$ – функція, що задає міру відмінності між розподілами, $\rho$ – межа допустимої відмінності.

За допомогою параметра $\rho$ дозволяємо лише такі домішки, що виглядають як звичайні дані за форматами і частотами. Мета порушника це збільшити помилку або зіпсувати узгодженість прогнозів загалом чи у вибраних групах клієнтів. Мета захисту: тримати $R_\varepsilon(f)$ низьким і стежити щоб відхилення за межі $\rho$ не проходили непоміченими. Для цього використовуємо індикатори відхилень, карантин підозрілих партій і навчання з меншими вагами для підозрілих записів

***Архітектура CRM-конвеєра та поверхні атаки***

Розглянемо, як дані проходять шлях від події клієнта до рішення моделі. Конвеєр складається з послідовних кроків. Спочатку відбувається збирання подій із форм на сайті, з мобільних застосунків, з кол-центру та через партнерські інтерфейси. Тут найчастіше і з'являються підробні звернення, бот-трафік і фальшиві підтвердження дій, бо подію створює сама людина або партнер і це важко відрізнити від справжнього сигналу [12].

Далі йде збагачення і уніфікація ідентичностей. Це крок, де записи з різних джерел з'єднують у один профіль. Помилка тут створює плутанину, коли дані різних людей зливаються в один профіль або один клієнт раптом має два профілі. Такі збої непомітні, але згодом впливають на ознаки та навчання. Потім події потрапляють у сховище, де зберігається історія. Ризик полягає у зміні схем полів, назви та типи можуть змінитися без попередження, і тоді розрахунки починають давати інший результат, хоча зовні все виглядає як завжди.

Після цього формуємо набір ознак з двох гілок. У пакетній гілці рахують агрегати за вікнами часу. У потоковій гілці ознаки оновлюються одразу після події. Навіть невелика домішка на вході може помітно змінити такі агрегати. У потоковій

гілці інколи спрацьовують приховані тригери, коли рідкісна послідовність подій змушує модель реагувати не так, як слід. Окремо формується мітка, тобто відповідь, на якій навчають модель. Тут небезпека у навмисних або випадкових інверсіях, коли мітка не відповідає реальній події. Далі йде навчання і валідація, зберігання артефактів і самих моделей у реєстрах. Якщо валідація налаштована неправильно або у вибірку потрапляють одні й ті самі впливові записи, модель видається точною на перевірці, але помиляється в роботі.

Останній крок це розгортання. Рішення віддаються пакетами або в реальному часі. Тоді виникають петлі зворотного зв'язку. Модель змінює комунікації з клієнтами, ці зміни повертаються у дані, і викривлення підсилюється. Під навантаженням сервіси можуть деградувати, і аномалії стають менш помітними [13-15].

На рис. 1 показано мінімальну схему такого конвеєра. Над вузлами позначені типові місця ризику. Ін'єкції на вході з'являються при збиранні подій. Конфлікти зв'язування виникають під час збагачення і уніфікації. Дрейф схем стосується сховища подій. Похідні ознаки вказують на чутливість агрегатів у сховищі ознак. Інверсії тригерів стосуються потокової гілки. Інверсії міток стосуються етапу формування відповідей. Окремо показано моніторинг і оркестрацію, які з'єднують усі кроки і дають змогу відтворити шлях даних та вчасно зупинити підозрілий сегмент. Така декомпозиція потрібна, щоб чітко бачити, де саме може з'явитися домішка і як її зупинити до того, як вона вплине на рішення моделі.

**Методологія захисту.** Запропонована методологія працює як захист у глибину і вбудовується у наявний CRM-конвеєр без перебудови. Мета – своєчасно помічати домішку у даних, зменшувати її вплив під час навчання, безпечно повертати систему до стабільної роботи. На рис. 2 показано три шари цієї побудови, зовнішній шар відповідає за попередні перевірки, середній за виявлення відхилень, внутрішній за стійке навчання та реагування.


Рис. 2. Багатошарова схема захисту

Спочатку діють попередні перевірки. Фіксуються джерела подій, наявність підписів або токенів, часові мітки, схеми полів, узгодженість ідентифікаторів. Для кожного постачальника визначається еталонний профіль значень і порівнюється

поточний розподіл із референтним. Партії даних, що виходять за звичні межі, спрямовуються у карантин з обмеженим використанням у навчанні. Усі перетворення записуються у журнал походження, що спрощує аудит і дотримання правил захисту даних.

Далі працює виявлення відхилень як ансамбль незалежних сигналів. Використовуються стійкі статистики для агрегованих ознак, вони менше реагують на поодинокі дивні значення, показують нетипові напрями у багатовимірних даних і підсвічують штучні скупчення та приховані тригери. Перевіряється локальна узгодженість міток та ознак за щільнісними співвідношеннями. Для індивідуальних записів розрахується показник впливу, що допомагає швидко підняти найпроблемніші точки в черзі на перегляд і карантин:

$$s(z) = \left\| H^{-1} \nabla_\theta \left[ \ell(f_\theta(x_z), y_z) \right] \right\|_2, \quad H = \nabla_\theta^2 R_{emp}(f_\theta) + \lambda \nabla_\theta^2 \Omega(f_\theta). \tag{3}$$

Значення $s(z)$ це сила з якою один запис здатний змінити параметри моделі. Матриця $H$ відображає кривизну цільової функції та стабілізуючий штраф, тому зменшує вплив одиничних дивних точок. Чим більше $s(z)$, тим вищий пріоритет для перевірки і тим імовірніше ізоляція такого запису. Виходи всіх детекторів нормуються і зводяться в один індикатор ступеня зараження, пороги налаштовуються за квантилями на чистих періодах, що допомагає тримати під контролем хибні тривоги.

Наступний крок це стійке навчання, яке зменшує внесок підозрілих прикладів без втрати точності на чистих даних. Використовується цільова функція з вагами для окремих прикладів і з порогом великої помилки $\tau$:

$$L_\tau(f) = \frac{1}{n} \sum_{i=1}^{n} w_i(\tau) \ell(f(x_i), y_i), \tag{4}$$

де $w_i(\tau) = 1\{\ell(f(x_i), y_i) \le \tau\}$ - обрізаний ризик, $w_i(\tau) = \Psi\left( \frac{\ell(f(x_i), y_i)}{\tau} \right)$ - м'яке обрізання типу Х'юберта.

Поріг $\tau$ відсікає приклади з надто великою помилкою, такі приклади просто не враховуються. У варіанті типу Х'юбера (швейцарського статистика, який запропонував робастну втрату, корисну коли у даних трапляються викиди) їхній вплив плавно зменшується за допомогою функції ψ, що запобігає різким стрибкам під час навчання. Поріг $\tau$ налаштовується на чистих відрізках історії, ваги $w_i$ вибиоаються залежними від інтегрального індикатора ступеня зараження, тому підозрілі групи автоматично впливають менше. Після оновлень виконується перевірка калібрування, тобто вирівнювання передбачених імовірностей з фактичними частотами.

Реагування організовано як керована послідовність кроків. Якщо інтегральний індикатор перевищує поріг, ізолюються відповідні канали та партії, виконується відкат до останньої контрольної точки, запускається тіньове перенавчання на очищених підмножинах, результати порівнюються з робочою версією у форматі A/B, після підтвердження стабільності рішення повертається в експлуатацію. За наявності підозри на приховані тригери у потоковій гілці додатково санітуються

відповідні ознаки і проводяться прості стрес тести. Усі дії автоматично протоколюються для аудиту.

**Організація CRMPGuard.** В CRMPGuard усі дії зведено в один виробничий контур без перебудови процесів. Спочатку йде моніторинг походження даних і ансамблеве виявлення відхилень, далі інтегральний індикатор ступеня зараження, після умовного переходу працює блок реагування з обмеженням впливу підозрілих записів, тіньовим навчанням і керованим відкатом (рис. 3). Верхня частина рисунка відповідає моніторингу та виявленню. Нижня частина показує реагування і відновлення. Пунктир між ними позначає порогове рішення за індикатором. Якщо індикатор нижче порога, конвеєр працює у штатному режимі. Якщо індикатор вище порога, відбувається перехід у нижній блок. Там послідовно виконуються адаптивне відсікання, тіньове навчання з карантином даних, відновлення або відкат. Це – керований контур CRMPGuard.

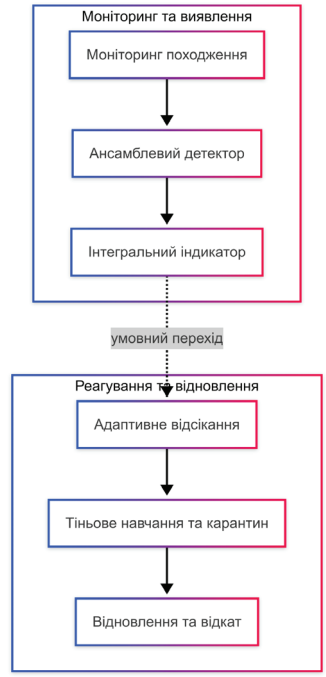


Рис. 3. CRM-конвеєр

Алгоритмічне ядро використовує дві групи сигналів. Перша група виявляє нетипові структури у просторах ознак за спектральними підписами та локальними перевірками узгодженості. Друга група оцінює внесок окремих записів у зміну параметрів моделі за допомогою показника впливу $s(z)$. Сигнали нормуються, усереднюються у часі за ковзними квантилями і зводяться в інтегральний індикатор. Це дає просту шкалу ризику від чистого режиму до підозри на зараження.

Неперервний контроль індикатора виконується накопичувальним тестом зміни з контрольованою частотою хибних тривог.

$$R_t = (1 + R_{t-1})\Lambda_t;\ \Lambda_t = \frac{f_1(z_t)}{f_0(z_t)};\ R_0 = 0; \quad \text{сигнал при } R_t \geq A(\alpha). \tag{5}$$

Величина $R_t$ накопичує випадки про перехід від нормального стану до зараженого. Множник $\Lambda_t$ є відношенням правдоподібностей. $f_0$ описує типові значення індикатора у чистому режимі. $f_1$ описує значення, які очікуємо при зараженні. Обидві щільності оцінюємо з даних. Для $f_0$ беремо чисті періоди. Для $f_1$ використовуємо підсилені зрізи або робастні наближення. Поріг $A(\alpha)$ задає бажану частоту хибних тривог. Щоб короткі піки не заважали, застосовується незначне згладжування поверх $R_t$.

Режим реагування реалізує адаптивне відсікання підозрілих прикладів у навчанні, тіньове перенавчання на очищених підмножинах і A/B порівняння зі сталою версією. Якщо нова версія стабільніша і краще відкалібрована, її повертають у роботу. Інакше виконується відкат до останньої контрольної точки. Для ризику прихованих тригерів додатково очищаються відповідні ознаки, проводяться прості стрес тести і перевіряється сталість рішень на синтетичних патернах.

**Верифікація стійкості та модельні приклади.**

У цьому розділі показуємо як працює детектор у контрольованих умовах і що означають його налаштування. Припускаємо невелику домішку у даних, стабільні правила формування ознак і регуляризацію, яка стримує різкі зміни параметрів. Індикатор ступеня зараження будується з кількох сигналів і нормується за чистими періодами. Пороги беремо з цих самих періодів, щоб кількість хибних тривог залишалася керованою. Далі використовуємо стандартну оцінку якості детектора ROC, це крива чутливість проти рівня хибної тривоги.

Орієнтири для практики такі. Для $\varepsilon = 0.5\ldots3$ % і при $\alpha = 1\ldots5\%$ середня затримка становить від двох до шести кроків. Це зручно для оперативних рішень у виробництві. Якщо потрібно забезпечити рідкісну хибну тривогу, наприклад $\alpha<1\%$, затримка буде довшою. Її можна частково зменшити посиленням чутливості у спектральному модулі та в оцінках впливу, але потрібно враховувати додаткові спрацювання.

Модельні сценарії відповідають трьом типам ризику у CRM середовищі: локальне інвертування міток у вузькому сегменті, приховані тригери у потоковій гілці ознак, колективні маніпулятивні відгуки ззовні. У першому випадку поверхня найкрутіша, бо з'являється виразний локальний напрям у просторі ознак. У другому випадку допомагає квазіперіодичність подій, індикатор накопичує докази тригерів і затримка падає вже при помірних значеннях $\varepsilon$. У третьому випадку головну роль відіграє концентрація енергії у верхніх сингулярних компонентах, тому чутливість зростає після короткого періоду адаптації.

Рис. 4 узагальнює поведінку детектора. Поверхня показує компроміс. Якщо дозволяємо більший α, сигнал зростає швидше і затримка зменшується. Якщо $\varepsilon$ більша, детектор також спрацьовує швидше. Коли ε майже нульова, затримка природно зростає, бо зміни ледь помітні.

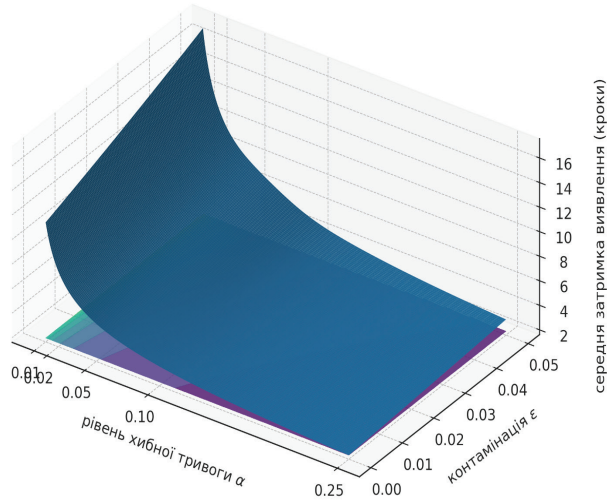


Рис. 4. ROC-криві детектора при різних ε

Робочу точку зручно вибирати за рис. 4. Спочатку визначаємо прийнятний максимум хибних тривог. Потім оцінюємо очікувану частку домішки у нашому домені. Перетин цих двох величин на поверхні дає прогнозну затримку. Якщо вона надто велика, підвищуємо α у розумних межах або збільшуємо вагу сигналів в інтегральному індикаторі. Якщо затримка задовільна, фіксуємо пороги і використовуємо їх у правилах переходу між штатним режимом, підвищеною увагою і критичним станом. Додаємо нагадування з оригіналу. Приріст ризику моделі масштабується приблизно лінійно з ε, а очікувана похибка калібрування зменшується до порядку O($\varepsilon$) після адаптивного відсікання. Це узгоджується з поведінкою поверхні на рисунку і пояснює, чому захисні кроки тримають якість під контролем.

**Аналітичні результати.**

Узагальнимо ефект CRMPGuard як вибір порога обрізання впливає на якість і калібрування. Припускаємо стандартні умови стабільності навчання тоді приріст ризику моделі масштабується приблизно лінійно з часткою домішки $\varepsilon$. Після адаптивного відсікання підозрілих прикладів очікувана похибка калібрування як узгодженість прогнозованих імовірностей з фактичними частотами, зменшується до O($\varepsilon$).

Рис. 5 демонструє два пов'язані ефекти. На верхньому зображено якість за AUC, (area under curve), тобто площа під ROC кривою, ROC це крива чутливість проти рівня хибної тривоги, у функції порога обрізання $\tau$ для різних рівнів контамінації $\varepsilon$. Крива має виразну вершину біля $\tau^*$. У цій точці якість максимальна на чистих даних і залишається стабільною за малою домішкою. Якщо τ зменшити дуже сильно, відсіюється багато корисних прикладів і якість падає. Якщо τ збільшити дуже сильно, підозрілі приклади впливають на навчання і якість теж погіршується.

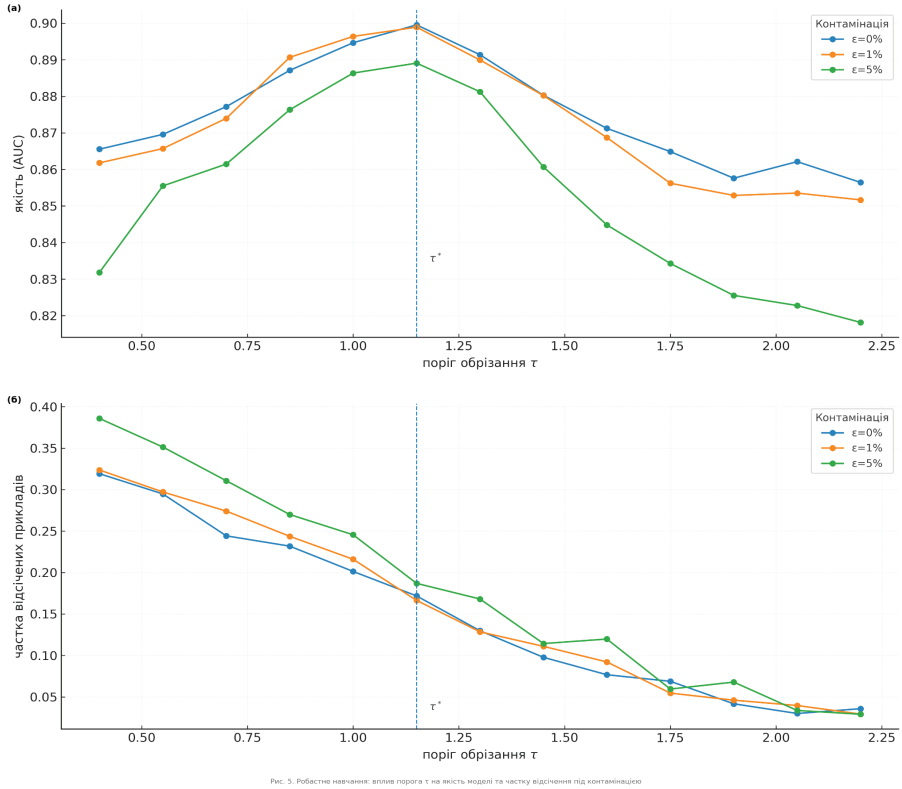


Рис. 5. Калібрування до/після застосування CRMPGuard

Нижня частина рис. 5 показує частку відсічених прикладів у функції τ. Чим більший τ, тим менше прикладів прибираємо. У точці $\tau^*$ частка відсічених прикладів помірна, водночас верхня панель фіксує максимум AUC. Для більшої домішки $\varepsilon$ крива якості опускається, однак максимум усе ще спостерігається поблизу тієї самої робочої точки. Це підтверджує, що вибір порога за чистими періодами переноситься на помірні рівні зараження.

Практично корисно тримати $\tau$ у вузькому околі $\tau^*$. Такий вибір дає хороший компроміс між точністю на чистих даних і стійкістю у присутності домішки. Після навчання варто виконувати перевірку калібрування і корекцію, щоб діаграма надійності наближалася до бісектриси, а показник ECE (expected calibration error) зменшувався. У сукупності ці кроки підтверджують головний висновок (рис. 5): якість і калібрування залишаються керованими, а залежність помилки від ε має передбачуваний вигляд.

**Обговорення.**

Отримані межі стійкості і приклади показують, що вплив домішків можна тримати під контролем за коректних налаштувань. Водночас висновки спираються на кілька передумов. Використано обмежені втрати і регуляризацію, яка робить мінімум цільової функції чітким та стійким до шуму. Припускаємо, що числові ознаки

мають скінченні другі моменти. У глибоких мережах це не завжди виконується, а у даних що мають розподіли з важкими хвостами великі значення трапляються частіше. У таких випадках варто додатково оцінювати кривизну поверхні втрат локально, віддавати перевагу розв'язкам із плоским мінімумом, застосовувати обмеження за дистанцією Вассерштейна, тобто за мірою близькості між розподілами, а також використовувати медіанні та інші стійкі оцінки, де медіана замінює середнє і менш чутлива до викидів.

Індикатор ступеня зараження зручний у щоденній роботі, проте потребує регулярної перевірки. Потрібно оновлювати еталонні чисті періоди, перевіряти стабільність списку записів за показником впливу та стежити за узгодженістю між потоковою і пакетною гілками ознак. Якщо застосовується карантин підозрілих партій, слід запобігати накопиченню перекосу. Для цього корисні короткі розморожування, введення нейтральних контрольних прикладів і перевірка, чи не з'явилася залежність рішень від самого факту ізоляції даних.

Практичний компроміс задається порогом для інтегрального індикатора та порогом обрізання під час навчання. Малий поріг підвищує чутливість і дає більше спрацювань. Великий поріг зменшує навантаження, але підвищує ризик пропустити слабку домішку. Доцільно тримати обидва пороги поряд із значеннями, отриманими на чистих періодах, і перевіряти їх на відкладених зрізах. За наявності сезонності або акцій варто мати окремі еталони для різних режимів трафіку.

З погляду відповідності вимогам до захисту персональних даних важлива простежуваність. Журнал походження має зберігати джерела даних, перетворення, версії ознак і моделей, а також рішення про ізоляцію та повернення у роботу. Це дозволяє відтворити повну послідовність від даних до дії, оцінити вплив домішки і обґрунтувати вибір порогів.

У підсумку CRMPGuard надає керований спосіб зменшувати вплив домішки у виробничих CRM середовищах. Межі стійкості пояснюють, чому приріст ризику масштабується з часткою домішки, а поєднання індикатора, обрізаних і м'яко обрізаних втрат та контрольованого реагування підтримує стабільність калібрування і прогнозів у реальних умовах.

**Висновки та напрями подальших робіт.** У роботі формалізовано загрозу зараження у CRM конвеєрах, введено метрики впливу через зміну ризику і близькість розподілів, подано рамку CRMPGuard, яка поєднує перевірку походження, ансамблеве виявлення і стійке навчання з перевіреним циклом реагування. З моделі встановлено межі стійкості. Приріст ризику масштабується приблизно лінійно з часткою домішки $\varepsilon$, а очікувана похибка калібрування ECE, зменшується до порядку $O(\varepsilon)$ після адаптивного відсікання і корекції. Операційний контур з карантином, тіньовим навчанням і перевіркою у двох гілках забезпечує відтворюваність і простежуваність.

Обмеження пов'язані з припущеннями про поведінку функції помилки і стабільність регуляризації, а також з характером оцінювання впливу окремого запису. У даних з важкими хвостами або за різких змін домішки потрібні додаткові перевірки і налаштування.

Подальші роботи включають сертифіковані гарантії для нелінійних моделей і для обмежень за відстанню Вассерштейна, спільну оптимізацію порогів детектора та порога обрізання як єдиної задачі, інтеграцію диференційної приватності, тобто способу захисту даних з контрольованим шумом, з аудитом походження, навчання з урахуванням справедливості, онлайн оновлення під нестаціонарною часткою домішки з адаптивним виявленням змін, включення каузальних цілей, тобто цілей про причинний ефект у таргетингових рішеннях, публікацію відкритих синтетичних потоків CRM для відтворюваних бенчмарків. Ці напрямки мають зміцнити переносимість CRMPGuard і розширити діапазон умов, у яких гарантії стійкості зберігаються.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. P. Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” in EUROCRYPT 1999, LNCS 1592, 1999.
2. T. ElGamal, “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” IEEE Transactions on Information Theory, vol. 31, no. 4, 1985.
3. C. Gentry, “Fully Homomorphic Encryption Using Ideal Lattices,” in Proceedings of STOC, 2009.
4. C. Gentry, A Fully Homomorphic Encryption Scheme, Ph.D. thesis, Stanford University, 2009.
5. Z. Brakerski and V. Vaikuntanathan, “Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages,” in CRYPTO, 2011.
6. Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(Leveled) Fully Homomorphic Encryption without Bootstrapping,” in ITCS, 2012.
7. J. Fan and F. Vercauteren, “Somewhat Practical Fully Homomorphic Encryption,” IACR ePrint 2012/144, 2012.
8. Z. Brakerski, “Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP,” in CRYPTO, 2012.
9. J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic Encryption for Arithmetic of Approximate Numbers,” in ASIACRYPT, 2017.
10. L. Ducas and D. Micciancio, “FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second,” in EUROCRYPT, 2015.
11. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, “TFHE: Fast Fully Homomorphic Encryption over the Torus,” Journal of Cryptology, vol. 33, 2020.
12. S. Halevi and V. Shoup, “HElib – Algorithms and Design,” IBM Research Report, 2014–2020.
13. Microsoft Research, Microsoft SEAL: Simple Encrypted Arithmetic Library, 2015–2024.
14. C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating Noise to Sensitivity in Private Data Analysis,” in TCC, 2006.
15. V. Costan and S. Devadas, “Intel SGX Explained,” Foundations and Trends in Electronic Design Automation, 2016.

**REFERENCES**

1. P. Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” in EUROCRYPT 1999, LNCS 1592, 1999.
2. T. ElGamal, “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” IEEE Transactions on Information Theory, vol. 31, no. 4, 1985.
3. C. Gentry, “Fully Homomorphic Encryption Using Ideal Lattices,” in Proceedings of STOC, 2009.
4. C. Gentry, A Fully Homomorphic Encryption Scheme, Ph.D. thesis, Stanford University, 2009.
5. Z. Brakerski and V. Vaikuntanathan, “Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages,” in CRYPTO, 2011.
6. Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(Leveled) Fully Homomorphic Encryption without Bootstrapping,” in ITCS, 2012.
7. J. Fan and F. Vercauteren, “Somewhat Practical Fully Homomorphic Encryption,” IACR ePrint 2012/144, 2012.
8. Z. Brakerski, “Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP,” in CRYPTO, 2012.
9. J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic Encryption for Arithmetic of Approximate Numbers,” in ASIACRYPT, 2017.
10. L. Ducas and D. Micciancio, “FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second,” in EUROCRYPT, 2015.
11. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, “TFHE: Fast Fully Homomorphic Encryption over the Torus,” Journal of Cryptology, vol. 33, 2020.
12. S. Halevi and V. Shoup, “HElib – Algorithms and Design,” IBM Research Report, 2014–2020.
13. Microsoft Research, Microsoft SEAL: Simple Encrypted Arithmetic Library, 2015–2024.
14. C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating Noise to Sensitivity in Private Data Analysis,” in TCC, 2006.
15. V. Costan and S. Devadas, “Intel SGX Explained,” Foundations and Trends in Electronic Design Automation, 2016.

**doi: 10.32403/1998-6912-2025-2-71-13-25**

## PROTECTING CRM MODELS FROM DATA INFECTION

O. V. Tymchenko, V. O. Chornyak

*Lviv Polytechnic National University,*
*12, Stepana Bandery St, Lviv, 79000, Ukraine*
*oleksandr.v.tymchenko@lpnu.ua, vovan41nsou@gmail.com*

*The article discusses a simple but dangerous problem. CRM data sometimes contains a small admixture of records that look plausible, pass the usual checks, but quietly*

*shift the model solutions. Since event collection occurs from forms on the website, from mobile applications, from the call center or through partner interfaces, this is where fake calls, bot traffic and false confirmations of actions most often appear, because the event is created by a person or a partner and it is difficult to distinguish it from a real signal. As a result, the system is more likely to make mistakes about who to write or call, what discount to give, how to distribute the budget. This leads to unnecessary costs, worse customer retention and lower forecast accuracy. We offer a practical CRMP-Guard framework. It works on top of existing processes and does three things. First. Checks the origin and plausibility of the data and sends suspicious batches to quarantine for verification. Second. It looks for atypical clusters and individual records that have too much impact on training, and reduces their contribution. Third. It updates the model on cleaned subsets in a safe loop, compares the results in two branches, and only then returns the solution to work. All steps are recorded for audit and compliance with personal data protection requirements.*

*We present the results in an understandable form. If the impurity is small, the model error increases in approximately the same proportion. After cutting off suspicious examples, the error decreases. That is, even in the presence of impurities, the quality of solutions is kept under control. Example. About one percent of fake reviews appear in a small segment. The indicator increases, suspicious records are isolated, the model is retrained on cleaned data, and a check in two branches shows the restoration of the accuracy and consistency of predictions. This means fewer false contacts and discounts to the wrong customers, more stable campaign operation, and faster recovery from incidents.*

***Keywords:*** *CRM models, data contamination, origin verification, anomaly detection, learning with limited exposure to suspicious records, prediction quality.*

*Стаття надійшла до редакції 15.10.2025.*
*Received 15.10.2025.*