

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Системное программирование

Михалев Андрей Дмитриевич

Эффект биометрического зверинца в алгоритмах локального сопоставления

Магистерская диссертация

Научный руководитель:
Сартасов С. Ю.

Рецензент:
к. ф.-м. н. Мельников А. А.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Software Engineering

Andrey Mikhalev

Biometric menagerie effect in local matching algorithms

Master's Thesis

Scientific supervisor:
Stanislav Sartasov

Reviewer:
Alexander Melnikov

Saint-Petersburg
2017

Оглавление

Введение	6
1. Описание предметной области	8
1.1. Регистрация в биометрической системе	8
1.2. Сопоставление шаблонов	8
1.3. График FAR/FRR	9
Постановка задачи	12
2. Обзор литературы	13
2.1. Обзор решений задачи сопоставления отпечатков	13
2.2. Эффект биометрического зверинца	14
2.3. Обзор дактилоскопических алгоритмов локального сопоставления	15
2.3.1. Алгоритм Minutia Cylinder-Code	15
2.3.2. Алгоритм Feng	16
2.4. Обзор алгоритмов консолидации	17
3. Выявление эффекта биометрического зверинца на глобальном уровне	20
3.1. Построение множеств статистических оценок для множеств глобальных <i>genuine</i> и <i>impostor</i> оценок	20
3.2. Описание классов биометрического зверинца	21
3.3. Эксперимент Ягера-Данстона для глобальных оценок сопоставления	23
3.4. Ограничения подхода Ягера-Данстона	24
4. Выявление эффекта биометрического зверинца на локальном уровне	26
4.1. Описание алгоритма выделения <i>genuine</i> и <i>impostor</i> оценок из LSM	27
4.2. Особенности реализации	28

4.2.1. Выбор лучших оценок локального сопоставления из <i>LSM</i>	28
4.2.2. Алгоритм параллельных линий	29
4.2.3. Выделение <i>genuine</i> оценок из <i>LSM</i>	30
4.3. Крайний случай работы алгоритма выделения <i>genuine</i> оценок	30
5. Результаты экспериментов	31
5.1. Используемые технологии	31
5.2. Проведение эксперимента	31
5.3. Эксперимент Ягера-Данстона на глобальном уровне . . .	32
5.4. Эксперимент Ягера-Данстона на локальном уровне . . .	33
5.5. Преимущества локального уровня над глобальным . . .	36
5.5.1. Описание алгоритма фильтрации <i>плохих</i> локальных структур	36
5.5.2. Результаты фильтрации	38
5.6. Выводы	41
6. Сравнение оригинальных и отфильтрованных данных для <i>MCC</i>	42
6.1. Описание оригинальных данных	42
6.2. Описание отфильтрованных данных	42
6.3. Описание отфильтрованных данных, учитывающих локальные структуры с $G_{set} = \emptyset$	43
6.4. Описание отфильтрованных данных, учитывающих только локальные структуры с $G_{set} \neq \emptyset$	43
6.5. Результаты сравнения baseline и фильтрованных данных	44
6.5.1. Baseline и данные с $G_{set} = \emptyset$	44
6.5.2. Baseline и данные с $G_{set} \neq \emptyset$	50
6.5.3. После фильтрации <i>Worms</i>	51
6.5.4. После фильтрации <i>Chameleons</i>	52
7. Выводы	55

8. Заключение	57
Список литературы	59

Введение

В современном мире информационные технологии являются неотъемлемой частью повседневной жизни. Во многих ее сферах требуется безопасный доступ к информационным системам и частной информации, который могут обеспечить биометрические системы аутентификации, имеющие многочисленные и разнообразные сферы применения: доступ к рабочим местам, режимным учреждениям, хранилищам ценных ресурсов, защита информации, обеспечение безопасности банковских операций и многое другое. Все они позволяют проверить личность пользователя, ограничить доступ к данным для посторонних лиц. Более того, они не подлежат кражам, потерям или копированию, как пароли или электронные носители информации.

Существуют различные биометрические параметры [20]:

- физиологические - радужная оболочка глаза, сетчатка глаза, отпечатки пальцев, геометрия руки, геометрия лица, термограмма лица;
- поведенческие - голос, почерк;
- комбинированные.

В настоящее время широко распространены биометрические системы аутентификации, использующие один из видов биометрического доступа - отпечаток пальца - благодаря его уникальности, неизменности и высокой точности. Методы распознавания отпечатков пальцев ежедневно применяются в криминалистике, а системы контроля доступа, использующие отпечатки пальцев, являются одними из самых востребованных биометрических систем. В течение последних десятилетий они всё чаще используются для верификации и идентификации пользователей с целью ограничения доступа к различного вида ресурсам и обеспечения безопасности хранения и передачи важной информации.

Такое повсеместное использование систем распознавания отпечатков обеспечило актуальность проблемы их распознавания и в настоящее

время продолжает поддерживать её. Различные подходы к решению данной задачи изучаются уже давно. На данный момент существует обширное количество достаточно эффективных решений, однако признать решенным вопрос распознавания отпечатков не представляется возможным, поскольку разработка более точных алгоритмов, минимизирующих накладные расходы, до сих пор не завершена.

Проблему распознавания отпечатков можно свести к задаче сопоставления двух экземпляров с помощью определённого алгоритма. Алгоритмы сопоставления принимают решение о принадлежности отпечатков одному источнику на основе некоторого порогового значения, что неизбежно приводит к ошибкам.

Одной из главных причин ошибок в биометрической системе может служить эффект биометрического зверинца. Он изучается уже на протяжении 20 лет и на текущий момент его исследование до сих пор остаётся перспективным в области биометрии.

1. Описание предметной области

1.1. Регистрация в биометрической системе

Перед биометрической системой может быть поставлена одна из задач - идентификация или верификация. Обе они могут быть сведены к одной - задаче сопоставления отпечатков. Для её решения необходимо в первую очередь зарегистрировать пользователей системы (рис. 1), создав их *шаблоны*, и сохранить их базе данных [17].

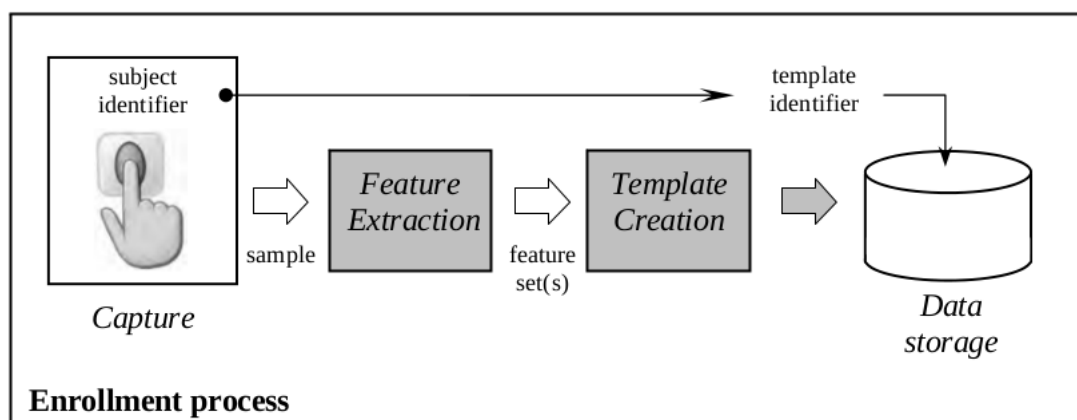


Рис. 1: Процесс регистрации пользователей в биометрической системе (из [17]).

Шаблон формируется для каждого отпечатка. Для каждого пользователя создается несколько шаблонов для одного отпечатка, но сохраняется только один (например, лучший).

1.2. Сопоставление шаблонов

При запросе доступа к системе будет происходить сопоставление между шаблоном, предоставленным некоторым лицом, и одним или множеством шаблонов из базы данных биометрической системы. При сопоставлении шаблонов алгоритмы вычисляют метрику, значение которой сравнивается с пороговым значением биометрической системы. Если посчитанное значение больше порогового значения, то пользователь получает доступ к системе, иначе – не получает.

Существует два типа оценок сопоставления [17]:

- *Impostor score* – оценка сопоставления двух шаблонов отпечатков разных пользователей;
- *Genuine score* – оценка сопоставления двух шаблонов отпечатков одного и того же пользователя.

При сопоставлении шаблонов в дактилоскопических алгоритмах неизбежно возникает два типа ошибок [17]:

- *False Match* – ошибка I рода, в таком случае пара сопоставляемых шаблонов от разных пальцев считается истинно сопоставленной;
- *False Non-Match* – ошибка II рода, в таком случае пара сопоставляемых шаблонов одного считается ложно сопоставленной;

Для *False Match* и *False Non-Match* ошибок определены следующие вероятности [17]:

- *False Match (Acceptance) Rate (FAR / FMR)* – вероятность ошибки I рода, т.е. ложного совпадения отпечатков.
- *False Non-Match (Rejection) Rate (FRR / FNMR)* – вероятность ошибки II рода, т.е. ложного несовпадения отпечатков.

Термины *genuine score*, *impostor score*, *FAR* и *FRR* применимы не только к глобальным оценкам схожести шаблонов, но и к оценкам схожести произвольных локальных структур, построенных в рамках этих шаблонов.

1.3. График *FAR/FRR*

Решение о совпадении отпечатков принимается на основе некоторого порогового значения t [17], которое определяет чувствительность системы. Его изменение влияет на связанные между собой *FAR* и *FRR*. Например, при уменьшении порога понижается *FRR* и повышается *FAR*.

График, на котором FAR/FRR представляет собой ROC -кривую (*Reciever Operating Characteristic*) зависимость количества ошибок FAR от FRR или наоборот. Для его построения выбирается порог t , относительно которого считаются данные ошибки. Алгоритм 1 демонстрирует пример вычисления FAR и FRR в зависимости от порогового значения $threshold$ для массивов $genuine$ и $impostor$ оценок.

Algorithm 1 Подсчёт количества ошибок в зависимости от порогового значения

Input: $genuine_scores, impostor_scores, threshold\ t$

Output: $FAR(t), FRR(t)$

```

1: function CALCERRORS( $genuine\_scores, impostor\_scores, t$ )
2:    $impostor\_count = 0$ 
3:    $genuine\_count = 0$ 
4:   for all  $score \in impostor\_scores$  do
5:     if  $score > t$  then
6:        $impostor\_count++ = 1$ 
7:   for all  $score \in genuine\_scores$  do
8:     if  $score < t$  then
9:        $genuine\_count++ = 1$ 
10:   $FAR = impostor\_count/size(impostor\_scores)$ 
11:   $FRR = genuine\_count/size(genuine\_scores)$ 
12: return  $FAR(t), FRR(t)$ 

```

Точка на графике 2, в которой $FAR(t) = FRR(t)$, является EER -точкой (*Equal-Error Rate*). Она отвечает за показатель, называемый производительность системы. Чем ближе к началу координат её положение, тем меньше количество ошибок системе, и тем качественнее распознавание и с точки зрения безопасности, и с точки зрения удобства.

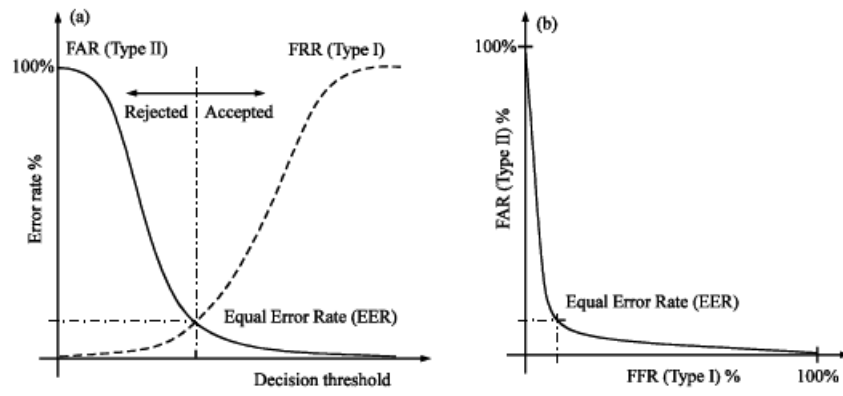


Рис. 2: Кривые FAR и FRR в зависимости от порогового значения и соответствующая *ROC*-кривая (из [24]).

Постановка задачи

Основной целью данной работы является изучение влияния *эффекта биометрического зверинца* в дактилоскопических алгоритмах сопоставления и методы борьбы с ним. Для достижения поставленной цели в рамках текущего исследования были поставлены следующие задачи:

1. Сделать обзор дактилоскопических алгоритмов сопоставления.
2. Сделать обзор эффекта биометрического зверинца.
3. Повторить эксперимент Ягера-Данстона для выявления эффекта биометрического зверинца.
4. Выявить ограничения подхода Ягера-Данстона.
5. Поставить эксперимент Ягера-Данстона на локальном уровне дактилоскопических алгоритмов сопоставления.
6. Экспериментально подтвердить, что алгоритмы консолидации добавляют в систему эффект биометрического зверинца.
7. Предложить метод регистрации пользователей, учитывающий эффект биометрического зверинца.

Решения перечисленных задач позволят ответить на следующие вопросы:

1. Наблюдается ли *эффект биометрического зверинца* на локальном уровне?
2. Лишены ли *эффекта биометрического зверинца* глобальные оценки, полученные путём объединения оценок сопоставления локальных структур шаблонов?
3. Можно ли использовать *эффект биометрического зверинца* в качестве критерия качества алгоритма сопоставления?

2. Обзор литературы

2.1. Обзор решений задачи сопоставления отпечатков

На данный момент доступно огромное количество разнообразных решений задачи сопоставления [34]. Большинство из них основаны на *минуциях* - уникальных для каждого отпечатка пальца точках, в которых изменяется структура папиллярных линий [17]. Как показано в [7], задача сопоставления отпечатков имеет два подхода к решению:

- *Однофазные алгоритмы сопоставления (алгоритмы глобального сопоставления)* - сопоставляют двухмерный точечный шаблон, осуществляя глобальное выравнивание, которое приводит к оптимальному пространственному и направленному составлению пар минуций. В результате алгоритм даёт *глобальную оценку* сопоставления двух шаблонов.
- *Двухфазные алгоритмы сопоставления (алгоритмы локального сопоставления)*:
 1. *Первая фаза(локальный уровень)* заключается в сопоставлении локальных структур, определённых для каждого шаблона. Итогом первой фазы является *матрица локальных схожестей*, каждый элемент которой - *локальная оценка* сопоставления двух локальных структур из разных шаблонов.
 2. *Вторая фаза(глобальный уровень)* заключается в объединении *локальных оценок* сопоставления с целью получения *глобальной оценки* сопоставления двух шаблонов.

Однофазные алгоритмы сопоставления, использующие глобальный подход, вычислительно сложны и недостаточно устойчивы к нелинейному растяжению отпечатков [7]. Поэтому в последние пару десятилетий основное внимание уделяется изучению алгоритмов локального

сопоставления [34]. Согласно [7], они инвариантны по отношению к глобальным преобразованиям (переносу, повороту и т.д.) и поэтому могут выполнять операцию сопоставления без предварительного глобального выравнивания.

2.2. Эффект биометрического зверинца

В 1998 г. Дж. Доддингтон опубликовал исследование [30] по распознаванию говорящих в системах распознавания речи. Результаты показали, что пользователи биометрической системы склонны иметь определенные тенденции при их распознавании. Он сформулировал нестрогую классификацию пользователей, которая основана на сопоставлении шаблонов, и ввёл термин *biometric menagerie / biometric zoo / биометрический зверинец*.

Позднее, в 2007 и 2010 гг., Н. Ягер и Т. Данстон в [32, 33] расширили исходную классификацию Дж. Доддингтона, которая состояла из 4 классов (*Sheeps, Goats, Lambs, Wolves*) и базировалась исключительно на *genuine* и *impostor* оценках. В своих работах Ягер и Данстон рассмотрели **соотношения** между *genuine* и *impostor* оценками, и ввели четыре дополнительных класса (*Chameleons, Phantoms, Doves, Worms*). Кроме того, они провели эксперимент по выявлению эффекта биометрического зверинца - принадлежность пользователей системы к классам биометрического зверинца [33].

Классификация Доддингтона и её расширение Ягером и Данстоном присутствует в биометрических системах распознавания: по отпечаткам, речи, по лицам, по радужной оболочке глаза, по рисунку вен, по онлайн подписи [32, 26, 21, 9, 3, 18]. В работах [26, 27, 29, 16, 35, 1] рассматриваются способы группировки пользователей системы по их оценкам сопоставления и принадлежность этих групп к классам биометрического зверинца. В [5] авторы предложили обобщение и доказательство существования классов биометрического зверинца, в [19] с использованием вероятностного подхода с уровнем значимости 0.05 доказано существование классов Ягера и Данстона и показано отсутствие стати-

стически значимых доказательств существования классов Доддингтона. В [28] рассматривается концепция нечёткой логики в применении к классам биометрического зверинца, в [2] проводится выявление классов Доддингтона, Ягера и Данстона.

В 2013 году было показано [25], что эффект биометрического зверинца напрямую связан с тем, что малое количество пользователей могут являться причиной большого количества ошибок при распознавании в биометрической системе. В частности, такое меньшинство приводит к непропорциональному распределению FAR и FRR ошибок среди всех пользователей системы.

Однако во всех работах, посвященных исследованию данного эффекта, внимание уделяется исключительно оценкам сопоставления шаблонов. Все гипотезы и выводы применяются только к оценкам сопоставления шаблонов. т.е. на *глобальном уровне*. В данной работе предлагается новый подход - исследование эффекта биометрического зверинца на *локальном уровне*.

2.3. Обзор дактилоскопических алгоритмов локального сопоставления

Алгоритмы локального сопоставления успешно справляются с решением задачи сопоставления отпечатков. На данный момент огромное количество разнообразных подходов [34]. Опираясь на упомянутый обзор [34], основное внимание в данной работе уделяется наиболее точным: *Bozorth3* [31], *MCC* [7], *Deng* [10] и *Feng* [14]. Однако два алгоритма из четырёх не рассматриваются, т.к. *Bozorth3* не имеет открытой реализации, *Deng* требует дополнительные входные данные - тип минуций.

2.3.1. Алгоритм Minutia Cylinder-Code

Minutia Cylinder-Code (MCC) – алгоритм локального сопоставления, основанный на минуциях [7]. Он имеет ряд преимуществ над остальными алгоритмами локального сопоставления [34, 7]. Его особенность заключается в том, что он связывает локальную структуру с каждой

минуцией шаблона, которая кодирует пространственные и «направленные» отношения между минуциями и её *fixed-based* соседями.

Локальная структура представляется в виде *цилиндра*, основание и высота которого определяют пространственную и «направленную» информацию [7]. Каждый цилиндр линеаризуется в одномерный вектор и цилиндры сопоставляются путем нахождения корреляции между каждой парой векторов. Значения корреляции, отражающие локальные оценки сопоставления двух минуций, заносятся в результирующую матрицу локальных схожестей $\Gamma \in [0, 1]$.

К матрице Γ применяются алгоритмы консолидации [7]: *LSS*, *LSA*, *LSS-R*, *LSA-R*, которые позволяют получить глобальную оценку сопоставления двух шаблонов из $[0, 1]$.

2.3.2. Алгоритм Feng

Feng – алгоритм локального сопоставления, основанный на минуциях [14]. Он представляет собою особый подход к сопоставлению на локальном уровне благодаря тому, что использует сразу два типа локальных структур:

- *Texture-based* дескрипторы - содержат в себе «направленную» и частотную характеристики области вокруг каждой минуции;
- *Minutiae-based* дескрипторы - содержат ближайших соседей минуции в области фиксированного радиуса.

Комбинация оценок сопоставления дескрипторов каждого типа в отдельности даёт оценку локального сопоставления для двух произвольных минуций. заносятся в результирующую матрицу локальных схожестей $\Gamma \in [0, 1]$.

В данной работе была реализована упрощённая, но не менее точная версия алгоритма *Feng*, в которой использовались только *Minutiae-based* дескрипторы [14]. Результаты сопоставления дескрипторов были занесены в результирующую матрицу локальных схожестей $\Gamma \in [0, 1]$.

К матрице Γ применяется жадный алгоритмы консолидации [14], который позволяет получить глобальную оценку сопоставления двух шаблонов из $[0, 1]$.

2.4. Обзор алгоритмов консолидации

Алгоритмы консолидации являются завершающим шагом алгоритмов локального сопоставления. Они позволяют перейти от локального уровня сопоставления к глобальному, собирая определённым образом лучшие локальные оценки сопоставления минущий из матрицы Γ и формируя общую глобальную оценку двух шаблонов из $[0, 1]$.

Как правило, каждый алгоритм локального сопоставления предлагает свой метод консолидации [34]. Алгоритм *Feng* предлагает перед использованием жадного алгоритма консолидации проводить нормализацию оценок сопоставления локальных структур для двух шаблонов в $[0, 1]$ по следующей формуле [14]:

$$s_{norm}(i, j) = \frac{s(i, j) * (N_1 + N_2 - 1)}{\sum_{k=1}^{N_1} s(k, j) + \sum_{k=1}^{N_2} s(i, k) - s(i, j)}, \text{ где:}$$

- $s(i, j)$ - локальная оценка сопоставления i -й и j -й локальных структур первого и второго шаблонов соответственно;
- N_1, N_2 - количество локальных структур в сопоставляемых шаблонах;
- $\sum_{k=1}^{N_1} s(k, j)$ - сумма оценок сопоставления всех локальных структур первого шаблона с j -й локальной структурой второго шаблона;
- $\sum_{k=1}^{N_2} s(i, k)$ - сумма оценок сопоставления i -й локальной структуры первого шаблона со всеми локальными структурами второго шаблона.

Если рассмотреть две локальные структуры построенные алгоритмом *Feng*: i -ю и j -ю из первого и второго шаблонов соответственно, и предположить, что:

- они хорошо сопоставлены друг с другом, т.е. имеют высокую локальную оценку $s(i, j)$;
- i -я локальная структура первого шаблона плохо сопоставилась со всеми локальными структурами второго шаблона, кроме j -й:

$$s(i, k) \ll s(i, j), \forall k \neq j$$

- j -я локальная структура второго шаблона плохо сопоставилась со всеми локальными структурами первого, кроме j -й:

$$s(k, j) \ll s(i, j), \forall k \neq j$$

то при делении числителя и знаменателя на значение $s(i, j) \neq 0$, нормализованное значение $s_{norm}(i, j)$ может оказаться > 1 , что противоречит цели нормализации алгоритма *Feng*. Поэтому, жадный алгоритм консолидации, применяемый в качестве последнего шага в *Feng* не рассматривается в данной работе. Основное внимание уделяется алгоритмам консолидации, которые применялись к *MCC*.

Рассмотрим один из алгоритмов консолидации для *MCC* - *LSS* [7]. Пусть имеется два шаблона $A = \{a_1, a_2, \dots, a_{n_A}\}$ и $B = \{b_1, b_2, \dots, b_{n_B}\}$, тогда из [7]:

- $\gamma(a, b)$ - оценка локальной схожести между цилиндрами, соответствующими минущиям $a \in A$ и $b \in B$,
 $\gamma : A \times B \rightarrow [0, 1]$
- $\Gamma \in [0, 1]^{n_A \times n_B}$ - матрица локальных схожестей, где $\Gamma[r, c] = \gamma(a_r, b_c)$

Метод *LSS* сортирует все оценки локальных схожестей из матрицы Γ и выбирает n_P лучших. Пусть \mathcal{P} - множество выбранных n_P пар индексов минущий [7]:

$$\mathcal{P} = \{(r_t, c_t)\}, t = 1, \dots, n_P, 1 \leq r_t \leq n_A, 1 \leq c_t \leq n_B$$

Величина $n_{\mathcal{P}}$ определяется по следующей формуле:

$$n_{\mathcal{P}} = \min_{n_{\mathcal{P}}} + \lfloor (Z(\min\{n_A, n_B\}, \mu_{\mathcal{P}}, \tau_{\mathcal{P}})) * (\max_{n_{\mathcal{P}}} - \min_{n_{\mathcal{P}}}) \rfloor,$$

где $\mu_{\mathcal{P}}, \tau_{\mathcal{P}}, \min_{n_{\mathcal{P}}}, \max_{n_{\mathcal{P}}}$ - параметры, $\lfloor \cdot \rfloor$ - оператор округления, $Z(v, \mu_{\Psi}, \tau_{\Psi})$ - сигмоида, зависящая от двух параметров (μ_{Ψ} и τ_{Ψ}), гарантирующая, что её значение лежит в отрезке $[0, 1]$ и определяющаяся по формуле:

$$Z(v, \mu, \tau) = \frac{1}{1 + \exp^{-\tau(v-\mu)}}$$

Общая оценка схожести двух шаблонов может быть получена из следующей формулы:

$$S(A, B) = \frac{\sum_{(r,c)} \Gamma[r, c]}{n_{\mathcal{P}}}$$

Остальные три алгоритма консолидации, LSA , $LSS-R$, $LSA-R$, подробно описаны в [7].

3. Выявление эффекта биометрического зверинца на глобальном уровне

В работах [32, 33] Ягер и Данстон определили классы биометрического зверинца и провели эксперимент (далее эксперимент Ягера-Данстона) по выявлению принадлежности пользователей биометрической системы к этим классам.

Входные данные для эксперимента Ягера-Данстона - статистические оценки для множеств глобальных *genuine* и *impostor* оценок, соответствующих сопоставлениям шаблонов.

Выходные данные - факт присутствия или отсутствия классов биометрического зверинца в биометрической системе.

3.1. Построение множеств статистических оценок для множеств глобальных *genuine* и *impostor* оценок

Н. Ягер и Т. Данстон рассмотрели множество пользователей системы \mathcal{P} и множество оценок S . Для каждой пары пользователей $j, k \in \mathcal{P}$ существует множество $S(j, k) \subset S$, содержащее оценки сопоставлений для одного из образцов j -го пользователя с зарегистрированными шаблонами, принадлежащим пользователю k .

Для каждого пользователя системы определяются два множества оценок:

- $G_k = S(k, k)$ - *genuine* оценки пользователя k , соответствующие тому, насколько хорошо пользователь сопоставляется сам с собой.
- $I_k = S(j, k) \cup S(k, j)$ - *impostor* оценки пользователя k для всех $j \neq k$, соответствующие тому, насколько хорошо пользователь сопоставляется с другими пользователями.

Для каждого из множеств оценок пользователя k строится статистика [33]:

- $g_k = \overline{G_k}$ - средняя *genuine* оценка из множества G_k для k -го пользователя.
- $i_k = \overline{I_k}$ - средняя *impostor* оценка из множества I_k для k -го пользователя.

Определяется $\mathcal{G} = \bigcup_{k \in \mathcal{P}} g_k$ – множество средних *genuine* оценок для всех пользователей. Множество \mathcal{G} упорядочивается по увеличению g_k , тогда согласно [33]:

- $\mathcal{G}_H \subset \mathcal{P}$ – наибольшие 25% элементов множества \mathcal{G} , т.е. пользователи с самыми высокими *genuine* статистиками.
- $\mathcal{G}_L \subset \mathcal{P}$ – наименьшие 25% элементов множества \mathcal{G} , т.е. пользователи с самыми низкими *genuine* статистиками.
- Аналогичным образом определяются \mathcal{I}_L и \mathcal{I}_H .

3.2. Описание классов биометрического зверинца

Базовые классы биометрического зверинца, введенные Доддингтоном, определяются схожим образом в работах в терминах *genuine* и *impostor* оценок [30, 32, 33]:

- **Sheeps** – большинство пользователей системы. Они легко распознаются биометрической системой.
- **Goats** – пользователи, сопоставление с которыми затруднительно. Они характеризуются низкими *genuine* и *impostor* оценками сопоставления и приносят в систему непропорциональное увеличение *FRR*.
- **Lambs** – пользователи, которых легко имитировать. Они характеризуются высокими *impostor* оценками при сопоставлении других пользователей с ними и приносят в систему непропорциональное увеличение *FAR*.

- **Wolves** – пользователи, которые легко имитируют других пользователей. Они характеризуются высокими *impostor* оценками сопоставления с другими пользователями системы и приносят в систему непропорциональное увеличение *FAR*.

Дополнительные 4 класса определяются с помощью множеств \mathcal{G}_H , \mathcal{G}_L , \mathcal{I}_H , \mathcal{I}_L в терминах *соотношений* между множествами *genuine* и *impostor* оценок.

- **Chameleons** – пользователи системы, для которых характерны высокие *genuine* и высокие *impostor* оценки: $\mathcal{G}_H \cap \mathcal{I}_H$. Они приносят в систему непропорциональное увеличение *FAR*.
- **Phantoms** – пользователи системы, для которых характерны низкие *genuine* и низкие *impostor* оценки: $\mathcal{G}_L \cap \mathcal{I}_L$. Они приносят в систему непропорциональное увеличение *FRR*.
- **Doves** – пользователи системы, для которых характерны высокие *genuine* и низкие *impostor* оценки: $\mathcal{G}_H \cap \mathcal{I}_L$. Сопоставление их отпечатков очень редко приводит к ошибкам сопоставления.
- **Worms** – пользователи системы, для которых характерны низкие *genuine* и высокие *impostor* оценки: $\mathcal{G}_L \cap \mathcal{I}_H$. Они приносят в систему непропорциональное количество *FAR* и *FRR* ошибок.

Все 8 классов представляются на плоскости на рис. 3.

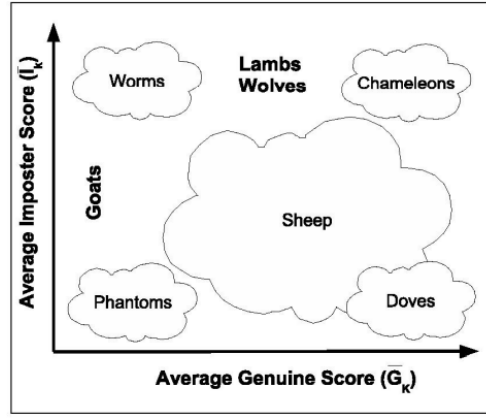


Рис. 3: Расположение классов относительно *genuine* и *impostor scores* (из [32]).

3.3. Эксперимент Ягера-Данстона для глобальных оценок сопоставления

Эксперимент Ягера-Данстона позволяет определить наличие или отсутствие классов биометрического зверинца во множестве пользователей биометрической системы с некоторым уровнем значимости. Ожидаемое количество пользователей каждого из классов определяется по следующей формуле: $p \times |P|$, $p = (1/4)^2$, при условии, что оценки каждого конкретного пользователя независимы [33].

Рассмотрим в качестве примера класс *chameleons* C (для остальных классов аналогично), представляющий пользователей (назовём их хамелеонами), которые имеют высокие *genuine* и высокие *impostor* оценки ($|\mathcal{G}_H \cap \mathcal{I}_H| \gg 1/16 \times |P|$). Пусть ν - ожидаемое количество хамелеонов в системе, $\nu = |C|$. Сформулируем гипотезу H_0 , которая утверждает, что если *genuine* и *impostor* оценки конкретного пользователя независимы, то вероятность того, что пользователь является хамелеоном, есть $p = 1/16$. Для проверки гипотезы считается вероятность того, что количество хамелеонов больше или равно предполагаемого значения ν [33]:

$$f(\nu, n, p) = \sum_{i=\nu}^n \binom{n}{i} p^i (1-p)^{n-i},$$

где $n = |\mathcal{P}|$ - количество испытаний в эксперименте. Гипотеза H_0 отклоняется, если $f(\nu, n, p) < \alpha$, где $\alpha = 0.05$ - уровень значимости.

Нулевая гипотеза имеет двухстороннюю альтернативу ($p \neq 1/16$), поэтому вероятность того, что количество хамелеонов меньше или равно ожидаемому ν , считается аналогичным образом и условия отклонения гипотезы H_0 сохраняются. Таким образом, возможны две ситуации: гипотеза H_0 отклоняется, если существует значительно малое или значительно большое количество хамелеонов. Следовательно, можно с уровнем значимости $\alpha = 0.05$ утверждать два факта [33]:

- если найдено *значительно малое количество* хамелеонов, то можно говорить о том, что класс хамелеонов есть, но количество его представителей настолько мало, что данный *класс не влияет на систему в целом*,
- если найдено *значительно большое количество* хамелеонов, то можно говорить о том, что класс хамелеонов есть, и количество его представителей настолько велико, что он *оказывает влияние на систему*.

Такой метод проверки гипотезы имеет два преимущества: он непараметрический и имеет небольшие накладные расходы [33].

3.4. Ограничения подхода Ягера-Данстона

Согласно описанию эффекта биометрического зверинца $S(j, k)$ - *множество глобальных оценок* сопоставления, поскольку $S(j, k)$ содержит оценки сопоставлений для одного из шаблонов j -го пользователя с шаблонами, принадлежащими пользователю k .

При проведении эксперимента Ягера-Данстона рассматривались множества $\mathcal{G}_H, \mathcal{G}_L, \mathcal{I}_H, \mathcal{I}_L$, принадлежность к которым помогает выявить эффект биометрического зверинца. Они последовательно строились из множеств $S(j, k)$:

1. Определялись множества *genuine* и *impostor* оценок: $G_k = S(k, k)$ и $I_k = S(j, k) \cup S(k, j), \forall j \neq k$.

2. Строились статистики $g_k = \overline{G_k}$, $i_k = \overline{I_k}$.
3. Определялись множества $\mathcal{G} = \bigcup_{k \in \mathcal{P}} g_k$ и $\mathcal{I} = \bigcup_{k \in \mathcal{P}} i_k$
4. Определялись $\mathcal{G}_H \subset \mathcal{P}$, $\mathcal{G}_L \subset \mathcal{P}$, $\mathcal{I}_H \subset \mathcal{P}$, $\mathcal{I}_L \subset \mathcal{P}$

Таким образом, в подходе Ягера-Данстона с использованием множеств \mathcal{G}_H , \mathcal{G}_L , \mathcal{I}_H , \mathcal{I}_L для *глобальных оценок сопоставления* выявлялся эффект биометрического зверинца на *глобальном уровне*.

Но в то же время мы можем построить множества \mathcal{G}_H , \mathcal{G}_L , \mathcal{I}_H , \mathcal{I}_L для *локальных оценок сопоставления*, что позволит выявить эффект биометрического зверинца не среди пользователей системы, а среди локальных структур шаблонов пользователей системы, т.е. на *локальном уровне*.

4. Выявление эффекта биометрического зверинца на локальном уровне

Эффект биометрического зверинца на локальном уровне может быть выявлен в ходе эксперимента Ягера-Данстона, если в качестве входных данных вместо оценок сопоставления шаблонов пользователей подать оценки сопоставления локальных структур шаблонов, которые могут быть получены на первом этапе в ходе работы двухфазных алгоритмов сопоставления. Такие алгоритмы принимают на вход два шаблона A и B , для каждого из них строят локальные структуры для всех минут. Затем алгоритмы выполняют сопоставление локальных структур и записывают результаты в матрицу локальных схожестей (Local Similarity Matrix, LSM) Γ .

Матрица $\Gamma_{\{j,k\}}$ - *множество локальных оценок*, содержащее результаты сопоставления локальных структур одного из шаблонов j -го пользователя с локальными структурами шаблона, принадлежащих пользователю k . Её строки соответствуют локальным структурам шаблона j , столбцы - локальным структурам шаблона k . На локальном уровне $\Gamma_{\{j,k\}}$ является аналогом *множества глобальных оценок* $S(j, k)$, содержащего оценки сопоставлений для одного из шаблонов j -го пользователя с шаблонами, принадлежащими пользователю k .

Для проведения эксперимента Ягера-Данстона необходимо получить локальные аналоги множеств $G_k = S(k, k)$ и $I_k = S(j, k) \cup S(k, j), \forall j \neq k$, которые будут содержать *genuine* и *impostor* оценки сопоставления локальных структур. В рамках данной работы был разработан алгоритм, который позволяет выделить множества G_k и I_k для локальных структур из LSM . Основываясь на локальных множествах G_k и I_k , можно расширить ограничение стандартного подхода Ягера-Данстона, построив множества $\mathcal{G}_H, \mathcal{G}_L, \mathcal{I}_H$ и \mathcal{I}_L , что позволит выявить эффект биометрического зверинца для локальных оценок сопоставления, т.е. на локальном уровне.

4.1. Описание алгоритма выделения *genuine* и *impostor* оценок из *LSM*

Входными данными для алгоритма является матрицы локальных схожестей $\Gamma_{\{j,k\}} \in [0, 1]^{n \times m}$, где n - число локальных структур в шаблоне j -го пользователя, m - число локальных структур в шаблоне k -го пользователя.

Поскольку матрицы схожести строятся для всевозможных пар шаблонов, то для двух произвольных шаблонов A и B формируются две матрицы локальных схожестей $\Gamma_{\{A,B\}} \in [0, 1]^{n_a \times n_b}$ и $\Gamma_{\{B,A\}} \in [0, 1]^{n_b \times n_a}$, и $\Gamma_{\{A,B\}} = (\Gamma_{\{B,A\}})^T$. Далее устанавливается соответствие между строками/столбцами $\Gamma_{\{A,B\}}$, $\Gamma_{\{B,A\}}$ и локальными структурами:

1. Номера строк матрицы $\Gamma_{\{A,B\}}$ соответствуют номерам локальных структур шаблона A , номера столбцов матрицы - номерам локальных структур шаблона B .
2. Номера строк матрицы $\Gamma_{\{B,A\}}$ соответствуют номерам локальных структур шаблона B , номера столбцов матрицы - номерам локальных структур шаблона A .
3. Поскольку $\Gamma_{\{A,B\}} = (\Gamma_{\{B,A\}})^T$, то достаточно рассматривать только строки матриц, соответствующие строками матриц Γ .

Для примера рассмотрим произвольную i -ю строку матрицы $\Gamma_{\{A,B\}}$, соответствующую некоторой локальной структуре шаблона A . Из её элементов можно выделить *genuine* и *impostor* оценки сопоставления со всеми локальными структурами шаблона B . Аналогично для любой i -й строки матрицы $\Gamma_{\{B,A\}}$ можно выделить *genuine* и *impostor* сопоставления со всеми локальными структурами шаблона A .

Тогда для некоторой i -й локальной структуры шаблона A множества *genuine* G_j и *impostor* I_j оценок могут быть получены так:

$$G_j = \bigcup_{k=1}^m \bigcup_{l=1}^{n_k} \Gamma_{\{A,k\}}(i, l) \mid \Gamma_{\{A,k\}}(i, l) - \textit{genuine},$$

$$I_j = \bigcup_{k=1}^m \bigcup_{l=1}^{n_k} \Gamma_{\{A,k\}}(i, l) \mid \Gamma_{\{A,k\}}(i, l) - not\ genuine, A \neq k$$

где m - количество сопоставлений шаблона A с другими шаблонами, n_k - количество локальных локальных структур в k -м шаблоне, $\Gamma_{\{A,k\}}(i, l)$ - оценка сопоставления i -й локальной структуры шаблона A с l -й локальной структурой шаблона k .

4.2. Особенности реализации

Задачей алгоритма является выделение *genuine* и *impostor* оценок для локальных структур из $\Gamma_{\{A,B\}}$. *Impostor* оценки для локальных структур выделяются из матриц схожестей двух шаблонов от разных пальцев. Выделение *genuine* оценок происходит в два этапа. Как привило, *genuine* оценки должны иметь более высокие значения, чем *impostor*, хотя это не всегда так. Поэтому первым шагом алгоритма является применение к матрице локальных схожестей $\Gamma_{\{A,B\}}$ алгоритма консолидации, который даст T_{set} - множество лучших оценок сопоставления локальных структур шаблонов A и B , претендующих на то, чтобы оказаться *genuine*. После чего с помощью *алгоритма параллельных линий* из T_{set} выбираются истинные оценки, которые формируют G_{set} - множество *genuine* оценок для локальных структур шаблона A . Аналогичным образом из матрицы $\Gamma_{\{B,A\}}$ можно получить *genuine* оценки для локальных структур шаблона B .

4.2.1. Выбор лучших оценок локального сопоставления из *LSM*

Как и на глобальном уровне, где множество глобальных *genuine* оценок $G_k = S(k, k)$ определяется как множество оценок сопоставления шаблонов от одного и того же пальца, для построения G_k на локальном уровне рассматриваются матрицы локальных схожестей двух разных шаблонов одного и того же пальца.

Пусть для таких двух шаблонов матрицей схожести будет $\Gamma_{\{A,B\}}$. К ней применяется алгоритм консолидации, из результатов работы кото-

рого можно выделить множество из n_p позиций лучших оценок сопоставления, которые могут быть *genuine*:

$$T_{set} = \{(i, j) \mid i \in 1...n_A, j \in 1...n_B, \Gamma_{\{A,B\}}(i, j) - top\ score\},$$

где n_p - параметр алгоритма консолидации, n_A, n_B - количество локальных структур в шаблонах A и B , $i(j)$ - номер строки (столбца), минуции и соответствующей ей локальной структуре в шаблоне A (B).

4.2.2. Алгоритм параллельных линий

Данный алгоритм работает со множеством T_{set} . Для каждой пары $(i, j) \in T_{set}$ отмечаются i и j минуции на соответствующих шаблонах A и B , расположенных рядом, и между ними строятся отрезки.

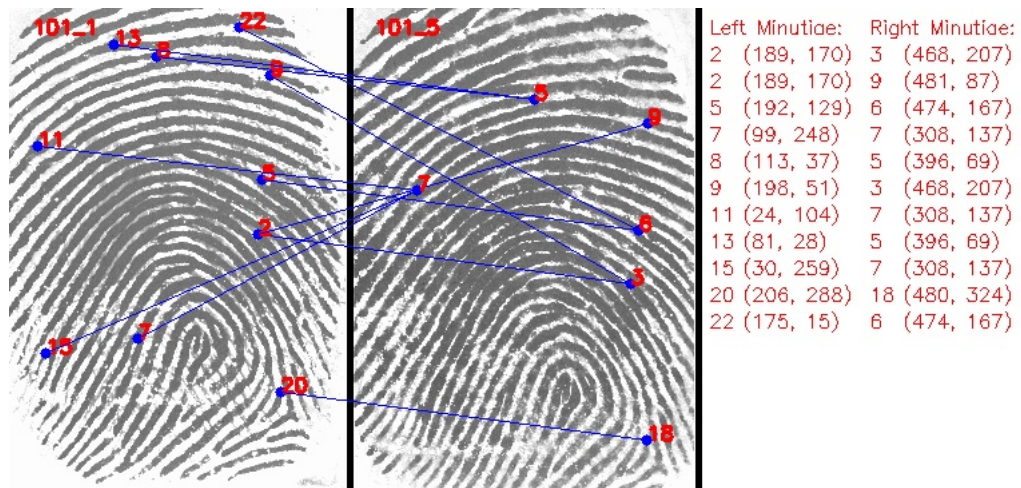


Рис. 4: Отрезки между парами минуций, для которых сопоставление локальных структур, соответствующим этим минуциям, попало в число n_p лучших.

С использованием построенных отрезков формируется *множество пар истинных сопоставлений* M_{set} . Пара минуций (и соответствующих им локальных структур) считается истинно сопоставленной, если отрезок, соединяющий эти минуции, "параллелен" как минимум $\frac{n_p}{2}$ отрезкам. "Параллельность" для отрезков определяется по формуле: $(0^\circ \leq |angle_1 - angle_2| \leq 2^\circ) \parallel (178^\circ \leq |angle_1 - angle_2| \leq 180^\circ)$, где $angle = \arctan(\frac{y_1 - y_0}{x_1 - x_0})$, $(x_0, y_0), (x_1, y_1)$ - координаты концов отрезка.

4.2.3. Выделение *genuine* оценок из *LSM*

После нахождения множеств T_{set} и M_{set} , выполняется их пересечение, с целью отбрасывания потенциально ложных сопоставлений, найденных алгоритмом консолидации и имеющих высокие оценки схожести:

$$G_{set} = T_{set} \cap M_{set}$$

Каждый элемент множества $(i, j) \in G_{set}$ - пара номеров *истинно сопоставленных* локальных структур в матрице $\Gamma_{\{A,B\}}$, определяющий *genuine* оценку для i -й структуры в матрице $\Gamma_{\{A,B\}}$.

Таким образом можно получить множества *genuine* и *impostor* оценок G_k и I_k на локальном уровне. По ним построить \mathcal{G}_H , \mathcal{G}_L , \mathcal{I}_H и \mathcal{I}_L и провести эксперимент Ягера-Данстона по той же методологии, но с целью выявления эффекта биометрического зверинца на локальном уровне.

4.3. Крайний случай работы алгоритма выделения *genuine* оценок

В худшем случае, если алгоритм параллельных линий не нашел истинных сопоставлений, т.е. множество M_{set} оказалось пустым, то и $G_{set} = T_{set} \cap M_{set}$ будет пусто. Тогда для i -й локальной структуры, имеющей $G_{set} = \emptyset$, множество *genuine* оценок G_j будет пустым. При построении множества *impostor* оценок такой проблемы не возникнет, и множество I_j никогда не будет пустым.

Таким образом, результат работы алгоритма выделения *genuine* оценок представляет собой множество локальных структур двух типов:

- локальные структуры, у которых $G_{set} = \emptyset$;
- локальные структуры, у которых $G_{set} \neq \emptyset$.

5. Результаты экспериментов

5.1. Используемые технологии

Входными данными для проведения эксперимента выступают базы данных отпечатков пальцев: *FVC2000*, *FVC2002*, *FVC2004* [11, 12, 13]. Каждая база имеет 4 выборки по 80 изображений отпечатков для 10 разных пальцев, по 8 отпечатков на каждый палец. Всего 12 выборок и 960 отпечатков.

Для извлечения минуций и построения шаблонов используется программное обеспечение *Griaule Fingerprint SDK* [6], которое находит в исходном изображении отпечатка координаты минуций и их направление.

Согласно обзору [34] в качестве двухфазных алгоритмов были выбраны наиболее точные: *MCC* [7], *Feng* (упрощённая версия базового алгоритма с *minutiae-based* локальными структурами) [14], *Bozorth3* [31] и *Deng* [10]. Однако два алгоритма из четырёх не рассматриваются, т.к. *Bozorth3* не имеет открытой реализации, а *Deng* дополнительно требует тип минуций. Так же из [7] были взяты четыре алгоритмы консолидации *LSS*, *LSA*, *LSS-R*, *LSA-R*.

Реализация [15, 8, 22] алгоритма *MCC* взята из *MCC* [4], реализации *minutiae-based* версии алгоритма *Feng* и алгоритмов консолидации *LSS*, *LSA*, *LSS-R*, *LSA-R* написаны на языке *Python* [23].

5.2. Проведение эксперимента

Из отпечатков извлекаются минуции, на основе которых строятся локальные структуры, и с помощью алгоритмов локального сопоставления вычисляются матрицы локальных схожестей. Далее проводятся эксперименты Ягера-Данстона для глобальных и локальных оценок сопоставления.

Поскольку эксперимент Ягера-Данстона подразумевает статистическую обработку данных, то была применена модификация эксперимента с целью получить устойчивую к выбросам статистику и сравнить её

с базовой, а именно, для построения множеств *genuine* G_k и *impostor* I_k оценок была выбрана медиана.

5.3. Эксперимент Ягера-Данстона на глобальном уровне

Перед проведением эксперимента на глобальном уровне к матрицам локальных схожестей были применены алгоритмы консолидации, которые дали *genuine* и *impostor* оценки сопоставления шаблонов. Данные оценки стали входными данными для эксперимента Ягера-Данстона, который показал результаты, отраженные в таблицах 1 и 2. Строки таблицы соответствуют названиям классам биометрического зверинца столбцы - комбинации локального алгоритма сопоставления с некоторым алгоритмом консолидации и статистикой, которая применялась к *genuine* и *impostor* оценкам. Число в ячейке таблицы отражает в скольких выборках из 12 был найден класс, соответствующий данной строке.

MCC								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	2	1	2	3	1	2	1	2
Phantomes	0	0	0	0	0	0	0	0
Doves	0	0	0	0	0	0	1	1
Worms	0	0	0	0	0	0	0	0

Таблица 1: Результаты для алгоритма *MCC* на глобальном уровне

Feng								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	10	8	11	10	7	3	2	4
Phantomes	5	6	7	7	5	6	5	4
Doves	1	1	0	0	1	1	0	0
Worms	0	0	0	0	0	0	0	0

Таблица 2: Результаты для *minutia-based* версии алгоритма *Feng* на глобальном уровне

Из результатов, представленных в таблицах 1 и 2, можно сделать следующие выводы:

- Алгоритмы *MCC* и *Feng* приносят эффект биометрического зверинца в глобальный уровень.
- Оба алгоритма всегда находят класс *chameleons*, который характеризуется высокими *genuine* и высокими *impostor* оценкам. Стоит подумать над тем, как бороться с представителями этого класса.
- Алгоритм *Feng* по сравнению с *MCC* приносит эффект биометрического зверинца в большее количество выборок, что позволяет сделать вывод о том, что локальные структуры *Feng* чаще служат причиной непропорционального распределения ошибок в биометрической системе.
- Случаи, когда находится класс *Doves*, нам не страшны, и даже полезны. Представители этого класса - лучшие пользователи биометрической системы, поскольку они имеют высокие *genuine* и низкие *impostor* оценки.
- Существенных различий между разными алгоритмами консолидации и статистками, применяемыми к одному алгоритму локального сопоставления, не наблюдается.
- Для алгоритма *MCC* с точки зрения эффекта биометрического зверинца алгоритм консолидации *LSS-R* является наихудшим.
- Для алгоритма *Feng* с точки зрения эффекта биометрического зверинца лучшим алгоритмом консолидации является *LSA-R*.

5.4. Эксперимент Ягера-Данстона на локальном уровне

Для данного эксперимента обработка матриц локальных схожестей происходит иначе. Данные матрицы являются множеством локаль-

ных оценок сопоставления локальных структур шаблонов. Из множества локальных оценок с помощью специального алгоритма выделяются *genuine* и *impostor* оценки. Далее проводится эксперимент Ягера-Данстона по выявлению классов биометрического зверинца среди локальных структур шаблонов пользователей для каждой статистики (матожидания и медианы).

Поскольку результаты выделения *genuine* и *impostor* оценок могут отличаться ($(G_{set} = \emptyset) \parallel (G_{set} \neq \emptyset)$), то результат эксперимента будет разным.

При рассмотрении зверинца для всех локальных структур, в том числе и таких, для которых $(G_{set} = \emptyset)$, может случиться так, что они будут являться представителями классов *Worms* и *Phantomes*, т.к. истинных сопоставлений для них не найдено. Такой случай отражен в таблицах 3, 4 с обозначениями, аналогичными таблицам 1, 2.

MCC								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	0	0	0	0	0	0	0	0
Phantomes	12	12	12	12	12	12	12	12
Doves	0	0	0	0	0	0	0	0
Worms	12	12	12	12	12	12	12	12

Таблица 3: Результаты для алгоритма *MCC* на локальном уровне с учётом локальных структур с $G_{set} = \emptyset$

Feng								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	0	4	0	3	0	6	0	4
Phantomes	12	12	12	12	12	12	12	12
Doves	0	0	0	0	0	0	0	0
Worms	12	12	12	12	12	12	12	12

Таблица 4: Результаты для *minutia-based* версии алгоритма *Feng* на локальном уровне с учётом локальных структур с $G_{set} = \emptyset$

При рассмотрении зверинца только для таких локальных структур, у которых которых множество $(G_{set} \neq \emptyset)$, найденные классы зверинца распределяются совсем иначе, что продемонстрировано в таблицах 5, 6 с обозначениями, аналогичными таблицам 1, 2.

MCC								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	1	2	2	4	1	2	3	5
Phantomes	0	0	5	5	0	0	8	10
Doves	6	6	2	2	7	6	2	1
Worms	2	1	1	1	2	1	0	0

Таблица 5: Результаты для алгоритма *MCC* на локальном уровне с учётом локальных структур с $G_{set} \neq \emptyset$

Feng								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	10	11	12	11	11	12	12	12
Phantomes	7	10	9	12	8	12	11	12
Doves	0	4	0	6	0	4	0	5
Worms	0	5	1	6	0	3	0	12

Таблица 6: Результаты для *minutia-based* версии алгоритма *Feng* на локальном уровне с учётом локальных структур с $G_{set} \neq \emptyset$

Из полученных результатов можно сделать следующие выводы:

- Рассмотрение эффекта биометрического зверинца для локальных структур с ($G_{set} = \emptyset$) даёт практически одинаковую картину для двух алгоритмов. Наличие классов *Chameleons* и *Phantomes* в этом случае не так страшно, как повсеместное наличие класса *Worms* во всех 12 выборках.
- Рассмотрение эффекта биометрического зверинца для локальных структур с ($G_{set} \neq \emptyset$) позволяет сделать следующие наблюдения:
 - Алгоритмы *MCC* и *Feng* приносят эффект биометрического зверинца в локальный уровень.
 - Алгоритм *Feng* справляется хуже *MCC* и в большем количестве выборок находит классы зверинца. В тоже время, появление класса *Worms* можно избежать с помощью выбора нужной статистики.
 - Для алгоритма *MCC* наблюдаются существенные различия между алгоритмами консолидации *LSS*, *LSA* и *LSS-R*, *LSA-*

R. Алгоритмы *LSS-R*, *LSA-R* привносит большее количество классов *Phantomes* и *Chameleons*.

- Для алгоритма *Feng* существенных различий между применяемыми алгоритмами консолидации не наблюдается.
- Найденные классы *Doves* - это хорошо, т.к. представители этого класса - лучшие пользователи для биометрической системы с точки зрения распознавания.

5.5. Преимущества локального уровня над глобальным

Эксперимент Ягера-Данстона на локальном уровне показал наличие эффекта биометрического зверинца. Отметим, что в обзоре литературе были рассмотрены различные работы, посвященные исследованию данного эффекта на глобальном уровне. Однако ни в одной работе не представляется возможным избавить систему даже от одного *плохого* пользователя, принадлежащего некоторому классу биометрического зверинца.

Преимущество предложенного подхода заключается в том, что на локальном уровне можно предпринять попытку избавить систему не от шаблонов, а от *плохих* локальных структур этих шаблонов. Конечно же избавить не от всех до единого, а только от тех, от которых сможем. Поскольку локальные структуры, принадлежащие классу *Worms*, приносят два типа ошибок - *FAR* и *FRR*, то в первую очередь необходимо отфильтровать шаблоны именно от них.

5.5.1. Описание алгоритма фильтрации *плохих* локальных структур

Фильтрация выполняется для выборки, в которой найден некоторый класс. Входными данными для данного алгоритма являются:

- Массив шаблонов из выборки. Каждый элемент массива - список:

- номер шаблона;
 - номер локальный структуры;
 - текущее количество структур в шаблоне;
 - общее количество структур в шаблоне.
- Информация о присутствии класса в данной выборке после проведения эксперимента Ягера-Данстона на локальном уровне.

Algorithm 2 Фильтрация локальных структур, принадлежащих классу *Worms*

Input: *fingers_array*, *worms_found*

Output: *filtered_fingers_array*

```

1: function FILTER_CLASS(fingers_array, worms_found)
2:   while worms_found do
3:     worms_found = False
4:     for all finger_data  $\in$  fingers_array do
5:       struct = finger_data[1] ▷ local_struct from finger
6:       amount1 = finger_data[2] ▷ total local_structures amount
7:       amount2 = finger_data[2] – 1 ▷ total amount – worms-like struct
8:       if (struct is Worm) & (amount2/amount1 > 3/4) then
9:         Remove struct from fingers_array
10:        worms_found = True
11:        Continue
12:       if not worms_found then
13:         break
14:     worms_found = YAGERDUNSTONE(fingers_array)
15: return fingers_array

```

Для некоторой выборки алгоритм 2 рассматривает список шаблонов с их локальными структурами (далее список локальных структур) и результат эксперимента Ягера-Данстона, дающий информацию о наличии класса *Worms*. В цикле проверяется условие того, класс в выборке найден. Далее из списка локальных структур ищется первая, ко-

торая является представителем класса *Worms*. Если она найдена, то проверяется, что при её удалении из шаблона, в нём останется не менее 75% локальных структур. Данная проверка обеспечивает возможность фильтровать *плохие* структуры, не удаляя полностью шаблон пользователя из выборки. Если *плохая* структура не найдена, то цикл прерывается, алгоритм завершает работу. Иначе, проводится эксперимент Ягера-Данстона для отфильтрованного списка локальных структур и выполняется следующая итерация цикла.

Применение описанного выше алгоритма обеспечивает фильтрацию *плохих* локальных структур шаблонов текущей выборки без удаления самого шаблона, минимизирует число представителей класса *Worms*. К результатам этой фильтрации можно применить её ещё раз, но уже для других классов, однако при повторении процесса класс *Worms* не должен появляться.

5.5.2. Результаты фильтрации

К результатам локального уровня, учитывающим локальные структуры с $G_{set} = \emptyset$ (таб. 3, 4), был применён алгоритм фильтрации локальных структур, принадлежащих классу *Worms*, результаты которого приведены в таблицах 7 и 8. Обозначения в данных таблицах аналогичны таблицам 1, 2.

	MCC							
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	0	1	0	1	0	1	1	2
Phantomes	12	12	12	12	12	12	12	12
Doves	0	0	0	0	0	0	0	0
Worms	2	4	1	1	2	4	1	1

Таблица 7: Результаты фильтрации для алгоритма *MCC* на локальном уровне с учётом локальных структур с $G_{set} = \emptyset$

Feng								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	5	5	7	9	6	7	7	11
Phantomes	12	12	12	12	12	12	12	12
Doves	0	0	0	0	0	0	0	0
Worms	12	12	5	9	9	11	6	9

Таблица 8: Результаты фильтрации для *minutia-based* версии алгоритма *Feng* на локальном уровне с учётом локальных структур с $G_{set} = \emptyset$

Из полученных результатов можно сделать следующие выводы:

- Не во всех выборках были ликвидированы классы *Worms*, поскольку главным условием фильтрации было сохранение в каждом шаблоне выборки не менее 75% локальных структур.
- Выборки, обработанные алгоритмом *MCC*, удалось достаточно хорошо отфильтровать от представителей класса *Worms*.
- Выборки, обработанные алгоритмом *Feng*, не удалось хорошо отфильтровать, более того, во многих выборках выявились классы *Chameleons*.
- Для обоих алгоритмов наблюдается появление класса *Chameleons*. Это вызвано тем, что при фильтрации классов *Worms* снижается величина I_h , используемая в эксперименте Ягера-Данстона, что приводит к появлению классов *Chameleons*.
- Для алгоритма *MCC* существенных различий между применяемыми алгоритмами консолидации и статистками не наблюдается.
- Для алгоритма *Feng* видны некоторые различия между применяемыми алгоритмами консолидации. Алгоритм *LSS* показал себя лучше всего, *LSA-R* - хуже всего.
- Почти во всех случаях использование математического ожидания в качестве статистики обеспечило чуть меньшее количество выявленных классов.

К результатам локального уровня, учитывающим только локальные структуры с $G_{set} \neq \emptyset$ (таб. 5, 6), был применён алгоритм фильтрации локальных структур, принадлежащих классу *Worms*, а после - алгоритм фильтрации *Chameleons*. Результаты двух последовательных фильтраций приведены в таблицах 9 и 10. Обозначения в данных таблицах аналогичны таблицам 1, 2.

MCC								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	0	0	0	0	0	0	0	0
Phantomes	0	0	5	5	0	0	8	10
Doves	6	6	2	2	7	6	2	1
Worms	0	0	0	0	0	0	0	0

Таблица 9: Результаты фильтрации для алгоритма *MCC* на локальном уровне с учётом локальных структур с $G_{set} \neq \emptyset$

Feng								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	3	10	1	7	1	7	0	3
Phantomes	6	10	8	12	6	12	11	12
Doves	0	5	0	7	0	7	0	7
Worms	0	0	0	0	0	0	0	0

Таблица 10: Результаты фильтрации для *minutia-based* версии алгоритма *Feng* на локальном уровне с учётом локальных структур с $G_{set} \neq \emptyset$

Из полученных результатов можно сделать следующие выводы:

- Во всех выборках были ликвидированы классы *Worms*.
- Выборки, обработанные алгоритмом *MCC*, удалось полностью отфильтровать и от классов *Chameleons*.
- В выборках, обработанных алгоритмом *Feng*, получилось отбросить классы *Worms*, но не удалось отфильтровать и от классов *Chameleons*.
- После фильтрации в результаты *MCC* и *Feng* не выявлены новые классы, кроме *Doves*.

- Для алгоритма *MCC* наблюдаются различия между применяемыми алгоритмами консолидации, статистики ведут себя похожим образом.
- Для алгоритма *Feng* видны некоторые различия между применяемыми алгоритмами консолидации, медиана находит больше классов биометрического зверинца.

5.6. Выводы

После проведения эксперимента Ягера-Данстона для локальных структур шаблонов исходных выборок, выявления эффекта биометрического зверинца на локальном уровне и фильтрации *плохих* локальных структур были получены урезанные шаблоны для тех выборок, где фильтрация была успешной. Далее предполагается подняться от локального уровня обратно на глобальный, т.е. построить матрицы локальных схожестей для урезанных шаблонов, провести консолидацию и сравнить FAR/FRR зависимость между начальными и урезанными данными.

В следующей главе рассматривается сравнение оригинальных и урезанных данных, обрабатываемых с помощью алгоритма *MCC*. Данным, обработанным алгоритмом *Feng*, внимание не уделяется, поскольку результат их фильтрации оказался хуже, чем для *MCC*.

6. Сравнение оригинальных и отфильтрованных данных для *MCC*

6.1. Описание оригинальных данных

Исходными данными являются 12 выборок: *DB_2000_DB1*, *DB_2000_DB2*, *DB_2000_DB3*, *DB_2000_DB4*, *DB_2002_DB1*, *DB_2002_DB2*, *DB_2002_DB3*, *DB_2002_DB4*, *DB_2004_DB1*, *DB_2004_DB2*, *DB_2004_DB3*, *DB_2004_DB4*. Каждая содержит по 10 разных отпечатков, по 8 шаблонов на каждый - 80 отпечатков. Для всех пар шаблонов, не содержащих один и тот же шаблон, применяется алгоритм *MCC*, предоставляющий матрицы локальных схожестей для каждого сопоставления, которые обрабатываются алгоритмами консолидации *LSS*, *LSS-R*, *LSA*, *LSA-R*, формирующими глобальные оценки каждому сопоставлению. Всего 6320 сопоставлений на одну выборку: 560 genuine и 5760 impostor оценок. Далее эти данные обозначаются как *baseline*.

6.2. Описание отфильтрованных данных

Исходные данные всё те же 12 выборок: *DB_2000_DB1*, ..., *DB_2004_DB4*. Для каждой выборки проводится эксперимент Ягера-Данстона на локальном уровне с целью выявления классов биометрического зверинца. Далее в зависимости от того, обнаружен ли класс *Worms*, выполняется фильтрация шаблонов выборок от *плохих* локальных структур, с сохранением в шаблонах не менее 75% структур от общего количества. После проводится ещё один этап фильтрации, но уже для класса *Chameleons*. Далее для всех пар шаблонов, не содержащих один и тот же шаблон, применяется алгоритм *MCC*, предоставляющий матрицы локальных схожестей для каждого сопоставления, которые обрабатываются алгоритмами консолидации *LSS*, *LSS-R*, *LSA*, *LSA-R*, формирующими глобальные оценки каждому сопоставлению. Всего 6320 сопоставлений на одну выборку: 560 genuine и 5760 impostor оце-

НОК.

6.3. Описание отфильтрованных данных, учитывающих локальные структуры с $G_{set} = \emptyset$

Для этого набора данных был:

- отфильтрован класс *Worms* в шаблонах тех выборок, в которых это удалось;
- обнаружен класс *Chameleons* в результате фильтрации *Worms*;
- вновь найден класс *Phantomes* во всех 12 выборках для всех методов, т.к. он не подлежал фильтрации.

6.4. Описание отфильтрованных данных, учитывающих только локальные структуры с $G_{set} \neq \emptyset$

Пронумеруем выборки *DB_2000_DB1*, ..., *DB_2004_DB4* от 1 до 12. В каждую ячейку таблицы 11 запишем номера выборок, в которых был **отфильтрован** некоторый класс.

Таблица с найденными классами								
	LSS		LSS-R		LSA		LSA-R	
	mean	median	mean	median	mean	median	mean	median
Chameleons	№ 4	№ 8, 12	№ 4, 11	№ 1, 8, 11, 12	№ 4	№ 8, 12	№ 4, 11, 12	№ 1, 8, 11, 12
Phantomes								
Doves								
Worms	№ 3, 9	№ 3	№ 3, 9	№ 3	№ 3	№ 3		

Таблица 11: Таблица с найденными классами

Класс *Worms* **отфильтрован** для:

- комбинаций обеих статистик с методами *LSS*, *LSA*, *LSS-R* в выборке *DB2000_DB3*;
- комбинации математического ожидания и методов *LSS*, *LSA* в выборке *DB2004_DB1*.

Класс *Chameleons* **отфильтрован** для:

- комбинации математического ожидания с *LSS*, *LSA*, *LSS-R*, *LSA-R* в выборке *DB2000_DB4*;
- комбинации обеих статистик с *LSA-R* и *LSS-R* в *DB2004_DB3*;
- комбинации медианы с *LSS*, *LSA*, *LSS-R*, *LSA-R* в выборках *DB2002_DB4*, *DB2004_DB4*;
- комбинации медианы с *LSS-R*, *LSA-R* в выборке *DB2000_DB1*;
- комбинации математического ожидания с *LSA-R* в выборке *DB2004_DB4*.

Класс *Phantomes* присутствует в 5 выборках для *LSS-R* и *LSA-R*, фильтрация его представителей не производилась.

6.5. Результаты сравнения baseline и фильтрованных данных

6.5.1. Baseline и данные с $G_{set} = \emptyset$

Для каждой выборки, фильтрация класса *Worms* для которой была успешной/неуспешной, были построены два, расположенных рядом графика, отображающих *ROC*-кривые, демонстрирующие результаты после фильтрации. Каждый график для некоторой выборки содержит 4 *baseline*-кривые и кривые, демонстрирующие ошибки, полученные в результате сопоставления отфильтрованных шаблонов. Название кривой соответствует методу, с помощью которого были получены отфильтрованные шаблоны. Графики слева соответствуют методам, использующим математическое ожидание, справа - медиану.

Для выборки *DB_2000_DB1* графики 5, 6 наглядно показывают, что комбинация *LSS* и *LSA* с любой статистикой показывает худший *EER*. Этот же показатель для *LSS-R* и *LSA-R* с медианой лучше, чем с математическим ожиданием. Более того, метод *LSA-R_median* немного лучше базового, *LSS-R_median* сопоставим с базовым.

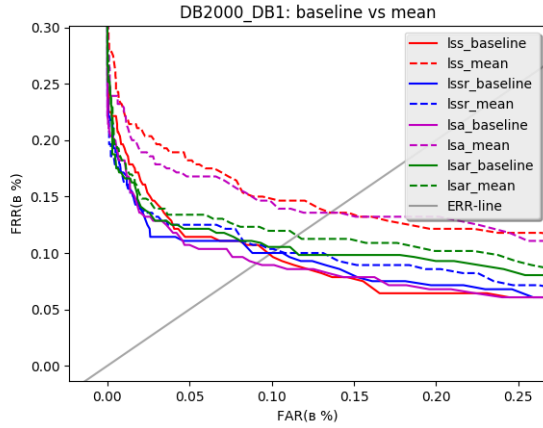


Рис. 5: DB2000_DB1

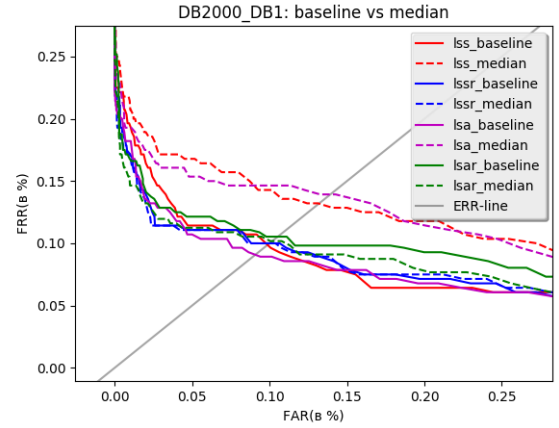


Рис. 6: DB2000_DB1

Для выборки *DB_2000_DB2* графики 7, 8 демонстрируют, что комбинация *LSS* и *LSA* с любой статистикой показывает худший *EER*, поскольку класс *Worms* присутствует в данной выборке для этих методов. Методы *LSS-R* и *LSA-R* для обоих статистик сопоставимы с базовым, но не лучше их.

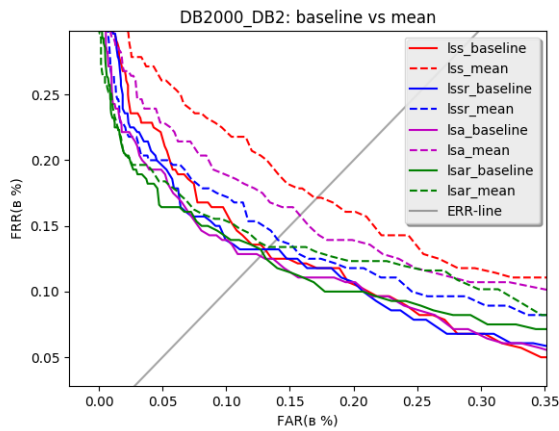


Рис. 7: DB2000_DB2

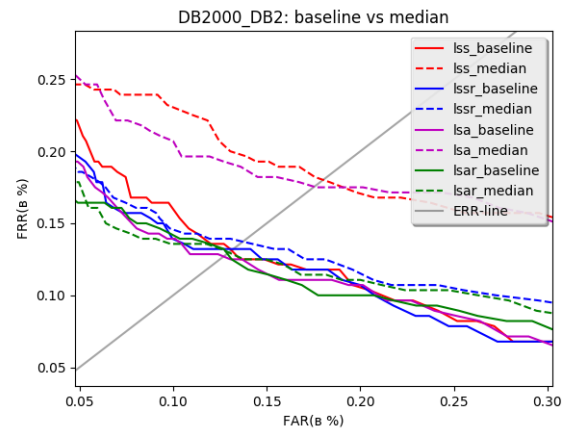


Рис. 8: DB2000_DB2

В выборке *DB_2000_DB3* для всех методов найден класс *Worms*. На графиках 9, 10 хорошо видно, что комбинация предложенные методы с любой статистикой показывает худший *EER*, как и базовый метод *LSA-R*. Однако комбинация *LSS-R* с любой статистикой приближается к базовым, но не лучше их.

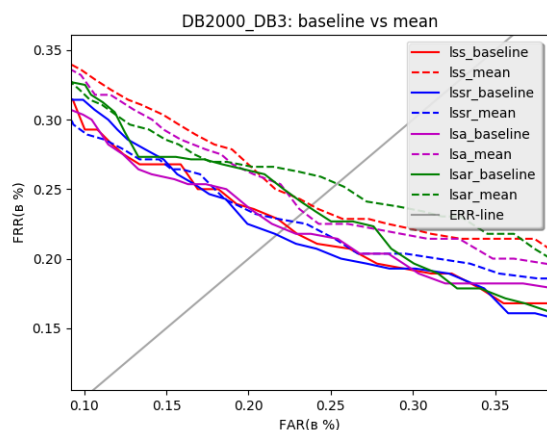


Рис. 9: DB2000_DB3

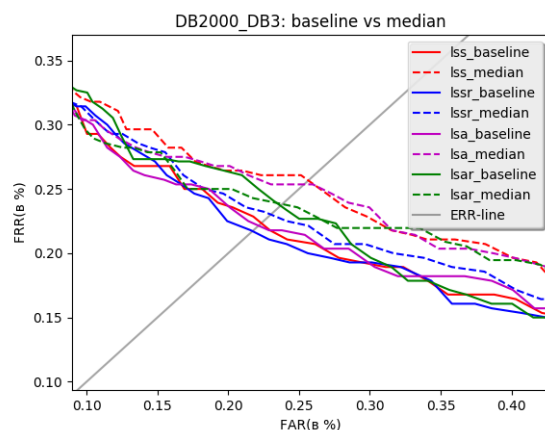


Рис. 10: DB2000_DB3

Для выборки *DB_2000_DB4* по графикам 11, 12 можно сделать вывод, что комбинация *LSA-R* с математическим ожиданием - лучшая, сопоставимая с базовым *LSA-R*. Остальные предложенные методы с любой статистикой не лучше базовых.

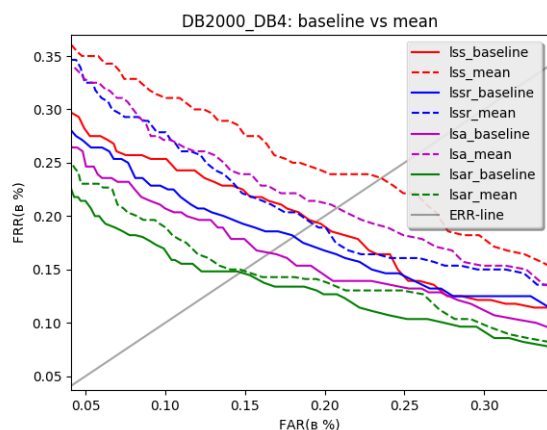


Рис. 11: DB2000_DB4

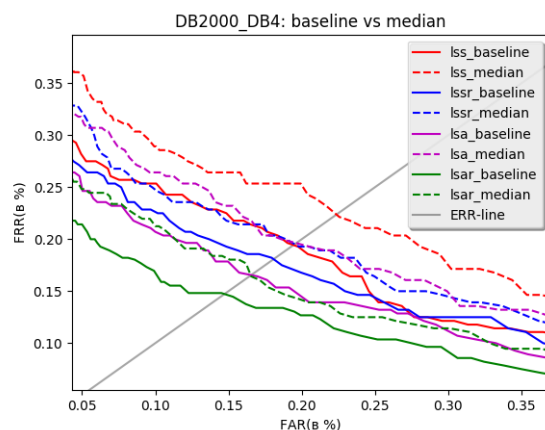


Рис. 12: DB2000_DB4

Для выборки *DB_2002_DB1* на графиках 13, 14 выделяется высокий показатель *EER* для *LSS* и *LSA* с любой статистикой. Комбинации *LSA-R* и *LSS-R* с математическим ожиданием - лучшие, сопоставимы с базовым.

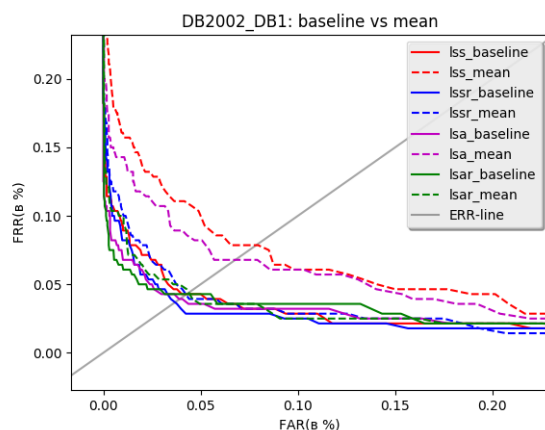


Рис. 13: DB2002_DB1

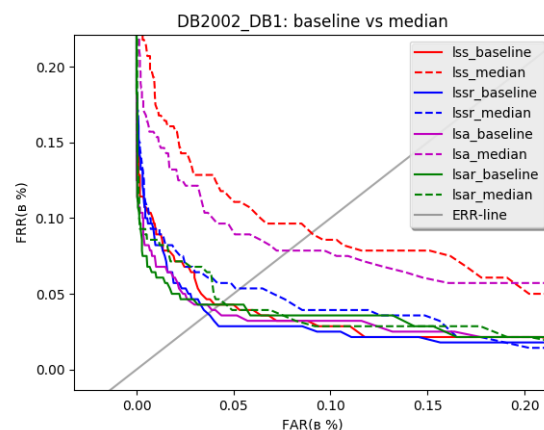


Рис. 14: DB2002_DB1

Для выборки *DB_2002_DB1* на графиках 15, 16 показано, что *LSS* и *LSA* дают худший *EER* с любой статистикой. Метод *LSA-R* независимо от статистики показывает хороший *EER* и сопоставим с базовым.

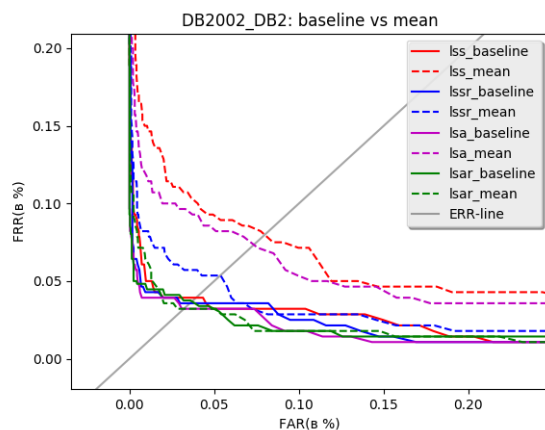


Рис. 15: DB2002_DB2

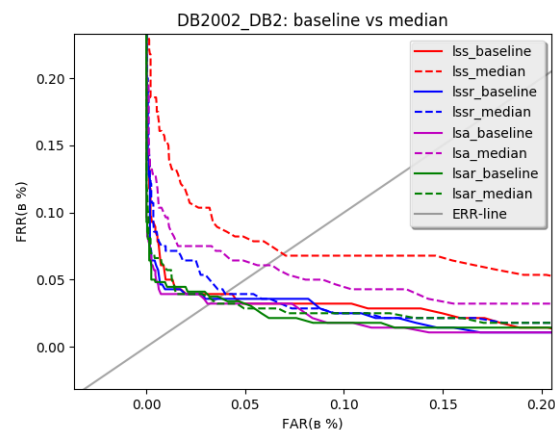


Рис. 16: DB2002_DB2

Для выборки *DB_2002_DB3* на графиках 17, 18 отражены худшие показатели для предложенных методов, метод *LSA-R* независимо от статистики лучший, тем не менее он хуже базовых.

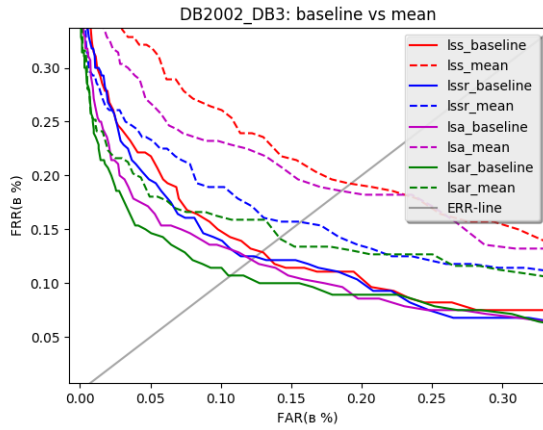


Рис. 17: DB2002_DB3

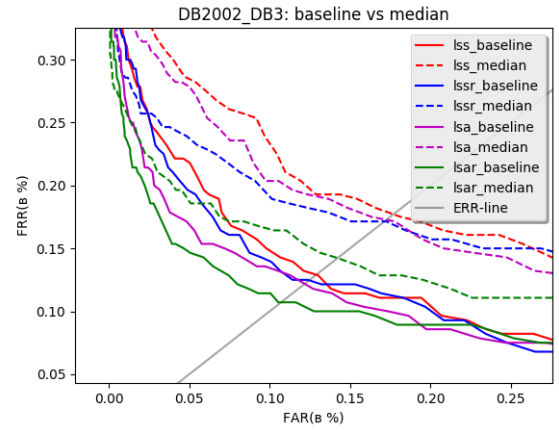


Рис. 18: DB2002_DB3

Для выборки *DB_2002_DB4* на графиках 19, 20 приведены худшие показатели для *LSS* и *LSA* с любой статистикой. Метод *LSS-R* с медианой приближается к базовому, *LSA-R* независимо от статистики даёт наилучший *EER*.

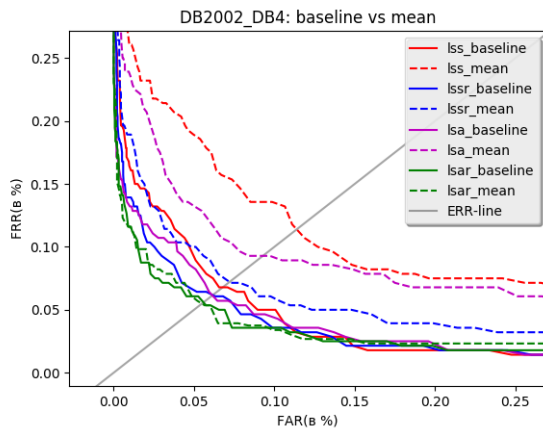


Рис. 19: DB2002_DB4

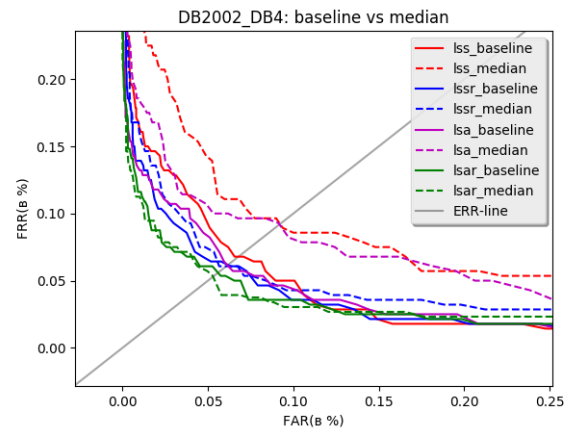


Рис. 20: DB2002_DB4

Для выборки *DB_2004_DB1* на графиках 21, 22 видно, что предложенные методы хуже базовых. Выделяется базовый *LSA-R*, дающий лучший показатель *EER*. Предложенные версии *LSA-R* сильно хуже базового метода.

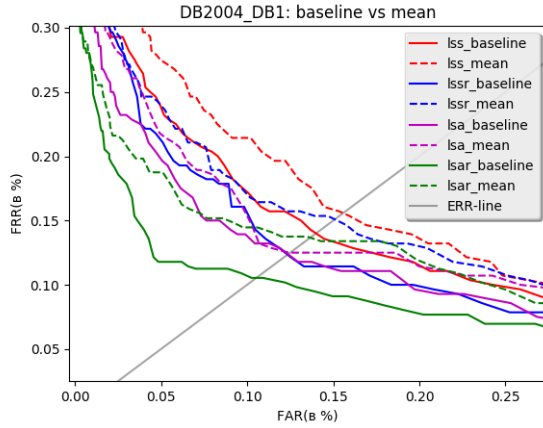


Рис. 21: DB2004_DB1

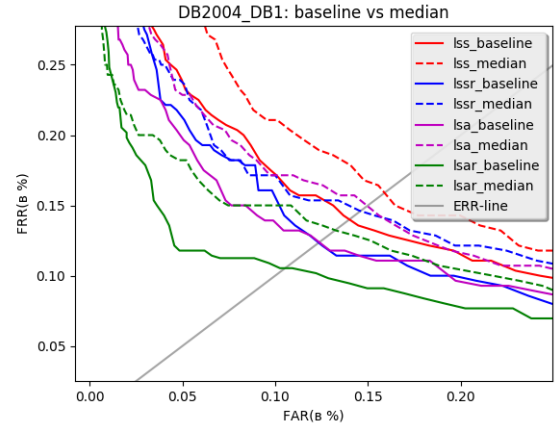


Рис. 22: DB2004_DB1

Для выборки *DB_2004_DB2* на графиках 23, 24 видно, что предложенные методы хуже базовых. Выделяется оригинальный *LSA-R*, предоставляющий наименьший *EER*.

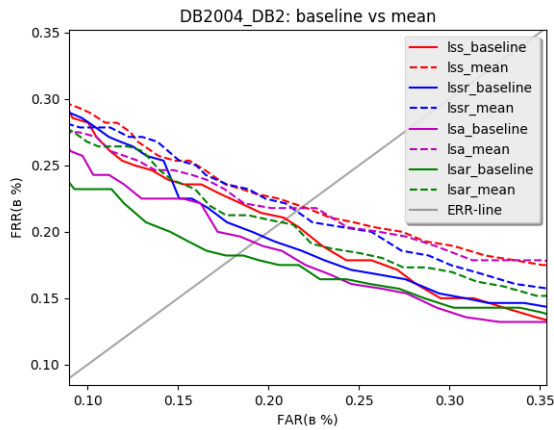


Рис. 23: DB2004_DB2

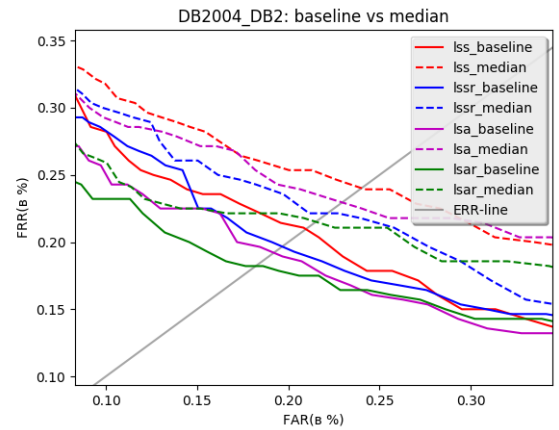


Рис. 24: DB2004_DB2

В выборке *DB_2004_DB3* для *LSS* и *LSA* с любой статистикой найден класс *Worms*, для *LSS-R* с медианой - класс *Chameleons*. На графиках 25, 26 наблюдается, что *LSA-R* вне зависимости от статистики сопоставим с базовым, как и *LSS-R*. Кроме того, как предложенная, так и базовая версии *LSA-R* дают наилучший *EER*.

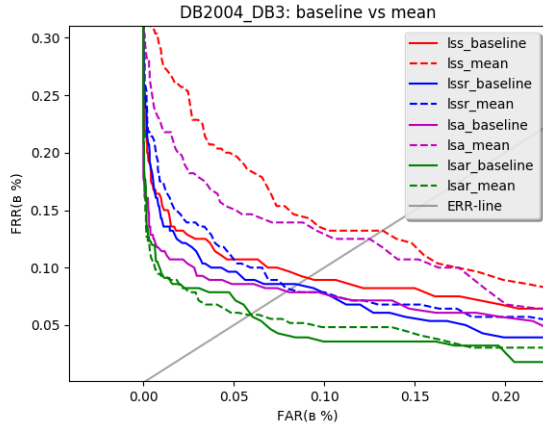


Рис. 25: DB2004_DB3

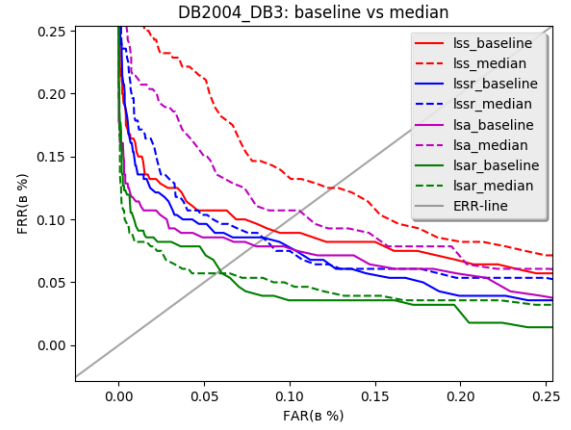


Рис. 26: DB2004_DB3

В выборке *DB_2004_DB4* для всех методов с медианой найден класс *Chameleons*. Графики 27, 28 демонстрируют, что использование медианы даёт лучший EER для всех методов по сравнению с математическим ожиданием. Кроме того, *LSS-R* с медианой приближается к базовому *LSS-R*, а *LSA-R* с медианой совсем немного выигрывает у базовой.

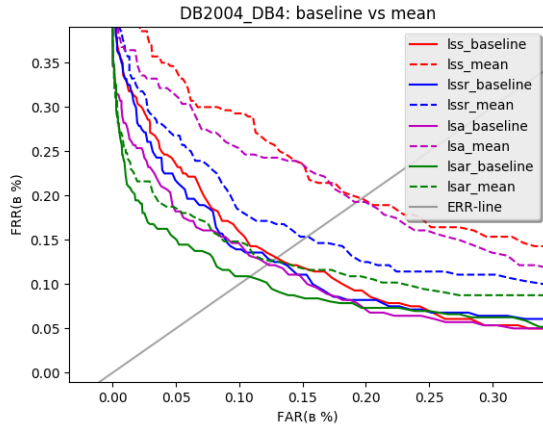


Рис. 27: DB2004_DB4

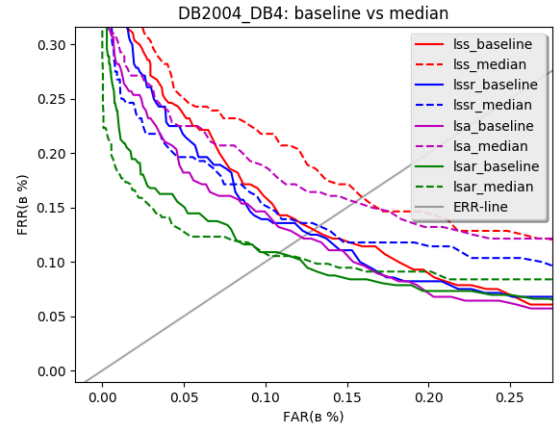


Рис. 28: DB2004_DB4

6.5.2. Baseline и данные с $G_{set} \neq \emptyset$

Для тех выборок, в которых была проведена фильтрация, были построены *ROC*-кривые, демонстрирующие результаты после фильтрации. Каждый график для некоторой выборки содержит *baseline*-кривые и кривые, демонстрирующие ошибки, полученные в результате сопо-

ставления отфильтрованных шаблонов. Название кривой соответствует методу, с помощью которого были получены отфильтрованные шаблоны.

6.5.3. После фильтрации *Worms*

Для выборки *DB2000_DB3* графиках 29, 30 продемонстрировано, что с использованием медианы *LSS* и *LSA* достигают показателя *EER* выше, чем *LSS-R* и остальные базовые методы, с которыми *LSS-R* сопоставим. При использовании математического ожидания *LSS* и *LSA* работают чуть хуже базовых, а *LSS-R* всё так же хорош.

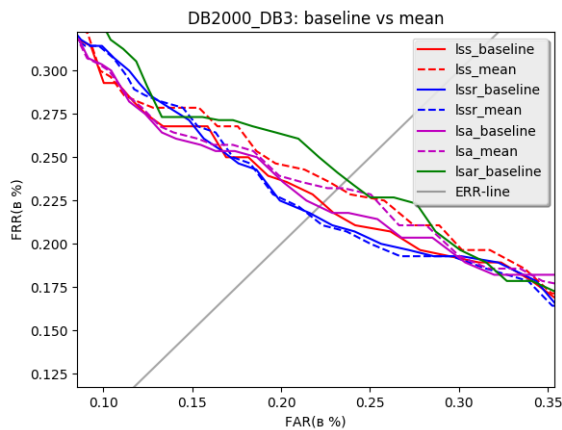


Рис. 29: DB2000_DB3

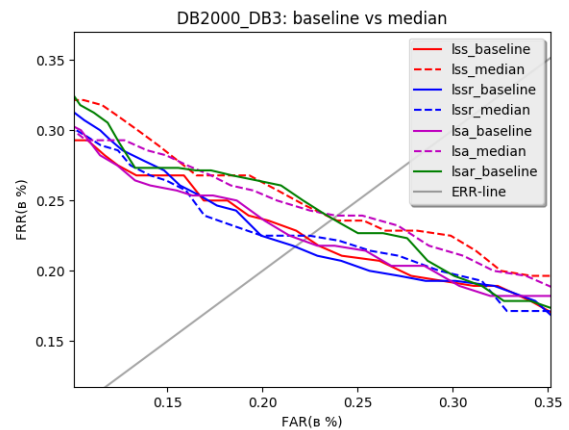


Рис. 30: DB2000_DB3

Для выборки *DB2004_DB1* на графике 31 хорошо видно, что с использованием математического ожидания методы *LSS* и *LSA* работают так же хорошо, как и базовые.

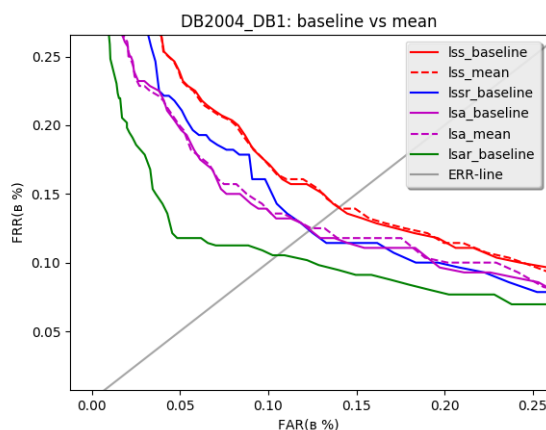


Рис. 31: DB2004_DB1

6.5.4. После фильтрации *Chameleons*

Для выборок *DB2000_DB1*, *DB2002_DB4* на графиках 32, 33 показано, что с использованием медианы *LSS*, *LSA*, *LSS-R* и *LSA-R* работают чуть лучше базовых.

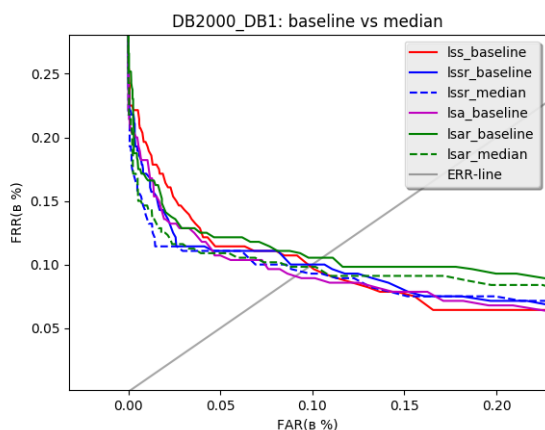


Рис. 32: DB2000_DB1

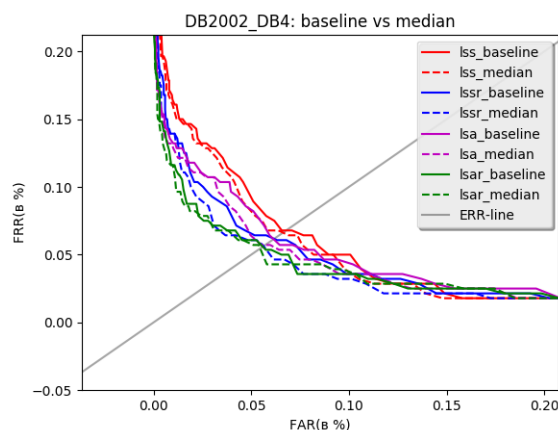


Рис. 33: DB2002_DB4

Для выборки *DB2000_DB4* на графиках 34, 35 показано, что версии методов *LSS*, *LSA*, *LSS-R* и *LSA-R* со своими статистиками работают чуть лучше базовых.

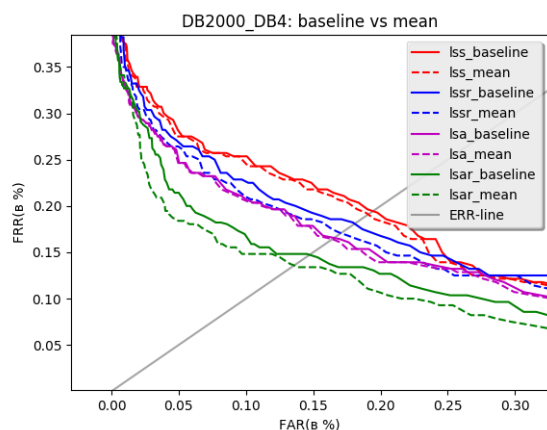


Рис. 34: DB2000_DB4

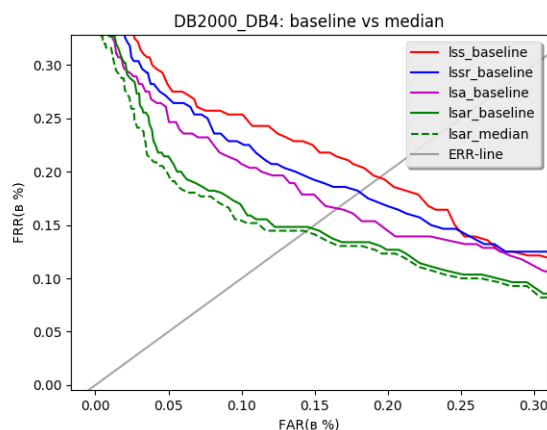


Рис. 35: DB2000_DB4

Для выборки *DB2004_DB3* на графиках 36, 37 показано, что версии методов *LSS-R* и *LSA-R* со любой статистикой работают лучше базовых.

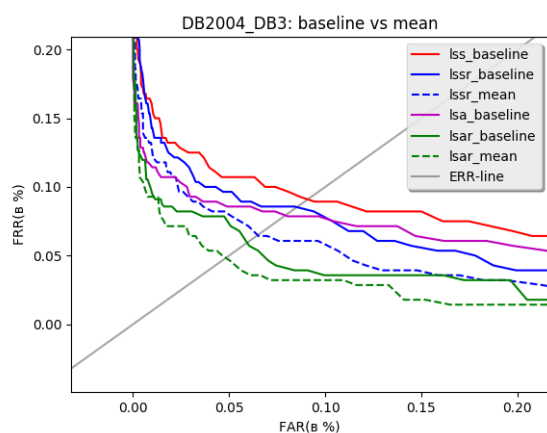


Рис. 36: DB2004_DB3

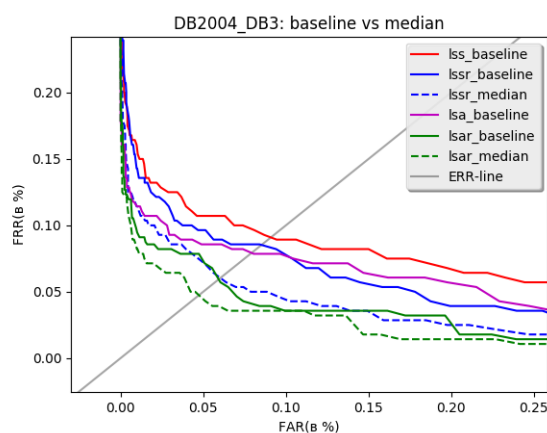


Рис. 37: DB2004_DB3

Для выборки *DB2004_DB4* на графиках 38, 39 показано, что методы *LSS*, *LSA*, *LSS-R* и *LSA-R* с медианой работают лучше базовых. Комбинация *LSA-R* с математическим ожиданием сопоставима с базовым методом.

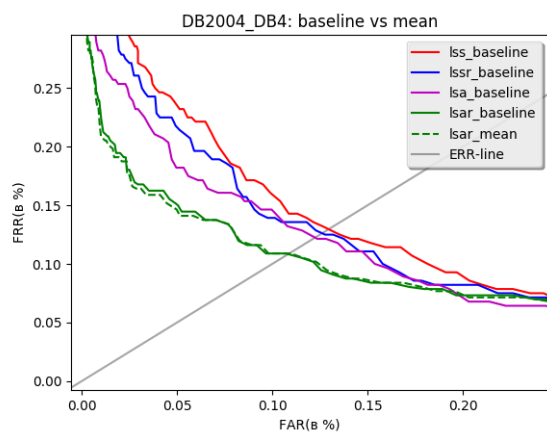


Рис. 38: DB2004_DB4

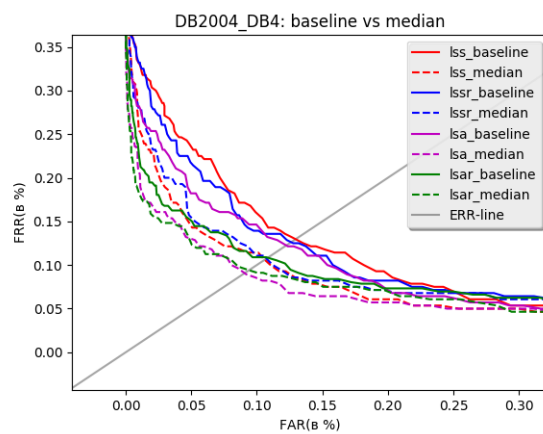


Рис. 39: DB2004_DB4

7. Выводы

В данной работе была поставлена задача выявления эффекта биометрического зверинца в алгоритмах локального сопоставления. Она была решена с помощью проведения эксперимента Ягера-Данстона для локальных структур шаблонов пользователей биометрической системы. После обработки 12 выборок, были приведены таблицы, в которых отражено наличие классов зверинца, что позволило сделать вывод о присутствии исследуемого эффекта на локальном уровне. Далее был предложен метод фильтрации локальных структур шаблонов, приносящих данный эффект. По результатам фильтрации были построены графики зависимости ошибок FAR/FRR , которые позволяют сделать следующие выводы:

- Эффект биометрического зверинца присутствует в алгоритмах локального сопоставления на локальном уровне.
- Расширение подхода Ягера-Данстона до локального уровня позволяет более детально проанализировать природу биометрического зверинца и задуматься о методах борьбы с ним.
- Алгоритм определения истинных и ложных сопоставлений локальных структур требует переосмысления, поскольку после его работы остаётся большое количество локальных структур, не имеющих *genuine* сопоставлений и попадающих в классы *Worms* и *Phantomes*.
- Идея фильтрации шаблонов от *плохих* локальных структур имеет право на жизнь. В полученных результатах по нескольким выборкам наблюдались отличные результаты для предложенных методов с разными статистиками. Определяющим фактором успеха для них являлось входное отфильтрованное множество данных.
- Фильтрация, применяемая к классам биометрического зверинца, использующего $G_{set} = \emptyset$, не помогает избавиться от эффекта и из-

бежать увеличение количества ошибок. Фильтрация же зверинца, в основе которого лежит $G_{set} \neq \emptyset$, имеет хороший потенциал.

- Применение различных статистик при проведении эксперимента Ягера-Данстона позволяет сделать два противоречивых утверждения. Первое, применение медианы чаще привносит классы биометрического зверинца, однако она в большинстве случаев показывает лучшие результаты по сравнению с математическим ожиданием. Второе, применение математического ожидания реже привносит классы биометрического зверинца, однако она в большинстве случаев показывает худшие результаты по сравнению с медианой.
- Из рассмотренных алгоритмов консолидации в большинстве случаев хорошо проявили себя те, которые используют этап релаксации [7]: *LSA-R*, *LSS-R*. Тем не менее, все типы алгоритмов приносят эффект биометрического зверинца на глобальный уровень.
- Эффект биометрического зверинца может быть использован в качестве критерия качества алгоритма сопоставления.
- Полученные результаты не предоставляют однозначный ответ о том, как бороться с эффектом биометрического зверинца, но дают возможности для более детального анализа этого эффекта как с точки зрения используемых алгоритмов, так и с точки зрения методом математической статистики.

8. Заключение

В данной работе была поставлена задача подробно изучить эффект биометрического зверинца в алгоритмах локального сопоставления. Он является одной из основных причин возникновения ошибок в биометрических системах. За основу был взят подход Ягера-Данстона, позволяющий выявить данный эффект для оценок сопоставления шаблонов пользователей биометрической системы, т.е. на глобальном уровне. Далее были выявлены ограничения подхода Ягера-Данстона (выявление эффекта для оценок сопоставления шаблонов, полученных алгоритмами локального сопоставления на второй фазе) и предложен новый метод - обнаружение классов зверинца для оценок сопоставления локальных структур шаблонов (полученных алгоритмами локального сопоставления на первой фазе). Основой для предложенного метода является алгоритм определения истинных и ложных сопоставлений локальных структур, результаты которого позволили выявить эффект биометрического зверинца на локальном уровне. После чего был предложен метод борьбы с этим эффектом при помощи фильтрации локальных структур шаблонов, принадлежащих классам зверинца. К фильтрованным шаблонам были применен алгоритм локального сопоставления и получены оценок сопоставления фильтрованных шаблонов. Далее был проведен подробный сравнительный анализ процентного содержания ошибок для выборок содержащих оригинальные и фильтрованные шаблоны. Таким образом, подробный анализ эффекта биометрического зверинца для локальных структур шаблонов позволил внести в процесс регистрации пользователей промежуточный этап - фильтрацию плохих структур, которые привносят данный эффект на локальный уровень.

Поставленные в работе задачи были решены, однако работа носит экспериментальный характер, поэтому на данный момент всё же остаётся несколько задач для дальнейшего исследования:

1. Модификация алгоритма параллельных линий.
2. Рассмотрение других алгоритмов локального сопоставления.

3. Более детально изучить алгоритмы консолидации.
4. Изменение критериев фильтрации или другой порядок фильтрации классов биометрического зверинца.

Список литературы

- [1] Algorithm to estimate biometric performance change over time / Norman Poh, Josef Kittler, Chi-Ho Chan, Medha Pandit. — Vol. 4. — 2015. — P. 236–245.
- [2] Araújo Rayron Victor Medeiros de. A probabilistic analysis of the biometrics menagerie existence: case study in fingerprint data. — 2016.
- [3] BMDT: An optimized method for Biometric Menagerie Detection / He Zheng, Liao Ni, Ran Xian et al. // Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on / IEEE. — 2015. — P. 1–8.
- [4] Biometric System Laboratory University of Bologna. Minutia Cylinder-Code SDK v2.0, 2015.
- [5] Biometric zoos: Theory and experimental evidence / Mohammad Nayeem Teli, J Ross Beveridge, P Jonathon Phillips et al. // Biometrics (IJCB), 2011 International Joint Conference on / IEEE. — 2011. — P. 1–8.
- [6] Biometrics Griaule. Fingerprint SDK. 2009.
- [7] Cappelli Raffaele, Ferrara Matteo, Maltoni Davide. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2010. — Vol. 32, no. 12. — P. 2128–2141.
- [8] Cappelli Raffaele, Ferrara Matteo, Maltoni Davide. Fingerprint indexing based on minutia cylinder-code // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2011. — Vol. 33, no. 5. — P. 1051–1057.
- [9] DeCann Brian, Ross Arun. Relating roc and cmc curves via the biometric menagerie // Biometrics: Theory, Applications and Systems

- (BTAS), 2013 IEEE Sixth International Conference on / IEEE. — 2013. — P. 1–8.
- [10] Deng Huimin, Huo Qiang. Minutiae matching based fingerprint verification using delaunay triangulation and aligned-edge-guided triangle matching // Audio-and Video-Based Biometric Person Authentication / Springer. — 2005. — P. 357–372.
 - [11] FVC2000: Fingerprint verification competition / Dario Maio, Davide Maltoni, Raffaele Cappelli et al. // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2002. — Vol. 24, no. 3. — P. 402–412.
 - [12] FVC2002: Second fingerprint verification competition / Dario Maio, Davide Maltoni, Raffaele Cappelli et al. // Pattern recognition, 2002. Proceedings. 16th international conference on / IEEE. — Vol. 3. — 2002. — P. 811–814.
 - [13] FVC2004: Third fingerprint verification competition / Dario Maio, Davide Maltoni, Raffaele Cappelli et al. // Biometric Authentication. — 2004. — P. 31–35.
 - [14] Feng Jianjiang. Combining minutiae descriptors for fingerprint matching // Pattern Recognition. — 2008. — Vol. 41, no. 1. — P. 342–352.
 - [15] Ferrara Matteo, Maltoni Davide, Cappelli Raffaele. Noninvertible minutia cylinder-code representation // IEEE Transactions on Information Forensics and Security. — 2012. — Vol. 7, no. 6. — P. 1727–1737.
 - [16] Group-specific score normalization for biometric systems / Norman Poh, Josef Kittler, Ajita Rattani, Massimo Tistarelli // Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on / IEEE. — 2010. — P. 38–45.

- [17] Handbook of fingerprint recognition / Davide Maltoni, Dario Maio, Anil Jain, Salil Prabhakar. — 2009.
- [18] Houmani Nesma, Garcia-Salicetti Sonia. On Hunting Animals of the Biometric Menagerie for Online Signature // PloS one. — 2016. — Vol. 11, no. 4. — P. e0151691.
- [19] Improving biometric verification systems by fusing Z-norm and F-norm / Messaoud Bengherabi, Farid Harizi, Norman Poh et al. // Chinese Conference on Biometric Recognition / Springer. — 2012. — P. 278–287.
- [20] Jain Anil K. Biometric authentication // Scholarpedia. — 2008. — Vol. 3, no. 6. — P. 3716.
- [21] Klare Brendan F, Jain Anil K. Face recognition: Impostor-based measures of uniqueness and quality // Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on / IEEE. — 2012. — P. 237–244.
- [22] M. Ferrara D. Maltoni, Cappelli R. A Two-Factor Protection Scheme for MCC Fingerprint Templates // International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany. — 2014.
- [23] Minutiae-based Feng algorithm. Consolidation algorithms. — URL: https://github.com/Montura/biometric_zoo/blob/master/FengMatching/feng_algorithm.py.
- [24] Online hand signature verification: a review / S Sayeed, A Samraj, R Besar, J Hossen // Journal of Applied Sciences. — 2010. — Vol. 10. — P. 1632–1643.
- [25] Paone Jeffrey Richard. Liberating the biometric menagerie through score normalization improvements : Ph.D. thesis / Jeffrey Richard Paone ; Citeseer. — 2013.

- [26] Poh Norman, Kittler Josef. A methodology for separating sheep from goats for controlled enrollment and multimodal fusion // Biometrics Symposium, 2008. BSYM'08 / IEEE. — 2008. — P. 17–22.
- [27] Poh Norman, Kittler Josef. A biometric menagerie index for characterising template/model-specific variation // Advances in biometrics. — 2009. — P. 816–827.
- [28] Popescu-Bodorin Nicolaie, Balas Valentina Emilia, Motoc Iulia Maria. The Biometric Menagerie—A Fuzzy and Inconsistent Concept // Soft Computing Applications. — 2013. — P. 27–43.
- [29] Ross Arun, Rattani Ajita, Tistarelli Massimo. Exploiting the “doddington zoo” effect in biometric fusion // Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on / IEEE. — 2009. — P. 1–7.
- [30] Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation : Rep. / DTIC Document ; Executor: George Doddington, Walter Liggett, Alvin Martin et al. : 1998.
- [31] User's guide to NIST biometric image software (NBIS) / Craig I Watson, Michael D Garris, Elham Tabassi et al. — 2007.
- [32] Yager Neil, Dunstone Ted. Worms, chameleons, phantoms and doves: New additions to the biometric menagerie // Automatic Identification Advanced Technologies, 2007 IEEE Workshop on / IEEE. — 2007. — P. 1–6.
- [33] Yager Neil, Dunstone Ted. The biometric menagerie // IEEE transactions on pattern analysis and machine intelligence. — 2010. — Vol. 32, no. 2. — P. 220–230.
- [34] A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation /

Daniel Peralta, Mikel Galar, Isaac Triguero et al. // Information Sciences. — 2015. — Vol. 315. — P. 67–87.

- [35] A user-specific and selective multimodal biometric fusion strategy by ranking subjects / Norman Poh, Arun Ross, Weifeng Lee, Josef Kittler // Pattern Recognition. — 2013. — Vol. 46, no. 12. — P. 3341–3357.