БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2025. – № 3 (118). – 74–92

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004 DOI: 10.17212/2782-2230-2025-3-74-92

ИССЛЕДОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ SQL-ИНЪЕКЦИЙ*

В.Г. ЛАПИН[1], Ю.А. АНДРУСЕНКО[2], Д.Э. СУРЕНКОВ[3], Д.А. ФЕДАШ[4], А.А. СОЛОМЯНКО[5]

[1] *355017, г. Ставрополь, ул. Пушкина, 1, Северо-Кавказский федеральный университет, доцент кафедры вычислительной математики и кибернетики. E-mail: vitlx@yandex.ru*

[2] *355017, г. Ставрополь, ул. Пушкина, 1, Северо-Кавказский федеральный университет, старший преподаватель кафедры вычислительной математики и кибернетики. E-mail: iuandrusenko@ncfu.ru*

[3] *355017, г. Ставрополь, ул. Пушкина, 1, Северо-Кавказский федеральный университет, лаборант кафедры вычислительной математики и кибернетики. E-mail: dsurenkov2004@gmail.com*

[4] *355017, г. Ставрополь, ул. Пушкина, 1, Северо-Кавказский федеральный университет, лаборант кафедры вычислительной математики и кибернетики. E-mail: dfedash@ncfu.ru*

[5] *355017, г. Ставрополь, ул. Пушкина, 1, Северо-Кавказский федеральный университет, лаборант кафедры математического анализа алгебры и геометрии. E-mail: artemixol@xmail.ru*

В настоящей статье исследуются современные методы машинного обучения для обнаружения SQL-инъекций – одной из наиболее распространенных угроз в области кибербезопасности. Рассматриваются подходы, основанные на нейронных сетях, методе главных компонент и использовании платформы KNIME. Проведен анализ эффективности различных алгоритмов, а также предложены рекомендации по их применению в реальных сценариях. Результаты исследования демонстрируют потенциал машинного обучения для повышения уровня защиты веб-приложений от атак типа SQL-инъекций. Нейронные сети и алгоритмы, основанные на PCA, показали значительную точность в классификации вредоносных запросов. Использование платформы KNIME позволило упростить процесс разработки и тестирования моделей, что делает ее перспективным инструментом для специалистов в области кибербезопасности. Перспективным направлением будущих исследований является интеграция предложенных методов с другими

* Статья получена 12 мая 2025 г.

технологиями, такими как анализ поведения пользователей и обработка естественного языка, для повышения точности обнаружения атак.

Ключевые слова: SQL-инъекция, машинное обучение, нейронные сети, PCA, KNIME, кибербезопасность, обнаружение атак, система защиты веб-приложений, кибератака.

ВВЕДЕНИЕ

SQL-инъекции остаются одной из наиболее опасных и распространенных угроз для веб-приложений, позволяя злоумышленникам получать несанкционированный доступ к базам данных. Традиционные методы защиты, такие как валидация входных данных и использование параметризованных запросов, не всегда обеспечивают достаточный уровень безопасности. В связи с этим возрастает интерес к применению методов машинного обучения для автоматического обнаружения и предотвращения SQL-инъекций.

Целью данной работы является исследование эффективности современных алгоритмов машинного обучения, включая нейронные сети и методы снижения размерности, для решения задачи классификации SQL-запросов. Особое внимание уделяется практическому применению платформы KNIME для создания и тестирования моделей. Актуальность исследования обусловлена необходимостью разработки более надежных и адаптивных систем защиты, способных противостоять постоянно эволюционирующим угрозам.

1. ОБЗОР СУЩЕСТВУЮЩИХ РАБОТ

SQL-инъекция – это кибератака, при которой злоумышленник внедряет вредоносный SQL-код в запросы к базе данных, чтобы получить несанкционированный доступ к информации, изменить или удалить данные. Этот метод остается одним из самых распространенных и опасных из-за уязвимостей в веб-приложениях, которые некорректно обрабатывают пользовательский ввод (рис. 1).

Первые упоминания об SQL-инъекциях появились в конце 1990-х годов, когда веб-разработчики начали активно использовать базы данных для хранения информации. В 1998 году Джефф Форристал (Jeff Forristal) описал уязвимости в динамических SQL-запросах, а в 2000-х годах атаки стали массовыми.

Одним из первых известных случаев эксплуатации SQL-инъекции была атака на Microsoft SQL Server в 2002 году, когда червь SQL Slammer использовал уязвимость в обработке запросов, что привело к масштабным DDoS-атакам.

Рис. 1. Обобщающая схема

Типы SQL-инъекций

Классическая SQL-инъекция – внедрение кода через поля ввода (логин, пароль, поиск).

Слепая SQL-инъекция (Blind SQLi) – когда приложение не выводит ошибки, но злоумышленник может извлекать данные по времени ответа или логике поведения.

Инъекция на основе UNION – использование оператора UNION для объединения легитимного запроса с вредоносным.

Таймлайн развития SQL-инъекций представлен в табл. 1.

Таблица 1

Таймлайн развития SQL-инъекций

Год	Название периода	Описание
1998	Первые упоминания	Первые документальные свидетельства о SQL-инъекциях в публичных форумах и рассылках
2003	Ранние исследования	SQL-инъекции начинают обсуждаться в узких кругах специалистов по безопасности

Окончание табл. 1

Год	Название периода	Описание
2004	Массовое распространение	SQL-инъекции становятся популярным методом атак. Появляются первые автоматизированные инструменты
2011	Борьба с инъекциями	Активное внедрение защитных мер: параметризованные запросы, ORM, WAF
2016	Усложнение атак	Появление слепых SQL-инъекций, эксплуатация NoSQL-инъекций. Развитие методов обхода защиты

2. ДАТАСЕТ И ЕГО СВОЙСТВА

Web Network – этот набор данных с коллекцией журналов сетевого трафика является ценным ресурсом для разработки решений веб-безопасности, которые классифицируют веб-запросы как хорошие или плохие. Анализируя пути к URL-адресам, заголовки и параметры, а также уделяя пристальное внимание конкретным ключевым словам, связанным с атаками, специалисты по безопасности и автоматизированные системы могут быстрее обнаруживать и предотвращать веб-атаки. Список «плохих слов» служит важным инструментом для выявления подозрительных запросов, что в конечном итоге помогает создавать более безопасные веб-приложения и среды. Подробная характеристика датасета представлена в табл. 2.

Таблица 2

Характеристика датасета

Название	Тип данных	Значение
method	Категориальный	HTTP-метод запроса (например, GET, POST)
path	Категориальный	Путь запроса (URL-адрес без домена)
single_q	Количественный	Количество одинарных кавычек (') в запросе
double_q	Количественный	Количество двойных кавычек (") в запросе
dashes	Количественный	Количество дефисов (-) в запросе

Окончание табл. 2

Название	Тип данных	Значение
braces	Количественный	Количество фигурных скобок ({}) в запросе
spaces	Количественный	Количество пробелов в запросе
percentages	Количественный	Количество знаков процента (%) в запросе
semicolons	Количественный	Количество точек с запятой (;) в запросе
angle_brackets	Количественный	Количество угловых скобок (< >) в запросе
special_chars	Количественный	Количество специальных символов в запросе
path_length	Количественный	Длина пути запроса (количество символов)
body_length	Количественный	Длина тела запроса (количество символов)
badwords_count	Количественный	Количество запрещенных слов в запросе
class	Бинарный	Класс запроса (1 – подозрительный, 0 – нормальный)

3. МОДЕЛИРОВАНИЕ

В этом разделе раскрыты характеристики стилей, используемых в данном документе.

Исследование проведено на операционной системе Windows 10 Pro.

Платформа KNIME v. 5.4.1

3.1. ИССЛЕДУЕМЫЕ МОДЕЛИ (НАЗВАНИЯ НА РУССКОМ ЯЗЫКЕ)

1. Decision Tree – это контролируемый алгоритм обучения, который можно использовать как для классификационного, так и для регрессионного анализа. Его называют деревом решений, потому что он использует древовидную модель, где каждое решение ответвляется от других решений. Дерево решений начинается с одного узла, который затем разделяется на возможные результаты. Каждый из этих результатов приводит к дополнительным узлам, которые разветвляются на другие возможности. Это продолжается до тех пор, пока не будет достигнут результат решения [3].

2. Random Forests – это тип ансамблевого метода обучения, в котором для решения проблемы используются несколько моделей обучения. Для случайных лесов модель создает «лес» деревьев решений, обычно обученных на различных подмножествах исходных данных. Алгоритм случайного леса выбирает наблюдения и признаки для построения нескольких деревьев решений. Затем он объединяет голоса из разных деревьев решений, чтобы определить окончательный класс тестового объекта (для классификации) или берет среднее значение выходов разных деревьев (для регрессии) [3].

3. Gradient Boosted Trees – это мощная техника ансамблевого машинного обучения, используемая как для регрессии, так и для классификации. Этот метод строит модель прогнозирования в форме ансамбля слабых предиктивных моделей, обычно деревьев решений. Основной идеей градиентного бустинга является последовательное добавление к ансамблю новых моделей, каждая из которых учитывает и исправляет ошибки предыдущих моделей [10].

4. Tree Ensemble – изучает ансамбль деревьев решений (например, варианты случайного леса). Обычно каждое дерево строится с различным набором строк (записей) и/или столбцов (атрибутов). Выходная модель описывает ансамбль моделей деревьев решений и применяется в соответствующем узле предиктора с использованием выбранного режима агрегации для агрегации голосов отдельных деревьев решений [4].

5. Naive Bayes – это метод классификации, основанный на так называемой теореме Байеса. По сути, он предполагает, что возникновение признака совершенно не коррелирует с возникновением другого признака в пределах класса [6].

6. Fuzzy Rule – изучает модель нечетких правил на маркированных числовых данных, используя смешанное формирование нечетких правил в качестве базового алгоритма обучения (также известного как алгоритм RecBF-DDA). Этот алгоритм генерирует правила на основе числовых данных, которые являются нечеткими интервалами в пространствах с более высокой размерностью [7].

7. PNN – вероятностная нейронная сеть (PNN) на основе метода DDA (динамическая корректировка спада) на маркированных данных с использованием конструктивного обучения вероятностных нейронных сетей в качестве базового алгоритма [8].

8. RProp MLP – реализация алгоритма RProp для многослойных сетей прямого распространения. RProp выполняет локальную адаптацию обновлений веса в соответствии с поведением функции ошибки [23].

9. K Nearest Neighbor – алгоритм классификации и регрессии, основанный на гипотезе компактности, которая предполагает, что расположенные близко друг к другу объекты в пространстве признаков имеют схожие значения целевой переменной или принадлежат к одному классу [11].

3.2. АНАЛИЗ МГК

Для того чтобы не подавать на вход модели большое количество ненужной информации, уберем данные, которые не повлияют на определение атаки. Для этого используем встроенный алгоритм МГК.

МГК (метод главных компонент) – это метод уменьшения размерности данных, который преобразует множество коррелированных признаков в меньшее число некоррелированных переменных (главных компонент). Сохраняет максимальную дисперсию (информативность) исходных данных. Используется для визуализации, ускорения обучения моделей и борьбы с переобучением. Все модели были протестированы с различными настройками PCA, и был выбран лучший результат.

3.2.1. RANDOM FORESTS

Был проведен анализ со значениями от 1 до 14 (рис. 2).

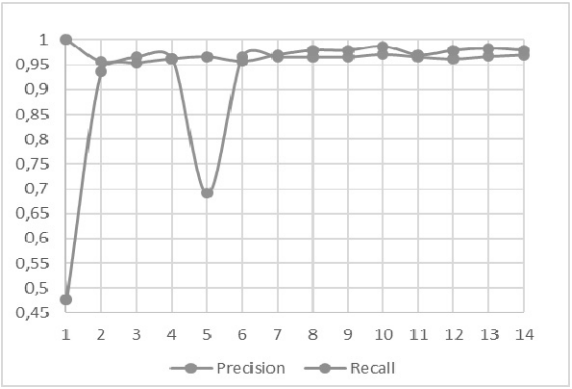

Рис. 2. Анализ PCA Random Forest

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с восемью колонками, значение Recall 0,944723618; значение Precision 0,899521531.

3.2.2. DECISION TREE

Был проведен анализ со значениями от 1 до 14 (рис. 3).

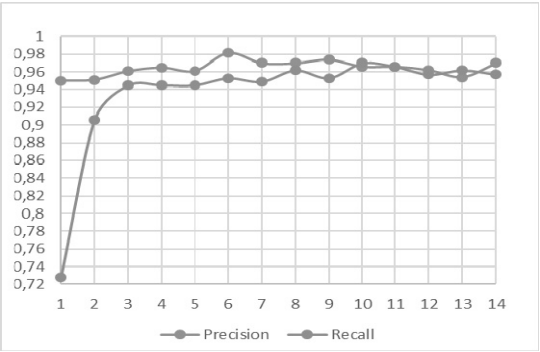

Рис. 3. Анализ PCA Decision Tree

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с 53 колонками, значение Recall 0,924623116; значение Precision 0,910891089.

3.2.3. TREE ENSEMBLE

Был проведен анализ со значениями от 1 до 14 (рис. 4).

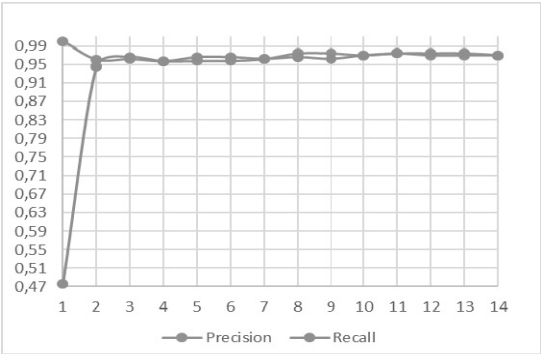

Рис. 4. Анализ PCA Tree Ensemble

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с 38 колонками, значение Recall 0,904761905; значение Precision 0,954773869.

3.2.4. GRADIENT BOOSTED TREES

Был проведен анализ со значениями от 1 до 14 (рис. 5).

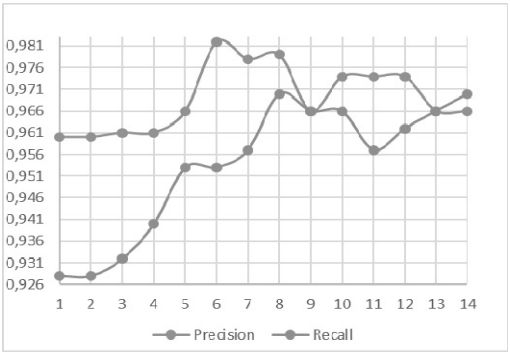

Рис. 5. Анализ PCA Gradient Boosted Trees

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с 20 колонками, значение Recall 0,939698492; значение Precision 0,903381643.

3.2.5. K NEAREST NEIGHBOR

Был проведен анализ со значениями от 1 до 14 (рис. 6).

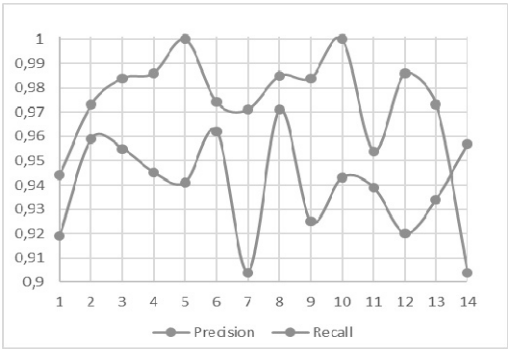

Рис. 6. Анализ PCA K Neighbor

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с восемью колонками, значение Recall 0,944723618; значение Precision 0,899521531.

3.2.6. NAIVE BAYES

Был проведен анализ со значениями от 1 до 14 (рис. 7).

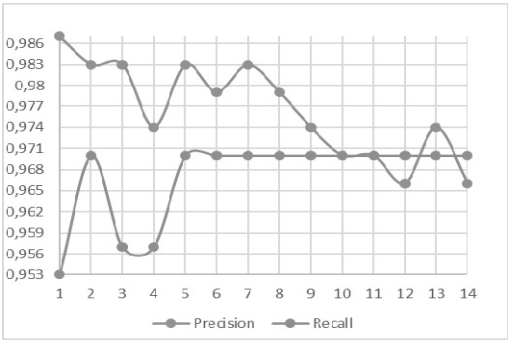

Рис. 7. Анализ PCA Naive Bayes

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с 20 колонками, значение Recall 0,929648241; значение Precision 0,872641509.

3.2.7. FUZZY RULE

Был проведен анализ со значениями от 1 до 14 (рис. 8).

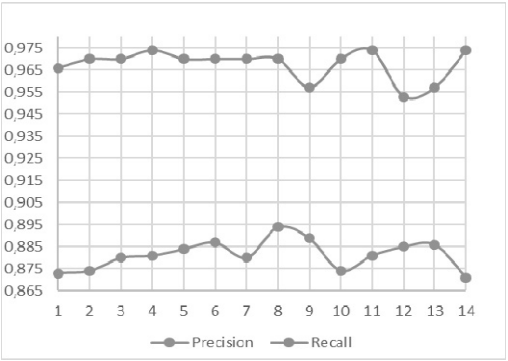

Рис. 8. Анализ PCA Fuzzy Rule

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с восемью колонками, значение Recall 0,979899497; значение Precision 0,573529412.

3.2.8. PNN

Был проведен анализ со значениями от 1 до 14 (рис. 9).

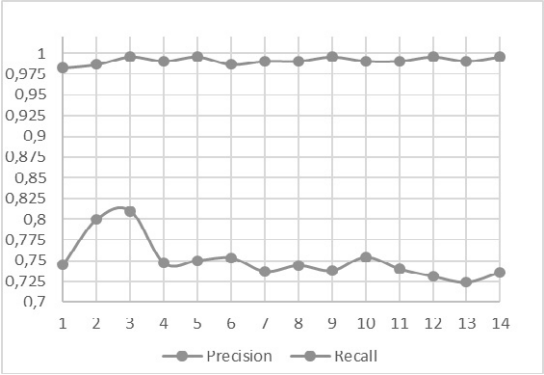

Рис. 9. Анализ PCA PNN

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с восемью колонками, значение Recall 0,969849246; значение Precision 0,853982301.

3.2.9. RPROP MLP

Был проведен анализ со значениями от 1 до 14 (рис. 10).

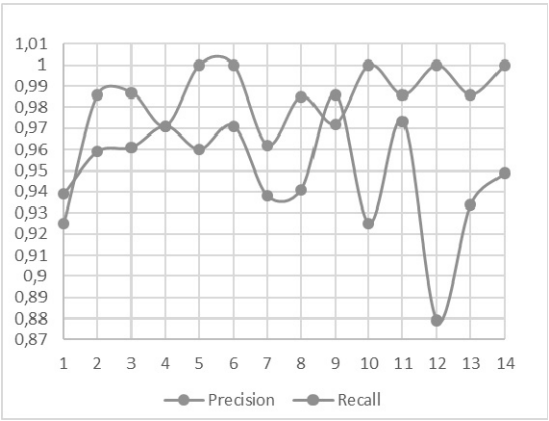

Рис. 10. Анализ PCA RProp MLP

По графику можем сделать вывод, что для данной модели наилучшим результатом является PCA с 17 колонками, значение Recall 0,944723618; значение Precision 0,926108374.

Данные лучших значений PCA каждой модели в табл. 3.

Таблица 3

Лучшие значения PCA

Модель	PCA	Recall	Precision
Naive Bayes	2	0,983	0,97
PNN	3	0,996	0,81
RProp MLP	8	0,985	0,941
Fuzzy Rule	8	0,97	0,894
K Nearest Neighbor	8	0,971	0,985
Random Forest	14	0,979	0,97
Decision Tree	10	0,97	0,966
Tree Ensemble	11	0,974	0,974
Gradient Boosted Trees	8	0,97	0,979

По табл. 3 можно сделать вывод, что пятью лучшими моделями являются Naive Bayes, K Nearest Neighbor, Random Forest, Gradient Boosted Trees, Tree Ensemble.

3.3. АНАЛИЗ ЛУЧШИХ МОДЕЛЕЙ

В каждой модели были исследованы их внутренние характеристики с использованием лучшего значения PCA. Были взяты значения «1», что соответствует вредоносному приложению.

3.3.1. NAIVE BAYES

В данной модели исследовался параметр Default probability (рис. 11).

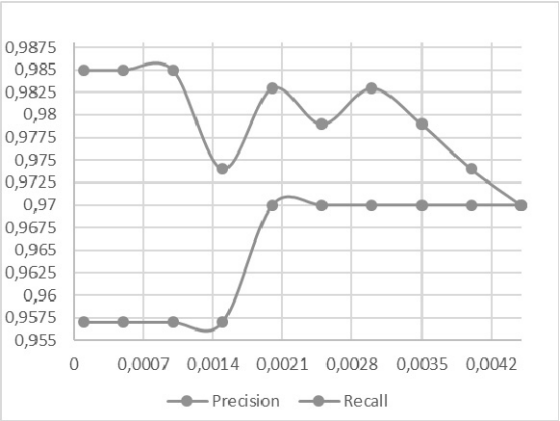

Рис. 11. Исследование параметра Default probability

Из рис. 11 делаем вывод, что лучшим является результат 0,002 со стандартным отклонением Recall 0,97 и Precision 0,983.

3.3.2. RANDOM FOREST

В данной модели исследовался параметр Tree depth (рис. 12).

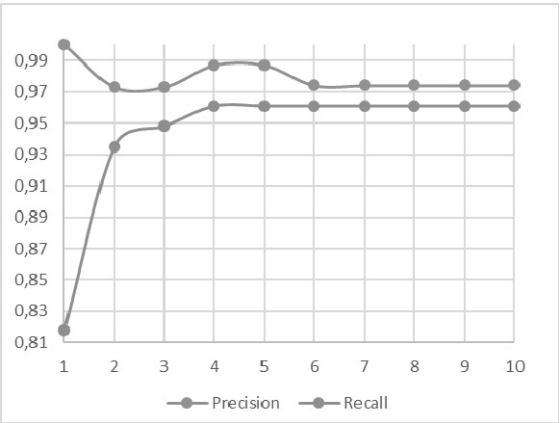

Рис. 12. Исследование параметра Tree depth

Из рис. 12 делаем вывод, что лучшим является результат глубины 4 со значениями Recall 0,987 и Precision 0,961.

3.3.3. K NEAREST NEIGHBOR

В данной модели исследовался параметр Number of neighbours to consider (k) (рис. 13).

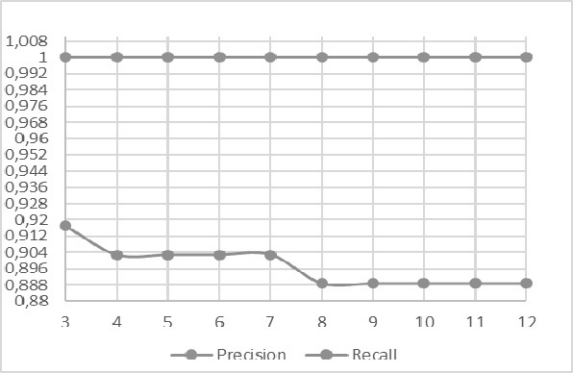

Рис. 13. Исследование параметра Number of neighbours to consider (k)

Из рис. 13 делаем вывод, что лучшим является результат соседа 3 со значениями Recall 0,917 и Precision 1.

3.3.4. GRADIENT BOOSTED TREES

В данной модели исследовался параметр Limit number of levels (tree depth) (рис. 14).

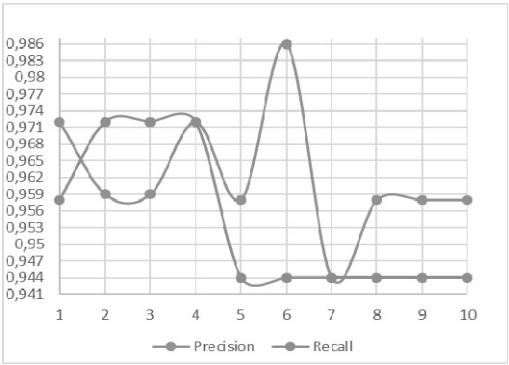

Рис. 14. Исследование параметра Limit number of levels (tree depth)

Из рис. 14 делаем вывод, что лучшим является результат глубины 3 со значениями Recall 0,972 и Precision 0,959.

3.3.5. TREE ENSEMBLE

В данной модели исследовался параметр Limit number of levels (tree depth) (рис. 15).

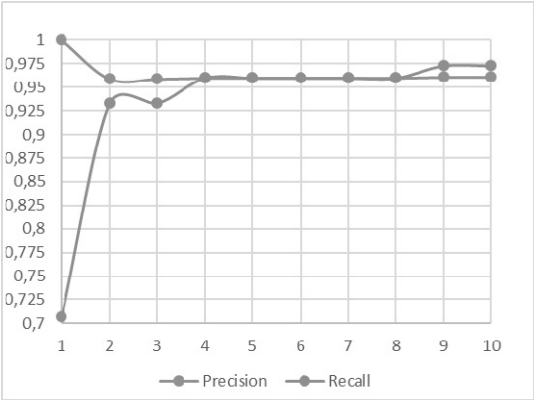

Рис. 15. Исследование параметра Limit number of levels (tree depth)

Из рис. 15 делаем вывод, что лучшим является результат глубины 9 со значениями Recall 0,973 и Precision 0,961.

Данные лучших значений каждой модели показаны в табл. 4.

Таблица 4

Лучшие значения моделей

Модель	Recall	Precision	AUC
Naive Bayes	0,97	0,983	0,973
Random Forest	0,987	0,961	0,975
K Nearest Neighbor	0,917	1	0,958
Gradient Boosted Trees	0,972	0,959	0,814
Tree Ensemble	0,973	0,961	0,968

4. БОРЬБА С ПЕРЕОБУЧЕНИЕМ

Для борьбы с переобучением и оценки устойчивости модели использовался ROC-анализ (Receiver Operating Characteristic).

ROC Curve строит ROC-кривую и вычисляет AUC (Area Under Curve), который показывает, насколько хорошо модель различает классы. Чем ближе AUC к значению «1», тем лучше модель, а значение около 0,5 указывает на случайное угадывание.

ЗАКЛЮЧЕНИЕ

Проведенное исследование подтвердило высокую эффективность методов машинного обучения для обнаружения SQL-инъекций. Нейронные сети и алгоритмы, основанные на PCA, показали значительную точность в классификации вредоносных запросов. Использование платформы KNIME позволило упростить процесс разработки и тестирования моделей, что делает ее перспективным инструментом для специалистов в области кибербезопасности.

Результаты работы могут быть применены для создания более надежных систем защиты веб-приложений, а также для дальнейшего изучения возможностей машинного обучения в контексте кибербезопасности. Перспективным направлением будущих исследований является интеграция предложенных методов с другими технологиями, такими как анализ поведения пользователей и обработка естественного языка, для повышения точности обнаружения атак.

СПИСОК ЛИТЕРАТУРЫ

1. Что такое Knime и как его использовать. – URL: https://sky.pro/media/chto-takoe-knime-i-kak-ego-ispolzovat/ (дата обращения: 03.09.2025).

2. *Abualkibash M.* Machine learning in network security using KNIME analytics // arXiv preprint arXiv:2001.11489. – URL: https://arxiv.org/abs/2001.11489 (accessed: 03.09.2025).

3. *Emery J.* Decision trees and random forests in KNIME. – URL: https://www.phdata.io/blog/decision-trees-and-random-forests-in-knime/ (accessed: 03.09.2025).

4. KNIME Hub: Tree ensemble classification. – URL: https://hub.knime.com/knime/extensions/org.knime.features.ensembles/latest/org.knime.base.node.mine.treeensemble2.node.learner.classification.TreeEnsembleClassificationLearnerNodeFactory2 (accessed: 03.09.2025).

5. Logistic Regression in Machine Learning. – URL: https://www.mastersindatascience.org/learning/machine-learning-algorithms/logistic-regression/ (accessed: 03.09.2025).

6. Naive Bayes Algorithm Overview. – URL: https://databasecamp.de/en/ml/naive-bayes-algorithm (accessed: 03.09.2025).

7. KNIME Hub: Fuzzy Basis Function Learner. – URL: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.bfn.fuzzy.FuzzyBasisFunctionLearnerNodeFactory (accessed: 03.09.2025).

8. KNIME Hub: Radial Basis Function Learner. – URL: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.bfn.radial.RadialBasisFunctionLearnerNodeFactory (accessed: 03.09.2025).

9. KNIME Hub: Neural Networks with RProp. – URL: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.neural.rprop.RPropNodeFactory2 (accessed: 03.09.2025).

10. Вопрос на EasyOffer. – URL: https://easyoffer.ru/question/5476 (accessed: 02.04.2025).

11. KNIME Hub: K Nearest Neighbor. – URL: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.knn.KnnNodeFactory2 (accessed: 03.09.2025).

Лапин Виталий Геннадьевич, доцент кафедры вычислительной математики и кибернетики Северо-Кавказского федерального университета. Основные направления научных исследований – построение и обеспечение безопасности медицинских информационных систем, системы и методы обнаружения и предотвращения сетевых атак, машинное обучение. E-mail: vitlx@yandex.ru

Андрусенко Юлия Алексеевна, старший преподаватель кафедры вычислительной математики и кибернетики Северо-Кавказского федерального университета. Область научных интересов – информационная безопасность, правовые основы информационной безопасности, компьютерная графика. E-mail: iuandrusenko@ncfu.ru

Суренков Данила Эдуардович, лаборант кафедры вычислительной математики и кибернетики Северо-Кавказского федерального университета. Область научных интересов – информационная безопасность, машинное обучение. E-mail: dsurenkov2004@gmail.ru

Федаш Диана Алексеевна, лаборант кафедры вычислительной математики и кибернетики Северо-Кавказского федерального университета. Область научных интересов – информационная безопасность. E-mail: dfedash@ncfu.ru

Соломянко Артем Алексеевич, лаборант кафедры математического анализа алгебры и геометрии Северо-Кавказского федерального университета. Область научных интересов – математические методы и современные компьютерные технологии для анализа данных. E-mail: artemixol@xmail.ru

DOI: 10.17212/2782-2230-2025-3-74-92

Research of machine learning methods for detecting SQL injections*

V.G. Lapin[1], Yu.A. Andrusenko[2], D.E. Surenkov[3], D.A. Fedash[4], A.A. Solomyanko[5]

[1] *North Caucasus Federal University, 1 Pushkina Street, Stavropol, 355017, Associate Professor of the Department of Computational Mathematics and Cybernetics. E-mail: vitlx@yandex.ru*
[2] *North Caucasus Federal University, 1 Pushkina Street, Stavropol, Russian Federation, 355017, Senior Lecturer of the Department of Computational Mathematics and Cybernetics. E-mail: iuandrusenko@ncfu.ru*
[3] *North Caucasus Federal University, 1 Pushkina Street, Stavropol, Russian Federation, 355017, laboratory assistant of the Department of Computational Mathematics and Cybernetics. E-mail: dsurenkov2004@gmail.com*
[4] *North Caucasus Federal University, 1 Pushkina Street, Stavropol, Russian Federation, 355017, laboratory assistant of the Department of Computational Mathematics and Cybernetics. E-mail: dfedash@ncfu.ru*
[5] *North Caucasus Federal University, 1 Pushkina Street, Stavropol, Russian Federation, 355017, laboratory assistant at the department of mathematical analysis, algebra, and geometry. E-mail: artemixol@xmail.ru*

This article explores modern machine learning techniques for detecting SQL injections, one of the most common cybersecurity threats. It examines approaches based on neural networks, principal component analysis, and the use of the KNIME platform. The article analyzes the effectiveness of various algorithms and provides recommendations for their implementation in real-world scenarios. The findings demonstrate the potential of machine learning to enhance the security of web applications against SQL injection attacks. Neural networks and PCA-based algorithms have shown significant accuracy in classifying malicious requests. The use of the KNIME platform has simplified the process of model development and testing, making it a promising tool for cybersecurity professionals. Future research directions include integrating these methods with other technologies such as user behavior analysis and machine learning.

Keywords: SQL injection, machine learning, neural networks, PCA, KNIME, cybersecurity, attack detection, web application protection system, cyberattack

* *Received 12 May 2025.*

REFERENCES

1. What is Knime and how to use it. (In Russian). Available at: https://sky.pro/media/chto-takoe-knime-i-kak-ego-ispolzovat/ (accessed 03.09.2025).

2. Abualkibash M. *Machine learning in network security using KNIME analytics*. Available at: https://arxiv.org/abs/2001.11489 (accessed 03.09.2025).

3. Emery J. *Decision trees and random forests in KNIME*. Available at: https://www.phdata.io/blog/decision-trees-and-random-forests-in-knime/ (accessed 03.09.2025).

4. *KNIME Hub: Tree ensemble classification*. Available at: https://hub.knime.com/knime/extensions/org.knime.features.ensembles/latest/org.knime.base.node.mine.treeensemble2.node.learner.classification.TreeEnsembleClassificationLearnerNodeFactory2 (accessed 03.09.2025).

5. *Logistic Regression in Machine Learning*. Available at: https://www.mastersindatascience.org/learning/machine-learning-algorithms/logistic-regression/ (accessed 03.09.2025).

6. *Naive Bayes Algorithm Overview*. Available at: https://databasecamp.de/en/ml/naive-bayes-algorithm (accessed 03.09.2025).

7. *KNIME Hub: Fuzzy Basis Function Learner*. Available at: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.bfn.fuzzy.FuzzyBasisFunctionLearnerNodeFactory (accessed 03.09.2025).

8. *KNIME Hub: Radial Basis Function Learner*. Available at: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.bfn.radial.RadialBasisFunctionLearnerNodeFactory (accessed 03.09.2025).

9. *KNIME Hub: Neural Networks with RProp*. Available at: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.neural.rprop.RPropNodeFactory2 (accessed 03.09.2025).

10. *The question is on EasyOffer*. Available at: https://easyoffer.ru/question/5476 (accessed 02.04.2025).

11. *KNIME Hub: K Nearest Neighbor*. Available at: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.knn.KnnNodeFactory2 (accessed 03.09.2025).

Для цитирования:

Исследование методов машинного обучения для обнаружения SQL-инъекций / В.Г. Лапин, Ю.А. Андрусенко, Д.Э. Суренков, Д.А. Федаш, А.А. Соломянко // Безопасность цифровых технологий. – 2025. – № 3 (118). – С. 74–92. – DOI: 10.17212/2782-2230-2025-3-74-92.

For citation:

Lapin V.G., Andrusenko Yu.A., Surenkov D.E., Fedash D.A., Solomyanko A.A. Issledovanie metodov mashinnogo obucheniya dlya obnaruzheniya SQL-in"ektsii [Research of machine learning methods for detecting SQL injections]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2025, no. 3 (118), pp. 74–92. DOI: 10.17212/2782-2230-2025-3-74-92.