Проект по курсу
“Теория графов и ее приложения

Выполнил:
Мовчан Максим Владимирович

Свойства сетей (для статических графов)

1. Число вершин, число рёбер, плотность (отношение числа рёбер к максимально возможному числу рёбер), число компонент слабой связности, долю вершин в максимальной по мощности компоненте слабой связности.
2. Для наибольшей компоненты слабой связности вычислить/оценить значения радиуса, диаметра сети, 90 процентиля расстояния (геодезического) между вершинами графа. Оценку провести на основании
    a. вычисления расстояний между 500 (1000) случайно выбранными вершинами из наибольшей компоненты слабой связности;
    b. вычисления расстояний по подграфу "снежный ком" (snowball sample), построенного по следующему принципу: выбирается небольшое начальное множество вершин (2 или 3), затем в граф добавляются все их соседи, затем соседи соседей и т.д., пока число вершин в подграфе не станет равным (примерно) заданному значению (например, 500 или 1000).

3. Для наибольшей компоненты слабой связности вычислить средний кластерный коэффициент сети $\overline{Cl} = \frac{1}{|V|}\sum_{u \in G} Cl_u$, где

$$Cl_u = \begin{cases} \frac{2L_u}{|\Gamma(u)| \cdot |\Gamma(u)-1|}, & |\Gamma(u)| \geqslant 2 \\ 0, & \text{иначе.} \end{cases},$$

где $\Gamma(u)$ – множество соседей вершины, $|\Gamma(u)| = k_u$ –степень вершины, L_u --- число ребер между соседями.

4. Коэффициент ассортативности по степени вершин $-1 \leq r \leq 1$ (коэффициент корреляции Пирсона)

$$r = \frac{\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2m}\right) k_i k_j}{\sum_{i,j}\left(k_i \delta_{ij} - \frac{k_i k_j}{2m}\right) k_i k_j} = \frac{R_e R_1 - R_2^2}{R_3 R_1 - R_2^2},$$

где $R_1 = \sum_i k_i = 2m$, $R_2 = \sum_i k_i^2$, $R_3 = \sum_i k_i^3$, $R_e = \sum_{i,j} A_{ij} k_i k_j$.

Результаты вычислений для датасетов

1. Socfb-Middlebury
2. socfb-Reed98
3. testgraph_1
4. testgraph_2
5. BA_bitA
6. RA_Rado
7. BO_bitOt
8. UC_UC

Датасет	Кол-во вершин	Кол-во ребер	Случайный подграф			Снежный ком			Средний кластерный коэффициент	Коэффициент ассортативности
			Диаметр	Радиус	90 пр. расстояни	Диаметр	Радиус	90 пр. расстояни		
Socfb-Middlebury	3075	124610	8	4	4.0	4	3	3.0	0.2816	0.078
socfb-Reed98	962	18812	6	3	3.0	4	2	3.0	0.3184	0.0234
testgraph_1	9	14	3	2	3.0	-	-	-	0.3999	-0.2037
testgraph_2	34	78	5	3	4.0	-	-	-	0.5706	-0.4756

Датасет	Кол-во вершин	Кол-во ребер	Случайный подграф			Снежный ком			Средний кластерный коэффициент	Коэффициент ассортативности
			Диаметр	Радиус	90 пр. расстоянии	Диаметр	Радиус	90 пр. расстоянии		
BA_bitA	3783	14124	12	2	6.0	6	3	3.0	0.1766	-0.1685
RA_Rado	148	1086	5	3	3.0	5	3	3.0	0.5608	-0.2584
BO_bitOt	5881	21492	13	2	7.0	4	2	3.0	0.1775	-0.1648
UC_UC	1899	13838	8	2	4.0	5	3	3.0	0.1093	-0.1878

Датасет socfb-Middlebury45

```
----1.1----
Количество вершин в графе: 3075
Количество ребер в графе: 124610.0
Плотность графа: 0.02636537230694363
Количество компонент слабой связности: 4
Доля вершин в максимальной по мощности компоненте слабой связности: 0.9980487804878049
----1.2----
Метрики расстояний на случайном подграфе:
Диаметр графа: 8
Радиус графа: 4
90 процентиль расстояния в графе: 4.0
Метрики расстояний на подграфе методом снежный ком:
Диаметр графа: 4
Радиус графа: 3
90 процентиль расстояния в графе: 3.0
----1.3----
Средний кластерный коэффициент: 0.28162939243642865
Коэффициент ассортативности: 0.07848305830139331
```

Датасет socfb-Reed98

```
----1.1----
Количество вершин в графе: 962
Количество ребер в графе: 18812.0
Плотность графа: 0.04069738513026754
Количество компонент слабой связности: 1
Доля вершин в максимальной по мощности компоненте слабой связности: 1.0
----1.2----
Метрики расстояний на случайном подграфе:
Диаметр графа: 6
Радиус графа: 3
90 процентиль расстояния в графе: 3.0
Метрики расстояний на подграфе методом снежный ком:
Диаметр графа: 4
Радиус графа: 2
90 процентиль расстояния в графе: 3.0
----1.3----
Средний кластерный коэффициент: 0.3183602272722795
Коэффициент ассортативности: 0.02343391176630079
PS C:\Users\Movchan MV\Desktop\graph tesssttt>
```

Датасет testgraph_1

```
----1.1----
Количество вершин в графе: 9
Количество ребер в графе: 13.0
Плотность графа: 0.3611111111111111
Количество компонент слабой связности: 1
Доля вершин в максимальной по мощности компоненте слабой связности: 1.0
----1.2----
Метрики расстояний на компоненте:
Диаметр графа: 3
Радиус графа: 2
90 процентиль расстояния в графе: 3.0
----1.3----
Средний кластерный коэффициент: 0.39999999999999997
Коэффициент ассортативности: -0.20370370370370416
PS C:\Users\Mavchan.MV\Desktop\graph_testtttt>
```

Датасет testgraph_2

```
----1.1----
Количество вершин в графе: 34
Количество ребер в графе: 78.0
Плотность графа: 0.13903743315508021
Количество компонент слабой связности: 1
Доля вершин в максимальной по мощности компоненте слабой связности: 1.0
----1.2----
Метрики расстояний на компоненте:
Диаметр графа: 5
Радиус графа: 3
90 процентиль расстояния в графе: 4.0
----1.3----
Средний кластерный коэффициент: 0.5706384782076824
Коэффициент ассортативности: -0.4756130976846141
```

Датасет BA_bitA

```
----1.1----
Количество вершин в графе: 3783
Количество ребер в графе: 14124.0
Плотность графа: 0.001974375888794159
Количество компонент слабой связности: 5
Доля вершин в максимальной по мощности компоненте слабой связности: 0.9978852762357917
----1.2----
Метрики расстояний на случайном подграфе:
Диаметр графа: 13
Радиус графа: 2
90 процентиль расстояния в графе: 7.0
Метрики расстояний на подграфе методом снежный ком:
Диаметр графа: 6
Радиус графа: 2
90 процентиль расстояния в графе: 3.0
----1.3----
Средний кластерный коэффициент: 0.1766290303590772
Коэффициент ассортативности: -0.16851576112150404
```

Датасет Ra_Rado

```
----1.1----
Количество вершин в графе: 148
Количество ребер в графе: 1086.0
Плотность графа: 0.09983452840595698
Количество компонент слабой связности: 1
Доля вершин в максимальной по мощности компоненте слабой связности: 1.0
----1.2----
Метрики расстояний на компоненте:
Диаметр графа: 5
Радиус графа: 3
90 процентиль расстояния в графе: 3.0
----1.3----
Средний кластерный коэффициент: 0.5707728115642402
Коэффициент ассортативности: -0.2584349998452466
```

Датасет BO_bitOt

```
----1.1----
Количество вершин в графе: 5881
Количество ребер в графе: 21492.0
Плотность графа: 0.001243020588612932
Количество компонент слабой связности: 4
Доля вершин в максимальной по мощности компоненте слабой связности: 0.9989797653460296
----1.2----
Метрики расстояний на случайном подграфе:
Диаметр графа: 13
Радиус графа: 2
90 процентиль расстояния в графе: 7.0
Метрики расстояний на подграфе методом снежный ком:
Диаметр графа: 4
Радиус графа: 2
90 процентиль расстояния в графе: 3.0
----1.3----
Средний кластерный коэффициент: 0.17750449405289517
Коэффициент ассортативности: -0.16483359451314525
```

Датасет UC_UC

```
----1.1----
Количество вершин в графе: 1899
Количество ребер в графе: 13838.0
Плотность графа: 0.007678601848568738
Количество компонент слабой связности: 4
Доля вершин в максимальной по мощности компоненте слабой связности: 0.9968404423380727
----1.2----
Метрики расстояний на случайном подграфе:
Диаметр графа: 8
Радиус графа: 2
90 процентиль расстояния в графе: 4.0
Метрики расстояний на подграфе методом снежный ком:
Диаметр графа: 5
Радиус графа: 3
90 процентиль расстояния в графе: 3.0
----1.3----
Средний кластерный коэффициент: 0.10939892385364355
Коэффициент ассортативности: -0.1877757871466803
```

Предсказания появления ребер в графе

Основная задача проекта – предсказать, появится ли ребро между парой вершин (u, v) к моменту времени t'', если на момент времени t^* ребро между этими вершинами отсутствовало, т. е. фактически, решается задача бинарной классификации.

Для предсказания необходимо сначала построить признаковое описание для каждого потенциального ребра (вектор/набор признаков) $X_{(u,v)}$, а также ответ $y_{(u,v)}$, который принимает значение $y_{(u,v)} = 1$, если ребро появляется в графе, и $y_{(u,v)} = 0$, иначе.

Построение векторов признаков для предсказания появления ребер в графе

Вычисляются статические признаки (Static topological features) для каждой пары вершин

1. Common Neighbours (CN);
2. Adamic-Adar (AA);
3. Jaccard Coefficient (JC);
4. Preferential Attachment (PA)

Вычисленные признаки хранятся в папке done

Common Neighbours (CN) The CN feature is equal to the number of common neighbours of two nodes.

$$CN_{\text{static}}(u, v) = |\Gamma(u) \cap \Gamma(v)| \quad (1)$$

Adamic-Adar (AA) The AA feature considers all common neighbours, favouring nodes with low degrees (Adamic and Adar 2003).

$$AA_{\text{static}}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log |\Gamma(z)|} \quad (2)$$

Jaccard Coefficient (JC) The JC feature is similar to the CN feature, but normalises for the number of unique neighbours of the two nodes.

$$JC_{\text{static}}(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \quad (3)$$

Preferential Attachment (PA) The PA feature takes into account the observation that nodes with a high degree are more likely to make new links than nodes with a lower degree.

$$PA_{\text{static}}(u, v) = |\Gamma(u)| \cdot |\Gamma(v)| \quad (4)$$

Построение векторов признаков для предсказания появления ребер в графе

Вычисляются взвешенные темпоральные признаки
(Weighted temporal topological features)

1. Common Neighbours temporal
2. Adamic-Adar temporal
3. Jaccard Coefficient temporal
4. Preferential Attachment temporal

Для весовых функций взят коэффициент l=0.2

Признаки без учета событий прошлого

$$AA_{\text{temporal}}(u,v) = \sum_{z \in \Gamma(v) \cap \Gamma(y)} \frac{wtf(u,z) + wtf(v,z)}{\log\left(1 + \sum_{x \in \Gamma(z)} wtf(z,x)\right)} \quad (8)$$

$$CN_{\text{temporal}}(u,v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} wtf(u,z) + wtf(v,z) \quad (9)$$

$$JC_{\text{temporal}}(u,v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{wtf(u,z) + wtf(v,z)}{\sum_{x \in \Gamma(u)} wtf(u,x) + \sum_{y \in \Gamma(v)} wtf(v,y)} \quad (10)$$

$$PA_{\text{temporal}}(u,v) = \sum_{a \in \Gamma(x)} wtf(u,x) \cdot \sum_{b \in \Gamma(y)} wtf(v,y) \quad (11)$$

$$w_{\text{linear}} = l + (1-l) \cdot \frac{t - t_{\min}}{t_{\max} - t_{\min}}$$

$$w_{\text{exponential}} = l + (1-l) \cdot \frac{\exp\left(3 \frac{t - t_{\min}}{t_{\max} - t_{\min}}\right) - 1}{e^3 - 1}$$

$$w_{\text{square root}} = l + (1-l) \cdot \sqrt{\frac{t - t_{\min}}{t_{\max} - t_{\min}}}$$

Задача бинарной классификации (появится ребро в графе или нет)

- Для обучения модели применяется алгоритм логистической регрессии — статистическая модель для прогнозирования вероятности возникновения некоторого события путём подгонки данных к логистической кривой.
- Классификатор выдаёт для каждого объекта вероятность того, что объект принадлежит к определенному классу. Далее, по принятому порогу объекты делятся на классы
- Для оценки качества построенной модели используется метрика AUC (Area Under the Receiver Operating Curve) – площадь под ROC AUC – кривой.
- ROC AUC - кривая строится на основании соотношения доли верно классифицированных объектов, обладающих некоторым свойством (TPR true positive rate) и доли объектов, не обладающих свойством, но ошибочно классифицированных как обладающие этим свойством (FPR false positive rate), при различных уровнях порога принятия решения.

Используем 75% данных для обучения и 25% для тестирования

TPR Доля рёбер, которые появились в графе $y_{(u,v)} = 1$ и которые классификатор отметил, как появившиеся в графе.

FPR Доля рёбер, которые не появились в графе, но которые были отмечены, как появившиеся.

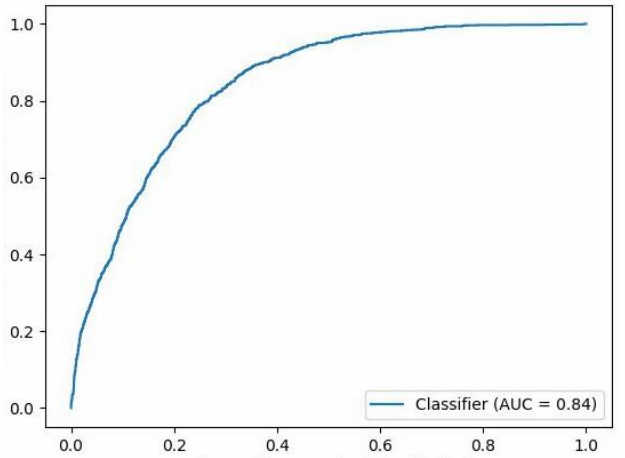
Стратегия

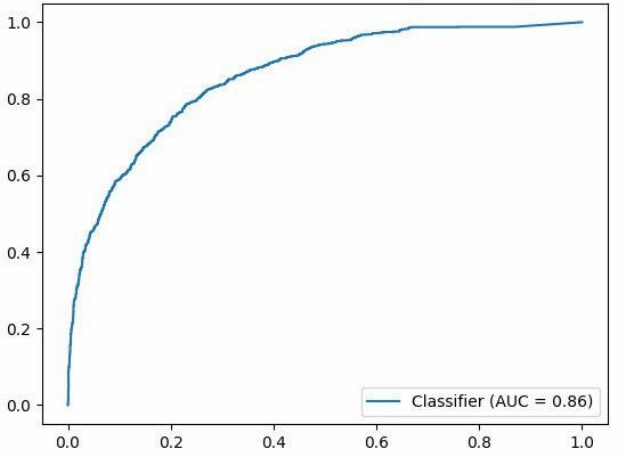
Будем беспорядочно и без повторений перебирать все пары вершин u,v так, чтобы выделить равные доли пар между которыми есть ребро в момент времени t (pos) и пар между которыми ребра нет (neg). Далее для каждой доли находим или статические признаки, или темпоральные (без учёта событий прошлого). Перемешиваем pos и neg доли, далее 75% данных используем для обучения и 25% для тестов.

Для больших сетей берем размер долей для статических признаков = 100 000. Для темпоральных признаков = 10 000. Для маленьких сетей перебираем все возможные пары без ограничений на размер доли.

После сравним наши значения с табличными из статьи.
В статье используется похожая стратегия, но пары вершин перебираются не беспорядочно, а на расстоянии 2 друг от друга

RA_Rado

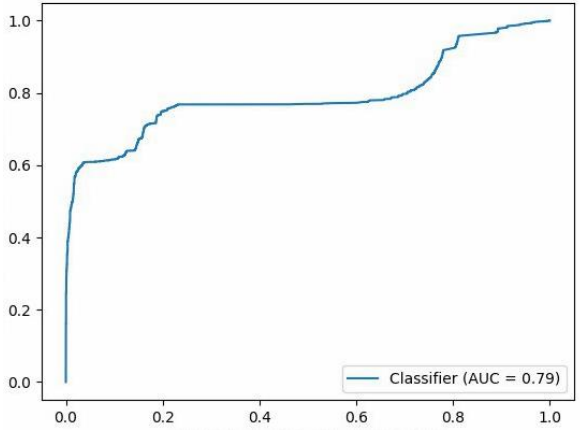


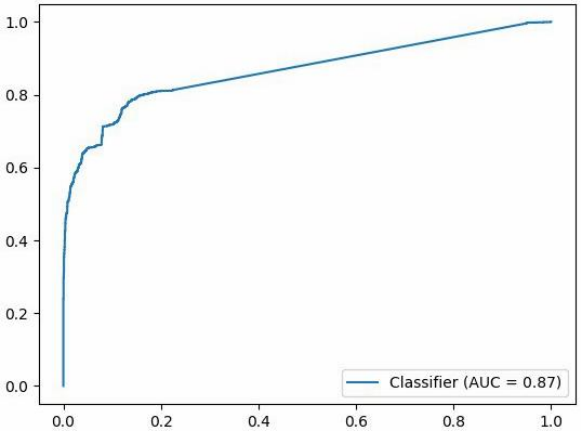


Static features	Test	Table
AUC	0.843	0.864

Temporal features	Test	Table
AUC	0.859	0.852

BA_bitA

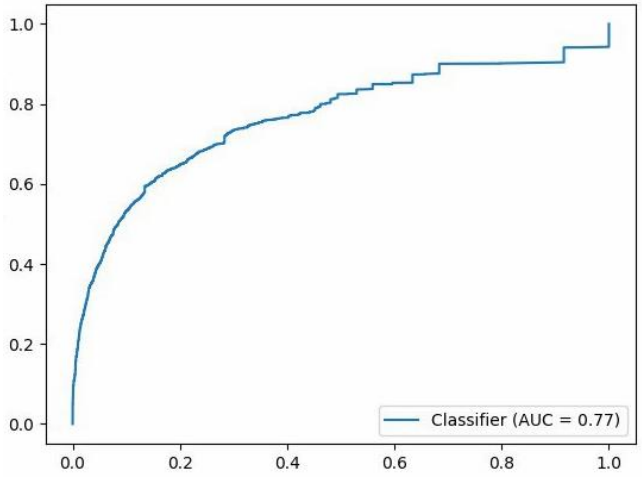




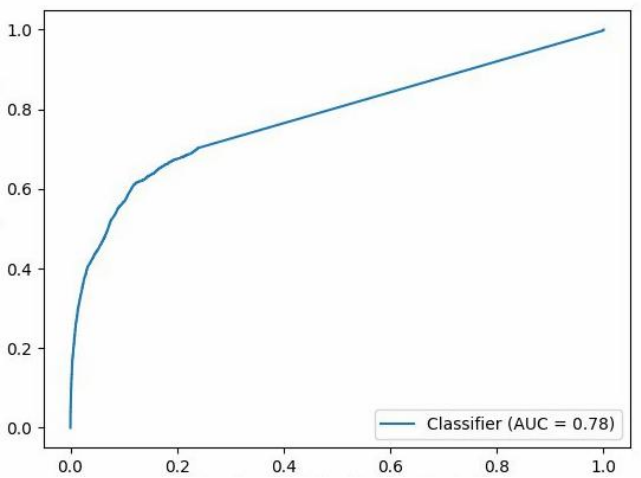
Static features	Test	Table
AUC	0.791	0.868

Temporal features	Test	Table
AUC	0.866	0.945

BO_bitOt

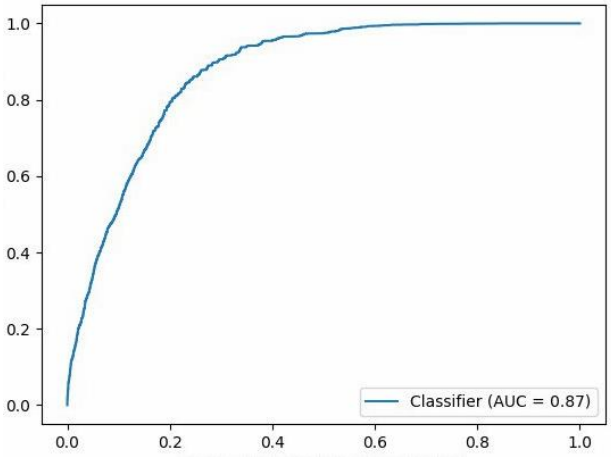


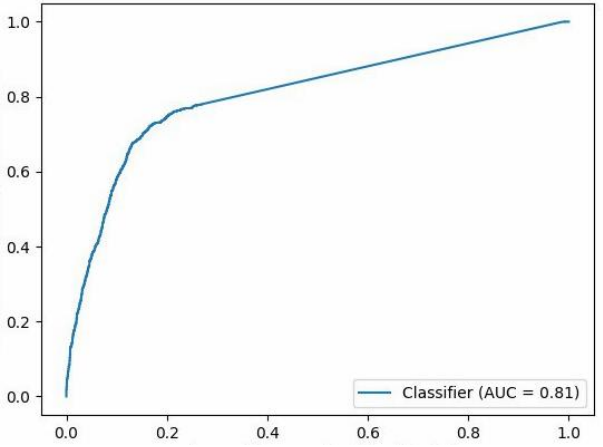
Static features	Test	Table
AUC	0.766	0.821



Temporal features	Test	Table
AUC	0.779	0.947

UC_UC

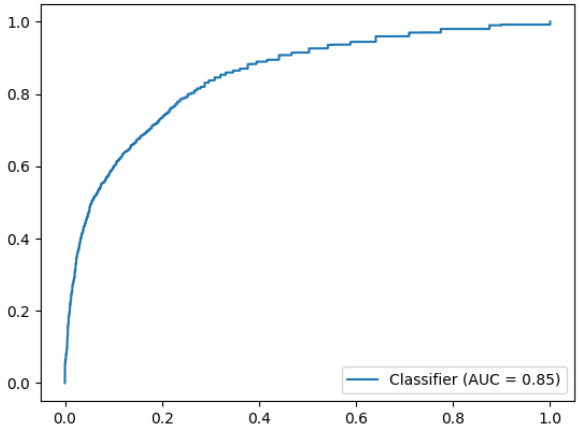


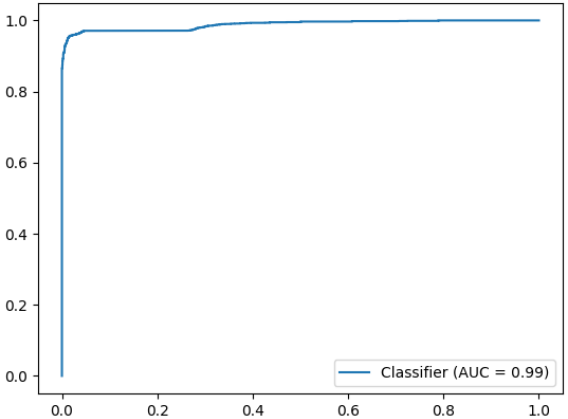


Static features	Test	Table
AUC	0.873	0.731

Temporal features	Test	Table
AUC	0.808	0.744

ma_SX-MO

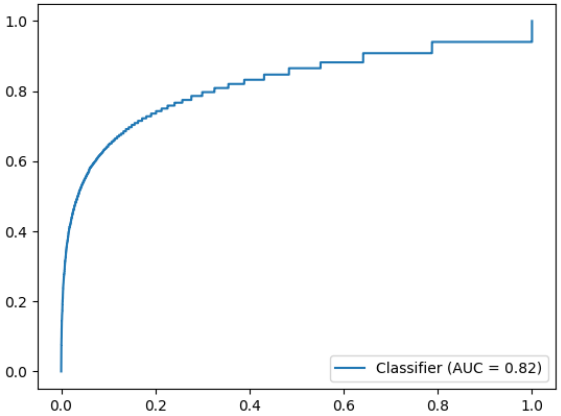


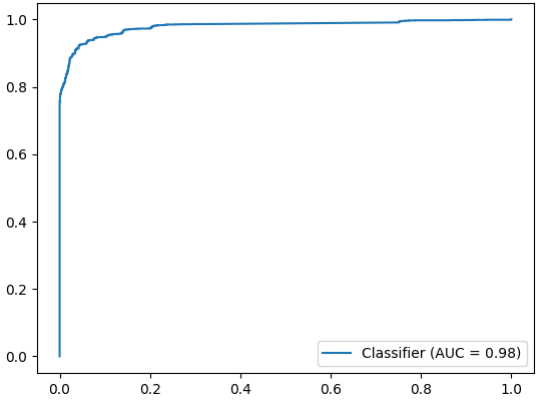


Static features	Test	Table
AUC	0.849	0.859

Temporal features	Test	Table
AUC	0.988	0.909

SU_SX-SU





Static features	Test	Table
AUC	0.820	0.921

Temporal features	Test	Table
AUC	0.979	0.946

Заключение

Мы можем заметить что для малых сетей наша стратегия с беспорядочным перебором работает хуже чем из статьи, но AUC близок к табличной. (BA_bitA, BO_bitOt)

Однако если мы переберем все возможные комбинации, не ограничивая размер долей. То стратегия будем лучше, чем табличная с ограничением в 10 000 пар и расстоянием 2. (RA_Rado, UC_UC)

Если же мы возьмем очень большие сети, то стратегия с беспорядочным перебором может быть вовсе неэффективна и давать неверную оценку (слишком большая AUC). Предположительно это происходит из за того что модель неправильно/неэффективно обучается на беспорядочном переборе(ma_SX-MO, SU_SX-SU)