ВЫЯВЛЕНИЕ АНОМАЛЬНОГО ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ ВУЗОВ

Котенко И.В.[1], Саенко И.Б.[2], Аль-Барри М.Х.[3]

***Ключевые слова**: информационная безопасность, инсайдер, компьютерная атака, база данных, SQL-инъекция, машинное обучение, классификатор, регистрационный журнал, набор данных.*

Аннотация

Цель: сформировать постановку задачи обеспечения требуемой точности выявления аномального поведения пользователей центров обработки данных вузов и получить ее решение с помощью методов машинного обучения.

Методы: системный анализ проблемы выявления аномального поведения пользователей центров обработки данных вузов и методы контролируемого машинного обучения.

Результаты: предложена оригинальная формальная постановка задачи обнаружения аномальных действий пользователей центров обработки данных вуза, ориентированная на применение методов машинного обучения; разработан подход к снижению размерности первоначального признакового пространства и реализующий его алгоритм, который основан на типизации имен таблиц данных, присутствующих в тексте SQL-запросов; проведена реализация предложенного подхода с использованием множества моделей машинного обучения; выполнена экспериментальная оценка предложенного подхода, которая подтвердила его высокую эффективность и позволила выявить наиболее приемлемые для решения данной задачи классификаторы, которыми являются дерево решений, метод k-ближайших соседей и многослойная нейронная сеть.

DOI: 10.21681/1994-1404-2023-1-62-71

Введение

В результате широкого распространения облачных сервисов существенно возросла популярность использования *центров обработки данных* (ЦОД) [1] в системах управления. ЦОД являются, по сути, хранилищами больших массивов разнородной информации. Они обеспечивают своим пользователям возможность совместного устойчивого и своевременного использования информационных ресурсов в интересах решения различных задач [2, 3]. Нарушители безопасности ЦОД могут быть как внутренними, так и внешними [4, 5]. Внутренние нарушители (инсайдеры) оказывают негативное влияние на безопасность ЦОД путем выполнения вредоносных действий, которые не удается зафиксировать имеющимися средствами защиты. Внешние нарушители, как правило, оказывают негативное влияние на безопасность ЦОД с помощью компьютерных атак различного вида.

При построении систем защиты информации ЦОД могут использоваться различные методы поиска аномалий, обладающие той или иной степенью эффективности. Обычно аномалии обнаруживаются в *сетевом трафике*. Для этой цели разработаны и используются различные сетевые средства защиты, например, системы обнаружения вторжений, межсетевые экраны, антивирусные средства и др. Однако аномалии сетевого трафика не в полной мере отражают неправильное или аномальное поведение пользователей при работе с *базами данных* ЦОД. Аномальное поведение пользователей ЦОД проявляется в виде обращения к базам данных

[1] ***Котенко Игорь Витальевич,*** *доктор технических наук, профессор, главный научный сотрудник и руководитель лаборатории проблем компьютерной безопасности Санкт-Петерургского Федерального исследовательского центра Российской академии наук, г. Санкт-Петербург, Российская Федерация.*
E-mail: ivkote@comsec.spb.ru

[2] ***Саенко Игорь Борисович,*** *доктор технических наук, профессор, ведущий научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского Федерального исследовательского центра Российской академии наук, г. Санкт-Петербург, Российская Федерация.*
E-mail: ibsaen@comsec.spb.ru

[3] ***Аль-Барри Мазен Хамед,*** *адъюнкт Военной академии связи имени маршала Советского Союза С.М. Буденного, г. Санкт-Петербург, Российская Федерация.*
E-mail: mazenb51@gmail.com

с неправильными, аномальными запросами. Аномальные запросы могут быть сформированы специальным образом, позволяющим произвести либо вредоносное изменение содержимого баз данных, либо несанкционированный доступ (НСД) к информации баз данных. Такие запросы являются особым типом компьютерных атак, которые называются *SQL-инъекциями* [6]. Кроме того, аномальные запросы могут иметь обычный вид и не содержать *SQL*-инъекций, но обращаться к неразрешенным областям баз данных. Защита от таких обращений обычно возлагается на имеющуюся в ЦОД систему разграничения доступа к базам данных. Для этого могут использоваться различные модели контроля доступа, такие как *ролевая* (*Role-Based Access Control*) или *атрибутивная* (*Attribute-Based Access Control*). Однако для сложных баз данных, содержащих большое количество таблиц данных, построить схему контроля доступа, полностью запрещающую аномальные запросы, является сложной задачей.

Предлагаемый *подход*, основанный на машинном обучении, предназначен для решения данной задачи. При этом в качестве исходных данных, на основе которых формируются применяемые в методах машинного обучения наборы данных, предлагается использовать регистрационные журналы баз данных. В этих журналах фиксируются тексты запросов, с которыми пользователи обращались к базам данных. Если базы данных основаны на *реляционной* модели, то запросы записываются на языке *SQL*. Если в ЦОД присутствуют базы данных других моделей (так называемые *NoSQL* базы данных), то возможна запись запросов на других языках.

В статье рассматривается возможность применения методов машинного обучения для обнаружения аномального поведения пользователей в ЦОД, используемого для хранения информации и решения на ее основе задач, связанных с учебным процессом в высшем учебном заведении (вузе). Выбор такого типа ЦОД объясняется двумя причинами [7]. *Во-первых*, в базах данных для учебного процесса содержится очень большое количество таблиц данных. Так, отдельные таблицы создаются для каждого преподавателя, каждого студента и для каждой учебной дисциплины. В результате, если использовать имена таблиц для формирования признакового пространства, то количество признаков в нем будет очень большим. Это делает применение машинного обучения невозможным или чрезвычайно затруднительным. *Во-вторых*, в ЦОД вуза существует достаточно большое количество инсайдерских угроз безопасности. В качестве потенциальных инсайдеров можно рассматривать студентов, многие из которых через некоторое время обучения начинают обладать навыками работы с *SQL*-запросами. Поэтому разработка для ЦОД вуза дополнительного рубежа безопасности, позволяющего обнаруживать аномальные запросы к его базам данных, является актуальной задачей. При этом следует заметить, что тематике обнаружения или анализа возможного вредоносного поведения пользователей ЦОД в настоящее время посвящено недостаточное количество работ.

Таким образом, предлагается новый подход к обнаружению аномального поведения пользователей ЦОД, основанный на использовании методов машинного обучения и их применении к регистрационным журналам баз данных.

Обзор работ по тематике исследования

Работы, связанные с тематикой обнаружения аномалий в работе ЦОД, можно разделить на две группы: по обнаружению аномалий в функционировании ЦОД и по обнаружению компьютерных атак типа *SQL*-инъекции.

Во всех работах *первой* группы подчеркивается, что процедуры обнаружения аномалий в работе ЦОД основываются на анализе регистрационных журналов. В [8] обращается внимание на то, что записи журналов в ЦОД являются стохастическими и нестационарными по своей природе. Поэтому эта работа предлагает подход, в котором атрибуты извлекаются из временных окон и используются для обучения и дообучения «на лету» классификатора, задействованного в процедуре анализа данных, в качестве которого используется развивающийся нечеткий классификатор Гаусса. В [9] для извлечения признаков из записей регистрационных файлов предлагается использовать методы обработки естественного языка, в частности, алгоритм *word2vec*, а для обнаружения аномальных регистрационных записей — автокодировщик с нейронной сетью вида *LSTM*. Это, несомненно, продуктивная идея, однако ее применение для обнаружения аномальных запросов к базам данных ЦОД вуза приводит к существенному увеличению размерности признакового пространства. Это значительно снижает эффективность применения методов глубокого обучения, к которым относятся *LSTM*-сети, и, в частности, значительно увеличивает время, требуемое на их обучение.

В [10] предлагается использовать методы неконтролируемого (*unsupervised*) машинного обучения для определения нормального и ненормального поведения систем охлаждения в ЦОД. Вопросы предотвращения враждебного влияния на обнаружение аномалий в ЦОД рассматриваются в [11]. Эта работа предлагает подход к оптимизации модели линейной регрессии с возможностью изменять данные на этапе обучения. В [12] предлагается выявлять аномалии в работе ЦОД путем сопоставления отклонений прогнозных и реальных данных с использованием различных методов машинного обучения. В [13] для обнаружения и классификации атаки в сетевом трафике ЦОД предлагается использовать модели линейной регрессии и случайного леса.

Несмотря на хорошие результаты, полученные в указанных выше работах по обнаружению аномалий, свойственных функционированию ЦОД, следует заметить, что в этих работах не рассматривались аномалии в *SQL*-запросах и обнаружение *SQL*-инъекций.

Этому посвящены работы *второй* группы, например, работы [14—18]. Так, в [14] подчеркивается, что *SQL*-инъекции стали возможными из-за отсутствия проверки вводимых запросов. Эта работа предлагает подход к обнаружению *SQL*-инъекций, основанный на выделении токенов запроса и их сравнении с зарезервированным словарем. В [15] предлагается предсказывать *SQL*-инъекции с помощью модели обучения ансамблю семантических запросов. В этой модели обучения использовался ансамбль из девяти базовых классификаторов, обеспечивающий максимальную точность прогнозирования с помощью голосования. В [16] для обнаружения аномалий в поведении пользователей ЦОД предлагается использовать многомерные статистические тесты. В [17] представлен метод обнаружения *SQL*-инъекций в веб-приложениях на основе сверхточной нейронной сети. Работа [18] предлагает подход к обнаружению *SQL*-инъекций, основанный на анализе реакции и состояния веб-приложения при различных атаках. В [19] рассматривается основанный на машинном обучении подход к предотвращению *SQL*-атак, в котором тестируется свыше 20 различных классификаторов и выбираются 5 наилучших. Эти идеи также использованы в нашей работе.

Постановка задачи

Рассмотрим вначале постановку задачи обнаружения аномальных *SQL*-запросов в ЦОД вуза на основе применения методов машинного обучения. Будем считать, что работа пользователей ЦОД сводится к обращениям к имеющимся в ЦОД базам данных с помощью запросов, составленных на языке *SQL*. Запросы к базам данных фиксируются в регистрационных журналах *системы управления базами данных* (СУБД). Примерами *SQL*-СУБД, которые являются открытыми и могут использоваться в вузовских ЦОД, являются *PostgreSQL*, *MySQL* и др.

Регистрационный журнал состоит из отдельных записей. Каждая запись отражает факт обращения некоторого пользователя к базе данных и содержит следующие поля: дату, время, идентификатор пользователя и текст *SQL*-запроса, который был сформирован пользователем и выполнен со стороны СУБД. Поэтому можно считать, что задача выявления аномального поведения пользователей ЦОД вуза сводится к обнаружению аномальных *SQL*-запросов к базам данных ЦОД, что приводит в конечном итоге к поиску аномальных записей в регистрационных журналах.

Если представить регистрационный журнал СУБД как набор данных, состоящих из записей, то возможная методика анализа такого набора данных на предмет выявления аномалий предполагает следующие этапы:

1) формирование множества признаков, которыми характеризуются *SQL*-запросы;

2) преобразование журнального набора данных в набор данных, записи которого содержат значения сформированных признаков;

3) формирование обучающей выборки, на которой будет осуществляться процесс машинного обучения;

4) использование обученных средств для непосредственного выявления аномальных запросов.

Исходными данными задачи являются:

- множество регистрационных журналов: $\boldsymbol{L} = \{L_1, L_2, ..., L_M\}$;
- множество пользователей: $\boldsymbol{U} = \{U_1, U_2, ..., U_N\}$;
- каждый журнал представляется в виде множества записей: $L_m = \{l_{mi}\}$;
- каждая запись журнала представлена в виде кортежа:

$$l_{mi} = <Date_{mi}, TimeStamp_{mi}, User_{mi}, Op_{mi}>,$$

где $Date_{mi}$ — дата i-го запроса в m-м журнале: $TimeStamp_{mi}$ — временна́я метка запроса; $User_{mi} \in U$ — пользователь запроса; Op_{mi} — текст *SQL*-запроса (*SQL*-инструкция);

- каждая *SQL*-инструкция может быть представлена в виде:

$$Op_{mi} = <Operator, \{Tables\}, \{Fields\}, \{Values\}>,$$

где *Operator* — оператор языка *SQL*; $\{Tables\}$ — множество имен таблиц, которые присутствуют в *SQL*-инструкции; $\{Fields\}$ — множество имен полей; $\{Values\}$ — множество значений полей;

- модели машинного обучения (бинарные классификаторы), которые наиболее часто используются для обнаружения аномалий в наборах данных [20, 21, 22];
- требования по обнаружению атак на базы данных типа *SQL*-инъекции: вероятность правильного обнаружения атаки: $P_{\text{det}} \geq 0{,}95$; вероятность пропуска атаки: $P_{\text{mis}} \leq 0{,}1$.

Для расчета вероятностей предлагается использовать следующие формулы:

$$P_{\text{det}} \approx TP/(TP + FP + FN) \quad (1)$$

$$P_{\text{mis}} \approx FN/(FN + TP) \quad (2)$$

где TP — количество правильно обнаруженных аномалий в наборе данных (True Positive); FP — количество ложно обнаруженных аномалий в наборе данных (*False Positive*); FN — количество ложно обнаруженных нормальных записей в наборе данных (*False Negative*).

В результате решения поставленной задачи требуется построить модель признакового пространства, характеризующую нормальную и аномальную деятельность пользователей БД при отсутствии и наличии атак, и определить методику обнаружения аномальных *SQL*-запросов на основе применения моделей бинарной классификации.

В предлагаемой *модели признакового пространства* признаки разделены на три категории. Признаки *первой* категории определяют количество вхождений того или иного ключевого слова языка *SQL* в *SQL*-инструкцию. Всего отобрано 30 ключевых слов-операторов, таких как SELECT, INSERT, UPDATE, DELETE, RENAME, CREATE, GRANT, ALTER и др.

Вторую категорию признаков составляют количества вхождений тех или иных сигнатур, свойственных *SQL*-инъекциям. Для этой цели были отобраны следующие сигнатуры: "Execute", "or", "txtUserId", "getRequestString", "1=1", "- -", "CHAR", "#" и ";". Появление таких сигнатур в *SQL*-запросах может быть вызвано реализацией *SQL*-инъекций. Всего было использовано 10 сигнатур.

Третью категорию формируют количества вхождений в записи тех или иных имен таблиц данных. В базе данных ЦОД вуза, которая использовалась для экспериментальной оценки предлагаемого подхода, присутствовало свыше 4 000 таблиц данных. Это было вызвано ненормализованным характером ее структуры. Около 2 000 таблиц хранили данные о преподавателях, по одной таблице на каждого преподавателя. Каждому учебному курсу, учебной дисциплине и учебной группе также соответствовала отдельная таблица данных.

Таблица 1

Состав признакового пространства

№ п/п	Категория	Название	Описание
1	1	SELECT_COUNT	Количество вхождений SELECT
2		INSERT_COUNT	Количество вхождений INSERT
…		…	…
30		HAVING_COUNT	Количество вхождений HAVING
31	2	Execute_COUNT	Количество вхождений "Execute"
32		"1=1" _COUNT	Количество вхождений "1=1"
…		…	…
40		txtUserId_COUNT	Количество вхождений "txtUserId"
41	3	Table_1_COUNT	Количество вхождений имени таблицы данных 1
42		Table_2_COUNT	Количество вхождений имени таблицы данных 2
…		…	…
181		Table_141_COUNT	Количество вхождений имени таблицы данных 141

Из-за такого большого количества таблиц были приняты два допущения. *Во-первых*, было решено при формировании признакового пространства не использовать имена полей и их значения, а ограничиться только именами таблиц данных. *Во-вторых*, было решено сократить количество учитываемых в признаковом пространстве имен таблиц, заменяя их общим типовым именем. Так, все имена таблиц для преподавателей были заменены на типовое имя "Teacher_Table", имена таблиц для учебных групп — "Group_Table" и др. В итоге количество учитываемых имен таблиц данных удалось снизить до 141.

Состав сформированного признакового пространства представлен в табл. 1. Из нее видно, что общее количество признаков стало равным 181. Из них 30 признаков относятся к *первой* категории, 10 — ко *второй* категории и 141 — к *третьей* категории.

Реализация предложенного подхода

Для реализации и проверки предлагаемого подхода был использован язык *Python v*.3.8.8 с наборами следующих библиотек: *sklearn, numpy, pandas, matplotlib, Scipy, Re, Pylab*, *Math*. Вычислительная среда была организована на ноутбуке *Jupyter*. В ЦОД вуза для создания базы данных использовалась СУБД *PostgreSQL v*. 13.4, работающая под операционной системой *Ubuntu v*. 13.4. Исследовались следующие наиболее популярные модели контролируемого (*supervised*) машинного обучения, которые являются бинарными классификаторами [20, 21]: машина опорных векторов (*SVM*), дерево решений (*DT*), логистическая регрессия (*LR*), случайный лес (*RF*), гауссов наивный Байес (*GNB*); метод *k*-ближайших соседей (*KNN*) и многослойная нейронная сеть (*NN*).

Для формирования набора данных, который применялся для обучения классификаторов, был выбран фрагмент регистрационного журнала, отображающий работу пользователей с базой данных в течение 15 минут. Всего в этом фрагменте первоначально находилось 82 192 инструкций. На рис. 1 показан вид отдельных инструкций, входящих в этот фрагмент.

Из рис. 1 видно, что запись данного фрагмента была произведена 18 января 2022 г. Она началась в 12:44:09

```
2022-01-18 12:44:09.749 UTC [1174] LOG:  database system is ready to accept connections
2022-01-18 12:44:10.986 UTC [1187] postgres@template1 LOG:  statement:
2022-01-18 12:44:37.957 UTC [1211] postgres@2122 LOG:  statement: SELECT DISTINCT "groups" FROM
"p_learn_plan" ORDER BY "groups"
...
2022-01-18 12:45:01.514 UTC [1230] postgres@2122 LOG:  statement: SELECT "potok_num" FROM "p_group"
WHERE groups='1123'
2022-01-18 12:45:01.514 UTC [1230] postgres@2122 LOG:  statement: SELECT "groups" FROM "p_group"
WHERE potok_num='112'
2022-01-18 12:45:01.567 UTC [1234] postgres@2122 LOG:  statement: SELECT "23" FROM "1123" ORDER BY
count;
...
2022-01-18 12:46:41.250 UTC [1276] postgres@2122 LOG:  statement: select * from pg_tables where
tablename='IvanovDA';
2022-01-18 12:46:41.254 UTC [1276] postgres@2122 LOG:  statement: select * from "IvanovDA" where
"count"='2'
...
2022-01-18 12:47:20.926 UTC [1276] postgres@2122 LOG:  statement: select "n_aud" from "D-0406" where
"prep"='5' and "groups"='5811'
...
2022-01-18 13:00:33.497 UTC [1656] postgres@2122 LOG:  statement: SELECT "groups" FROM "p_group"
WHERE potok_num='534'
2022-01-18 13:00:33.498 UTC [1587] postgres@2122 LOG:  statement: select "n_aud" from "D-2006" where
"prep"='38' and "groups"='3882'
```

Рис.1. Фрагмент регистрационного журнала базы данных ЦОД вуза

и закончилась в 13:00:33. С базой данных работало несколько пользователей. Пользователи, чьи запросы отражает рисунок, имели идентификаторы 1174, 1187, 1211, 1230, 1234, 1276, 1565, 1587. Запросы обращались к различным таблицам данных. Так, запрос со временем 12:44:37 обращался к системной таблице “p_learn_plan” (в ней находилась планирующая информация по учебному процессу). Имя этой таблицы стоит в инструкции после ключевого слова FROM. Другие запросы обращались к следующим таблицам: “p_group”, “1123”, “pg_tables”, “IvanovDA”, “D-0406”, “D-2006”. Таблицы “p_group” и “pg_tables” являются *системными*. Они были созданы системой при создании самой базы данных. Остальные таблицы являются *пользовательскими*. Пользовательские таблицы создаются пользователями с помощью команды CREATE в процессе работы с базой данных. Таблица “IvanovDA” содержит данные о преподавателе с именем «Д.А. Иванов». Таблица “1123” содержит данные об учебной группе с номером 1123. Таблицы “D-0406” и “D-2006” содержат данные об учебных дисциплинах, которые имеют идентификаторы Д-0406 и Д-2006 соответственно.

Всего в базе данных ЦОД вуза на момент создания набора данных, как было сказано, имелось свыше 2 000 таблиц с данными о преподавателях, около 1 000 таблиц с данными об учебных группах, несколько сотен таблиц с данными об учебных дисциплинах. Поэтому при формировании набора данных было принято решение о типизации таких имен таблиц, т. е. об их замене на типовые имена.

Эта процедура стала одним из начальных шагов разработанного алгоритма формирования набора данных, состоящего в следующем:

Шаг 1. Извлечение из фрагмента регистрационного журнала всех имен таблиц и формирование множества имен таблиц, используемых во фрагменте. Всего из фрагмента, представленного на рис. 1, было извлечено 310 имен таблиц.

Шаг 2. Формирование множества новых, типовых имен таблиц. Для имен таблиц с данными о преподавателях использовалось имя “Teacher_Table”, для имен таблиц с данными об учебных группах — имя “Group_Table” и др. Всего в это множество вошло 141 имя, включая имена системных таблиц.

Шаг 3. Замена исходных имен таблиц во фрагменте регистрационного журнала на типовые имена. При этом количество инструкций во фрагменте попрежнему равнялось 82 192.

Шаг 4. Формирование начальной версии набора данных в формате *CSV*. Для каждой инструкции из исходного фрагмента создавалась *CSV*-запись. Полями этой записи служили признаки, которые были включены в модель признакового пространства (см. *табл.* 1). Включение в начальную версию набора данных поля *Result*, значение которого играет роль метки нормальной или аномальной записи, причем *Result* = 0, если запись является нормальной, *Result* = 1, если запись является аномальной.

Шаг 5. Исключение из набора данных дублирующих записей и внесение в него аномалий.

Так как дата и время запроса на текущем этапе исследований были исключены из признакового пространства (это было сделано осознанно, чтобы проверить эффективность машинного обучения на структурах *SQL*-запросов), то в начальной версии набора данных после *Шага* 4 появилось много дублирующих записей, которые

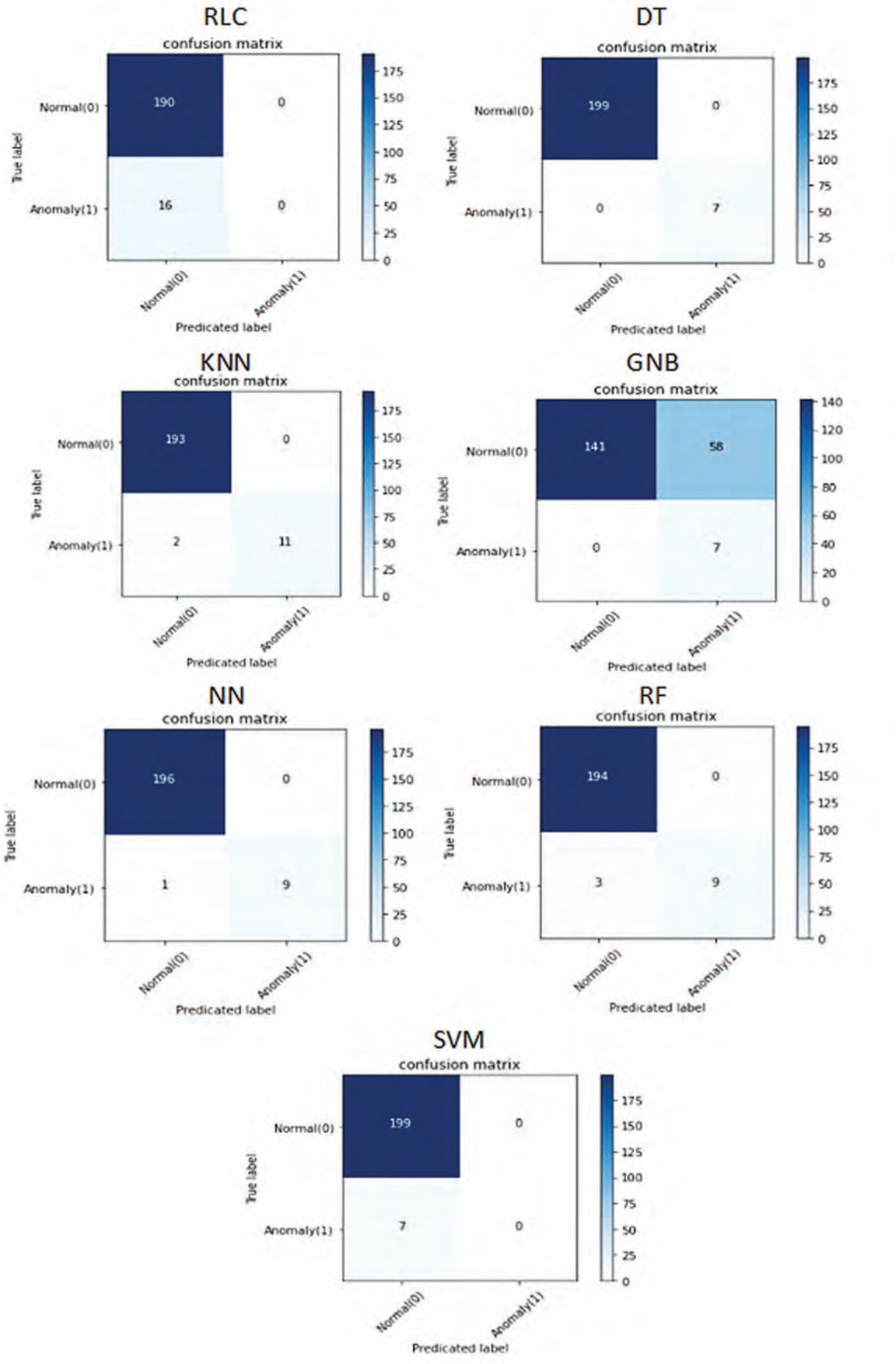

Рис.2. Результаты тестирования на различных классификаторах

никак не влияют на точность, но могут негативно влиять на скорость обнаружения аномалий. Поэтому на *Шаге* 5 такие записи удалялись. После их удаления в наборе данных осталось всего 1026 записей. Кроме того, несколько случайно выбранных записей были модифицированы таким образом, чтобы они соответствовали различными возможным аномалиям (*SQL*-инъекциям и попыткам несанкционированного доступа). Всего было модифицировано 50 таких записей, которые были помечены в поле *Result* как аномальные.

Набор данных, сформированный с помощью описанного выше алгоритма, далее подвергался анализу с помощью выбранных бинарных классификаторов.

Экспериментальные результаты

Экспериментальные исследования были разделены на два этапа — этап обучения и этап тестирования. *Первый* этап был посвящен обучению классификаторов и контрольной проверке точности их работы. На *втором* этапе использовались обученные классификаторы для непосредственного обнаружения аномалий.

На *первом* этапе набор данных, описанный выше, разбивался на две части. В *первую* часть, предназначенную для обучения, входили 80% записей. Во *вторую* часть, предназначенную для контрольного тестирования, вошли остальные 20% записей. Результаты тестирования (в виде матрицы ошибок) представлены на рис. 2.

Анализируя результаты (см. рис. 2), можно сделать следующие *выводы*. Наивысшую точность показал классификатор *DT*. Он без ошибок обнаружил как нормальные, так и аномальные записи. Удовлетворительные результаты продемонстрировали классификаторы *KNN*, *NN*, *RF*, которые без ошибок обнаружили все нормальные записи, а при обнаружении аномалий имели 10—25% ошибок. Классификаторы *RLC*, *SVM* также без ошибок обнаружили все нормальные записи, однако они не смогли правильно обнаружить ни одну аномалию. Наконец, классификатор *GNB* правильно обнаружил все аномалии, однако сделал очень большое количество ошибок при обнаружении нормальных записей — 29%.

На *втором* этапе обученные классификаторы использовались для тестирования нового набора данных, сформированного исходя из нового фрагмента регистрационного журнала. В этом фрагменте журнала содержалось 78 880 записей, записанных в течение 40 минут работы с базой данных. После его обработки рассмотренным выше алгоритмом в новом наборе данных осталось 1852 записи. В этот набор данных также было внесено 50 аномалий путем модификации имеющихся в нем записей.

Результаты тестирования нового набора данных, позволяющие оценить точность работы классификаторов, представлены в табл. 2.

Таблица 2

Результаты оценки точности работы классификаторов

Классификатор	TN	TP	FN	FP	P_{del}	P_{mis}
RLC	1807	5	45	0	0.10	0.90
DT	**1807**	**50**	**0**	**0**	**1.00**	**0.00**
KNN	**1807**	**48**	**2**	**0**	**0.96**	**0.04**
GNB	1256	50	0	551	0.08	0.00
NN	**1807**	**48**	**2**	**0**	**0.96**	**0.04**
RF	1807	38	12	0	0.76	0.24
SVM	1807	1	49	0	0.02	0.98

Как видно из табл. 2, из семи рассмотренных классификаторов только три удовлетворяют *требованиям* по обнаружению атак на базы данных, приведенным в постановке задачи. Таковыми классификаторами являются *DT, KNN, NN*. Для них выполняются требования по вероятностям P_{del} и P_{mis}. При этом классификатор *DT* является наилучшим. У классификатора *GBN* выполняются требования по P_{mis}, но он имеет очень плохие значения по вероятности P_{del}.

Остальные классификаторы (*RLC, RF, SVM*) не отвечают требованиям по P_{del} и по P_{mis}. Возможно, это связано с тем, что использовалась обучающая выборка очень малого объема. Однако для того, чтобы получить обучающую выборку объемом в несколько десятков или даже сотен тысяч записей, необходимо использовать фрагмент регистрационного журнала, соответствующий одному дню работы.

По этой же причине оказались нерезультативными исследования временны́х характеристик процессов обучения и тестирования. На использованном в экспериментах компьютере эти временны́е значения находились в пределах нескольких секунд, что соответствует «вычислительному шуму». Мы рассчитываем, что в дальнейших исследованиях на больших массивах данных удастся построить временны́е зависимости для всех используемых классификаторов.

Заключение

Таким образом, представлены: оригинальная формальная постановка задачи, алгоритмы, аспекты реализации и результаты экспериментальной оценки подхода к выявлению аномального поведения пользователей ЦОД вуза, основанного на эвристическом алгоритме снижения размерности признакового пространства наборов данных, используемых для машинного обучения, и применении моделей бинарной классификации. Исходными данными задачи являются множество регистрационных журналов, множество пользователей базы данных, множество отобранных бинарных классификаторов и требования по точности обнаружения аномальных *SQL*-запросов к базам данных ЦОД вуза. Результатом реализации предложенного подхода являются модель признакового пространства наборов данных, содержащих нормальные и аномальные записи со значениями сформированных признаков, и методика поиска аномальных запросов, содержащая этапы обучения и тестирования классификаторов.

Экспериментальная оценка предложенного подхода была проведена на реальных наборах данных, сформированных в ходе работы пользователей ЦОД вуза с базой данных учебного процесса, с использованием множества бинарных классификаторов, включающего классификаторы типов SVM, DT, LR, RF, GNB, BN, ANN. Результаты оценки подтвердили результативность предложенного подхода и его высокую эффективность. Три модели машинного обучения показали точность, отвечающую предъявляемым требованиям.

Дальнейшие исследования направлены на повышение точности обнаружения аномальных *SQL*-запросов за счет совершенствования параметров классификаторов и их комбинирования.

Рецензент: ***Лаута Олег Сергеевич,*** *доктор технических наук, профессор кафедры комплексного обеспечения информационной безопасности Государственного университета морского и речного флота имени адмирала С.О. Макарова, г. Санкт-Петербург, Российская Федерация.*
E-mail: laos-82@yandex.ru

Литература

1. Воронцов Д.А., Видманов Д.В. Центры обработки данных // Colloquium-Journal. 2020. № 7-1 (59). С. 15—16.
2. Котенко И.В., Саенко И.Б., Чернов А.В., Бутакова М.А. Построение многоуровневой интеллектуальной системы обеспечения информационной безопасности для автоматизированных систем железнодорожного транспорта // Труды СПИИ РАН. 2013. № 7 (30). С. 7—25.
3. Чешейко С.И. Особенности архитектуры центра обработки данных в медицинском учреждении // Информационные и телекоммуникационные технологии. 2021. № 49. С. 12—19.
4. Касенова Д.А. Необходимость обеспечения информационной безопасности центра обработки данных // Modern Science. 2021. № 10-1. С. 436—439.
5. Асадуллин Я.Я. Управление информационной безопасностью центра обработки данных // Защита информации. Инсайд. 2020. № 6 (96). С. 12—22.
6. Белянская О.В., Привалов А.Н. О модели угроз информационной безопасности в центрах обработки данных // Изв. Тульского гос. ун-та. Технические науки. 2021. № 9. С. 12—16.
7. Мартышкин А.И. Вариант реализации вычислительного кластера центра обработки данных на примере интернет-центра вуза // XXI век: итоги прошлого и проблемы настоящего плюс. 2022. Т. 11. № 1 (57). С. 28—33.
8. Decker L., Leite D., Giommi L., Bonacorsi D. Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach // 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). 2020. Pp. 1—8.
9. Shahid N., Ali Shah M. Anomaly detection in system logs in the sphere of digital economy // Competitive Advantage in the Digital Economy (CADE 2021). Online Conference. 2021. Pp. 185—190. DOI: 10.1049/icp.2021.2432 .
10. Nanekaran N.P., Esmalifalak M., Narimani M. Fast anomaly detection in micro data centers using machine learning techniques // 2020 IEEE 18th International Conference on Industrial Informatics (INDIN). 2020. Pp. 86—93. DOI: 10.1109/INDIN45582.2020.9442233 .
11. Deka P.K., Bhuyan M.H., Kadobayashi Y., Elmroth E. Adversarial impact on anomaly detection in cloud datacenters // 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC). 2019. Pp. 188—18809. DOI: 10.1109/PRDC47002.2019.00049 .
12. Chen J., Wang L., Hu Q. Machine learning-based anomaly detection of ganglia monitoring data in HEP data center // EPJ Web Conf. 2020. Vol. 245. Article No. 07061. DOI: 10.1051/epjconf/202024507061 .
13. Salman T., Bhamare D., Erbad A., Jain R., Samaka M. Machine learning for anomaly detection and categorization in multi-cloud environments // 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud). 2017. Pp. 97—103. DOI: 10.1109/CSCloud.2017.15 .

14. Hlaing Z. C. S. S., Khaing M. A detection and prevention technique on SQL injection attacks // 2020 IEEE Conference on Computer Applications (ICCA). 2020. Pp. 1—6. DOI: 10.1109/ICCA49400.2020.9022833 .
15. M G., H B P. Semantic query-featured ensemble learning model for SQL-injection attack detection in IoT-ecosystems // IEEE Transactions on Reliability. 2022. Vol. 71. No. 2. Pp. 1057—1074. DOI: 10.1109/TR.2021.3124331 .
16. Prarthana T.S., Gangadhar N.D. User behaviour anomaly detection in multidimensional data // 2017 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). 2017. Pp. 3—10. DOI: 10.1109/CCEM.2017.19 .
17. Xie X., Ren C., Fu Y., Xu J., Guo J. SQL injection detection for web applications based on elastic-pooling CNN // IEEE Access. 2019. Vol. 7. Pp. 151475—151481. DOI: 10.1109/ACCESS.2019.2947527 .
18. Xiao Z., Zhou Z., Yang W., Deng C. An approach for SQL injection detection based on behavior and response analysis // 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN). 2017. Pp. 1437—1442. DOI: 10.1109/ICCSN.2017.8230346 .
19. Hasan M., Balbahaith Z., Tarique M. Detection of SQL injection attacks: a machine learning approach // 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA). 2019. Pp. 1—6. DOI: 10.1109/ICECTA48151.2019.8959617 .
20. Браницкий А.А., Котенко И.В. Анализ и классификация методов обнаружения сетевых атак // Труды СПИИ РАН. 2016. № 2 (45). С. 207—244.
21. Котенко И.В., Саенко И.Б., Браницкий А.А., Паращук И.Б., Гайфулина Д.А. Интеллектуальная система аналитической обработки цифрового сетевого контента для защиты от нежелательной информации // Информатика и Автоматизация. 2021. № 4. С. 755—788. DOI: 10.15622/ia.20.4.1 .
22. Kotenko I., Saenko I., Branitskiy A. Framework for Mobile Internet of Things Security Monitoring based on Big Data Processing and Machine Learning // IEEE Access, 2018, Vol. 6, pp. 72714—72723. DOI: 10.1109/ACCESS.2018.2881 998 .

DETECTING ABNORMAL BEHAVIOUR OF USERS OF DATA PROCESSING CENTRES OF HIGHER EDUCATION INSTITUTIONS

Igor' Kotenko, *Dr.Sc. (Technology), Professor, Chief Researcher and Head of the Computer Security Problems Laboratory of the Saint Petersburg Federal Research Centre of the Russian Academy of Sciences, Saint Petersburg, Russian Federation.*
E-mail: ivkote@comsec.spb.ru

Igor' Saenko, *Dr.Sc. (Technology), Professor, Leading Researcher at the Computer Security Problems Laboratory of the Saint Petersburg Federal Research Centre of the Russian Academy of Sciences, Saint Petersburg, Russian Federation.*
E-mail: ibsaen@comsec.spb.ru

Mazen Hamed Al-Barri, *postgraduate student at the S. Budyonnyi Military Academy of Communications, Saint Petersburg, Russian Federation.*
E-mail: mazenb51@gmail.com

Keywords: *information security, insider, computer attack, database, SQL injection, machine learning, classifier, registration journal, data set.*

Abstract

Purpose of the paper: setting up the problem of ensuring the required precision of detecting abnormal behaviour of users of data processing centres of higher education institutions (HEI) and find its solution using machine learning methods.

Methods of study: system analysis of the problem of detecting abnormal behaviour of users of data processing centres of HEI and supervised machine learning methods.

Study findings: an original setup of the problem of detecting abnormal behaviour of users of data processing centres of a HEI is proposed which is oriented towards using machine learning methods. An approach to reducing the dimensionality of the initial feature space as well as an algorithm implementing it is developed based on typing of data table names present in SQL query texts. An implementation of the proposed approach using different machine learning models was carried out.

An experimental assessment evaluation of the proposed approach was performed which confirmed its high efficiency and made it possible to identify the most appropriate classifiers for solving this problem, which are: the decision tree, k-nearest neighbours method and multilayer neural network.

References

1. Vorontsov D.A., Vidmanov D.V. Tsentry obrabotki dannykh. Colloquium-Journal, 2020, No. 7-1 (59), pp. 15—16.
2. Kotenko I.V., Saenko I.B., Chernov A.V., Butakova M.A. Postroenie mnogourovnevoi intellektual'noi sistemy obespecheniia informatsionnoi bezopasnosti dlia avtomatizirovannykh sistem zheleznodorozhnogo transporta. Trudy SPII RAN, 2013, No. 7 (30), pp. 7—25.
3. Chesheiko S.I. Osobennosti arkhitektury tsentra obrabotki dannykh v meditsinskom uchrezhdenii. Informatsionnye i telekommunikatsionnye tekhnologii, 2021, No. 49, pp. 12—19.
4. Kasenova D.A. Neobkhodimost' obespecheniia informatsionnoi bezopasnosti tsentra obrabotki dannykh. Modern Science, 2021, No. 10-1, pp. 436—439.
5. Asadullin Ia.Ia. Upravlenie informatsionnoi bezopasnost'iu tsentra obrabotki dannykh. Zashchita informatsii. Insaid, 2020, No. 6 (96), pp. 12—22.
6. Belianskaia O.V., Privalov A.N. O modeli ugroz informatsionnoi bezopasnosti v tsentrakh obrabotki dannykh. Izv. Tul'skogo gos. un-ta. Tekhnicheskie nauki, 2021, No. 9, pp. 12—16.
7. Martyshkin A.I. Variant realizatsii vychislitel'nogo klastera tsentra obrabotki dannykh na primere internet-tsentra vuza. XXI vek: itogi proshlogo i problemy nastoiashchego plius, 2022, t. 11, No. 1 (57), pp. 28—33.
8. Decker L., Leite D., Giommi L., Bonacorsi D. Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach. 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020, pp. 1—8.
9. Shahid N., Ali Shah M. Anomaly detection in system logs in the sphere of digital economy. Competitive Advantage in the Digital Economy (CADE 2021). Online Conference, 2021, pp. 185—190. DOI: 10.1049/icp.2021.2432 .
10. Nanekaran N.P., Esmalifalak M., Narimani M. Fast anomaly detection in micro data centers using machine learning techniques. 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), 2020, pp. 86—93. DOI: 10.1109/INDIN45582.2020.9442233 .
11. Deka P.K., Bhuyan M.H., Kadobayashi Y., Elmroth E. Adversarial impact on anomaly detection in cloud datacenters. 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), 2019, pp. 188—18809. DOI: 10.1109/PRDC47002.2019.00049 .
12. Chen J., Wang L., Hu Q. Machine learning-based anomaly detection of ganglia monitoring data in HEP data center. EPJ Web Conf, 2020, Vol. 245, Article No. 07061. DOI: 10.1051/epjconf/202024507061 .
13. Salman T., Bhamare D., Erbad A., Jain R., Samaka M. Machine learning for anomaly detection and categorization in multi-cloud environments. 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), 2017, pp. 97—103. DOI: 10.1109/CSCloud.2017.15 .
14. Hlaing Z. C. S. S., Khaing M. A detection and prevention technique on SQL injection attacks. 2020 IEEE Conference on Computer Applications (ICCA), 2020, pp. 1—6. DOI: 10.1109/ICCA49400.2020.9022833 .
15. M G., H B P. Semantic query-featured ensemble learning model for SQL-injection attack detection in IoT-ecosystems. IEEE Transactions on Reliability, 2022, Vol. 71, No. 2, pp. 1057—1074. DOI: 10.1109/TR.2021.3124331 .
16. Prarthana T.S., Gangadhar N.D. User behaviour anomaly detection in multidimensional data. 2017 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2017, pp. 3—10. DOI: 10.1109/CCEM.2017.19 .
17. Xie X., Ren C., Fu Y., Xu J., Guo J. SQL injection detection for web applications based on elastic-pooling CNN. IEEE Access, 2019, Vol. 7, pp. 151475—151481. DOI: 10.1109/ACCESS.2019.2947527 .
18. Xiao Z., Zhou Z., Yang W., Deng C. An approach for SQL injection detection based on behavior and response analysis. 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), 2017, pp. 1437—1442. DOI: 10.1109/ICCSN.2017.8230346 .
19. Hasan M., Balbahaith Z., Tarique M. Detection of SQL injection attacks: a machine learning approach. 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2019, pp. 1—6. DOI: 10.1109/ICECTA48151.2019.8959617 .
20. Branitskii A.A., Kotenko I.V. Analiz i klassifikatsiia metodov obnaruzheniia setevykh atak. Trudy SPII RAN, 2016, No. 2 (45), pp. 207—244.
21. Kotenko I.V., Saenko I.B., Branitskii A.A., Parashchuk I.B., Gaifulina D.A. Intellektual'naia sistema analiticheskoi obrabotki tsifrovogo setevogo kontenta dlia zashchity ot nezhelatel'noi informatsii. Informatika i Avtomatizatsiia, 2021, No. 4, pp. 755—788. DOI: 10.15622/ia.20.4.1 .
22. Kotenko I., Saenko I., Branitskiy A. Framework for Mobile Internet of Things Security Monitoring based on Big Data Processing and Machine Learning. IEEE Access, 2018, Vol. 6, pp. 72714—72723. DOI: 10.1109/ACCESS.2018.2881 998 .