

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт: ИРЭ

Кафедра: Радиотехнических систем

Специальность:

11.04.01 – Радиотехника

ОТЧЕТ по практике

Наименование
практики:

Производственная практика: научно-
исследовательская работа

СТУДЕНТ

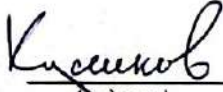
 / Масалкова Н.В. /
(подпись) (Фамилия и инициалы)


Группа ЭР-12м-20
(номер учебной группы)

**ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО
ПРАКТИКЕ**

ОТЛИЧНО

(отлично, хорошо, удовлетворительно, неудовлетворительно,
зачтено, не зачтено)

 / Куликов Р.С. /
(подпись) (Фамилия и инициалы члена комиссии)

 / . . . /
(подпись) (Фамилия и инициалы члена комиссии)

Москва
2021

Оглавление

Введение	3
1. Этапы обработки радиолокационной информации	5
2. Постановка задачи.....	7
3. Кластерный анализ.....	8
4. Алгоритмы кластеризации	12
4-1. Иерархическая кластеризация.....	12
4-2. Метод k- средних значений	13
4-3. Основанная на плотности пространственная кластеризация с шумом (DBSCAN)	14
4-4. Смешанная гауссовская модель	15
4-5. Спектральная кластеризация	16
5. Анализ радиолокационных импульсов с помощью алгоритмов кластеризации	19
5.1. Визуальный анализ кадра	19
5.2. Применение алгоритмов кластеризации к реальным измерениям.....	22
5.2.1. Иерархическая кластеризация.	22
5.2.2. Метод k-средних значений.	25
5.2.3. Основанная на плотности пространственная кластеризация с шумом (DBSCAN).	27
5.2.4. Смешанная гауссовская кластеризация.	31
5.2.5. Спектральная кластеризация.	35
Выводы	38
Список литературы.....	39

Введение

В современном мире от устройств радиолокации требуется, чтобы они работали в условиях, для которых характерны высокая скорость изменения внешней обстановки и большое количество объектов, подлежащих обнаружению [1]. Обработка столь большого количества информации в требуемые сжатые сроки не может быть эффективно осуществлена человеком-оператором [3]. Поэтому перед радиолокационными станциями (РЛС), являющимися основным источником информации, ставится задача автоматизации процессов обработки информации. Актуальность решения данной задачи в том, что в отличие от применявшихся ранее систем обработки, в которых конечное решение принимал человек-оператор, в этих образцах конечное решение принимается определенными алгоритмами [2].

Для разработки подобных систем потребовалось создание теории автоматической обработки радиолокационной информации (РЛИ). Существуют методы, способные осуществлять автоматическую обработку РЛИ [3-8]. Однако большинство данных методов базируются на положениях классической теории радиолокации, которая опирается на предположения, многие из которых не выполняются на практике. Необходимость использования этих предположений обусловлена трудностью формализации и математического описания всевозможных воздействий помех и других факторов, отсутствием единой методологии оценки систем обработки в различных воздушных и помеховых условиях. И оказалось, что при создании образцов РЛС с автоматической обработкой информации, классическая теория часто оказывается неприменима [8]. До сих пор нет универсальной теории для создания автоматической обработки РЛИ.

Действительно, современные радиолокационные средства в достаточно простых условиях успешно справляются со своими задачами, а в сложных условиях (нестационарных и негауссовых помех) их эффективность может резко снижаться. Снижается достоверность выдаваемой РЛС информации за счет появления большого числа ложных отметок и появления ложных траекторий. Их число может намного превышать число целей в зоне обзора, а использование недостоверной информации ведет к снижению эффективности, например, радиолокационных стрельбовых средств в несколько раз [6].

Непредсказуемость внешней обстановки и высокая динамика её изменения создают значительные трудности для формулировки алгоритмов и для обеспечения высокого качества их работы. Появляются и новые задачи, например, распознавание, кластеризация, классификация и анализ ситуаций. И классических методов решения таких задач нет [10].

В настоящее время актуальными являются исследования по повышению качества обработки информации за счет использования, например, статистики и машинного обучения

на определенных подсистемах обработки в зависимости от физической структуры входных и выходных сигналов, места их применения, характера изменения внешних условий. В этих условиях встает задача разделения подобных между собой этапов обработки информации в разнородных системах, эффективность которых могла бы быть резко повышена за счет применения алгоритмов машинного обучения. В данной работе будет рассмотрен один из этапов обработки информации с применением к нему алгоритмов машинного обучения.

1. Этапы обработки радиолокационной информации

Обработка РЛИ представляет собой наиболее важный комплекс задач РЛС. Назначение обработки подготовить к выдаче в требуемом виде полную, достоверную и современную информацию для потребителя о состоянии воздушной обстановки, появлении и местоположении воздушных средств объектов, параметрах их движения, возможных вариантах развития динамики изменения воздушно-помеховой обстановки. В ходе обработки РЛИ решается целый ряд задач, которые можно сгруппировать по близости объектов и применяемых методов в основные этапы. К основным этапам обработки радиолокационной информации можно отнести [8-9]:

- Формирование диаграммы направленности антенны для фазирования антенных решеток;
- Первичная (спектральная) обработка;
- Обнаружение радиолокационных объектов
- Вторичная (траекторная) обработка
- Распознавание объектов и ситуаций
- Анализ обстановки и адаптация

Решаемые на этих этапах задачи служат выполнению главной цели, для которой предназначен радиолокатор – выдаче полной, достоверной и своевременной информации потребителю о воздушных целях.

Большого внимания заслуживает такой этап обработки, как обнаружение отметок от целей и их кластеризация для дальнейшей вторичной (траекторной) обработки целей. Обнаружение радиолокационных целей обычно включает в себя два подэтапа: обнаружение импульсов, отраженных от целей, и их последующего пакетирования в ходе которого объединяются решения, принятые по каждому отдельному импульсу. Обнаружение целей должно проводиться автоматически, сохраняя высокие характеристики при работе в различных тактических условиях, при воздействии разнотипных помех, в том числе нестационарных, т.е. в существенно неопределенных условиях.

Рассмотрим некоторые методы, которые применяются для автоматического обнаружения воздушных целей в РЛС, а именно использование адаптивного порога обнаружения [11], использование фазовых соотношений в многобазовых сигналах (знаковые методы) и ранговые методы [4].

Данным методам присущи крупные недостатки, которые определяют практическую полную непригодность для РЛС, ведущих работу в реальных условиях. Рассмотрим подробнее: в реальных условиях при воздействии на РЛС помех нестационарного типа, а

также хотя и стационарных, но негауссовых помех таких, как отражения от подвижных и местных предметов, пассивные помехи, уровень ложных тревог может возрасти на 2...3 порядка, что недопустимо. Кроме того, при наличии в пределах опорной выборки сигналов, отраженных от соседних с обнаруживаемой целью, что характерно для обнаружения целей в группах, энергетические потери могут составлять 10...15 дБ и более [11]. Исключение составляют ранговые методы, обеспечивающие практически полную стабилизацию уровня ложных тревог, но эти методы при малой пачке имеют очень высокие потери даже при обнаружении одиночных целей, а при обнаружении групп целей эти потери катастрофически возрастают.

Для синтеза эффективных методов автоматического обнаружения необходимо привлечь статическую теорию различения гипотез. Синтезируемые методы должны эффективно работать в условиях воздействия основных помех. При этом необходимо определить совокупность параметров помех и обеспечить устойчивость и совокупность параметров.

Пакетирование импульсов проводится для формирования отметок от целей, на основе полученных нескольких импульсов от цели. Эта задача аналогична задаче кластеризации: необходимо из множества одиночных отметок выбрать несколько центров группирования, которые соответствуют обнаруживаемым целям. В итоге проведение группирования отметок в кластеры позволит улучшить алгоритмы вторичной обработки сигналов, а именно сопровождение траекторий групп целей. Группирование позволит сократить количество ложных траекторий примерно на порядок.

2. Постановка задачи

Глобальная цель данной работы является возможность повышения эффективности обработки информации в системах пассивной радиолокации за счет отождествления сигналов с целями путем обработки данных сигналов алгоритмами кластеризации.

Чтобы приблизиться к решению данной цели требуется рассмотреть существующие алгоритмы кластеризации, сравнить алгоритмы кластеризации, описать принципы работы алгоритмов кластеризации, смоделировать данные алгоритмы кластеризации и с их помощью проанализировать радиолокационные данные. Моделирование предлагается осуществлять в среде Matlab с помощью соответствующих встроенных функций.

Задача работы – ознакомиться с работой встроенных функций статистики и машинного обучения среды Matlab, проанализировать данные радиолокационных импульсов, применить к ним алгоритмы кластеризации, и осуществить поиск характерных закономерностей в этих данных.

3. Кластерный анализ

Чтобы решить поставленную задачу для начала требуется дать определение понятию кластеризация. *Кластеризация или кластерный анализ* — это задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы оказываются объекты с подобными параметрами, а объекты разных групп имеют максимальные отличия друг от друга.

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. Перед запуском алгоритма чаще всего требуется *нормализовать* данные или по-другому *стандартизировать* данные. Нормализация предполагает замену номинальных признаков так, чтобы каждый из них находился в диапазоне от нуля до единицы. Стандартизация же подразумевает предобработку данных, после которой каждый признак имеет среднее значение равное нулю и среднеквадратичное отклонение (СКО) равное единице. В итоге нормализация или стандартизация дает равную важность переменным во время кластеризации.
3. Вычисление значений *меры сходства* между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

Кроме того, кластерный анализ является общим методом машинного обучения без учителя для поиска шаблонов в наборе данных. Обучение без учителя используется, чтобы формировать группы из наборов данных, которые обычно заранее не размечены. Например, используются кластерный анализ для исследовательского анализа данных, чтобы найти скрытые шаблоны или группы объектов в немеченных входных данных.

Кластерный анализ создает группы или по-другому *кластеры*, данных. Объекты, которые принадлежат тому же кластеру, похожи друг на друга и отличны от объектов, принадлежащих другим кластерам. Чтобы определить сколько таких "подобных" и "отличных", используют меру по несходству или по-другому метрику расстояний. Кроме того, в зависимости от имеющегося набора данных требуется использовать масштабирование (или стандартизацию) переменных, чтобы дать равную оценку во время кластеризации.

Категоризация точек данных на основе их расстояния до соседних точек в этом же наборе данных является простым и эффективным способом кластеризации точек.

Решение данной задачи будет проводиться в среде Matlab, поэтому следует привести описание функций, с которыми будет проводиться работа.

В среде Matlab попарное расстояние между точками набора данных находится с помощью функции `pdist2`. Синтаксис данной функции:

`D = pdist2(X,Y,Distance)` – возвращает расстояние между каждой парой наблюдений в X и Y с используемой метрикой, заданной в `Distance`.

`D = pdist2(X,Y,Distance,DistParameter)` – возвращает расстояние с помощью метрики, заданной `Distance` и `DistParameter`. Можно задать `DistParameter` только, когда `Distance` 'seuclidean', 'minkowski', или 'mahalanobis'.

`D = pdist2(____,Name,Value)` – задает дополнительную опцию с помощью одного из аргументов пары "имя-значение" 'Smallest' или 'Largest' в дополнение к любому из аргументов в предыдущих синтаксисах.

Например,

`D = pdist2(X,Y,Distance,'Smallest',K)` – вычисляет расстояние с помощью метрики, заданной `Distance` и возвращает K наименьшие попарные расстояния до наблюдений в X для каждого наблюдения в Y в порядке возрастания.

`D = pdist2(X,Y,Distance,'Largest',K)` вычисляет расстояние с помощью метрики, заданной `Distance` и `DistParameter` и возвращает K самые большие попарные расстояния в порядке убывания.

`[D,I] = pdist2(____,Name,Value)` – также возвращает матричный I . Матричный I содержит индексы наблюдений в X соответствие расстояниям в D .

Требуется определить такое понятие как метрика расстояний. Метрика расстояний является функцией, которая задает расстояние между двумя наблюдениями. В среде Matlab функция попарного-расстояния `pdist2` поддерживает следующие метрики расстояния: Евклидово расстояние, квадратичное Евклидово расстояние, стандартизированное Евклидово расстояние, расстояние Махаланобиса, расстояние городского квартала (манхэттенское расстояние), расстояние Минковского, расстояние Чебышева, расстояние косинуса, корреляционное расстояние, расстояние Хемминга, расстояние Жаккара и расстояние Спирмена. Описание данных метрик сведено в Таблицу 1 и продолжение в Таблице 2.

Таблица 1. Метрики расстояний.

Значение	Формула	Описание
'euclidean'	$\rho(x, x') = \sqrt{\sum_{i=1}^N (x_i - x'_i)^2}$ <p>Где x и x' точки в N-мерном пространстве</p>	Значение по умолчанию. Наиболее распространенная функция для определения расстояния. Представляет собой геометрическое расстояние в многомерном пространстве между двумя точками. Евклидово расстояние является особым случаем расстояния Минковского, где $p = 2$
'sqeuclidean'	$\rho(x, x') = \sum_{i=1}^N (x_i - x'_i)^2$	Применяется для придания большего вес более отдаленных друг от друга объектов.
'seuclidean'	$\rho(x, x') = \sqrt{\sum_{i=1}^N S_i (x_i - x'_i)^2}$	Каждое координатное различие между наблюдениями масштабируется путем деления на соответствующий элемент стандартного отклонения, $S = \text{std}(x, 'omitnan')$
'mahalanobis'	$\rho(x, x') = (X_i - X'_i)C^{-1}(X_i - X'_i)^T$	Предполагает, что точки множества эллипсоидально (частный случай – сферически) распределены вокруг центра масс – расстояние безразмерно и масштабно-инвариантно. Ковариация – это численное выражение свойства ковариантности двух признаков точек. Свойство ковариантности означает, что признаки имеют тенденцию изменяться совместно. C – ковариационная матрица
'cityblock'	$\rho(x, x') = \sum_{i=1}^N x_i - x'_i $	Это расстояние является средним разностей по координатам. Для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т.к. они не возводятся в квадрат). Расстояние городского квадрата является особым случаем расстояния Минковского, где $p = 1$

'minkowski'	$\rho(x, x') = \sqrt[p]{\sum_{i=1}^N x_i - x_i' ^p}$	Параметрическая метрика на евклидовом пространстве, которую можно рассматривать как обобщение евклидова расстояния и расстояния городских кварталов.
'chebychev'	$\rho(x, x') = \max(\sum_{i=1}^N x_i - x_i')$	Это расстояние может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние городского квадрата является особым случаем расстояния Минковского, где $p = \infty$
'cosine'	$\rho(x, x') = \frac{\sum_{i=1}^N x_i \cdot x_i'}{\sqrt{\sum_{i=1}^N (x_i)^2} \cdot \sqrt{\sum_{i=1}^N (x_i')^2}}$	Метод сходства между двумя ненулевыми векторами пространства данных. Определен как косинус угла между ними, которых также совпадает с внутренним произведением тех же векторов, нормализованных к обоим, имеющим длину 1.

Таблица 2. Продолжение таблицы 1. Метрики расстояний.

Функция	Описание
'correlation'	Ещё называют скорректированным косинусным сходством. Использование данного метода компенсирует недостаток предыдущего метода путем вычитания среднего значения соответствующе точки набора.
'hamming'	Число позиций, в которых соответствующие точки двух наборов одинаковой длины различны.
'jaccard'	Используется для калибрования сходства и разнообразия из образцов наборов. Основан на пересечении и объединении данных
'spearman'	Является методом ранговой корреляции и используется для выявления и оценки тесноты связи между двумя рядами сопоставляемых количественных показателей.

4. Алгоритмы кластеризации

4-1. Иерархическая кластеризация

Иерархическая кластеризация группирует данные в различных масштабах, создавая дерево кластеров или дендрограмму (*dendrogram*). Дерево не является ни одним из наборов кластеров, а является скорее многоуровневой иерархией, в которой кластеры на одном уровне объединяются в кластеры на следующем. Эта многоуровневая иерархия позволяет выбирать требуемые уровень или шкалу кластеризации, которые наиболее подходят для решения поставленной задачи. Иерархическая кластеризация присваивает каждую точку данных набора к кластеру.

Приведем описание встроенных функций Matlab. Использование `clusterdata` позволяет выполнить иерархическую кластеризацию на неразмеченных входных данных. Функция `clusterdata` соединяет в себе следующие функции: `pdist`, `linkage`, и `cluster`, которые можно использовать отдельно для более детального анализа иерархической кластеризации. С помощью функции `dendrogram` можно построить график кластерного дерева.

Описание алгоритма

Чтобы реализовать иерархическую кластеризацию на наборе данных с помощью функций статистики и машинного обучения среды Matlab Statistics and Machine Learning Toolbox, требуется выполнить следующие шаги:

1. **Найти подобие или несходство между каждой парой объектов во входном наборе данных.** На этом шаге вычисляется расстояние между точками данных с помощью `pdist` функция. Данная функция поддерживает много различных способов нахождения расстояния, а именно, можно изменять метрику расстояния, для получения лучшего результата.
2. **Построить иерархическое кластерное дерево.** На этом шаге требуется соединить пары точек, которые находятся в непосредственной близости с помощью функции `linkage`. Эта функция использует информацию о расстоянии, сгенерированную на шаге, чтобы определить близость объектов друг другу. Когда объекты соединяются в бинарные кластеры, недавно сформированные кластеры группируются в большие кластеры, пока иерархическое дерево не сформируется.
3. **Определить, где сократить иерархическое дерево в кластеры.** На этом шаге используется функция `cluster`. Чтобы определить количество кластеров нужно обрезать иерархическое дерево по требуемому уровню, тогда нижняя часть

иерархического дерева и все точки ниже этого уровня присваиваются одному кластеру. Данный метод позволяет создать кластеры путем обнаружения естественных групп в иерархическом дереве или путем сокращения иерархического дерева по произвольно выбранному уровню.

4-2. Метод k - средних значений

Данный метод кластеризации считается наиболее простым, но в то же время ему не хватает точности. Суть данного алгоритма в следующем: метод разбивает множество элементов векторного пространства данных на заранее известное число кластеров k . Т.е. для данного алгоритма число кластеров k является входным аргументом. Работа алгоритма заключается в том, что алгоритм стремится минимизировать среднеквадратическое отклонение (или другую выбранную меру расстояний) для точек каждого кластера. На каждой итерации вычисляется заново центр масс для каждого кластера, полученного на предыдущей итерации, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения внутрикластерного расстояния. Происходит это за конечное число итераций, так как количество возможных разбиений конечного множества также конечно, а на каждом шаге суммарное квадратичное отклонение уменьшается, поэтому заикливание невозможно.

Функции, используемые в среде Matlab: `kmeans` – k - кластеризация. Есть ещё виды данной кластеризации, а именно: `kmedoids` – k - *medoids* кластеризация; `mahal` – расстояния Махаланобиса. Например, расстояние Махаланобиса учитывает корреляцию между переменными и инвариантно к масштабу.

В отличие от иерархической кластеризации, этот метод работает с фактическими наблюдениями (т.е. с входными данными напрямую, а не с мерой по несходству) и создается один уровень кластеров. Поэтому кластеризация k - средних значений чаще подходит для больших объемов данных, чем иерархическая кластеризация.

Функция `kmeans` получает данные и возвращает индекс кластера, которому присваивает каждое наблюдение, находит кластеры, в которых объекты максимально близки друг к другу по параметрам, и максимально далеки от объектов в других кластерах.

Каждый кластер в методе k - средних значений состоит из объектов члена и центроида (или по-обычному центра). В каждом кластере, `kmeans` минимизирует сумму расстояний между центроидом и всеми объектами члена кластера.

Можно варьировать аргументы пары "имя-значение" доступных для `kmeans`; например, можно задать начальные значения кластерных центроидов и максимальное количество итераций для алгоритма.

4-3. Основанная на плотности пространственная кластеризация с шумом (DBSCAN)

DBSCAN (*Density-based spatial clustering of applications with noise*, плотностной алгоритм пространственной кластеризации с присутствием шума) является основанным на плотности алгоритмом кластеризации, который идентифицирует кластеры произвольной формы и выбросы (шум) в данных. Во время кластеризации алгоритм DBSCAN определяет точки, которые не принадлежат ни одному из кластеров и относит к шуму. В отличие от метода *k*- средних значений, DBSCAN не требует предварительных знаний количества кластеров.

Функция в среде Matlab. `dbscan` - основанная на плотности пространственная кластеризация приложений с шумом (DBSCAN). Точка набора данных, которая будет присвоена кластеру, лежит условие, что окружность вокруг этой точки радиуса `epsilon` (`epsilon`) содержит, по крайней мере, минимальное количество соседей (`minpts`). Или, точка может лечь в окружении `epsilon` другой точки, которая удовлетворяет условиям `epsilon` и `minpts`.

Алгоритм DBSCAN идентифицирует три вида точек:

1. Базовая точка — точка в кластере, которая имеет, по крайней мере, `minpts` соседей в `epsilon` окрестности.
2. Пограничная точка — точка в кластере, которая имеет меньше, чем `minpts` соседей в `epsilon` окрестности.
3. Шумовая точка — выброс, который не принадлежит никакому кластеру.

Описание алгоритма

Для заданных значений радиуса `epsilon` и минимального количества соседей `minpts` требуемых для базовой точки, функция `dbscan` реализует метод DBSCAN можно следующим образом:

1. Алгоритм начинается с произвольной точки, которая ещё не была просмотрена.
2. Выбирается `epsilon` окрестность этой точки и, если она содержит `minpts` точек, то начинается формирование кластера, в противном случае точка помечается как шум. Эта точка может быть позже найдена в `epsilon` окрестности другой точки и, таким образом, может стать частью кластера.

3. Если точка найдена как основная точка, то точки в ϵ -окрестности также являются частью этого кластера. Таким образом, все точки, найденные в ϵ окрестность, добавляются вместе с их собственной ϵ окрестностью, если они также являются основными точками.
4. Вышеописанный процесс продолжается, пока не будет найден связанный по плотности кластер.
5. Процесс возобновляется. Выбирается и обрабатывается новая не рассмотренная точка, что ведёт к обнаружению следующего кластера или шума.

4-4. Смешанная гауссовская модель

Метод смешанной гауссовской модели (GMM) формирует кластеры по принципу суперпозиции многомерных гауссовских законов распределения плотности вероятности. Для каждого наблюдения данный метод определяет апостериорные плотности вероятности, которые указывают, что наблюдение имеет некоторую вероятность принадлежности к каждому кластеру.

По GMM можно выполнить *hard-кластеризацию* (жесткую кластеризацию), которая гарантирует высокую точность группировки, где все запросы привязываются друг к другу и в один кластер попадают только на сто процентов совместимые запросы, путем выбора компонента, который максимизирует апостериорную вероятность. *Hard-кластеризация* определяет точку из данных только в один кластер. И в итоге кластер образуется там, где существует наибольшая апостериорная вероятность.

Кроме *hard-кластеризации* можно использовать смешанную гауссовскую модель *soft-кластеризации* (нечетной или мягкой кластеризации). При данном методе присваивается вероятность попадания точки данных для каждого кластера, и эта вероятность указывает на силу связи точки данных с кластером. В отличие от методов *hard-кластеризации*, методы *soft-кластеризации* являются гибкими, поскольку они могут назначить точку данных нескольким кластерам.

GMM может быть более соответствующим методом, чем метод k - средних значений, когда кластеры имеют различные размеры и различные структуры корреляции в них.

Описание алгоритма

Смешанные гауссовские модели (GMMs) присваивают каждое наблюдение кластеру путем максимизации апостериорной вероятности, что точка данных принадлежит своему присвоенному кластеру. Создается объект GMM путем подбора модели к данным (`fitgmdist`) или настройкой значений параметров (`gmdistribution`). Используются следующие функции в среде Matlab.

`GMMModel = fitgmdist(X, k)` возвращает Гауссову модель распределения смеси с k компонентами адаптированными к данным X .

`GMMModel = fitgmdist(X, k, Name, Value)` возвращает Гауссову модель распределения смеси с дополнительными опциями, заданными парой аргументов `Name, Value`.

`gmdistribution` объектно-ориентированная память Гауссово распределение смеси, которая является многомерным распределением, которое состоит из многомерных компонентов распределения Гаусса. Каждый компонент задан его средним значением и ковариацией. Смесь задана вектором из смешивания пропорций, где каждая пропорция смешивания представляет часть данных, описанных соответствующим компонентом.

`gm=gmdistribution(mu, sigma)` создает `gmdistribution` объект модели с помощью заданных средних значений `mu` и ковариации (СКО) `sigma` с равными пропорциями смешивания.

`gm=gmdistribution(mu, sigma, p)` задает смесительные пропорции многомерных компонентов Распределения Гаусса.

Затем используются объектные функции, чтобы выполнить кластерный анализ (`cluster`, `posterior`, `mahal`), оценивается модель (`cdf`, `pdf`), и генерируются случайные варьируемые величины (`random`).

4-5. Спектральная кластеризация

Спектральная кластеризация использует спектр собственных значений матрицы сходства данных для осуществления снижения размерности перед кластеризацией в пространство меньших размерностей. Матрица сходства подается в качестве входа и состоит из количественных оценок относительно схожести каждой пары точек в данных. В низкой размерности данные в кластерах разделяются более широко, позволяя использовать далее алгоритмы кластеризации, например, метод k -средних значений. Низкая размерность данных основана на значимых собственных векторах матрицы Лапласа. Данная матрица является одним из способов представить график подобия, который отображает локальные отношения между точками данных, как неориентированный граф.

Использование функции `Matlab spectralcluster` позволяет выполнить спектральную кластеризацию на матрице входных данных или на матрице подобия. Функция `spectralcluster` требует изначально заданного числа кластеров k .

Однако можно проверить, что оценка для числа кластеров k правильна при помощи одного из следующих методов: посчитать количество нулевых собственных значений

Матрицы Лапласа. Кратность нулевых собственных значений является индикатором количества кластеров в исходных данных.

Описание алгоритма

Спектральная кластеризация является основанной на графике алгоритмом для нахождения k кластеров произвольной формы в данных.

С помощью функции `spectralcluster` кластеризация данных происходит следующим образом:

1. Для каждой точки данных в X , задается локальное окружение с помощью метода определения радиуса `epsilon` или с помощью метода самых близких соседей `knn`. Как задано в `SimilarityGraph` аргументом является пара "имя-значение". Затем находятся попарные расстояния $Dist_{ij}$ для всех точек i и j в этом окружении.
2. Преобразуются расстояния до мер по подобию с помощью преобразования ядра $S_{ij} = \exp\left(-\left(\frac{Dist_{ij}}{\sigma}\right)^2\right)$. Матрица S является матрицей подобию, и σ является масштабным коэффициентом для ядра. Для этого используется функция `KernelScale` аргументом является пара "имя-значение".
3. Вычисляется ненормированная Матрица Лапласа L , нормированная Матрица Лапласа случайного обхода L_{rw} или нормированная симметричная Матрица Лапласа L_s , в зависимости от значения `LaplacianNormalization` аргументом является пара "имя-значение".
4. Создается матрица $V \in \mathbb{R}^{n \times k}$, содержащая столбцы v_1, \dots, v_k , где столбцы являются собственными векторами k , которые соответствуют k самые маленькие собственные значения Матрицы Лапласа. При обработке каждой строки V необходимо выполнить кластеризацию точки n , например, с помощью метода k -средних значений, `ClusterMethod` аргументом является пара "имя-значение".
5. Присваиваются исходные точки в наборе X к тем же кластерам, как и их соответствующие строки в V .

В данном разделе были рассмотрены основные алгоритмы кластеризации, которые есть в разделе статистики и машинного обучения в среде Matlab. Сведем характеристики данных алгоритмов в общую Таблицу 3.

Таблица 3. Характеристики основных методов кластеризации.

Алгоритм кластеризации	Описание алгоритма кластеризации	Входные данные	Требование конкретного количества кластеров
Иерархическая кластеризация	Расстояние между объектами	Попарные расстояния между наблюдениями	Нет
К-средних	Расстояние между объектами и центроидами	Фактические наблюдения	Да
Смешанные гауссовские модель	Смесь нормальных распределений	Фактические наблюдения	Да
Спектральная кластеризация	График, представляющий связи между точками данных	Фактические наблюдения или матрица подобия	Да, но алгоритм также обеспечивает способ оценить количество кластеров
DBSCAN	Плотность областей в данных	Физические наблюдения или попарные расстояния между наблюдениями	Нет

5. Анализ радиолокационных импульсов с помощью алгоритмов кластеризации

В данной задаче алгоритмы кластеризации применяются к реальным радиолокационным сигналам, полученным с многопозиционного пассивного радиолокационного комплекса. Данный комплекс представляет собой систему, состоящую из четырех постов (рисунок 1), на которых осуществляется первичная обработка сигналов.

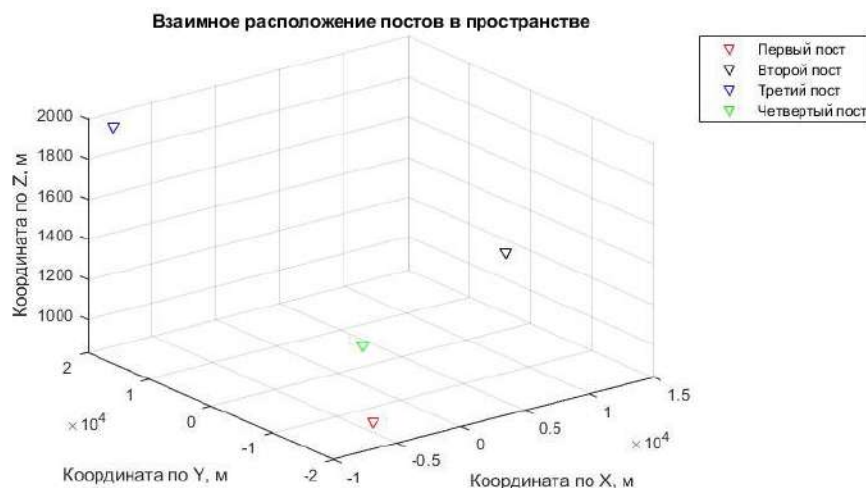


Рисунок 1. Радиолокационный комплекс, состоящий из четырех радиолокационных постов.

Алгоритмы кластеризации будут применяться для вторичной обработки. Требуется определить вид данных, которые будут использоваться для кластеризации. Информация после первичной обработки предоставляется в виде набора импульсов, пришедших на четыре поста, которые объединены в кадр длиной 12 мс. Каждый импульс кадра имеет следующий набор параметров: время прихода импульса относительно кадра uT , сек, абсолютное время прихода импульса T , сек в шкале времени системы, амплитуда Amp , частота $freq$, Гц, длительность dur , нс, период $period$, сек, азимут или направление зондирования антенны az , град. Азимут является детерминированным параметром и принимает значения в следующем интервале: $[22.5, 67.5, 112.5, 157.5, 202.5, 247.5, 292.5, 337.5]$. Каждый кадр формируется при одном угле зондирования. В итоге полный поворот антенны, а значит зондирование на 360 градусов, осуществляется за 96 мс.

5.1. Визуальный анализ кадра

Для анализа были взяты реальные измерения, которые накапливались на постах в течение 23 минут.

Построим зависимости значения частоты и значения длительности импульса от абсолютного времени прихода импульса по постам. Будем проверять алгоритмы кластеризации именно по этим параметрам: по частоте и по длительности импульса, так как это наиболее значимые параметры сигнала.

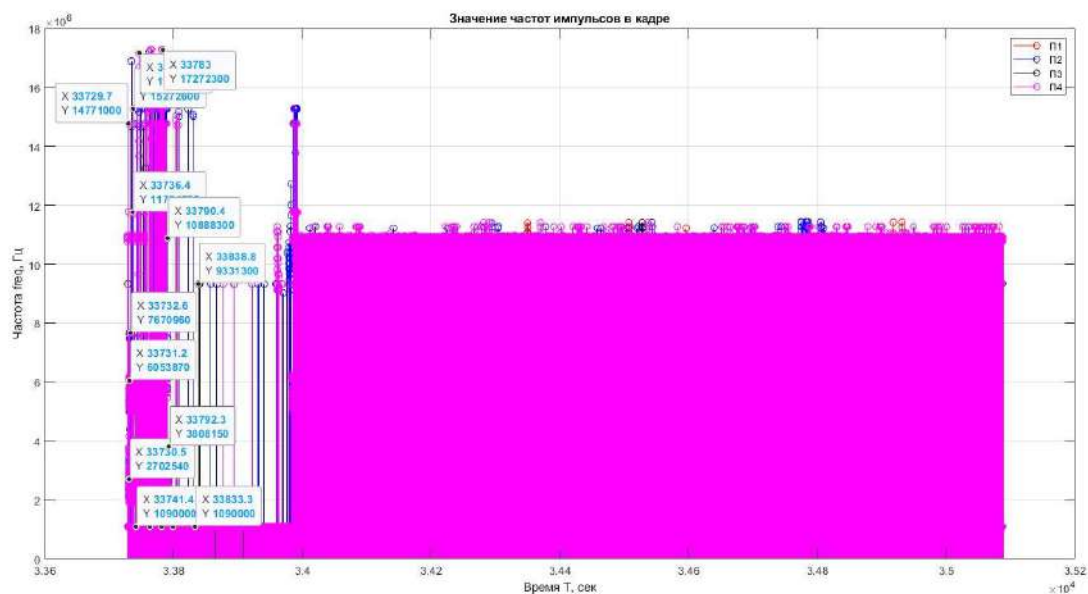


Рисунок 2. Зависимость частоты импульса от абсолютного времени прихода импульса. Время накопления 23 минуты. Выделен участок накопления до 3.5 минут.

Анализируя рисунок 2 можно сделать следующие выводы. Явно видна частота по уровню ровно 1,090 МГц, так как частота не флуктуирует, делаем вывод, что данные сигналы приходят от одной цели. На рисунке 2, подробнее выделили характерные значения частот до 3.5 минут (по рисунку 2 до 3.38×10^4).

Явно виден уровень, где значения частот флуктуируют около 10.8 МГц, на четвертом посту. Делаем предположение о цели, которая стоит вблизи четвертого поста. Также замечаем уровень частот около 9.4 МГц, причем на данной частоте импульсы отождествляются на четырех постах.

Из визуального анализа делаем вывод, что есть определенная цель, которую довольно легко отождествить, а именно это набор импульсов на частоте 1,090 МГц. Поэтому построим следующий график, а именно удалим из набора импульсы на данной частоте (рисунок 3).

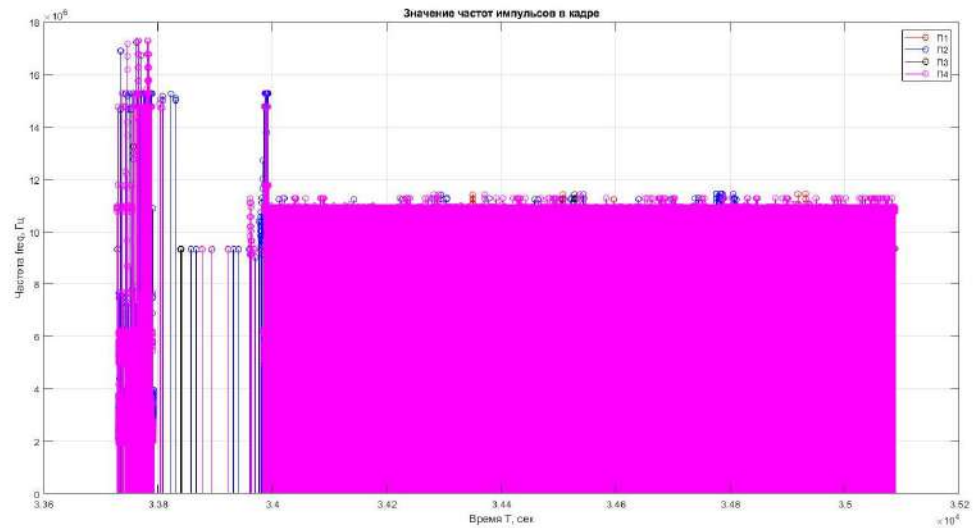


Рисунок 3. Зависимость частоты импульса от абсолютного времени прихода импульса.

Время накопления 23 минуты. Без сигналов частотой 1.090 МГц.

Теперь построим зависимость длительности импульса от абсолютного времени прихода импульса в кадре по четырем постам (рисунок 4).

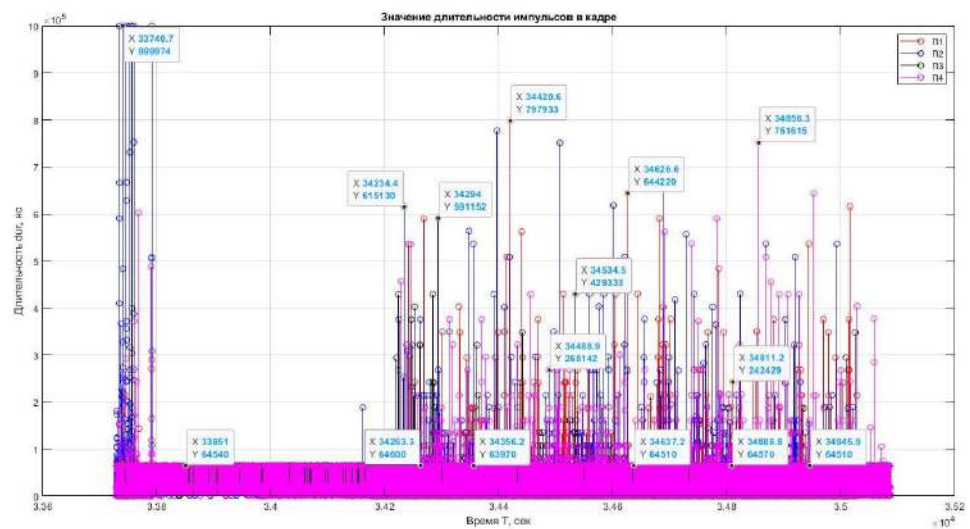


Рисунок 4. Зависимость длительности импульсов от абсолютного времени прихода импульсов.

Выделим участок, где видим постоянное значение длительности импульса, а именно около 64.5 мкс. Также выделим «звездочками» импульсы со значением частоты 1,090 МГц (рисунок 5).

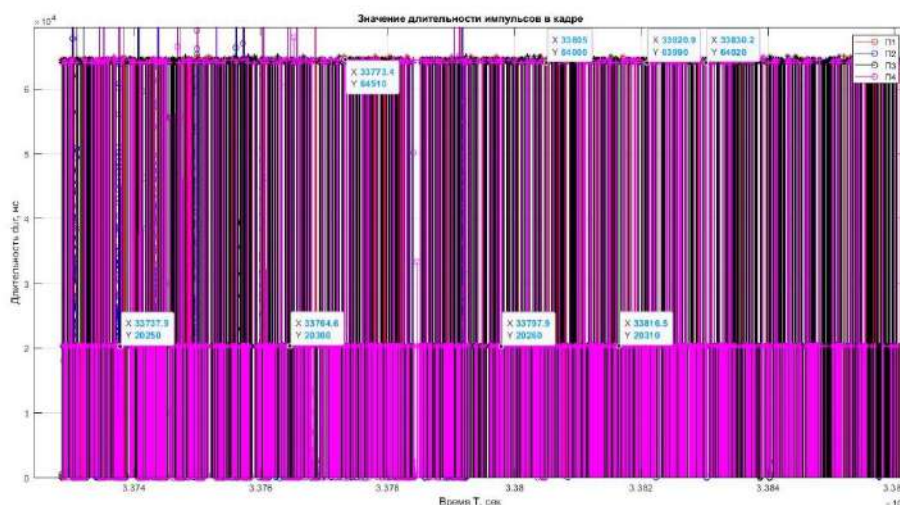


Рисунок 5. Выделен участок со значениями длительностей импульсов, где частота импульса 1.090 МГц.

Оказывается, импульсы на частоте 1.090 МГц имеют разные длительности. А значит скорее всего это разные стационарные цели. Предполагается, что данную особенность должен будет выделить алгоритм кластеризации.

5.2. Применение алгоритмов кластеризации к реальным измерениям.

После визуального анализа переходим к решению поставленной задачи, а именно ознакомиться с работой встроенных функций статистики и машинного обучения среды Matlab и применить к реальным радиолокационным импульсам алгоритмы кластеризации.

Набор для кластеризации представляет собой следующее: в один массив собираются импульсы со всех четырех постов в зависимости от времени прихода импульса в шкале системы. Параметрами импульса являются частота и длительность. Берутся первые 50000 импульсов. Анализ происходит при таком количестве из-за того, что большее число импульсов при кластеризации занимает слишком много памяти и возникает ошибка при расчетах на вычислительной машине.

5.2.1. Иерархическая кластеризация.

Иерархическая кластеризация позволяет исследовать различные группировки в имеющихся данных путем создания иерархического дерева кластеров.

Функции иерархической кластеризации в среде Matlab.

`pdist` – попарное расстояние между точками данных

`linkage` – возвращает кластеры, полученные иерархической кластеризацией

`dendrogram` – графическое построение иерархического дерева.

Из визуального анализа сделали выводы о том, как меняется частота, и увидели характерные уровни.

Отсюда для начала применим иерархическую кластеризацию для первых 50000 импульсов, пришедших на первый пост, и сделаем условное ограничение по уровню на 20 кластеров. Построим иерархическое дерево для данного набора данных.

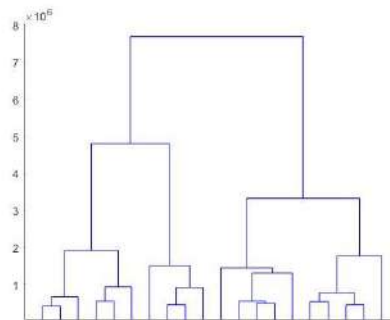


Рисунок 6. Иерархическое дерево на 20 кластеров.

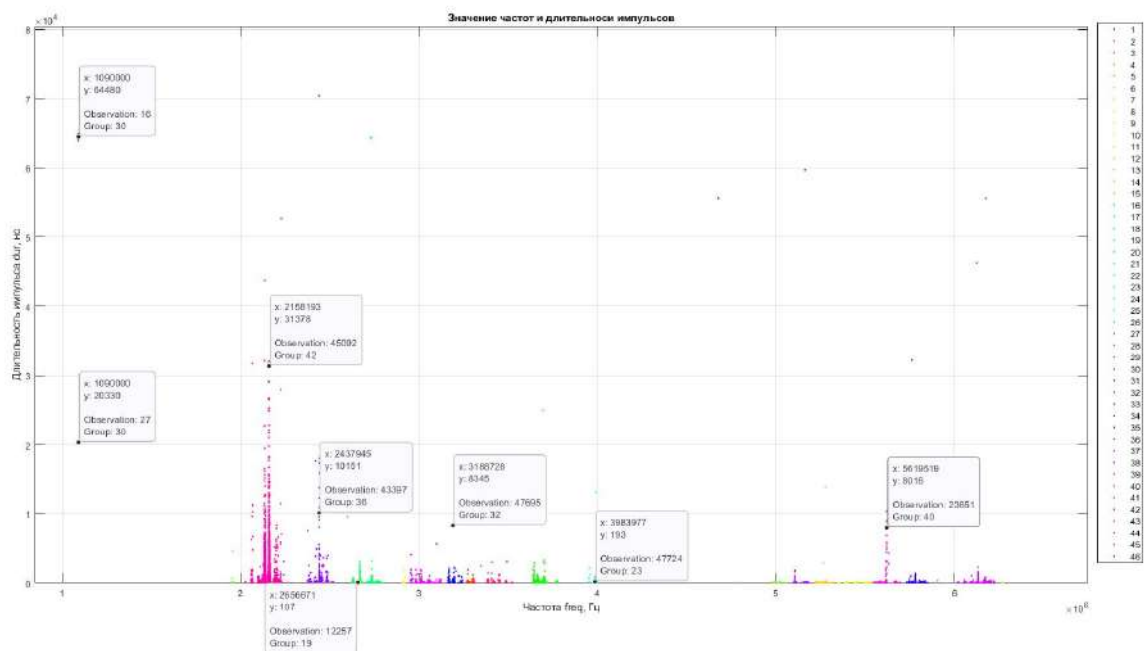


Рисунок 7. Характерные значения частот в кластерах. Приблизнен участок частот до 7 МГц.

Алгоритмы кластеризации выделили разные частоты. Были отмечены частоты около 2 МГц, которые явно видны на рисунке 2. А также набор частоты по уровню 10.8 МГц и её уровни флуктуации. На рисунке 7 видно, что данная частота флуктуирует дискретно.

Далее алгоритмы кластеризации применяются по данным, параметры которых являются частота импульса и длительность импульса. А именно формируется следующий

массив данных: складываются все импульсы с четырех постов и берутся параметры частоты и длительности импульсов первых 50000 импульсов. Построим иерархическое дерево для данного набора данных, ограничимся уровнем в 50 кластеров.

Из рисунка 5 видно, что корневой узел в этом дереве намного выше, чем остальные узлы и это показывает, что существует две многочисленные, отличные от других группы наблюдений. В каждой из больших групп видно, что более низкие уровни проявляются, когда рассматриваются меньшие и меньшие уровни по метрике расстояния. В итоге существует много разных уровней групп различных размеров и в различных степенях отчетливости. Аналитически можно предположить, что данные две большие группы – это импульсы на частоте 1.090 МГц и остальные импульсы на других частотах.

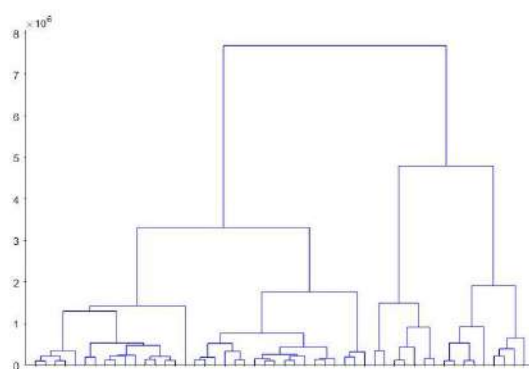


Рисунок 8. Иерархическое дерево по первым 50000 импульсам на 50 кластеров.

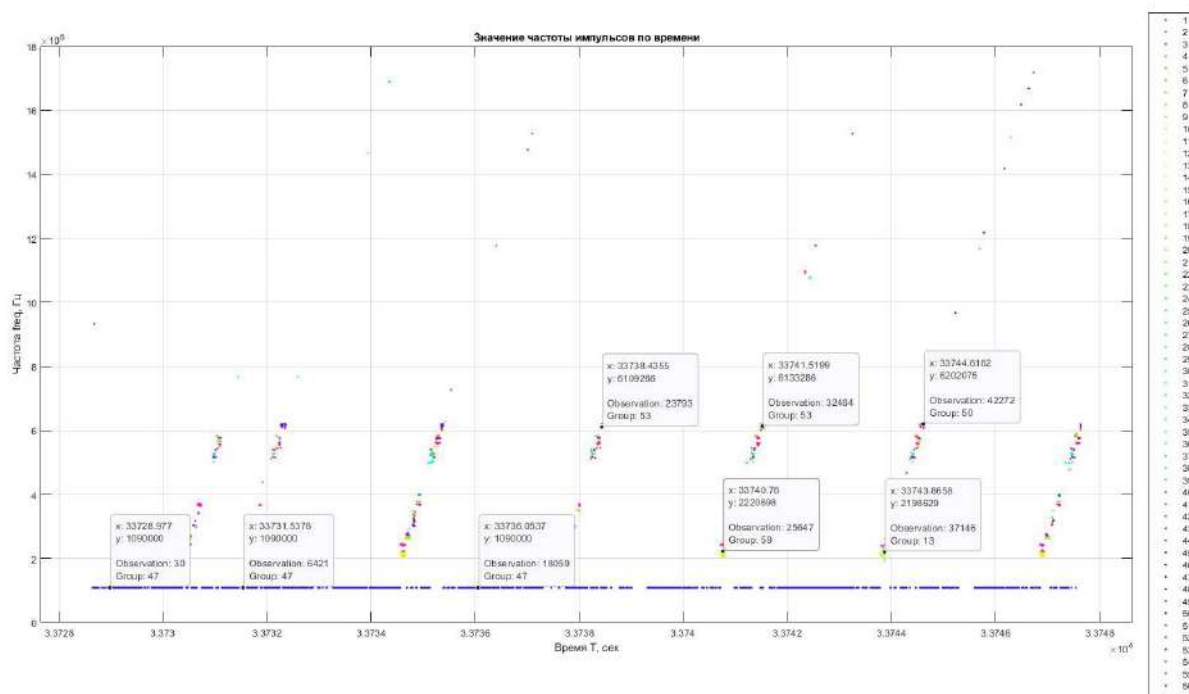


Рисунок 9. Зависимость частоты от абсолютного времени прихода импульсов в кадре в шкале системы.

Был применен иерархический алгоритм кластеризации на данный набор данных. Анализируя рисунок 9 видно следующее: алгоритм выделил в один кластер уровень частоты 1.090 МГц, чего и требовалось ожидать. В итоге данный алгоритм иерархической кластеризации с данной метрикой расстояний выполняет кластеризацию на набор данных по значению частоты.

5.2.2. Метод k -средних значений.

Функция `kmeans` в среде `Matlab` выполняет кластеризацию k -средних значений, с помощью итеративного алгоритма, который присваивает объекты кластерами так, чтобы сумма расстояний от каждого объекта до его кластерного центроида, по всем кластерам, была минимумом. С кластеризацией k -средних значений необходимо задать количество кластеров, на которое требуется разделить набор данных. Для анализа возьмем $k = 9$.

Для анализа того, насколько хорошо разделяются получившиеся кластеры имеет место построить график контура. График контура отображает меру того, как близко каждая точка в одном кластере к точкам в соседних кластерах.

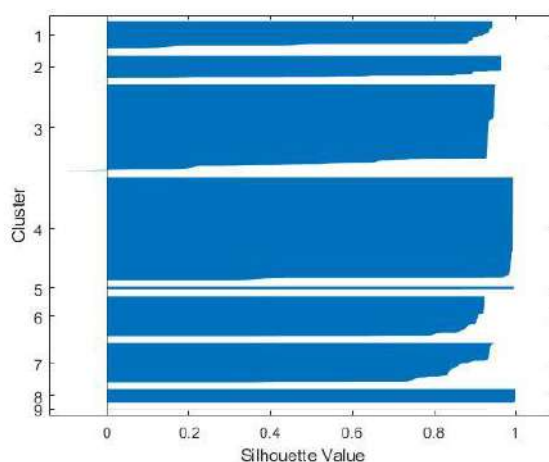


Рисунок 10. График контура, для девяти кластеров. Метрика – Евклидово расстояние.

По рисунку 10 можно сказать следующее: большинство точек в кластерах имеет значение контура, больше чем 0.8, это указывает на то, что токи хорошо разделяются от соседних кластеров.

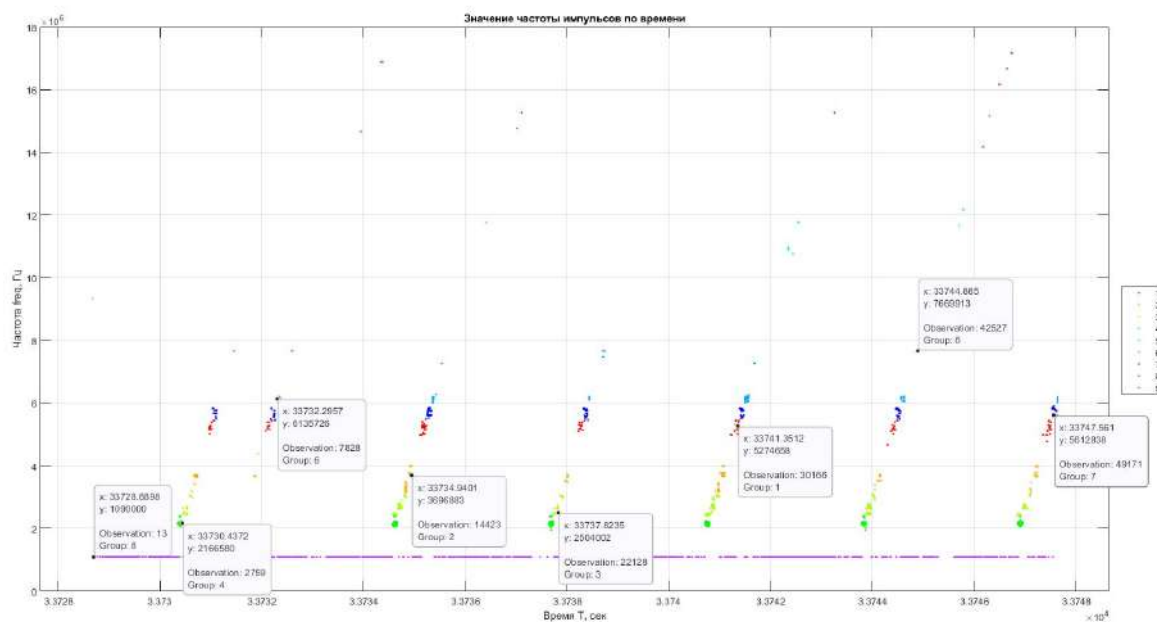


Рисунок 11. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы.

Изменим метрику расстояния с Евклидова расстояния на косинусное расстояние. Расстояние косинуса отличается тем, что игнорирует абсолютные расстояния и определяет только относительные расстояния.

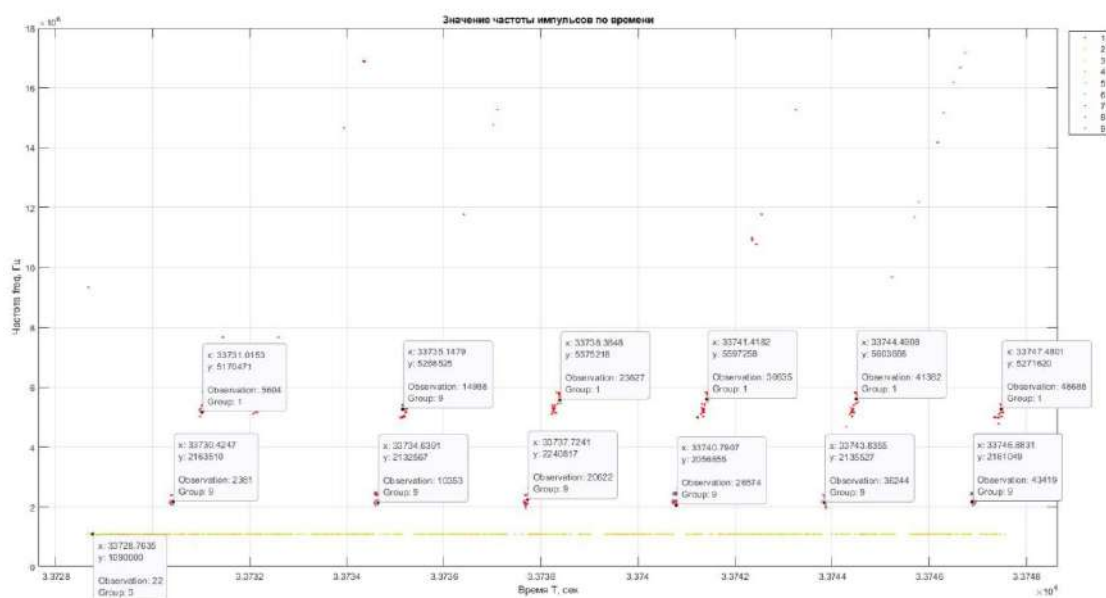


Рисунок 12. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Метрика – косинусное расстояние.

По рисункам 12 делаем следующие выводы: существует два кластера на частоте 1.090 МГц с разными средними значениями длительности импульса. Что и предполагалось при

визуальном анализе данных. Ранее используя Евклидову метрику расстояния не удалось добиться разделение на два кластера эти наборы данных. В итоге при использовании косинусного расстояния алгоритм в основном выполняют кластеризацию по значению длительности импульса.

При использовании косинусной метрики расстояний не учитываются абсолютные значения и был получен такой результат. Тогда требуется учесть тот момент, что данные вначале подавались не нормированными. Тогда проверим алгоритм кластеризации на нормированных данных (нормировка на максимальное значение частоты и длительности импульса) с Евклидовой метрикой расстояний.

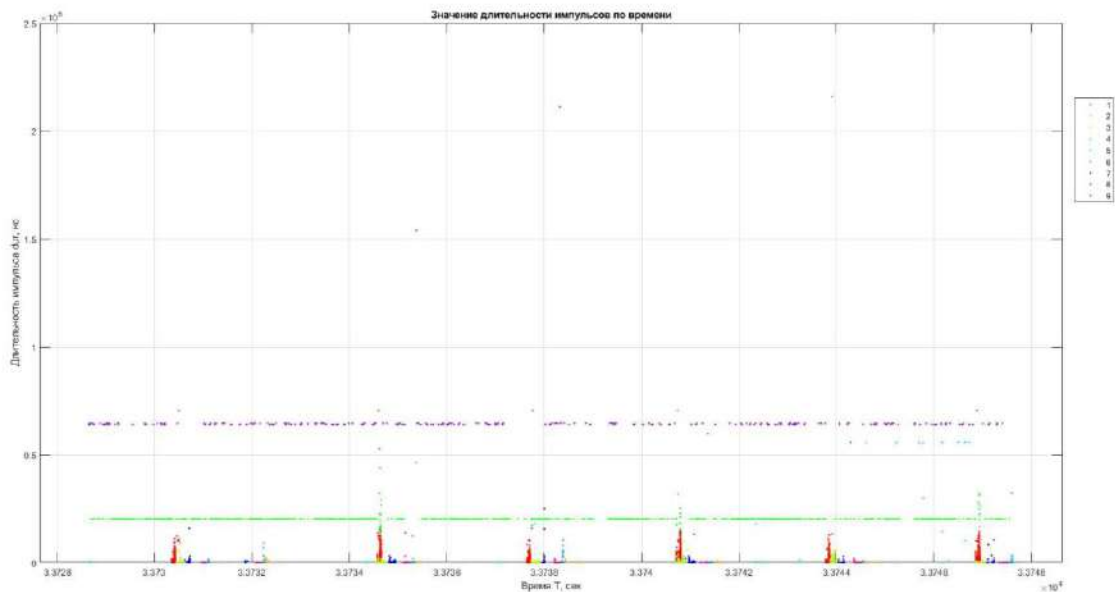


Рисунок 13. Зависимость длительности импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Метрика – евклидово расстояние, данные нормированы.

На рисунке 13 видим, что если изначально данные нормированы, то на частоте 1.090 МГц действительно выделяется два кластера. Делаем вывод, что нормирование данных в задачах кластеризации является важным моментом при решении задачи.

5.2.3. Основанная на плотности пространственная кластеризация с шумом (DBSCAN).

Выполним кластеризацию методом `dbscan`. Чтобы решить данную задачу необходимо подобрать значение `minpts` – число минимальных соседей. Данный параметр обычно считается большим или равным размерности входных данных плюс один. Для данной задачи число минимальных соседей представляет собой число импульсов в кластере.

Для начала выберем `minpts = 100`. Далее нужно подобрать значение для `epsilon` окрестности. Существует следующая стратегия для оценки значения `epsilon` – это

построение графика взаимных попарных расстояний для набора входных данных. График содержит «колено» - резкий перегиб. Расстояние, которое соответствует этому перегибу, обычно является хорошим выбором для ϵ , т.к. эта область, где далее точки становятся зашумленными. Выполним на ненормированных данных.

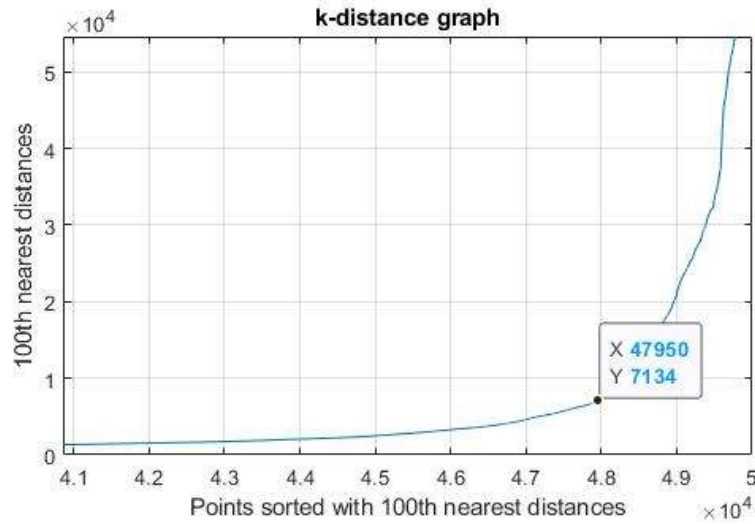


Рисунок 14. График k-расстояний.

По графику взаимных расстояний выберем значение ϵ примем равным 7200.

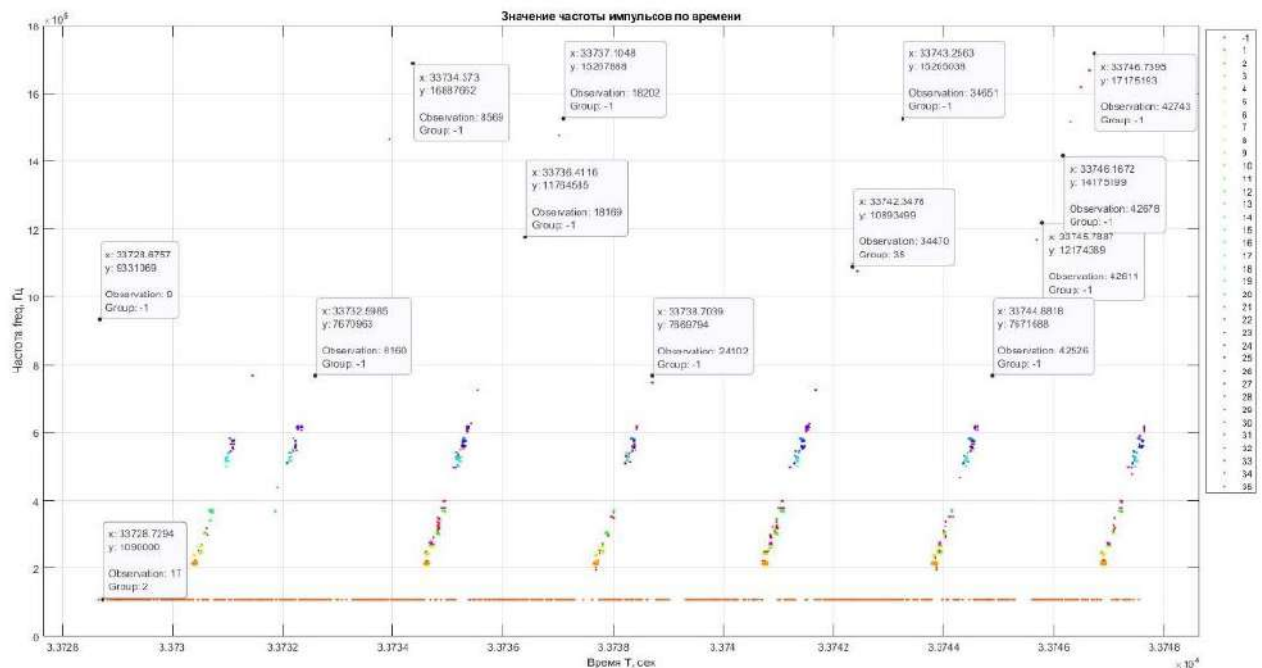


Рисунок 15. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы.

Отличительная особенность данного алгоритма в том, что не требуется задавать изначально, как входной аргумент, число кластеров. Также алгоритм выделяет выбросы или

по-другому шумы в отдельный кластер обозначенный за -1 (рисунок 18). Видно импульсы с значениями частот равными 1.090 МГц образуют только один кластер.

Запустим данный алгоритм кластеризации с метрикой косинусного расстояния. Оставим число минимальный соседей также равным $\text{minpts} = 100$.

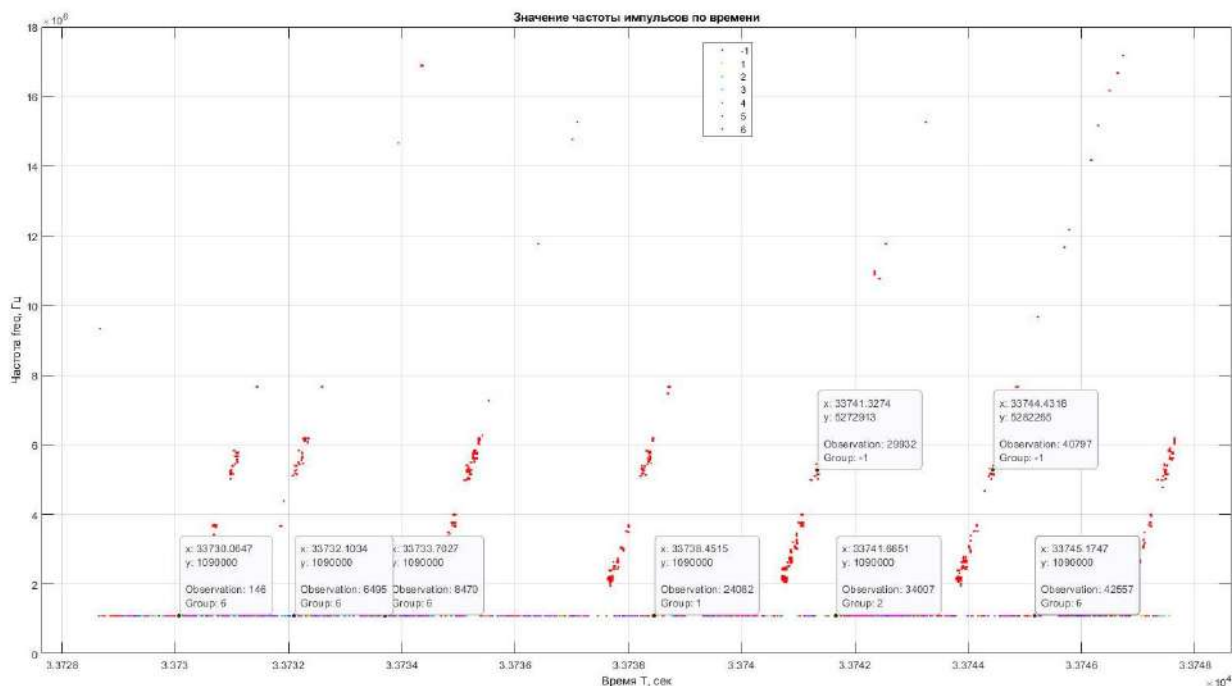


Рисунок 16. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Метрика – косинусное расстояние. $\text{Epsilon} = 1.64\text{e}-08$.

Анализируя рисунок 16 получаем следующее: на частоте 1.090 МГц выделяется 6 кластеров и все они на уровне 2.5 мкс по длительности импульса. Все остальные импульсы алгоритм относит к шумовым данным. Возможно действительно на частоте 1.090 МГц импульсы с большой длительностью являются шумовыми, так как могут быть вызваны переотражением.

Выполним нормировку входных данных и для анализа поведения алгоритма кластеризации будем варьировать число минимальных соседей minpts и значение окрестности точки epsilon .

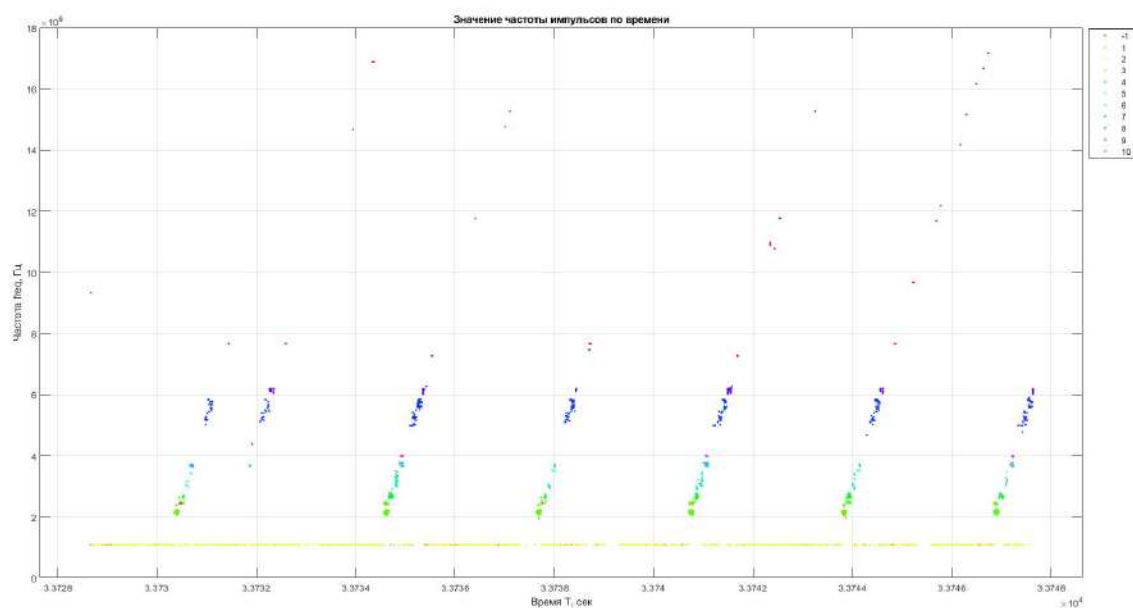


Рисунок 17. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Minpts = 100, Epsilon = 0.005.

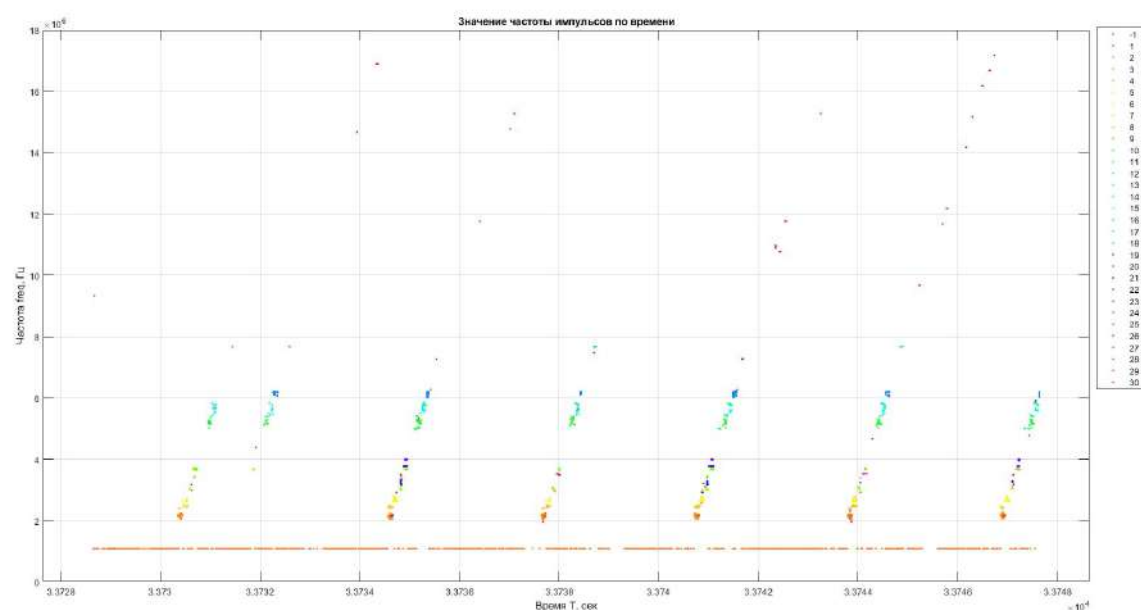


Рисунок 18. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Minpts = 10, Epsilon = 0.0015.

Анализируем рисунки 17-18 и говорим о следующем: уменьшение числа соседей приводит к тому, что в один кластер попадает уже меньшее количество импульсов, также это происходит из-за уменьшения окрестности точки, в итоге кластеры становятся меньше, но число кластеров увеличивается. В основном разбиение на кластеры происходит по значению частоты импульса.

Алгоритм DBSCAN легко варьирует своими параметрами `minpts` и `epsilon` для получения разных результатов кластеризации. Плюс данного метода кластеризации в том, что он выделяет шумовые точки, что является актуальным для задачи радиолокации.

5.2.4. Смешанная гауссовская кластеризация.

Смешанная гауссовская модель (GMM) формирует кластеры как смесь многомерных нормальных компонентов плотности. Для заданного набора данных GMM определяет апостериорные вероятности каждой плотности компонентов (или кластер). Апостериорные вероятности указывают, что наблюдение имеет некоторую вероятность принадлежности каждому кластеру.

Кластеризация GMM может вместить кластеры, которые имеют различные размеры и структуры корреляции в них. Поэтому в определенных приложениях, кластеризация GMM может быть наиболее подходящей, чем, например, кластеризация методом k -средних значений. Как и много других методов кластеризации, GMM требует заранее заданного количества кластеров. Количество кластеров задает количество компонентов в GMM.

Для реализации метода GMM применяются следующие методы.

Рассматривается ковариационная матрица данных. Требуется задать диагональные или полные ковариационные матрицы и начальные условия. Алгоритм максимизации ожидания (ЕМ) используется в GMM. Как в методе k -средних значений, ЕМ чувствителен к начальным условиям и может сходиться к локальному оптимуму. Можно задавать собственные начальные значения для параметров, а также задать начальные кластерные присвоения для точек данных. Иначе можно позволить им быть выбранными случайным образом или использовать вспомогательно метод k -средних значений. Также если данные имеют сильные выбросы, потребуется нормализовать данные, чтобы усреднить оценку.

Количество компонентов k в GMM определяет количество подпопуляций или кластеры. Каждый Гауссов компонент имеет ковариационную матрицу. Геометрически, структура ковариации определяет форму эллипсоида доверия, определяющий кластер. Можно задать, являются ли ковариационные матрицы для всех компонентов диагональными или полными, и имеют ли все компоненты ту же ковариационную матрицу. Каждая комбинация требований определяет форму и ориентацию эллипсоидов.

Для примера зададим 20 компонентов GMM и тысячу итераций для алгоритма ЕМ. Также для метода потребуется задать порог вероятности для области доверия (99%), который определяет длину главных и малых осей эллипса доверия. Далее нужно задать тип ковариации,

который определяет ориентацию осей эллипса доверия. Используются следующие ковариационные матрицы:

Диагональные ковариационные матрицы указывают, что данные являются некоррелированными. Форма области доверия ближе к кругу. Главные и малые оси параллельны или перпендикулярны осям X и Y .

Полные ковариационные матрицы допускают корреляцию данных без ограничения на ориентацию замещающих знаков относительно осей X и Y . Эта спецификация может вызвать сверхподбор кривой.

Разделяемые ковариационные матрицы указывают, что все компоненты имеют ту же ковариационную матрицу, т.е. одну и ту же область доверия. Все замещающие знаки одного размера и имеют ту же ориентацию.

Неразделяемые ковариационные матрицы указывают, что каждый компонент имеет свою собственную ковариационную матрицу. Размер и ориентация всех эллипсов могут отличаться.

Выполним кластеризацию на анализируемых данных и отобразим графически (рисунок 19-21).

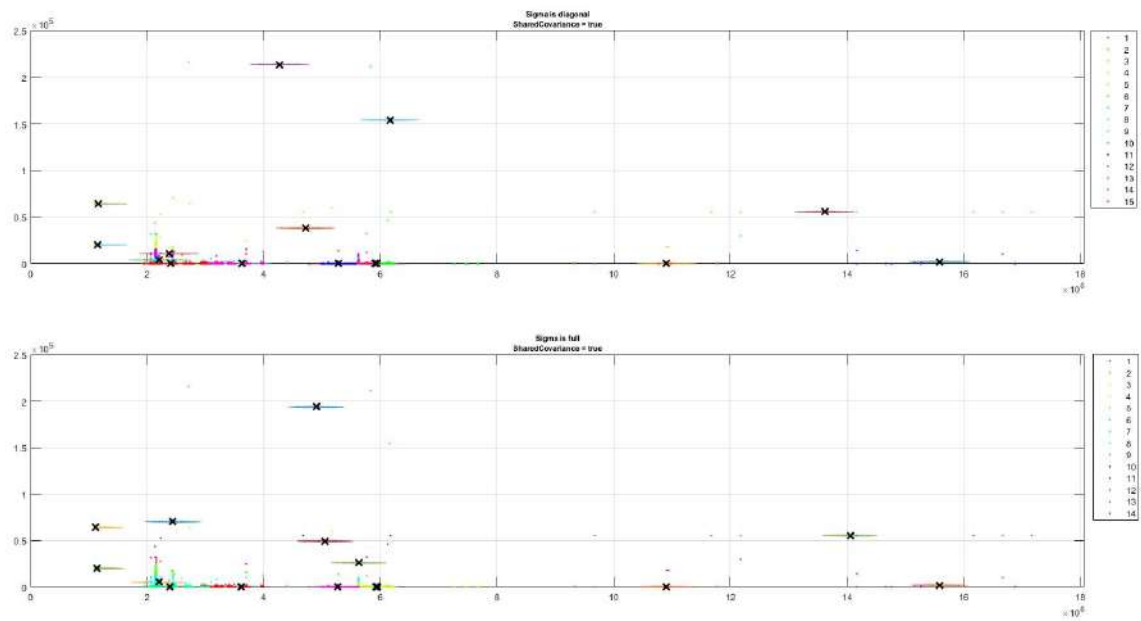


Рисунок 19. Зависимость длительности импульса от частоты импульса. Ненормированные данные.

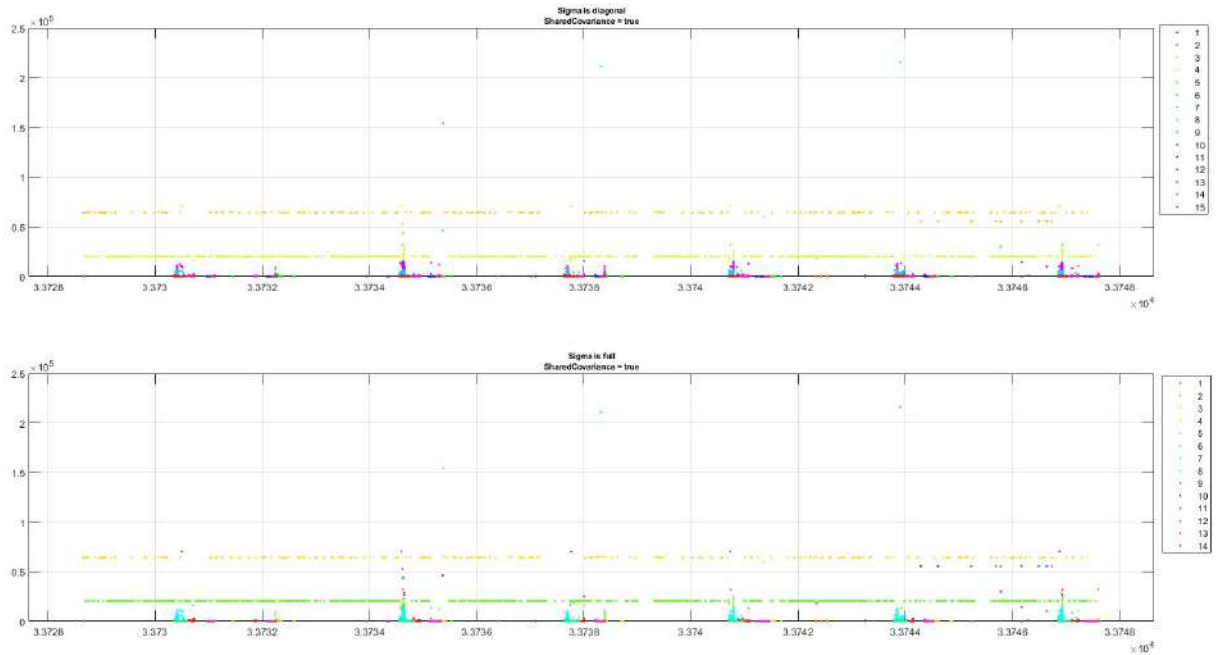


Рисунок 20. Зависимость длительности импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Ненормированные данные.

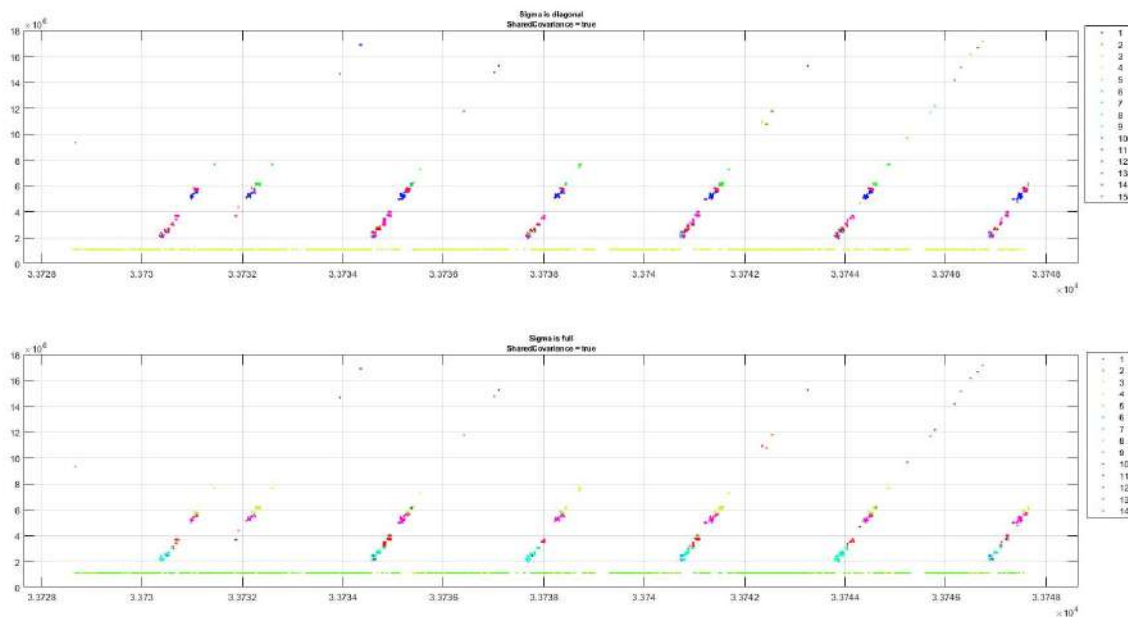


Рисунок 21. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Ненормированные данные.

Проанализируем результат. Во время итерации ЕМ, рассчитались только диагональная и разделяемая ковариационная матрица, которые считаются из учета того, что данные некоррелированные. Не рассчитались матрицы, которые допускают корреляцию данных, это возникает из-за того, что во время итерации алгоритма ЕМ вероятность стремиться к бесконечности. Это проблема бывает в следующих моментах: рассчиталось больше предсказанных меток, чем есть точек данных, задается подбор кривой слишком многими компонентами, переменные высоко коррелируются. Также `cluster` не всегда сохраняет кластерный порядок, что видно на рисунках (19-21). В итоге получили деление на 15 и 14 кластеров. На частоте 1.090 МГц выделяется два кластера, которые соответствуют двум разным длительностям импульса.

Запустим данный алгоритм кластеризации при тех же начальных условиях, но уже на нормированных данных (рисунок 22).

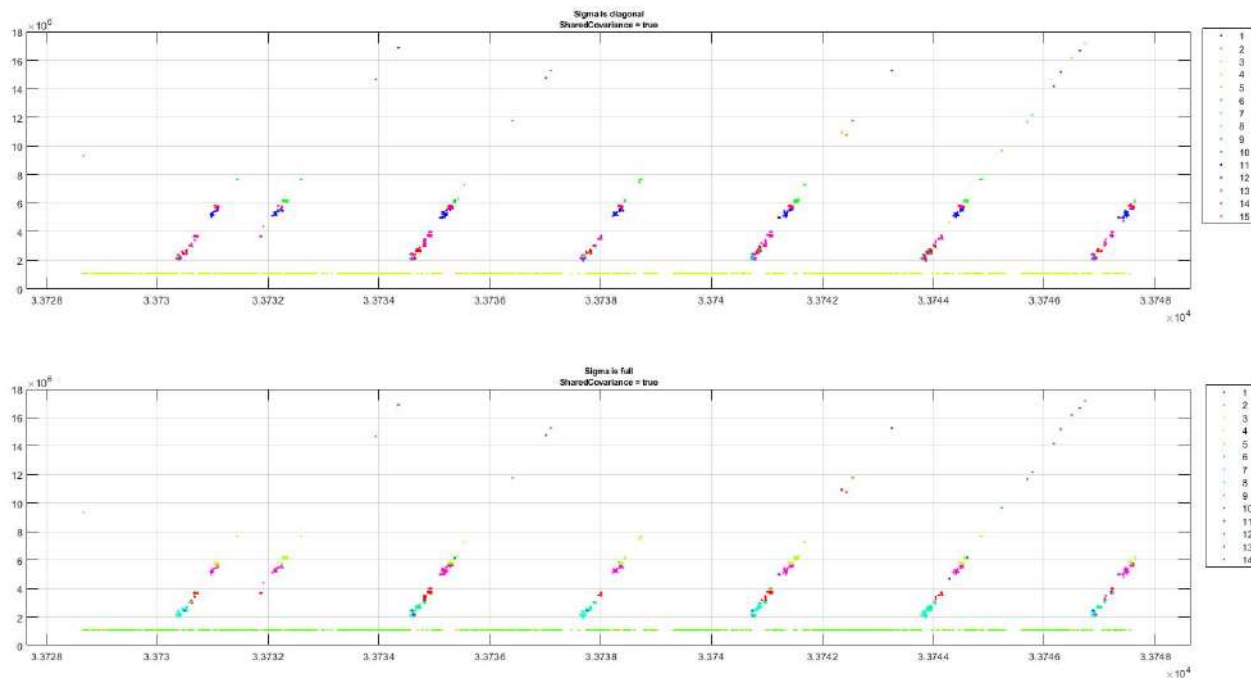


Рисунок 22. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Нормированные данные.

Получаем аналогичные результаты, как и ранее (рисунок 21).

5.2.5. Спектральная кластеризация.

Спектральная кластеризация является основанным на графике алгоритмом для нахождения k кластеров произвольной формы в данных.

Функции в Matlab. Использование `spectralcluster` выполнит спектральную кластеризацию на входных данных или на матрице подобия графика подобия, выведенного из данных. Также этот алгоритм требует заранее заданного количества кластеров. Однако алгоритм для спектральной кластеризации также обеспечивает способ оценить количество кластеров в ваших данных.

Алгоритм спектральной кластеризации оценивает количество кластеров с помощью собственных значений Матрицы Лапласа и выполняет спектральную кластеризацию на наборе данных.

Оценим количество кластеров в данных при помощи собственных значений Матрицы Лапласа и выполним спектральную кластеризацию на наборе данных.

Вычислим 20 собственных значений величины Матрицы Лапласа при помощи `spectralcluster`. По умолчанию алгоритм использует нормированную Матрицу Лапласа случайного обхода.

Выполним спектральную кластеризацию на ненормированных наблюдениях при помощи функции `spectralcluster`. Задаем значение равное $k = 20$ кластеров.

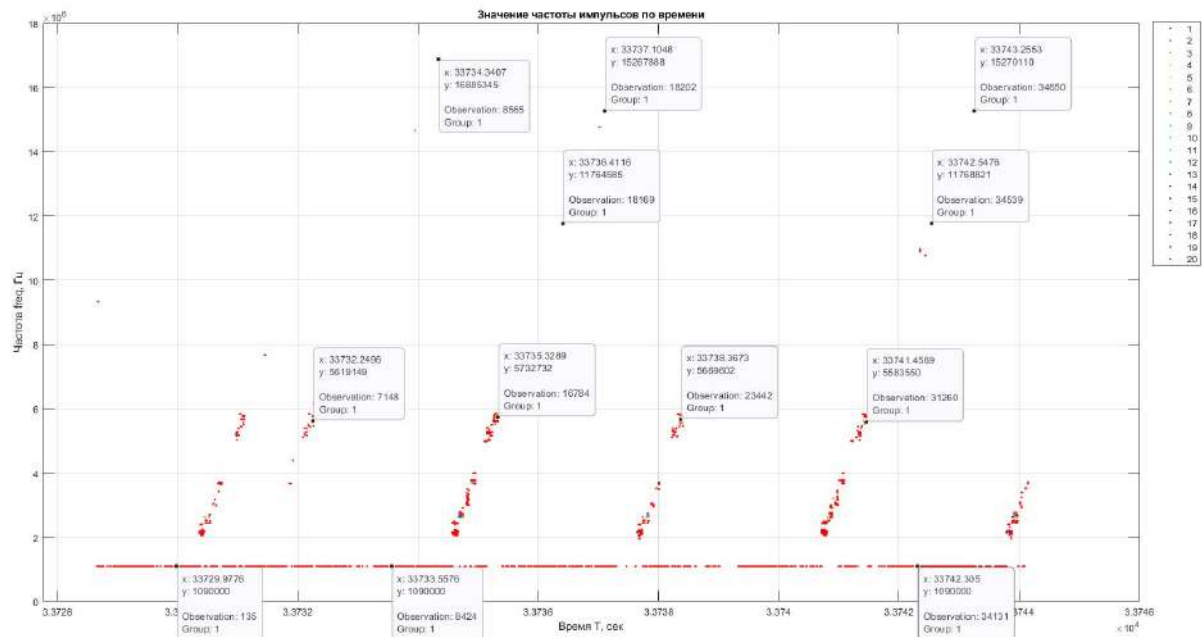


Рисунок 23. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Ненормированные данные. 20 кластеров.

Анализируя рисунок 23, получаем, что большую часть импульсов алгоритм отнес к первому кластеру и все остальные девятнадцать кластеров это маленькие скопления импульсов в интервале частот от 2 МГц и до 4 МГц.

Повторим спектральную кластеризацию на нормированных данных при тех же начальных условиях.

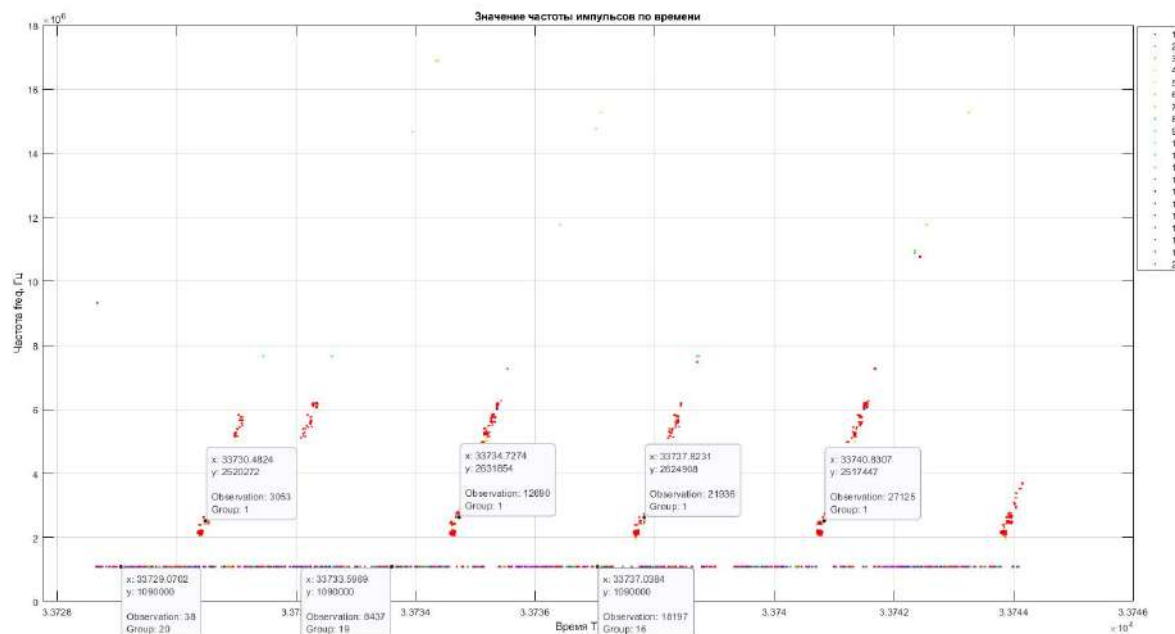


Рисунок 24. Зависимость частоты импульса от абсолютного времени прихода импульсов в кадре в шкале системы. Нормированные данные. 20 кластеров.

На нормированных данных спектральная кластеризация сработала явнее. Выделились кластеры на частоте 1.090 МГц, причем заметно, что на этой частоте при характерном одном значении длительности импульсов выделяется множество кластеров. При этом скопление импульсов на частотах выше 2 МГц в основном образуют первый кластер.

Выводы

В ходе выполнения научно-исследовательской работы были рассмотрены этапы обработки радиолокационной информации. Было сказано, что применение алгоритмов статистики и машинного обучения к системам обработки радиолокационной информации считается актуальным и многообещающим направлением. В данной работе было предложено применить алгоритмы кластеризации на этапе обнаружения отметок от целей. Решение такой задачи позволит в будущем подбирать более эффективные алгоритмы вторичной (траекторной) обработки целей.

Для решения поставленной задачи были рассмотрены методы кластеризации данных, а именно: иерархическая кластеризация, метод k-средних значений, основанная на плотности пространственная кластеризация с шумом (DBSCAN), смешанная гауссовская модель и спектральная кластеризация. Была описана основная теория по данным методам и алгоритм работы.

Алгоритмы кластеризации применялись на реальных радиолокационных импульсах. Требовалось проанализировать реальные записи с многопозиционного пассивного радиолокационного комплекса на поиск характерных повторений с целью объединения импульсов в кластеры и дальнейшего отождествления.

Анализируя проделанную работу можно сделать следующие выводы: алгоритмы кластеризации выделяют импульсы в группы, где импульсы наиболее похожи между собой. В итоге в таких кластерах легче осуществить отождествление межпостовое, так как выборка из импульсов становится меньше, что уменьшает вычислительный процесс, увеличивает точность отождествления импульсов.

Список литературы

1. Галушкин А.И. Нейрокомпьютеры (Нейрокомпьютеры и их применение. Кн. 3). М.: ИПРЖР, 2000. 528 с.
2. Barton D.K. Radar system analysis and modeling. – Artech House, Boston, MA, 2005.
3. Попов Г.П. Инженерная психология в радиолокации. М.: Сов. Радио, 1971. 143с.
4. Кузьмин С.З. Цифровая обработка радиолокационной информации. М.: Сов. радио, 1967. 400 с.
5. Левин Б.Р. Теоретические основы статистической радиотехники. 3-е изд. перераб. и доп. М.: Радио и связь, 1989. 656 с.
6. Перов А.И. Оптимальный алгоритм дискретного сопровождения многих целей идентификацией измерений // Радиотехника, No1, 2003.
7. Петухов СИ., Степанов А.Н. Эффективность ракетных средств ПВО. М.:Воениздат, 1976. 104 с.
8. Радиолокационные системы: Основы построения и теория. Справочник / Под ред.Я.Д. Ширмана. М.: ЗАО "МАКВИС", 1998. 828 с.
9. Татузов А.Л., Чухлеб Ф.С. Использование нейросетевой технологии при обработке радиолокационной информации // Информационные Технологии. 1999, No 1, С. 25-33.
10. Татузов А.Л. Нейронные сети в задачах радиолокации. Кн. 28. – М.: Радиотехника, 2009. – 432 с.: ил. (Научная серия «Нейрокомпьютеры и их применение»)
11. Dillard G.M. Mean-level detection of nonfluctuating signals // IEEE Trans., 1974, v.AES-10, no.6 (Nov. 1974), pp.795-799.