Подход к обнаружению вторжений на основе федеративного обучения

Е. С. Новикова, С. А. Голубев
Санкт-Петербургский Федеральный исследовательский центр Российской академии наук
novikova@comsec.spb.ru, ser9800@gmail.com

***Аннотация*. Применение федеративного обучения (ФО) для построения систем обнаружения вторжений позволяет решить проблему использования данных с ограниченным доступом, которые не могут быть выложены в открытый доступ или переданы третьим лицам, для формирования аналитических моделей выявления вторжений. Данный подход позволяет расширить множество данных, используемых для обучения интеллектуальных моделей, и повысить тем самым уровень детектируемости разнородных атак. В настоящей работе предлагается архитектура системы обнаружения вторжений (СОВ), которая построена с учетом принципов федеративного обучения. Учитывая, что применение ФО предъявляет определенные требования к вычислительным ресурсам компонентов СОВ и пропускной способности сети, предложена методика оценки эффективности СОВ на основе ФО, учитывающая не только точность обнаружения атак, но и ресурсоемкость процесса обучения.**

Ключевые слова: обнаружение вторжений, федеративное обучение, машинное обучение, горизонтальное разделение данных, оценка эффективности

I. Вступление

В последнее время исследователи уделяют большое внимание федеративному обучению (ФО) как способу построения аналитических моделей в распределенных системах обнаружения вторжений [1]–[4]. Данная вычислительная парадигма позволяет обеспечить конфиденциальность данных, используемых для обучения модели, и представляет собой естественный способ расширения набора обучающих данных за счет включения нескольких наборов данных. Эксперименты показали, что модели обнаружения атак, обученные в федеративном режиме на нескольких наборах данных, имеют более высокий уровень обнаружения ранее неизвестных атак по сравнению с моделями, обученными только на одном наборе данных.

Федеративное обучение – это распределенное совместное машинное обучение, позволяющее обучать модели локально на устройствах, которые генерируют данные, не передавая их в общее хранилище, как это обычно происходит при классической схеме машинного обучения. Глобальная модель рассчитывается с учетом параметров локальных моделей. Системы ФО состоят из трех основных компонентов: (1) клиентов, которые генерируют данные и обучают на них локальные модели; (2) агрегирующего сервера, который координирует процесс обучения и вычисляет глобальную модель; и (3) вычислительно-коммуникационной среды. Явное присутствие последнего компонента в структуре ФО свидетельствует о том, что исключать его влияние на производительность системы ФО не представляется возможным. Таким образом, системы ФО обычно характеризуются тремя свойствами: схемой взаимодействия между клиентами, схемой разделения данных и вычислительными и сетевыми ресурсами, которыми обладают взаимодействующие клиенты.

Исследование выполнено за счет гранта Российского научного фонда № 22-21-00724, https://rscf.ru/project/22-21-00724.

Как правило, в СОВ на основе ФО используется централизованная топология связи, заключающаяся в наличии выделенного узла, который выполняет функции сервера агрегации [5]–[7]. Схема разделения данных определяет, каким образом набор атрибутов распределен между клиентами. Если клиенты имеют одинаковый набор атрибутов, а данные описывают разные объекты, то говорят о горизонтально разделенных данных. Следует отметить, что этот случай является наиболее часто используемой схемой разделения данных в СОВ на основе ФО. Случай вертикально разделенных данных, когда все клиенты имеют разные наборы атрибутов, но данные описывают один и тот же объект, гораздо сложнее и менее исследован в задачах обнаружения вторжений [8].

Однако недавнее исследование СОВ на основе ФО показало, что существуют определенные проблемы, связанные с практическим применением ФО для разработки СОВ. Одной из таких проблем является отсутствие единой методологии оценки эффективности СОВ [3]. Методика оценки должна учитывать характерные особенности ФО, такие как разбиение данных, распределение данных, функции агрегирования, вычислительные ресурсы и объемы оперативной и постоянной памяти взаимодействующих объектов, поскольку они могут существенно влиять не только на эффективность обнаружения атак, т. е. на непосредственные функциональные характеристики СОВ, но и на ее вычислительную производительность. В данной статье рассматривается проблема оценки СОВ на основе ФО, и предлагаются рекомендации по выполнению этой задачи.

Как правило, процедура оценки СОВ включает в себя функциональные критерии, такие как возможности по сбору, хранению данных, обнаружению и предотвращению атак [9]. К функциональным критериям безопасности относятся также такие показатели, как частота обнаружения и частота ложных срабатываний для известных типов атак и неизвестных атак для настроек по умолчанию, т. е. без обновления.

Помимо функциональных критериев безопасности, необходимо учитывать и критерии производительности [9]. Оценка эксплуатационных характеристик СОВ является сложной задачей, поскольку сильно зависит от продукта и алгоритмических решений, используемых в нем. Не существует общедоступных стандартов для тестирования производительности СОВ. Другой важный

вопрос связан с тем, что характеристики производительности и обнаружения взаимосвязаны: сложные и устойчивые алгоритмы обнаружения могут иметь низкие характеристики производительности с точки зрения требования к вычислительным ресурсам и памяти. Поэтому на ранней стадии разработки алгоритмов обнаружения вторжений можно учитывать такие характеристики, как среднее использование оперативной памяти и процессора. Между тем большинство научных работ посвящено оценке только эффективности обнаружения атак [2], [3].

Таким образом, *новизна* работы заключается в разработке архитектуры СОВ на основе ФО и формированию рекомендаций по оценке производительности СОВ на основе ФО с указанием параметров, влияющих на производительность на четырех различных уровнях, а именно: на уровне данных, на уровне настроек ФО, на уровне модели анализа и уровне ресурсов клиента.

Работа построена следующим образом. В разделе II представлена архитектура СОВ на основе ФО, определены место и роль ФО в системе. В разделе III представлены предлагаемые рекомендации. В разделе IV приводится описание сценария эксперимента, который может быть использован для оценки СОВ на основе ФО. В конце статьи приводятся выводы и указываются дальнейшие направления работы.

II. Архитектура СОВ на основе ФО

Типовая архитектура СОВ состоит из компонентов сбора данных (сенсоров), анализаторов данных (механизмов обнаружения), которые могут реализовывать различные стратегии обнаружения атак, в том числе основанные на правилах и машинном обучении, и компонента реагирования, принимающего на вход параметры конфигурации системы. База знаний для механизмов обнаружения должна постоянно обновляться производителем СОВ с учетом появления новых информационных угроз. В случае облачных СОВ потоки данных от датчиков безопасности перенаправляются в облачный операционный центр безопасности (SOC), что позволяет проводить глобальную аналитику безопасности, а также разрабатывать новые методы обнаружения вторжений [10].

Применение ФО для проектирования СОВ позволяет настраивать механизмы обнаружения на основе машинного обучения локально, т. е. с учетом локально генерируемых данных, и поддерживать их постоянное обновление без передачи данных на облачный сервер безопасности. Таким образом, использование ФО изменяет существующие потоки данных в среде СОВ. На рис 1 показаны основные компоненты системы обнаружения вторжений на основе ФО и потоки данных между ними.

Структура облачного сервера расширяется за счет компонентов, отвечающих за организацию ФО. Например, компонент выбора настроек ФО отвечает за определение стратегии выбора клиента в процессе обучения в зависимости от ресурсов, доступных агенту СОВ, и других атрибутов, таких как уровень доверия к этому агенту. Он также передает настройки для ФО, указывая тип модели анализа, ее архитектуру, количество раундов агрегирования, настройки для локального обучения в зависимости от конфигурации агента и т.д. Облачный сервер СОВ может быть также оснащен компонентом, отвечающим за обнаружение атак, специфических для ФО, к ним относятся атаки, отравляющие данные или модели.

Структура интеллектуального агента СОВ также дополняется компонентом, отвечающим за локальное обучение модели. Этот модуль отвечает за обновление аналитической модели, используемой в компоненте обнаружения атак. Функциональность остальных компонентов остается неизменной. Подробная функциональная схема компонента локального обучения приведена на рис. 2. Помимо модуля, отвечающего за локальное обучение модели, предлагается включить компонент выбора данных. Этот компонент осуществляет отбор и разметку новых данных в случае ограниченных возможностей по хранения данных, используемых для локального обучения. Проблема отбора данных в настоящее время недостаточно изучена, однако возможное решение можно найти в [11]. Компонент оценки рисков конфиденциальности данных является необязательным, его цель – оценка рисков конфиденциальности в процессе обучения.

III. Рекомендации по оценки эффективности СОВ на основе ФО

Рекомендации по экспериментальной оценке эффективности алгоритмов обнаружения вторжений на основе ФО частично основаны на руководстве, разработанном для оценки алгоритмов агрегирования ФО [12], и включают два аспекта:

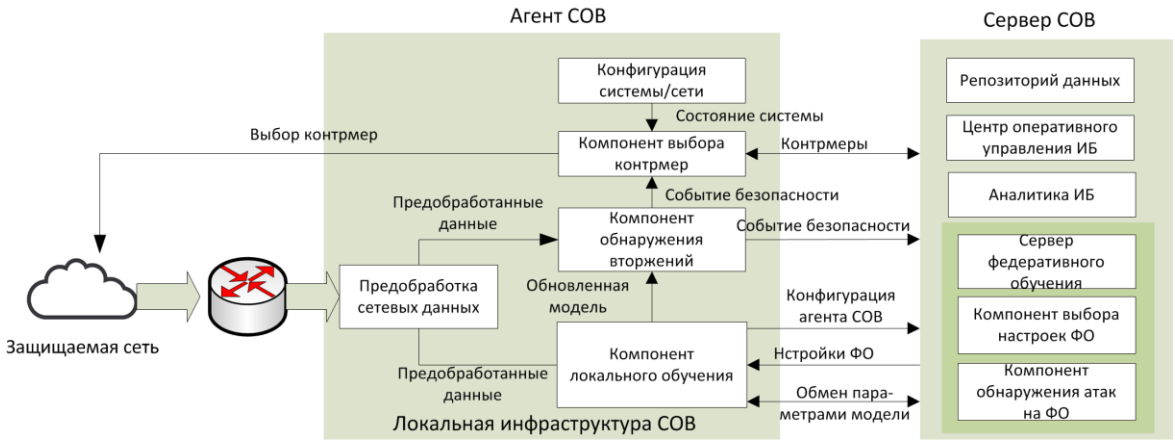

Рис. 1. Компоненты СОВ на основе ФО и основные потоки данных

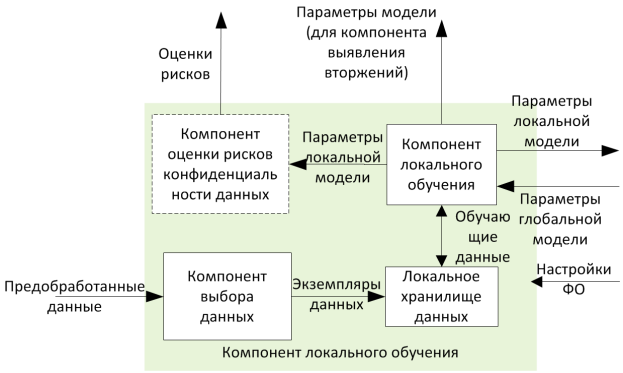

Рис. 2. Структура компонента локального обучения

1) определение параметров, влияющих на эффективность моделей машинного обучения и алгоритмов, используемых для обнаружения атак;

2) определение метрик, позволяющих отслеживать и сравнивать производительность систем СОВ на основе ФО и производительность машинного обучения.

В табл. I приведен набор параметров, влияющих на общую производительность подхода к обнаружению вторжений, построенного с помощью ФО. Они разделены на четыре группы по степени возможности манипулирования ими в процессе оценки: (1) данные, (2) настройки ФО, (3) параметры машинного обучения и (4) характеристики клиента.

Первая группа параметров характеризует такие аспекты распределения данных между клиентами, как схема разделения, размеры локальных наборов данных, разнородность представленных в них атак. Прямо или косвенно они характеризуют устойчивость алгоритма агрегирования ФЛ к неравномерному, неидентичному распределению данных.

Вторая группа параметров характеризует настройки ФО, и в основном описывает коммуникационную схему алгоритма агрегации локальных моделей: количество раундов вычисления глобальной модели, поддержка отключения/подключения устройств и требуемая синхронность обучения. Эти параметры существенно влияют на пропускную способность обучения и определяют требования к пропускной способности сети.

Уровень модели машинного обучения описывает параметры, специфичные для модели машинного обучения, они зависят от типа самой модели и могут включать число внутренних слоев, число свёрточных слоев, настройки используемых фильтров, функций активации и т.д.

Последняя группа параметров относится непосредственно к характеристикам вычислительных ресурсов и памяти клиентов.

Для мониторинга эффективности различных СОВ на основе ФО авторы предлагают использовать следующие параметры:

- производительность глобальной модели,
- время обучения,
- объем трафика, генерируемого в процессе обучения,
- потребление памяти,
- загрузка процессора.

Последние четыре параметра характеризуют процесс обучения. Этот процесс должен осуществляться регулярно в процессе функционирования СОВ, поэтому необходимо оценить его влияние на производительность системы в целом, чтобы определить вычислительные ресурсы и пропускную способность сети компонента СОВ, отвечающего за локальное обучение на основе ФО. Размер обучающего набора данных может быть использован для определения требований к локальному хранилищу.

IV. Экспериментальный сценарий по оценке эффективности СОВ на основе ФО

В этом разделе обсуждается сценарий эксперимента для оценки СОВ на основе ФО, в котором соблюдены

ТАБЛИЦА I. Параметры, влияющие на эффективность СОВ на основе ФО

Уровень параметров	Параметр	Описание
Уровень данных	Разделение данных	Параметр определяет, как данные будут разделены между клиентами; существует три типа разделения данных: горизонтальное, вертикальное и гибридное.
	Распределение данных	Параметр характеризует способность алгоритма обнаружения вторжений на основе ФО работать с разнородными неидентично распределенными данными. В случае систем СОВ различные схемы распределения связаны с различным распределением атак по аимодействующим клиентам.
	Различия в данных	Параметр характеризует различия в размерах наборов данных, принадлежащих разным клиентам, и характеризует способность системы работать с разными наборами данных.
Уровень настроек ФО	Скорость обмена данными	Значение параметра зависит от числа раундов аггрегирования, необходимых для вычисления глобальной модели. Он также определяет объем передаваемого трафика при обучении глобальной модели.
	Количество клиентов	Параметр характеризует масштабируемость системы обнаружения вторжений на основе ФО и, соответственно, ее способность работать с различным количеством клиентов.
	Количество клиентов, отключающихся из обучения	Обычно данный параметр характеризует процент клиентов, которые могут быть отключены во время процесса обучения.
	Синхронность	Поддержка синхронного и асинхронного ФО обучения.
Уровень настроек машинного обучения	Настройки машинного обучения	Характеризует настройки моделей машинного обучения. Различные модели машинного обучения имеют различные параметры, влияющие также на частоту обнаружения атак, частоту ложных срабатываний и т.д.
	Вычислительные характеристики локального обучения	Этот параметр зависит от числа эпох локального обучения, и определяет требования к вычислительным ресурсам клиента.
Уровень настроек клиентов	Соотношение ресурсов клиентов	Данный параметр характеризует разницу (в %) вычислительных ресурсов клиентов.

предложенные рекомендации, и приводится описание экспериментального стенда.

Компонент ФО в СОВ реализован на языке Python и основан на программном фреймворке Flower [13]. Данный фреймворк не требователен к вычислительным ресурсам и является легко расширяемым, он поддерживает различные алгоритмы агрегирования, устойчивые к неидентично распределенным данным [14], [15]. Фреймворк ФО Flower также имеет встроенные возможности по мониторингу системных ресурсов в симуляционном режиме для предварительной оценки производительности.

Другой вопрос, который необходимо решить при оценке СОВ на основе ФО, связан с моделированием распределения данных между клиентами. Существует два основных подхода к решению этой проблемы. В первом подходе один набор данных разбивается по нескольким клиентам, и каждый клиент получает определенный тип атаки (или несколько типов атак). При втором подходе используются различные наборы данных с одним и тем же набором признаков, и каждый набор данных предполагается как данные клиента. В сценарии эксперимента предлагается рассмотреть оба случая.

В первом случае используется набор данных CICIDS [16]. Он описывает пять дней функционирования информационной системы, причем первый день характеризует нормальное поведение системы, а остальные четыре дня содержат различные типы атак. Предлагается разбить его на два набора данных таким образом, чтобы первый набор содержал атаки второго и третьего дня, второй – четвертого и пятого дней, а нормальные данные выбирались из первого дня для каждого набора данных. Для моделирования второго случая используются обработанные версии четырех различных наборов данных, а именно UNSW-NB15, BoT-IoT, ToN-IoT и CSE-CIC-IDS2018 [17]. Эти наборы данных содержат только характеристики сетевых потоков, поэтому их можно легко использовать для моделирования горизонтально разделенных данных на четырех клиентах. Поскольку распределение атак по клиентам в обоих случаях различно, в предлагаемом экспериментальном сценарии решается проблема моделирования неидентичных распределенных данных.

Клиенты и сервер ФО были развернуты в среде Yandex.Cloud. Использовались четыре виртуальные машины со следующими характеристиками: 8 ядер, 16 Гбайт ОЗУ и 30 Гбайт HDD. В тестовых экспериментах использовалась трехслойная сверточная нейронная сеть. В качестве функции агрегирования использовался классический алгоритм FedAvg с агрегированием параметров локальных моделей после каждой эпохи. Предварительные результаты первого экспериментального случая показали высокую скорость обнаружения атак: 99% точности на обоих клиентах, несмотря на различную схему распространения атак.

V. Заключение

В данной статье авторы рассматривают проблему оценки СОВ на основе ФО и предлагают рекомендации по оценке таких систем с учетом эффективности обнаружения атак в контексте требований к вычислительным ресурсам и памяти предлагаемого решения. В предлагаемых рекомендациях определены параметры, влияющие на эффективность обучения ФО, включая глобальную точность модели, а также предложен набор метрик для мониторинга и сравнения вычислительной эффективности различных алгоритмов.

Авторы также разработали сценарий эксперимента, который учитывает предложенные рекомендации, он включает определение наборов данных, моделирование разбиения данных и неидентичного распределения данных, выбор фреймворка ФО и настройки экспериментального стенда.

Дальнейшие исследования включают в себя тщательную оценку стратегий агрегации для описанного распределения данных, оценку контролируемых метрик. Еще одно направление исследований связано с определением стратегии отбора данных для локального обучения.

Список литературы

[1] P.V. Astillo, D.G. Duguma, H. Park, J. Kim, B. Kim, and I. You, “Federated intelligence of anomaly detection agent in iotmd-enabled diabetes management control system,” FGCS, vol. 128, pp. 395–405, 2022.

[2] E.M. Campos, P.F. Saura, A. Gonzalez-Vidal, J.L. Hernandez-Ramos, J.B. Bernabe´, G. Baldini, and A. Skarmeta, “Evaluating federated learning for intrusion detection in internet of things: Review and challenges,” Computer Networks, vol. 203, p. 108661, 2022.

[3] E. Fedorchenko, E. Novikova, and A. Shulepov, “Comparative review of the intrusion detection systems based on federated learning: Advantages and open challenges,” Algorithms, vol. 15, no. 7, 2022.

[4] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K.-K. R. Choo, and M. Nafaa, “Felids: Federated learning-based intrusion detection system for agricultural internet of things,” J. of Parallel and Distributed Computing, vol. 165, pp. 17–31, 2022.

[5] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, “Fluids: Federated learning with semi-supervised approach for intrusion detection system,” in 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 2022, pp. 523–524.

[6] Y. Qin and M. Kondo, “Federated learning-based network intrusion detection with a feature selection approach,” in 2021 Int. Conference on Electrical, Com., and Comp. Eng. (ICECCE), 2021, pp. 1–6.

[7] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, “Iotdefender: A federated transfer learning intrusion detection framework for 5g iot,” in 2020 IEEE 14th Int. Conference on Big Data Science and Engineering (BigDataSE), 2020, pp. 88–95.

[8] E. Novikova, E. Doynikova, and S. Golubev, “Federated learning for intrusion detection in the critical infrastructures: Vertically partitioned data use case,” Algorithms, vol. 15, no. 4, 2022.

[9] K.A. Scarfone and P.M. Mell, “Sp 800-94. guide to intrusion detection and prevention systems (idps),” Gaithersburg, MD, USA, Tech. Rep., 2007.

[10] F. A. Saputra, M. Salman, J. A. N. Hasim, I. U. Nadhori, and K. Ramli, “The next-generation nids platform: Cloud-based snort nids using containers and big data,” BD and CC, vol. 6, no. 1, 2022.

[11] C. Gong, Z. Zheng, F. Wu, Y. Shao, B. Li, and G. Chen, “To store or not? online data selection for federated learning with limited storage,” in Proc. of the ACM Web Conference 2023, p. 3044–3055.

[12] A. Mulay, B. Gaspard, R. Naidu, S. Gonzalez-Toral, V. S. T. Semwal, and A. Manish Agrawal, “Fedperf: A practitioners’ guide to performance of federated learning algorithms,” in NeurIPS 2020 vol. 148. PMLR, 11 Dec 2021, pp. 302–324.

[13] “Flower a friendly federated learning framework,” accessed on 12.07.2023. [Online]. Available: https://flower.dev/

[14] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, “Fedbn: Federated learning on non-iid features via local batch normalization,” CoRR, vol. abs/2102.07623, 2021.

[15] S.J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konecny, S. Kumar, and H. B. McMahan, “Adaptive federated optimization,” CoRR, vol. abs/2003.00295, 2020.

[16] I. Sharafaldin, A. Habibi Lashkari, and A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” 01 2018, pp. 108–116.

[17] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, “Netflow datasets for machine learning-based network intrusion detection systems,” in Big Data Technologies and Applications, Cham: Springer International Publishing, 2021, pp. 117–135.