Анализ структуры и характеристики многослойных автокодировщиков, используемых для обнаружения компьютерных атак

О.И. Шелухин, А.В. Ванюшина

Московский технический университет связи и информатики, г. Москва

Аннотация: Анализируются возможности использования нейронных сетей типа многослойного автокодировщика (МАК) для улучшения характеристик обнаружения компьютерных атак. Рассмотрена структура МАК, предназначенного для сокращения размерности больших массивов данных, подлежащих обработке в задачах обнаружения компьютерных атак. Анализируется использование различных функций активации нейронов сети и наиболее часто применяемые функции потерь, определяющие качество реконструкции оригинала. Рассмотрен алгоритм оптимизации параметров автокодировщика, позволяющий ускорить обучение модели, снизить вероятность его переобучения и минимизировать функцию потерь.
Ключевые слова. Нейронные сети, слои, нейроны, функция потерь, функция активации, мобильные приложения, атаки, гиперпараметры, оптимизация, машинное обучение

Постановка задачи

За последнее десятилетие было предложено множество методов машинного обучения для улучшения эффективности обнаружения компьютерных атак (КА) [1-3]. Одним из популярных подходов [4] является использование искусственной нейронной сети (ИНС) для выполнения проверки сетевого трафика. Преимущество использования ИНС заключается в том, что она хорошо обучается. Это дает ИНС большое преимущество в эффективности обнаружения по сравнению с другими алгоритмами машинного обучения [2,5].

Распространенный подход к использованию ИНС в качестве систем обнаружения вторжений (СОВ) заключается в обучении классифицировать сетевой трафик как нормальный либо аномальный [6-8].

В основе обнаружения аномалий, в том числе КА с помощью ИНС, лежит способность восстанавливать входную информацию на выходе. Если сформировать в качестве обучающей выборку из нормальных данных, и

обучить ИНС воспроизводить эти данные на выходе, а после обучения подать на вход ИНС, например, КА, то ошибка реконструкции для входного нормального процесса будет значительно меньше, чем для входного аномального процесса. При превышении ошибки реконструкции заданного порогового значения принимается решение о принадлежности входного процесса к классу аномальных.

Для обнаружения аномальных процессов (например, компьютерных атак) хорошие результаты показывают ИНС, названные автокодировщиками (АК) [9-11].Использование таких нейронных сетей в ряде случаев приносит хорошие результаты, в том числе для извлечения признаков из наборов данных с целью улучшения обнаружения КА.

В работах [12,13] рассматриваются классификации нежелательных мобильных приложений, основанные на использовании многослойных автокодировщиков **(**МАК). При реализации признаки экземпляра обрабатываемых данных сначала отображаются на нейроны ансамбля. Затем каждый МАК пытается восстановить признаки экземпляра и вычисляет ошибку восстановления в среднеквадратичной ошибке (MSE, Mean Squared Error) либо (RMSE, Root Mean Squared Error). После чего значения MSE или RMSE передаются на выход МАК, который действует как нелинейный механизм голосования для ансамбля. Целью работы является анализ структуры и особенности функционирования многослойных автокодировщиков для обнаружения и классификации как компьютерных атак, так и аномалий мобильных приложений.

Модели АК

Автокодировщик это ИНС, которая сначала кодирует входной сигнал в некоторое скрытое состояние, размерность которого, как правило, меньше

размерности входного сигнала, а затем, из скрытого состояния снова разворачивает (декодирует) данные в другое, новое состояние. ИНС состоят из L слоев нейронов, где каждый i-й слой $l^{(i)}$ последовательно соединен через синапсы со связанными весами $W^{(i)}$. Основой для построения всех моделей АК является модель простого трехслойного автокодировщика. Это сеть прямого распространения с входным и выходным слоями, содержащими одинаковое число нейронов, и единственным внутренним (горловым) слоем, содержащим меньшее число нейронов, чем входной и выходной слои (см. рис. 1).

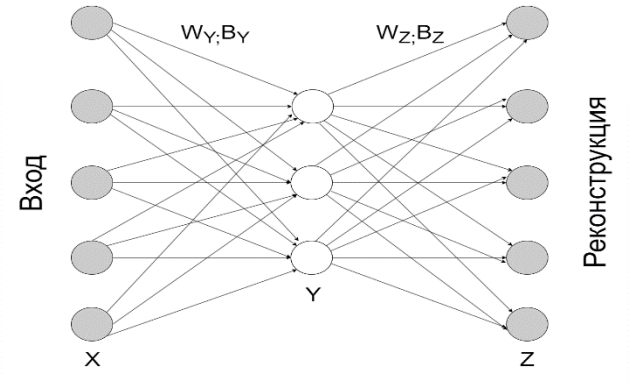

Рисунок 1. Структура трехслойного АК

Будем считать $\boldsymbol{X}_1.\boldsymbol{X}_2,\cdots\boldsymbol{X}_M \in \boldsymbol{R}^N$ - векторами входных данных, характеризующими анализируемые мобильные приложения. Тогда матрицу входных данных можно представить в виде $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots \mathbf{X}_\mathrm{M}]^\mathrm{T}$, в которой каждая строка представляет собой вектор обрабатываемых признаков (атрибутов) M анализируемых приложений, а число столбцов N характеризует размерность пространства признаков. В результате матрица X представляет собой матрицу размера N x M

$$\boldsymbol{X} = \begin{vmatrix} \boldsymbol{X}_{11} & \boldsymbol{X}_{12} & \cdots & \boldsymbol{X}_{1M} \\ \boldsymbol{X}_{21} & \boldsymbol{X}_{22} & \cdots & \boldsymbol{X}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{X}_{N1} & \boldsymbol{X}_{N2} & \cdots & \boldsymbol{X}_{NM} \end{vmatrix}, \boldsymbol{X} \in \boldsymbol{R}^{N \times M}.$$

Рассмотрим структуру многослойного автокодировщика, предназначенного для сокращения размерности больших массивов данных, подлежащих обработке. МАК [6,9,14] представляет собой специальный вид многослойной сети прямого распространения – многослойный симметричный персептрон, содержащий несколько внутренних слоев уменьшающегося размера и слой «бутылочная горловина» в середине сети. МАК - сеть производит тождественное преобразование входного слоя в выходной. В результате работы МАК - сети в горловом слое появляется вектор, компонентами которого являются «признаки» – обобщенные характеристики входного массива данных, извлеченные из исходных данных и содержащие дополнительную существенную и не избыточную информацию, определяющую входной массив данных в пространстве меньшей размерности, в так называемом скрытом (latent**)** пространстве.

Задачей латентного пространства является выделение важных признаков (атрибутов), которые будут использоваться для восстановления исходных данных при максимально малой размерности слоя.

Размерности скрытых слоев зависят от желаемой степени сжатия входных данных, количества признаков выборки и целевого значения размерности латентного пространства – параметра, который влияет на способность модели к обучению и реконструкции. Слой Y содержит меньшее количество информативных параметров обрабатываемого массива данных извлеченных в процессе работы АК. Меньшая размерность скрытого слоя может привести к более эффективному сжатию и выделению значимых

признаков. Однако при этом увеличивается риск потери информации, и наоборот.

Общая сложность реализации ИНС зависит от количества слоев и числа нейронов в каждом слое и, как правило, реализуется с использованием алгоритма стохастического градиентного спуска *(англ. Stochastic gradient descent, SGD)*. При этом сложность обучения ИНС на одном экземпляре примерно вдвое превышает сложность выполнения без использования *SGD*, что объясняется использованием метода обратного распространения.

Автокодировщики могут быть глубокими (иметь много скрытых слоев). При этом, более глубокие и широкие сети могут изучать более сложные процессы. Однако, поскольку обучение и реализация глубоких сетей могут быть дорогостоящими в вычислительном отношении, обычно рассматривается реализация АК с одним скрытым слоем.

Как показано на рисунке 1, наиболее простой АЕ представляет собой трехслойный персептрон, который имеет один скрытый слой и один выходной слой, с двумя ограничениями: матрица весов выходного слоя является транспонированной матрицей весов скрытого слоя $\widehat{\boldsymbol{W}}_Y = \widehat{\boldsymbol{W}}_Z^T = \widehat{\boldsymbol{W}}$ (т.е. веса фиксированы) и число количество выходных нейронов равно количеству входных. Значения нейронов скрытого слоя, называемые кодированием, получаются с помощью уравнения:

$$\boldsymbol{Y} = \boldsymbol{G}_\theta(\boldsymbol{X}) = \boldsymbol{F}\left(\widehat{\boldsymbol{W}}_Y\boldsymbol{X} + \boldsymbol{B}_Y\right); \boldsymbol{\theta} = \left\{\boldsymbol{W}_{Y,}\boldsymbol{B}_Y\right\}. \quad (1)$$

где $\boldsymbol{X}$ - входной вектор, **F** – функция активации нейронов сети, $\boldsymbol{B}_Y$ - вектор скрытых нейронных смещений, а $\boldsymbol{W}_{Y,}$ - матрица скрытых весов.

Задача функции кодирования $\boldsymbol{Y} = \boldsymbol{F}\left(\boldsymbol{X}, \widehat{\boldsymbol{W}}_Y, \boldsymbol{B}_Y\right)$ заключается в сжатии входного вектора. Операция декодирования характеризуется функцией декодирования $\boldsymbol{Z} = \boldsymbol{F}\left(\boldsymbol{Y}, \widehat{\boldsymbol{W}}_Z, \boldsymbol{B}_Z\right)$ и заключается в восстановлении входного «сжатого» вектора

$$\boldsymbol{Z} = \boldsymbol{G}_{\hat{\theta}}(\boldsymbol{Y}) = \boldsymbol{F}\left(\widehat{\boldsymbol{W}}_{\boldsymbol{Z}}\boldsymbol{Y} + \boldsymbol{B}_{\boldsymbol{Z}}\right);\ \hat{\boldsymbol{\theta}} = \left\{\boldsymbol{W}_{\boldsymbol{Z},}\boldsymbol{B}_{\boldsymbol{Z}}\right\}. \quad (2)$$

В формулах (1) и (2) $\widehat{\boldsymbol{W}}_{\boldsymbol{Y}}$ и $\widehat{\boldsymbol{W}}_{\boldsymbol{Z}}^{\boldsymbol{T}}$ это матрицы сетевых связей (матрицы весовых коэффициентов) кодировщика и декодировщика АК. Весовые коэффициенты $\widehat{\boldsymbol{W}}$ **это** векторы смещения $\boldsymbol{B}_{\boldsymbol{Y}}$ и $\boldsymbol{B}_{\boldsymbol{Z}}$, которые определяют важность каждого входного сигнала для вычисления выходных значений слоя.

Параметры $\boldsymbol{\theta} = \left\{\boldsymbol{W}_{\boldsymbol{Y},}\boldsymbol{B}_{\boldsymbol{Y}}\right\}$ и $\hat{\boldsymbol{\theta}} = \left\{\boldsymbol{W}_{\boldsymbol{Z},}\boldsymbol{B}_{\boldsymbol{Z}}\right\}$ представляют собой наборы параметров отображения.

Каждый нейрон имеет свое собственное смещение, не зависящее от входных данных, и настраивается в процессе обучения модели вместе с весами. Количество весов определяется количеством нейронов на предыдущем слое, а количество смещений – количеством нейронов на текущем. Общее количество параметров определяется соотношением:

$(\boldsymbol{input}_{\boldsymbol{dimension}} + \mathbf{1}) * \boldsymbol{cur}_{\text{dense\_dimension}}$, где input_dimension – размерность предыдущего слоя (или размер значения выборки) – значение веса, cur_dense_dimension – размерность текущего слоя – значение смещения.

Целью обучения АК является минимизация функции потерь, характеризующей разницу между входными $\boldsymbol{X}$ и выходными $\boldsymbol{Z}$ данными. Типичная функция потерь представляет собой среднеквадратическую ошибку:

$$\boldsymbol{L}(\boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{X} - \boldsymbol{Z}\|^2. \quad (3)$$

Соотношение (3) может быть преобразовано, используя уравнения (1) и (2) к виду:

$$\boldsymbol{L}(\boldsymbol{X}, \boldsymbol{Y}) = \left\|\boldsymbol{X} - \boldsymbol{F}(\widehat{\boldsymbol{W}}_{\boldsymbol{Z}}\left(\boldsymbol{F}\left(\widehat{\boldsymbol{W}}_{\boldsymbol{Y}}\boldsymbol{X} + \boldsymbol{B}_{\boldsymbol{Y}}\right)\right) + \boldsymbol{B}_{\boldsymbol{Z}})\right\|^2. \quad (4)$$

Функция потерь $\boldsymbol{L}(\boldsymbol{X}, \boldsymbol{Y})$ определяет качество реконструкции оригинала, так что выходная реконструкция должна быть как можно ближе к исходному входному вектору. Для повышения точности необходимо

минимизировать потери функции и обновлять параметры. Распространенные функции потерь приведены в таблице 1.

Таблица 1.

Функции потерь

Функция потерь $L(X,Y)$	Аналитический показатель	Название
$MSE(x,z)$	$\frac{1}{N}\sum_{i=1}^{M}(x_i - z_i)^2$	Среднеквадратичная ошибка (Mean Squared Error, MSE
RMSE(x, z)	$(\frac{1}{N}\sum_{i=1}^{M}(x_i - z_i)^2)^{1/2}$.	Корень квадратный из среднеквадратичной ошибки (Root Mean Squared Error, RMSE)
IRE_X	$\sqrt{\sum_{i=1}^{N}(x_i - z_i)^2}$,	Мгновенная ошибка реконструкции (Immediate Reconstruction Error, IRE)
MAE	$\frac{1}{N}\sum_{i=1}^{N}\lvert f(x_i) - y_i\rvert$	Абсолютная ошибка (Mean Absolute Error, MAE)
$C_{CE}(x,z)$	$-\sum_{k=1}^{n_x}(x\ln(z) + (1-x)\ln(1-z))$	кросс-энтропия (Перекрестная энтропия) (CE)

В структуре АК могут использоваться различные функции активации нейронов сети [11]. Нелинейность функции активации позволяет извлекать из исходных данных более существенные обобщенные характеристики в исходных данных устраняя как линейные, так и нелинейные корреляции. Наибольшее распространение получили функции активации, представленные в таблице 2.

Таблица 2.

Функции активации в нейронных сетях

Название	$F(x)$	Комментарии
LReLU	$f(x) = \max(ax, x)$	(Leaky Rectified Linear Activation), a=0,01…0,03.
ReLU	$\max(0, x)$	Rectified Linear Unit
Sigmoid	$\frac{e^x}{e^x + 1}$	Сигмоидальная функция
ELU	$\begin{cases} x & if\ x > 0 \\ a(e^x - 1) & if\ x \leq 0 \end{cases}$	Exponential Linear Unit; a –гиперпараметр,

Основная цель обучения МАК состоит в том, чтобы найти оптимальные параметры ($\boldsymbol{\theta}$ и $\widehat{\boldsymbol{\theta}}$), которые могут эффективно минимизировать разницу между входными и восстановленными выходными данными по всему обучающему набору:

$$\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{B}\} = arg_{\theta} min \boldsymbol{L}(\boldsymbol{X}, \boldsymbol{Y}). \quad (5)$$

Использование оптимизаторов позволяют ускорять обучение модели, снижать вероятность ее переобучения и минимизировать функцию потерь. Оптимизаторы определяют лучший набор параметров модели, таких, как матрицы весовых коэффициентов $\boldsymbol{W}$… и векторы смещений $\boldsymbol{B}_Y$ и $\boldsymbol{B}_Z$.

В процессе работы обычно используется оптимизатор Adam (Adaptive Moment Estimation**)** [12-14] работа которого заключается в постоянном сохранении и обновлении градиентов **P** (скорость изменения функции потерь по каждому параметру модели – вес, смещение) и квадратов градиентов **G.** Это позволяет алгоритму машинного обучения адаптироваться к скорости изменения параметров. Adam обновляет веса, используя градиенты и их квадраты, что позволяет в процессе обучения минимизировать функции потерь. Оптимизация смещений позволяет влиять на порог активации

нейрона, что, в свою очередь, влияет на результаты работы как слоя, так и модели в целом.

В процессе своей работы Adam за одну итерацию оптимизации выполняет следующие шаги:

1) Вычисляются экспоненциальное скользящее среднее (EMA exponential moving average). EMA – метод сглаживания, вычисляющий среднее значение последовательности, придавая больший вес более новым наблюдениям с учетом градиентов и квадратов градиентов при каждом новом наблюдении. Вычисления для градиентов $\mathbf{P}$ и квадратов градиентов $\mathbf{G}$ осуществляются с использованием соотношений:

$$\mathbf{P} = \boldsymbol{\beta_1} * \mathbf{P} + (\mathbf{1} - \boldsymbol{\beta_1}) * \boldsymbol{\nabla}\mathbf{J}(\theta), \quad (6)$$

$$\mathbf{G} = \boldsymbol{\beta_2} * \mathbf{G} + (\mathbf{1} - \boldsymbol{\beta_2}) * \left(\boldsymbol{\nabla}\mathbf{J}(\theta)\right)^{\mathbf{2}}, \quad (7)$$

где, β_1 иβ_2 – коэффициент затухания для EMA, θ – вектор параметров модели, $\nabla J(\theta)$ – градиент функции потерь по параметрам модели.

2) Корректируется значение первого и второго моментов:

$$\widehat{\boldsymbol{P}} = \frac{\boldsymbol{P}}{(1-\boldsymbol{\beta}_1)^t}; \quad (8)$$

$$\widehat{\boldsymbol{G}} = \frac{\boldsymbol{G}}{(1-\boldsymbol{\beta}_2)^t} \quad (9)$$

где t – номер итерации процесса оптимизатора.

3) Обновляются параметры модели – данный шаг происходит согласно формуле $\theta = \theta - \frac{\boldsymbol{\alpha}*\widehat{\boldsymbol{P}}}{\sqrt{\widehat{\boldsymbol{G}}}}$, где α – скорость обучения.

Размерность входного и выходного слоев определяются размером входных признаков. Размерности скрытых слоев зависят от желаемой степени сжатия входных данных, количества признаков выборки и целевого значения размерности латентного пространства – параметра, который влияет на способность модели к обучению и реконструкции.

Выводы

Проведенный анализ показал, что использование нейронных сетей типа многослойного автокодировщика позволяет улучшить характеристики обнаружения компьютерных атак за счет фильтрующих свойств. Рассмотрена структура МАК, предназначенного для сокращения размерности больших массивов данных, подлежащих обработке в задачах обнаружения компьютерных атак. Анализируется использование различных функций активации нейронов сети и наиболее часто применяемые функции потерь, которые определяют качество реконструкции.

Рассмотрен алгоритм оптимизации параметров автокодировщика, позволяющий ускорить обучение модели, снизить вероятность ее переобучения и минимизировать функцию потерь.

Литература

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA: The MIT Press, 2017. 767 p.

2. Damopoulos S. A., Menesidou, G., Kambourakis M., Papadaki N., Clarke S. Gritzalis Evaluation of anomaly-based ids for mobile devices using machine learning classifiers // Security and Communication Networks. 2012. №5(1). Pp. 3–14.

3. Yinhui Li, Jingbo Xia, Silan Zhang, Jiakai Yan, Xiaochuan Ai, and Kuobin Dai. An efficient intrusion detection system based on support vector machines and gradually feature removal method // Expert Systems with Applications. 2012. №39(1). Pp. 424–430.

4. Ranjan S. Machine learning based botnet detection using real-time extracted traffic features, US Patent 8682812B1, 2014.

5. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey. ACM Comput. 2009. Surv.. 41. 10.1145/1541880.1541882.

6. Demuth H., Beale M., Jess O., Hagan M. Neural network design. Martin Hagan, 2014.

7. Srivastav N., Challa R.K. Novel intrusion detection system integrating layered framework with neural network // In Advance Computing Conference (IACC). 2013 IEEE 3rd International. pp.682–689.

8. Naoum R., Abid N., AlSultani Z. An enhanced resilient backpropagation artificial neural network for intrusion detection system // International Journal of Computer Science and Network Security (IJCSNS). 2012. №12(3). p.11.

9. Golovko V. Neural network and artificial immune systems for malware and network intrusion detection // Studies in computational intelligence. – Heidelberg, 2010. – Vol. 263 : Advances in machine learning II. – pp. 485 – 513

10. Sung J.K., Jo W., Shon T. APAD: Autoencoder-based Payload Anomaly Detection for industrial IoE. December 2019 Applied Soft Computing 88:106017. DOI: 10.1016/j.asoc.2019.106017.

11. Baldi P. Autoencoders, unsupervised learning, and deep architectures //Proceedings of ICML workshop on unsupervised and transfer learning. 2012. Pp. 37 – 49.

12. Шелухин О.И., Барков В.В., Маторин Ф.А. Повышение эффективности классификации противоправных и нежелательных приложений в условиях фонового трафика с помощью автокодировщиков // Вестник Санкт-Петербургского государственного университета технологии и дизайна: Серия 1. Естественные и технические науки. 2023. № 3. С. 159–165.

13. Шелухин О.И., Зегжда Д.П., Раковский Д.И., Самарин Н.Н., Александрова Е.Б. Интеллектуальные технологии информационной безопасности. / Под ред. О. И. Шелухина. – М.: Горячая линия – Телеком, 2023. – 384с.: ил. ISBN 978-5-9912-1084-3.

14.Кузьмина М.Г. Многослойные сети-автоэнкодеры в задачах анализа и обработки гиперспектральных изображений // Препринты ИПМ им. М.В.Келдыша. 2021. № 28. 21 с. doi.org/10.20948/prepr-2021-28

References

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA: The MIT Press, 2017. 767 p.

2. Damopoulos S. A., Menesidou, G., Kambourakis M., Papadaki N., Clarke S. Gritzalis Evaluation of anomaly-based ids for mobile devices using machine learning classifiers. Security and Communication Networks. 2012. №5(1). Pp. 3–14.

3. Yinhui Li, Jingbo Xia, Silan Zhang, Jiakai Yan, Xiaochuan Ai, and Kuobin Dai An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Systems with Applications. 2012. №39(1). Pp. 424–430.

4. Ranjan S. Machine learning based botnet detection using real-time extracted traffic features, US Patent 8682812B1, 2014.

5. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey. ACM Comput. 2009. Surv.. 41. 10.1145/1541880.1541882.

6. Demuth H., Beale M., Jess O., Hagan M. Neural network design. Martin Hagan, 2014.

7. Srivastav N., Challa R.K. Novel intrusion detection system integrating layered framework with neural network. In Advance Computing Conference (IACC). 2013 IEEE 3rd International. Pp.682–689.

8. Naoum R., Abid N., AlSultani Z. An enhanced resilient backpropagation artificial neural network for intrusion detection system. International Journal of Computer Science and Network Security (IJCSNS). 2012. №12(3). p.11.

9. Golovko V. Neural network and artificial immune systems for malware and network intrusion detection. Studies in computational intelligence. Heidelberg, 2010. Vol. 263 : Advances in machine learning II. Pp. 485 – 513.

10. Sung J.K., Jo W., Shon T. APAD: Autoencoder-based Payload Anomaly Detection for industrial IoE. December 2019 Applied Soft Computing 88:106017. DOI: 10.1016/j.asoc.2019.106017.

11. Baldi P. Autoencoders, unsupervised learning, and deep architectures. Proceedings of ICML workshop on unsupervised and transfer learning. 2012. Pp. 37 – 49.

12. Sheluhin O.I., Barkov V.V., Matorin F.A. Vestnik Sankt-Peterburgskogo gosudarstvennogo universiteta tekhnologii i dizayna: Seriya 1. Estestvennye i tekhnicheskie nauki. 2023. № 3. Pp. 159–165.

13. Sheluhin O.I., Zegzhda D.P., Rakovskiy D.I., Samarin N.N., Aleksandrova E.B. Intellektual'nye tekhnologii informatsionnoy bezopasnosti. [Intelligent information security technologies]. M.: Goryachaya liniya – Telekom, 2023. 384 p. ISBN 978-5-9912-1084-3.

14. Kuz'mina M.G. Preprinty IPM im. M.V.Keldysha. 2021. № 28. 21 p. doi.org/10.20948/prepr-2021-28

Дата поступления: 7.10.2024

Дата публикации: 24.11.2024