



**УНИВЕРСИТЕТ ПО ХРАНИТЕЛНИ ТЕХНОЛОГИИ**

**гр. Пловдив**

**Катедра:** „Автоматика, информационна и управляваща техника”

**Специалност:** „Автоматика, информационна и управляваща техника ”

# **ДИПЛОМНА РАБОТА**

***Тема: Невронни класификатори на пчелен мед по  
критерий „ботанически произход“***

Дипломант:.....

/Кирил Нанков/

Ръководител катедра:.....

/доц. В. Начев/

Научен ръководител:.....

/проф. Д. Цанкова/

Пловдив

2019 г



## СЪДЪРЖАНИЕ

УВОД.....	4
ГЛАВА ПЪРВА – ИЗСЛЕДВАНЕ АВТЕНТИЧНОСТТА И КАЧЕСТВОТО НА ПЧЕЛНИЯ МЕД .....	6
1.1.    Стандарти и Нормативни уредби към пчелния мед .....	6
1.2.    Методи за изследване ботаническия произход на пчелния мед и тяхното приложение .....	7
1.2.1. Физико-химичен анализ.....	8
1.2.2. Поленов анализ .....	10
1.2.3. Спектрален анализ .....	11
1.2.4. Изграждане на профил на тежките метали в меда .....	12
1.2.5. Ядрено-магнитен резонанс (ЯМР).....	13
1.2.6. Други методи .....	14
ГЛАВА ВТОРА – СТАТИСТИЧЕСКИ И ИНТЕЛИГЕНТНИ КЛАСИФИКАТОРИ НА ПЧЕЛЕН МЕД ПО БОТАНИЧЕСКИ ПРОИЗХОД.....	15
2.1. Постановка на задачата за класифициране на мед по критерий „ботанически произход“ .....	15
2.2.    Теоритични сведения за използваните принципи и методи .....	17
2.2.1.    Снемане на абсорбционни спектрални характеристики със спектрофотометър .	17
2.2.2.    Метод на главните компоненти (МГК) .....	21
2.2.3.    Линеен Дискриминантен Анализ (ЛДА).....	24
2.2.4.    Изкуствени Невронни Мрежи (ИНМ) .....	26
2.3.    Реализация на ЛДА класификатор .....	30
2.4.    Реализация на ИНМ класификатор с права структура и Backpropagation обучаващ алгоритъм .....	33
ГЛАВА ТРЕТА – ВАЛИДАЦИЯ И СРАВНИТЕЛЕН АНАЛИЗ НА РАЗГЛЕДАНИТЕ КЛАСИФИКАТОРИ	35
3.1.    ЛДА класификатор по УВ спектър.....	38
3.2.    ЛДА класификатор по видимия спектър .....	40
3.3.    ИНМ класификатор с Backpropagation алгоритъм по УВ спектър .....	42
3.4.    ИНМ класификатор с Backpropagation алгоритъм по видимия спектър.....	45
3.5.    Анализ и сравняване на резултатите .....	48
ГЛАВА ЧЕТВЪРТА - ИЗВОДИ .....	49
ИЗПОЛЗВАНА ЛИТЕРАТУРА .....	50

Приложение 1.....	53
Приложение 2.....	58
Приложение 3.....	63
Приложение 4.....	69

## УВОД

Наименованието ”пчелен мед” се използва за сладък продукт, получен от медоносни пчели от нектара на растенията или от секрети на живите части на растенията, или от екскрементите на смучещите насекоми по растенията, които пчелите събират, трансформират чрез комбиниране на специфични вещества от организма си, отлагат, дехидратират, скадират и съхраняват в пчелни килийки до съзряване (наредба на МЗ от 2002г.). Или на кратко: Пчелният мед е ценен хранителен продукт, който се получава чрез преработката на нектара или сладките сокове по някои растения от пчелите.<sup>[1]</sup>

Още от древността медът е познат като един от продуктите с най-висока енергийна стойност. От хиляди години, та и до ден днешен медът се използва не само като храна, но и като лекарство за редица заболявания. В миналото медът е бил считан за „златен дар“, който може да гарантира дълголетието и здравето на хората.

По-късно през XX век с помощта на развитието на човечеството и науките са изяснени самият състав на меда, както и начините по които той въздейства на организма на човека. В днешно време все повече проучвания потвърждават антибактериалните свойства на меда и успешно се доказва тяхното целебно свойство при лечението на редица рани, гастро-интестинални , чернодробни и други видове заболявания.

Медът се дели на три вида според произхода си: нектарен, манов и смесен. Нектарият се произвежда от така нареченият нектарен сок на растенията, който всъщност представлява воден разтвор на захароза, глюкоза и фруктоза в различни съотношения. Той бива: монофлорен – когато в съдържанието му преобладава нектара на определено растение(липа, рапица, слънчоглед и др.) и полифлорен – когато в състава му участват нектарите на две или повече растения като е трудно да се определи кой от тях е с доминантно съдържание. Мановият мед се произвежда основно от

екскрементите на другите насекоми смучещи от растителните цветове или от секрети на живи части на цветовете. Това обуслава два вида мана – животинска(най-често се получава от древни насекоми) или растителна(секрети от листа и млади клонки). Смесеният мед се получава когато се смесят нектарният и мановият мед.

В състава си медът съдържа: Вода(16 - 20%); Въглехидрати(70-80%), по-голяма част от които са глюкоза и фруктоза; а по-малка част заема захарозата; Ензими(диастаза, инвертаза, естераза); Аминокиселини(основно пролин); Минерали(P, Fe, Mg, Mn, Ca, Cl, Cu, Na и др.); Витамини(предимно каротиноиди в минимални количества); Чужди примеси(полени, ветеринарномедицински препарати и антибиотици, тежки метали, пестициди, микроорганизми).

За натурален пчелен мед се смята медът, който не е загряван, не е миксиран, няма примеси от други храни и подсладители и който не е получен чрез преработката на захарен сироп, с който са захранвани пчелите по време на медосбор.<sup>[12]</sup> Добиването на естествен пчелен мед е бавен, труден и скъп процес свързан с много разходи. Поради тази причина често се срещат подправени(фалшифицирани) продукти на основата на пчелния мед. В общия случай това представлява мед към който са добавени захари и други продукти с цел увеличаване на количеството му и осигуряването на конкурентната му цена на пазара. Качеството (а и автентичността) може да се наруши и по други начини, като например следните: (1) „Захарният мед“, предлаган на пазара, се получава когато пчелното семейство усилено се подхранва със захарен сироп и като резултат, подправеният мед не съдържа много от характерните минерални вещества, аминокиселини и органични киселини; (2) Въпреки добрия си търговски вид (по-светъл цвят и рядка консистенция) предлаганият на пазара мед с различно съотношение на добавена към него глюкоза е фалшификат с влошено качество; (3) За възстановяване на течната

консистенция на кристализирал мед, той се загрява, а високата температура разрушава ензимите в него и влошава качеството му; (4) При събиране на прашец от култури в райони, замърсени с тежки метали и пестициди, малки количества от някои вредни елементи могат да преминат в меда.<sup>[12]</sup>

## **ГЛАВА ПЪРВА – ИЗСЛЕДВАНЕ АВТЕНТИЧНОСТТА И КАЧЕСТВОТО НА ПЧЕЛНИЯ МЕД**

### **1.1. Стандарти и Нормативни уредби към пчелния мед**

В България за първи път през 1957г. се полагат първите нормативни документи за стандартизиране на качеството на меда и методите за анализа му. Към ден днешен актуализацията на уредбите е обособила няколко държавни стандарта от които основни са: БДС 3050-80 – дефиниращ методите за анализ на пчелния мед; БДС 2673-77 – описващ изисквания на които трябва да отговаря качественият пчелен мед; БДС 2673-89 – определящ начините за взимане на проби от пчелен мед.

Интензивната търговия с пчелен мед през последните десетилетия наложи да се разработят международни стандарти. Още през 1969 г. Комисията по Продоволствения кодекс /Кодекс Алиментариус/ към Организацията на ООН по прехрана и земеделие /ФАО/ и Световната здравна организация /СЗО/ разработи и публикува регионални норми за меда за страните от Европа, включващи критериите, нормите и методите за анализ. По-късно /1974 г./ беше разработена и въведена Директива 74/409 на Европейския съюз, която не се различава съществено от регионалните норми на Кодекс Алиментариус. Комисията по Продоволствения кодекс актуализира неколkokратно стандартите за пчелния мед и предложи те да са валидни за целия свят.<sup>[14]</sup>

Общоевропейските усилия на страните на континента да унифицират и подобрят правилата и стандартите дефиниращи качествения мед, както и методите за анализа му наложиха създаването на Международна комисия по меда. Едно от постиженията на комисията е изготвяне на директива за меда (Директива 110/2001), описваща критериите за окачествяване и методите за анализ. Изхождайки от тази директива са въведени промени и в българското законодателство като се въведе Наредба за изискванията към пчелния мед, предназначен за консумация от човека влезла в сила от Август 2003г.

## **1.2. Методи за изследване ботаническия произход на пчелния мед и тяхното приложение**

С анализите на пчелния мед се цели да се определят качеството, растителният произход и възможните фалшификации.<sup>[13]</sup> Пчелният мед се фалшифицира сравнително лесно. Ботаническият произход (тоест източника на цветен нектар) на меда също се отнася към качествените му показатели. Класифицирани по ботанически произход в България се срещат най-вече: акациев; липов; манов; рапицов; слънчогледов; ментов; лавандулов; от магарешки бодил; детелинов; плодов. По-високата цена на някои видове монофлорен мед (акациев, липов, лавандулов и др.) изисква доказване на техния ботанически произход.

За удостоверяване на автентичността на пчелния мед се използват различни аналитични техники, като: изотопен, хроматографски и термичен анализ, ядрено-магнитен резонанс, сензорно тестване, микроскопски анализ на полена в меда.<sup>[12]</sup> По-долу повърхностно са засегнати тези и някои други методи и тяхното приложение.



### **1.2.1. Физико-химичен анализ**

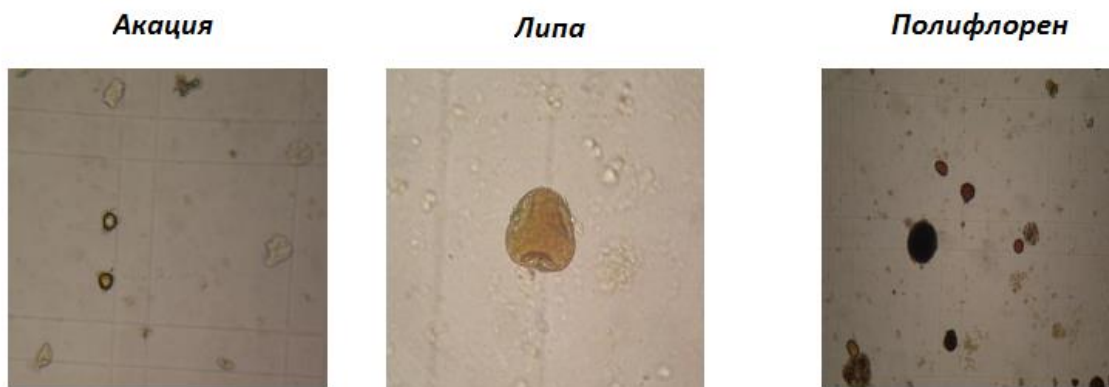
Това е анализ, който включва субективното определянето на органолептични свойства и химическият състав на меда. Органолептичните показатели са: цвят; вкус; аромат; консистенция; кристализация; чистота (наличие на примеси). Химичните свойства се определят от: съдържанието на захари; водното съдържание; неразтворимите примеси във вода; електропроводимостта; свободната киселинност; диастазната активност; количеството на хидроксиметилфурфурола; поленовата характеристика. Изискванията към физико-химичните характеристики са разписани в табличен вид в Наредбата за изискванията към пчелния мед (Табл. 1.1), предназначен за консумация от човека, а методите за изследването на тези характеристики е даден в Закона за пчеласрството, Наредба №48.

Качествени критерии и норми	БДС 2673 - 89	Дирек тива на ЕС	Codex alime- ntarius	Наредба МЗ, 2002 г. <sup>1</sup> и Наредба №9 на МЗГ от 2005 г. <sup>2</sup>
1.Водно съдържание, %, не повече от (за всички видове мед без посочените изключения).....	20	21	20	20 <sup>1:2</sup>
от пирен ( <i>Calluna</i> ) и <i>Trifolium</i> sp.....	-	23	-	-
от пирен ( <i>Calluna</i> ).....	-	-	23	-
от пирен ( <i>Calluna</i> ) и <i>индустриален</i> от останалите .....	-	-	-	23 <sup>1:2</sup>
<i>индустриален</i> мед от пирен ( <i>Calluna</i> ).....	-	-	-	25 <sup>1:2</sup>
2.Редуциращи захари,% ( <i>валиден до 01.01.2006г.</i> ), не по-малко: нектарен мед.....	68	65	-	68 <sup>2</sup>
манов .....	62	60	-	62 <sup>2</sup>
3.Захароза, %, не повече от:				
нектарен мед.....	5	5	5	5 <sup>1:2</sup>
акациев, лавандулов и манов мед... смесен (нектарен и манов).....	8 6	10 -	- -	8 <sup>2</sup> -
акациев, люцернов, <i>Banksia menziesii</i> , <i>Eucalyptus</i> <i>camadulensis</i> , <i>Citrus</i> spp.....	- -	- -	10 15	10 <sup>1</sup> 15 <sup>1</sup>
лавандулов, пореч ( <i>Borago officinalis</i> )..	-	-	15	15 <sup>1</sup>
4.Сума от фруктоза и глюкоза ( <i>след 01.01.2006г.</i> ), %, не по-малко от:				
нектарен мед .....	-	-	60	60 <sup>1</sup>
манов и смес от манов и нектарен мед.....	-	-	45	45 <sup>1</sup>
5.Неразтворими във вода вещества,%, не повече от.....	0.1	0.1	0.1	0.1 <sup>1:2</sup>
пресован пчелен мед.....	-	0.5	0.5	0.5 <sup>1</sup> / 0.1 <sup>2</sup>
6. Общ минерален състав (пепел), % не повече от:				
нектарен мед.....	0.5	0.5	-	- <sup>1</sup> / 0.5 <sup>2</sup>
манов мед .....	1.0	1.0	-	- <sup>1</sup> / 1.0 <sup>2</sup>
смесен (нектарен и манов) мед .....	0.8	-	-	-
7. Съдържание на свободни киселини (meq/kg), не повече от:				
Индустриален пчелен мед .....	-	80	80	80 <sup>1:2</sup>
Всички останали видове пчелен мед .....	40	50	50	50 <sup>1:2</sup>
8.Диастазна активност, ед. по Gote или Shade, не по-малко:	Gote	Shade	Shade	Shade <sup>1</sup> / Shade (Gote) <sup>2</sup>
всички видове мед ,без изброените по- долу.....	9	8	8	8
Акациев.....	8	-	-	-
медове с естествено ниска активност при не повече от 15 мг/кг ХМФ.....	-	3	3	3
9.Хидроксиметилфурфурол, mg/kg, не повече от: за всички видове без посочените по-долу.....	40	40	40	40 <sup>1:2</sup>
за непреработен мед.....	20	-	-	- <sup>1</sup> / 10 <sup>2</sup>
за мед от страни с тропичен климат или смеси с него .....	-	-	80	80 <sup>1:2</sup>
10.Електропроводимост, mS.sm <sup>-1</sup> -за нектарен мед, до.....	-	-	0.8	0,8 <sup>1:2</sup>
за манов и кестенов мед и смеси от тях, не по-малко.....	-	-	0,8	0,8 <sup>1:2</sup>
Изключения: манов пчелен мед от <i>арбутус</i> ( <i>Arbutus unedo</i> ), ерика ( <i>Erica</i> ), евкалипт, липа ( <i>Tilia</i> spp.), пирен, калуна ( <i>Calluna vulgaris</i> ), <i>лептоспермум</i> ( <i>Leptospermum</i> ), чаено дърво ( <i>Melaleuca</i> spp.)	-	-	не се нормира	не се нормира
10.Поленова характеристика - за монофлорен нектарен пчелен мед - съдържание на полен от съответното растение, %, не по-малко от:				
- Акациев.....	30			30 <sup>2</sup>
- Лавандулов.....	15			15 <sup>2</sup>
Други видове.....	40			40 <sup>2</sup>

Табл. 1.1 Физико-химични показатели и норми за пчелния мед<sup>[11]</sup>

### 1.2.2. Поленов анализ

Друг метод за определяне на растителния произход на меда е поленовият анализ, който също е включен в БДС. Поленовият анализ се извършва в специална лаборатория. В България това може да се направи в Централната лаборатория по ветеринарно-санитарна експертиза и екология (ЦЛВСЕЕ). Методът се състои в микроскопско изследване на меда. Наблюдават се поленовите зърна под микроскоп. Те се характеризират с определена форма, размер и строеж на външната си обвивка. На Фиг. 1.1 са показани снимки на поленови зърна в пчелния мед.



*Фиг. 1.1 Поленови зърна в пчелния мед*

Видът на поленовите зърна определя определя растителния произход на прашеца, а така и ботаническият произход на меда. Процентът поленови зърна в изследваната проба се изчислява по специална формула.

За да се приеме меда за монофлорен /от определено растение/, поленът от това растение трябва да е не по-малко от 40% от общата количество полен. За акациевия и липовия мед долната граница е 30%, а за лавандуловия - 15%.<sup>[14]</sup>

Поленовият анализ е приложен в изследователски проект на екип от академичния състав на Университета по Хранителни Технологии в Пловдив

(Д. Цанкова, Ж. Стойчев, М. Хаджикинова, Г. Терзийски, С. Лекова(ХТМУ-София)) и съдействието на техни студенти. Те съставят база данни от микроскопски снимки на различни видове поленови зърна, извличат характерните им оптични признаци за разпознаване с прилагане на филтрите на Габор, данните се минимизират по метода на главните компоненти, след което се синтезира същинският класификатор на принципа на Линеиния дискриминантен анализ. Методът дава успеваемост от 90.32% успешно разпознати проби.<sup>[25]</sup>

### **1.2.3. Спектрален анализ**

Спектроскопията е физичен метод за анализ на обекти, който се основава на способностите на обектите да поглъщат или отразяват лъчи, звуци или частици изпуснати от източника на спектъра. Съществуват различни видове спектроскопия в зависимост от обекта на изследване(атомна, молекулна, масова, ядрена) и типа на излъчването(оптична, фотоелектронна, радиовълнова, микровълнова, рентгенова, звукова).

Пчелният мед също притежава способността да преработва попадналите върху него оптични лъчи и точно това му свойство бива най-често използвано за научни разработки и опити за определяне на ботаническия му произход. Изследваните от учените методи се различават по използвания спектъра за снемане на спектралните характеристики на пробите. Най-често той бива инфрачервен, видим и ултравиолетов.

Инфрачервените спектрални характеристики като критерий за определяне на ботаническия произход на хърватски пчелния мед са успешно изследвани от група Хърватски преподаватели(Lidija Svecnjak, Nikola Biliskov, Dragan Bubalo, Domagoj Barisic) от Факултета по Агрокултура към Университета в Загреб.<sup>[19]</sup>

Спектралните характеристики от флуоресцентната спектроскопия на тридесет и два броя от три вида мед пък са използвани в изследване на академичен екип от преподаватели от Университета по Хранителни Технологии в Пловдив(Д. Цанкова, К. Николова, Т. Ефтимов(Университета на Квебек в Гатино)). Данните биват първично обработени по метода на главните компоненти, след което се съставят класификатори на основата на невронна мрежа тип многослоен перцептрон с Backpropagation обучаващ алгоритъм. Резултатите от изследването дават успеваемост на класификатора от 84.4%.<sup>[20]</sup>

#### **1.2.4. Изграждане на профил на тежките метали в меда**

Масова спектрометрия с индуктивно свързана плазма(ICP-MS) е метод за откриване на метали в много малки концентрации. Той комбинира високотемпературен източник на индуктивно свързана плазма и масова спектрометрия. Източникът на индуктивно свързаната плазма преобразува атомите на изследваната проба в йони, които след това биват разделени и измерени от масов спектрометър.

Група преподаватели(Mircea Oroian, Ancuta Prisacaru, Ana Leahu и др.) от факултета по Хранително инженерство в Румънския университет в Сучава разглеждат възможността за разпознаване на ботаническия и географския произход на няколко проби мед от различни видове мед, събрани от няколко региона в източна Румъния по количеството и вида на тежките метали(As, Cd, Cr, Cu, Fe, Hg, Mn, Ni, Pb, Zn) в състава им.<sup>[17]</sup>

Събраните данни по ICP-MS метода са обработени по метода на Линейния дискриминантен анализ. Резултатите които получават показват сравнително добра успеваемост на класифициране на меда по ботанически произход(80.8%) и ниска успеваемост при класифициране по географския произход(21.2%).

Подобни изследвания се правят и в Испания от група учени от факултета по Аналитична химия в Университета в Севиля. Които също доказват

зависимостта между наличието и концентрацията на тежките метали в състава на пчелния мед и неговият ботанически произход.<sup>[18]</sup>

### **1.2.5. Ядрено-магнитен резонанс (ЯМР)**

Ядрено-магнитният резонанс е физично явление при което атомни ядра с различен от нула спин при поставянето им под въздействието на магнитно поле поглъщат и повторно излъчват в резонанс. ЯМР спектроскопията се прилага успешно в редица сфери включително и при идентифицирането на ботаническия произход на пчелния мед. Проф. Светлана Симова заедно с колегите си от лабораторията по ЯМР от Института по органична химия при БАН работят по ЯМР спектроскопски метод за бързо и надеждно различаване на манов от полифлорен мед. Заедно са установили характерен химичен профил, позволяващ разграничаването на различни видове нектарен и манов мед от България, Мароко, Македония и Румъния.<sup>[15]</sup>

Разработено решение на базата на ЯМР представя и фирма “Bruker”. Чрез разработената от тях технология е възможно не само да се идентифицира ботаническия произход, но също така и географския произход на дванадесет вида монофлорен мед.<sup>[16]</sup>

Едно от предимствата на ЯМР метода за разпознаване на ботанически произход е, че той успешно може да класифицира дори и филтрирани медове, тоест такива при които липсват Polenovi зърна.

### 1.2.6. Други методи

Други използвани методи за изследване на автентичността, географския и ботаническия произход на пчелния мед са:

- **Изотопен анализ:** Кооперативните усилия на Австралийски учени от Университета Маккуори в Сидни и Австралийският измервателен институт изследват връзката между автентичността и географския произход на меда и съотношението на въглеродните изотопи.<sup>[21]</sup>
- **Електронен изкуствен език:** Такъв проект е разработен от кооперативен екип на Университета Сиди Мохамед Бен Абдулах в Мароко и Португалския университет в Алгарве. Електронният език „усеща“ основава на електрическата проводимост на меда, която се влияе от състава на различните видове мед.<sup>[22]</sup>
- **ДНК анализ:** Основава се на декодиране и идентифициране на ДНК на растенията от които е направен определения ботанически вид мед. Методът е изследван от Факултета по агрокултурни и хранителни науки към Университета в Болоня, Италия.<sup>[23]</sup>
- **Мултифакторен анализ:** Използват се комбинации от различни показатели и характеристики (водно съдържание, проводимост, киселинност, спектрални характеристики и др.) на меда за оценка на ботаническия произход. Тези характеристики се използват като променливи за съставяне на класификатор на основата на изкуствена невронна мрежа. Това са изследвали съвместно специалисти от университетски заведения и държавни институции в Испания и Португалия. Резултатите показват успеваемост от 95%.<sup>[24]</sup>

## **ГЛАВА ВТОРА – СТАТИСТИЧЕСКИ И ИНТЕЛИГЕНТНИ КЛАСИФИКАТОРИ НА ПЧЕЛЕН МЕД ПО БОТАНИЧЕСКИ ПРОИЗХОД**

### **2.1. Постановка на задачата за класифициране на мед по критерий „ботанически произход“**

Разглежда се задача за съставяне на автоматична система за разпознаване и класифициране на различни по ботаническия произход видове мед, чрез използване на статистически и интелигентни класификатори. В основата на разгледаните класификатори са различията в спектралните показатели между различните видове мед.

С помощта на спектрофотомер „Agilent Cary100“ се снемат спектралните характеристики на 111 проби мед, предварително окачествени от производителите им като: акациев – 34 проби; манов – 31 проби; липов – 46 проби. За минимизиране на грешките от измерване всички проби са третираны по един и същи начин. Всяка от пробите представлява предварително темперирани на стайна температура 10% разтвор на мед, поместен в измервателна кювета. Измерват се абсорбционните способности на всяка проба при облъчването и с монохроматичен сноп лъчи с дължина на вълната в диапазона от 180 до 900nm със стъпка 1nm. За целите на изследването данните биват разделени в две групи(спектъра): УВ спектър - от 180 до 380nm и Видим спектър - от 381 до 900nm.

След снемане, спектралните характеристики биват обработени по принципа на главните компоненти(PCA) с цел да се редуцира размерността на входните за класификатора величини, като в същото време се запази колкото е възможно по-голяма част от информацията. За целта се използват възможностите на софтуерния продукт „MatLab”.



Обработените по метода на главните компоненти данни в последствие биват използвани за създаване на математически модели на класификатори в средата на „MatLab”.

Реализират се следните симулационни модели на класификатори:

- Класификатор на основата на Линеен Дискриминантен Анализ, работещ с UV спектралните характеристики на пробите;
- Класификатор на основата на Линеен Дискриминантен Анализ, работещ със спектралните характеристики на пробите във видимия спектър;
- Класификатор на основата на Изкуствена Невронна Мрежа с права структура и Backpropagation обучаващ алгоритъм, работещ с UV спектралните характеристики на пробите;
- Класификатор на основата на Изкуствена Невронна Мрежа с права структура и Backpropagation обучаващ алгоритъм, работещ със спектралните характеристики на пробите във видимия спектър.

Целта е въз основа на сравнителен анализ между получените от четирите класификатора симулационни резултати да се определи кой от тях с най-голяма точност и адекватност успява да изпълни задачата за класифициране.

Оценяват се възможностите на различните класификационни алгоритми, прилагайки реалните данни на пробите пчелен мед.

## **2.2. Теоритични сведения за използваните принципи и методи**

### **2.2.1. Снемане на абсорбционни спектрални характеристики със спектрофотометър**

Спектралния анализ е метод за изследване на вещество, при който на анализ се подлагат лъчи, звуци или частици, изпуснати, погълнати, отразени или взаимодействащи по друг начин с изследвания предмет.

При фотометричните методи се използва избирателното поглъщане на светлина от молекулите на анализираното вещество и определянето на коефициентите на поглъщане на средата. При поглъщането на светлина енергията на светлинната вълна намалява при преминаването ѝ в дадено вещество. Част от енергията на вълната се преобразува във вътрешна енергия на веществото или в енергия на вторично лъчение, имащо друг спектрален състав и друга посока на разпространение (фотолуминесценция).<sup>[1]</sup>

При поглъщане на светлина атомите и молекулите на анализираното вещество преминават от основно във възбудено състояние. В зависимост от вида на частиците, поглъщащи светлина и начина на трансформиране на енергията се различават:

- Атомно-абсорбционен анализ, основава се на поглъщането на светлина от атомите на анализираното вещество.
- Молекулярно-абсорбционен анализ, т.е. анализ на поглъщане на светлина от молекулите на анализираното вещество в ултравиолетовия, видимия и инфрачервения спектър на светлината (спектрофотометрия, фотоколориметрия, ИЧ-спектроскопия).
- Анализ на поглъщане и разсейване на светлинната енергия от частиците на анализираното вещество(турбидиметрия, нефелометрия).

- Луминисцентен анализ, основава се на измерване на излъчвания, възникнали в резултат от разпределението на енергията на възбудените молекули на анализираното вещество.<sup>[1]</sup>

Поглъщането на светлината се извършва по експоненциален закон, който е формулиран от Буге и Ламберт по следния начин: абсорбираната част на светлина, пропусната през прозрачна еднородна среда не зависи от интензитета на светлинния поток, като всеки последователен слой от тази среда поглъща една и съща част от достигналата до него светлина.

Нека монохроматичен светлинен сноп с интензитет  $I_0$  пада нормално спрямо абсорбираща хомогенна среда с дебелина  $b$ . Намалението на интензитета  $dI_v$  (при определена честота  $\nu$ ), след преминаване през безкрайно тесен слой  $db$  ще се дава с израза:

$$dI_v = -\alpha_v I_v db$$

където  $\alpha_v$  е коефициент характеризиращ абсорбционната способност на средата при определена честота  $\nu$  и зависи от броя на поглъщащите частици на веществото. Знакът минус пред  $\alpha_v$  показва, че интензитетът на светлинния сноп се понижава.<sup>[2]</sup>

Горното уравнение представлява диференциален запис на закона на Буге-Ламберт. След неговото преобразуване и интегриране в интервал от 0 до  $b$  се получава:

$$\int_{I_0}^I \frac{dI_v}{I_v} = -\alpha_v \int_0^b db$$

което води до:

$$\ln I - \ln I_0 = -\alpha_v b$$

или

$$I = I_0 \exp(-\alpha_v b).$$

От Беер е било установено, че  $\alpha_v$  е пропорционален на концентрацията на разтвора. Следователно за обобщения закон на Буге-Ламберт-Беер се получава:

$$\ln\left(\frac{I}{I_0}\right) = -kbc ,$$

или

$$I = I_0 \exp(-kbc) .$$

При преминаване от натурален в десетичен логаритъм,  $\lg( )$ , се получават изразите

$$\lg\left(\frac{I}{I_0}\right) = -abc ,$$

или

$$\lg(I_0 / I) = ab \lg\left(\frac{I_0}{I}\right) = abc ,$$

където  $a$  ( $k = 2.303.a$ ) е молекулна характеристика, която не зависи от концентрацията на веществото.<sup>[2]</sup>

Отношението на интензитетите на преминалата и първоначално пропуснатата светлина  $I / I_0$  се нарича пропускливост  $T$  (Transmittance) и обикновено се дава в проценти:

$$T = \frac{I}{I_0} \cdot 100 [\%] .$$

Величината, която всъщност се измерва с обикновените спектрометри, се нарича абсорбция  $A$  (Absorbance); термините оптична плътност  $D$  или екстинкция  $E$  вече не се използват. Абсорбцията  $A$  е свързана с описаните до тук величини чрез зависимостта:

$$A = \lg \frac{1}{T} = \lg \frac{I_0}{I} = abc .$$

Когато концентрацията  $c$  е изразена в  $[\text{mol l}^{-1}]$ , а дебелината на слоя  $b$  в  $[\text{cm}]$ , абсорбируемостта  $a$  се превръща в моларна абсорбируемост  $\epsilon$ :

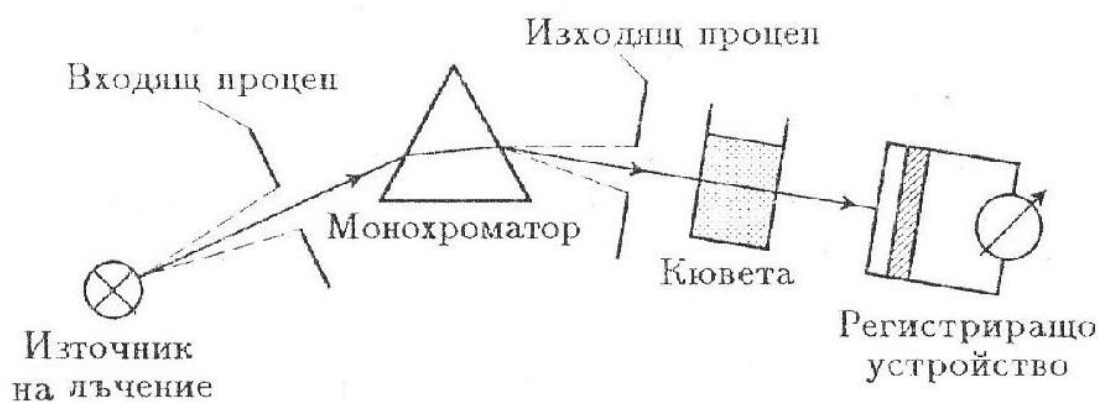
$$\epsilon = \frac{A}{b[\text{cm}] \cdot c[\text{mol.l}^{-1}]} ,$$

т.е.  $\epsilon$  има измерение  $[\text{l mol}^{-1} \text{ cm}^{-1}]$ .<sup>[2]</sup>

Моларната абсорбируемост е свързана с вероятността за преход; при голяма вероятност за преход  $\epsilon$  е висока и обратно. Стойността на  $\epsilon_{\max}$ , измерена в електронните абсорбционни спектри, се дава като характеристика към всяка ивица, наред с дължината на вълната на абсорбционния максимум.

При изследване на разтвори, те се поставят в кювети. Обикновено се използват две еднакви кювети – едната с изследвания разтвор, а другата – с разтворителя. По този начин може да се измерят  $I$  и  $I_0$ , участващи в закона на Буге-Ламберт-Беер.

Спектралните абсорбционни характеристики се измерват с уред наречен спектрофотометър. Основните структурни компоненти на спектрофотометра са: източник на светлина(в общия случай на бяла светлина); оптична система за насочване и фокусиране на светлината; монохроматор или спектрална решетка, за разлагане на бялата светлина; втора оптична система за насочване и фокусиране на разложената светлина; кювета държател в който се поставя изследваната проба; регистриращо устройство, което представлява фотодетектор(фотоклетка) превръщащата светлинния сигнал в електрически и галванометър за отчитане показанията на детектора. Принципната схема на спектрофотометра е показана на долната фигура(Фиг. 2.1).



Фиг. 2.1 Принципна схема на спектрофотометър

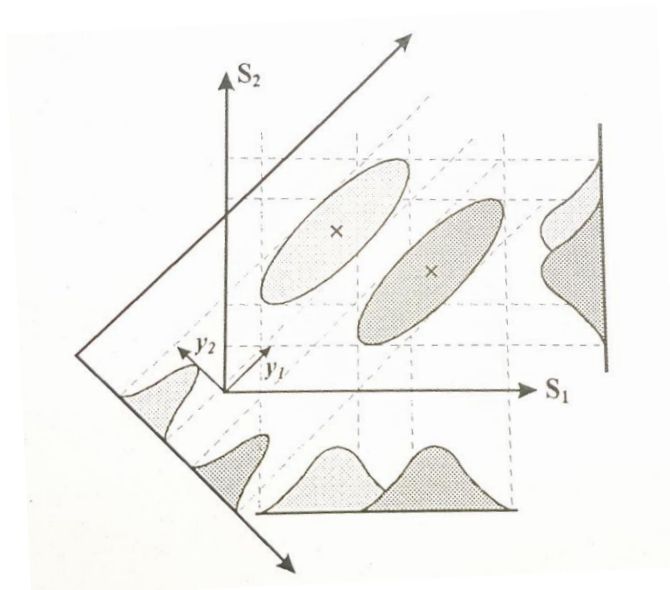
За анализ на ботаническия произход на меда са използвани данните снети от абсорбционния спектрален анализ на всяка от пробите мед снети с лабораторен спектрофотометър „Aligen Cary 100“.

### **2.2.2. Метод на главните компоненти (МГК)**

Методът на главните компоненти(МГК) е инструмент за редуциране на размерността. Прилага се за редуциране на голям брой променливи до по-малък такъв, който въпреки това да съдържа повечето от информацията която се съдържа в големия брой.

Първото описание на метода на главните компоненти е представено от Пиърсън през 1901г. При МГК се диагонализира ковариационната матрица на данните и по този начин се декорелира сигнала в преобразуваната област, минимизирайки средноквадратичната грешка при компресията на данните и обединяването на най-високо енергийните(с най-голяма дисперсия) от тях в най-малко на брой коефициенти на преобразуване. Методът намира направленията в пространството на променливите, по протежениет на които разстоянието между точките за съответните проби е най-голямо. Това позволява намиране линейните комбинации от изходните променливи, които допринасят най-много две проби да бъдат различни помежду си. Тези направления или комбинации се наричат главни компоненти. Изчисляват се итеративно като първият главен компонент носи най-много информация(обяснява най-голяма част от дисперсията в изходните данни), вторията носи максималния дял от остатъчната информация(тази, която не е била обяснена от първия главен компонент и т.н. Главните компоненти образуват нова координатна система от ортогонални вектори, подредени по възходящ ред. Новата координатна система позволява лесно интерпретиране на структурата на данните.<sup>[3]</sup>

На Фиг. 2.2 е показано, как два клъстера в равнината на признаците ( $S_1$  и  $S_2$ ) са неразделими по всяка от осите и как могат да се класифицират само с един главен компонент ( $y_2$ ).



Фиг. 2.2

Главната задача при МГК е разлагането на матрицата от данни  $X$  на две части: матрица на факторните резултати  $S$  и матрица на факторните тегла  $L$ . Това разлагане се осъществява на няколко стъпки.<sup>[4]</sup>

- 1) През пространство на обектите се прекарва нова ос, която да описва по-добре обектите, отколкото началните оси, т.е. остатъците (разлики между стойности по новите и стари оси) са най-малки, а вариациите в системата най-големи. Тази ос (променлива) се нарича първи „главен компонент“ или „латентен (скрит) фактор“ и често се означава като PC1. PC1 е линейна комбинация от старите оси, като всяка стара ос участва във формирането ѝ с определени, различни по стойност, тегла.

$$PC1 = l_1.x_1 + l_2.x_2 + \dots + l_m.x_m$$

- 2) Въвежда се следващо направление в пространството (нов фактор, нова ос), което е перпендикулярно на първото. Този фактор се означава като PC2. По начина на описване на обектите, втората ос не е равностойна на първата. PC1 описва най-голяма част от вариацията на системата, а всяка следваща ос описва все по-малка.

Теоретично, броят на нововъведените оси е равен на броя на старите оси, но на практика се избират по-малък брой нови оси.

- 3) Определянето на статистически значимите фактори е една от найтрудните задачи при прилагането на МГК. Най-често прилаганите методи са два: да се използват толкова фактора, колкото да опишат преобладаващата част от дисперсията на данните, обикновено до 90%; и да се използват само фактори, чиито собствени стойности ( $\lambda_i$ ) са значими, т.е.  $\lambda_i > 1$ .

МГК дава възможност да се намали броят на характеристиките, описващи системата от  $m$  на  $f$  ( $f < m$ ). Тази редукция на размерността е съпроводена с минимална загуба на информация, която се съдържа в матрицата (Фиг. 2.3):<sup>[4]</sup>

$$\begin{array}{|c|} \hline \mathbf{X} \\ \hline (m \times n) \\ \hline \end{array} = \begin{array}{|c|} \hline \mathbf{S} \\ \hline (m \times f) \\ \hline \end{array} \times \begin{array}{|c|} \hline \mathbf{L} \\ \hline (f \times n) \\ \hline \end{array} + \begin{array}{|c|} \hline \mathbf{E} \\ \hline (m \times n) \\ \hline \end{array}$$

Фиг. 2.3 Разлагане на входната матрица  $X$  на матрица на факторните резултати ( $S$ ), матрица на факторните тегла ( $L$ ) и матрица на остатъците ( $E$ )

$X$  – входна матрица

$S$  – матрица на факторните резултати – съдържат новите координати на обектите в пространството на главните компоненти

$L$  – матрица на факторните тегла – дефинира участието на всяка изходна



характеристика във формирането на даден фактор чрез стойностите на т. нар. факторни тегла (1)

$E$  – матрица на остатъците – съдържа загубата на информация при редукицията на размерността.<sup>[4]</sup>

Методът на главните компоненти се прилага често за извличане на високо-пространствени данни при разпознаването на образи. В много от случаите обаче не съществува линейна разделимост, тогава се използва модифицирания метод на главните компоненти, т.нар. нелинеен метод на главните компоненти.

Важно при използване на компонентния анализ(линеен или нелинеен) е да се направи разграничение между приложенията, дали ще бъде използван само за намаляване на размерността, за идентификация на определен набор от функции, въз основа на зададени критерии или др.<sup>[3]</sup>

### **2.2.3. Линеен Дискриминантен Анализ (ЛДА)**

Дискриминантният анализ е статистически метод, който въз основа на множество променливи характеризиращи дадени обекти извършва класифицирането им в две или повече предварително зададени различни една от друга групи. Групите се определят от стойности на една зависима променлива, наречена категорийна, а характеристиките на обектите се представят с променливи, наречени предиктори.<sup>[6]</sup> Всеки от обектите бива класифициран към определена група въз основа на информацията която носят предикторите. За първи път логиката на дискриминантният анализ е представена от Роналд Фишер през 1936г. В основата на метода е тезата, че обектите от една и съща група си приличат по множество показатели.

В линейния дискриминантен анализ (ЛДА) е частен случай на дискриминантният анализ. Той е метод, който използва линейна комбинация от характеристики, които класифицира два или повече обекта. Основното

изискване на което трябва да отговарят даните, за да може принципът на линейния дискриминантен анализ да е успешно приложим е, те да бъдат линейно разделими. Получената комбинация може да се използва като линейен класификатор или както по-често се случва – с цел дименсионен редуктор на променливите преди по-късно класифициране на обектите. ЛДА в основата си е много близък с анализа на вероятностите (ANOVA), методът на главните компоненти (МГК) и регресионния анализ.

Линейният дискриминантен анализ предполага, че разпределенията на количествените променливи (предикторите) вътре в класовете са нормални и се различават само по средните си стойности. Тогава процедурата произвежда оптимални решаващи правила. Разбира се, тя може да се използва и при случайна (по групиращата променлива) извадка, но появата на празен клас е недопустима.<sup>[7]</sup>

Ако са известни стойностите на поне една от групиращите променливи, това е съществена априорна информация - фиксирането на тази променлива е еквивалентно на задаването на нулева априорна вероятност за поне половината от класовете.<sup>[7]</sup>

Нека допуснем, че са известни вероятностите  $\{p(g)\}$ , груповите средни  $\{m(g)\}$  и вътрешно - груповата ковариационна матрица

$$C(g) = C, \quad (g = 1, 2, \dots, G)$$

Тогава по формулата на Бейс, апостериорната вероятност за класификация в класа  $g$  на наблюдението  $(x, .)$  ще бъде

$$q(g) = c.p(g).f(x, m(g), C) .$$

Тук  $f$  е плътността на нормалното разпределение със средна стойност  $m(g)$  и ковариационна матрица  $C$ , а  $c$  е нормираща константа

(такава, че  $Pq(g) = I$ ). Съгласно принципа за максимално правдоподобие, класифицира се по правилото:

$$\hat{g} = \max h : q(h) .$$

Класификационните правила могат да бъдат записани във вида:

$$p(\hat{g}).f(x, m(\hat{g}), C) \geq p(h).f(x, m(h), C) , \quad (h = 1, 2, \dots, G) .$$

След логаритмуване и съкращаване, получаваме:

$$b(\hat{g})'x + a(\hat{g}) \geq b(h)'x + a(h) , \quad (h = 1, 2, \dots, G) ,$$

Векторът  $b(g)$  и числото  $a(g)$  се получават по формулите:

$$b(h) = m(h)'C^{-1} , \quad a(h) = \log p(h) - m(h)'C^{-1}m(h) .$$

Оттук се вижда, че в горните неравенства участвуват линейни функции относно променливите и това обстоятелство е дало името на линейния дискриминантен анализ.<sup>[7]</sup>

ЛДА се използва в редица сфери където е нужен статистически анализ на данните като например разпознаване на образи и обучение на машини.

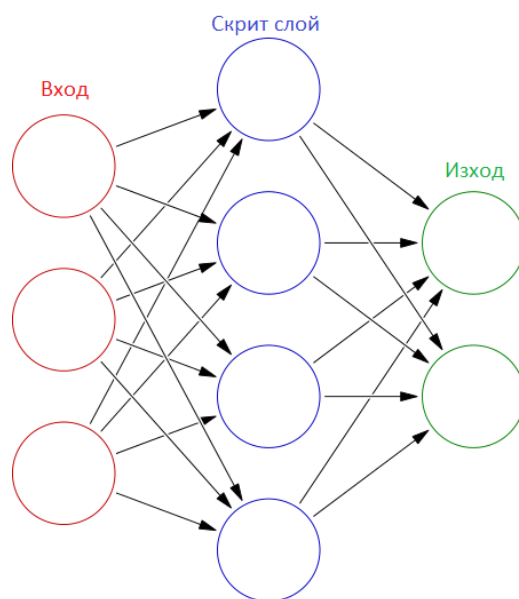
#### 2.2.4. Изкуствени Невронни Мрежи (ИНМ)

През 1943 г. Маккулок и Питс предлагат модел на неврон като двоичен прагов елемент. С това се слага началото на моделирането на изкуствени невронни мрежи (ИНМ), намиращи приложение в производственото планиране и управление на процеси, система за идентификация, проверка и прогнозиране и др.<sup>[8]</sup>

Изкуствените невронни мрежи (ИНМ) са математически обекти, които изобразяват едно векторно пространство в друго.<sup>[9]</sup> По аналогия на човешкия мозък, ИНМ са изградени от прости изчислителни елементи наречени

неврони. Всеки неврон приема сигнали от другите (под формата на числа), сумира ги, като сумата минава през активационна функция и така определя своята активация (степен на възбуда), която се предава по изходящите връзки към другите неврони. Всяка връзка има тегло, което, умножавайки се със сигнала, определя неговата значимост (сила). Теглата на връзките са аналогични на силата на синаптичните импулси, предавани между биологичните неврони. Отрицателна стойност на теглото съответства на потискащ импулс, а положителна – на възбуждащ.<sup>[10]</sup>

На долната фигура (Фиг. 2.4) е показана общата структура на една изкуствена невронна мрежа. В общия случай тя се състои от групи свързани помежду си възли разделени на слоеве. В общия случай слоеве са входен, скрит/и и изходен слой. Информацията в мрежата се предава от ляво на дясно(по посока на стрелките). Кръгчетата представляват отделни неврони всеки от които има собствена изчислителна функция. В реализациите на ИНМ сигналът който се предава между невроните е реално число, а изходът на всеки неврон се изчислява като нелинейна функция на сумата от входовете му.



Фиг. 2.4 Структура на ИНМ

Невронните мрежи се делят на няколко вида в зависимост от: начина на свързване между невроните; вида на използваната функция и стойността на сигналите, които се обработват от мрежата.

Засега най-много приложения в спектроскопията имат невронните мрежи, с права структура разпространяващи сигналите напред (forward feed) и коригиращи грешките в обратна посока (back propagation of errors). Невронните мрежи с обратно разпространение на грешките са изградени от няколко слоя неврони: входен слой, няколко скрити слоя и един изходен слой. На Фиг. 2.4 е изобразена невронна мрежа с един скрит слой. Всеки неврон получава „претеглени“ сигнали от всички свързани с него предходни неврони от входния слой и изпраща своя изходен (обработен) сигнал към всички неврони на следващия слой. Полученият сигнал на един неврон,  $Net_j$ , е сума от произведенията на коефициентите на мрежата (силата на връзките -  $W_{j,k}$ ) и изходните сигнали от предхождащия слой ( $Out_k$ ). Записано като уравнение това изглежда по следния начин:

$$Net_j = \sum W_{j,k} * Out_k$$

$Net_j$  се обработва по-нататък от т.н. изглаждаща (squashing) функция, за да се получи изходен сигнал в интервала 0.0 - 1.0. Могат да се използват различни функции, но най-често употребявани са приложенията, където сигналите се обработват по следната формула:

$$Out_j = 1 / [1 + \exp(-Net_j + offset_j)]$$

Преобразуващата функция превръща сигнала в числа между нула и единица. Без тази функция и скрития слой, ИНМ щеше да дава резултати, аналогични на линейната многопроменлива регресия, но наличието на един или повече скрити слоя, както и тази нелинейна преобразуваща функция водят до нелинейно изображение на входните в изходни вектори.

Когато невронните мрежи се използват за обработка на спектри, входните вектори се формират от спектралната крива или набор от други спектрални признаци, които се представят като един многомерен вектор. Изходните вектори се получават от информацията за структурата на съответните химични съединения; те също са вектори, но с координати нула или единица, показващи наличието (1) или отсъствието (0) на даден структурен фрагмент. Съответно броят на входните/изходните неврони съвпада с размерността на входните/изходните вектори.<sup>[9]</sup>

За да се получат спектро-структурните корелации между спектрите и съответните структурни дескриптори е необходимо невронната мрежа да бъде обучена с подходяща по обем и представителност извадка от спектри на съединения с известна структура, наречена обучаваща извадка. Входните вектори (спектрите) се пропускат последователно през мрежата, като получените изходни вектори се сравняват с целевите изходни вектори, отразяващи структурата на съединенията.<sup>[9]</sup>

Получените грешки се "разпространяват" обратно като това променя стойностите на коефициентите на мрежата:

$$W_{i,j}(new) = W_{i,j}(old) + \eta * \Delta W_{i,j}(calc) + \mu * \Delta W_{i,j}(old)$$

$\Delta W_{i,j}(calc)$  включва разпространението на грешката и се изчислява чрез параметри, специфични за даденият слой от неврони. Параметърът  $\eta$  се нарича скорост на обучение (learning rate) и определя големината на градиента, т.е. влияе върху сходимостта на оптимизационната процедура. Параметърът  $\mu$  се нарича моментен коефициент (momentum factor) и заедно с  $W_{i,j}(old)$  спомага за преодоляване на локалните минимума, като запазва стабилна посоката на преход към минимума – точно съответствие на реалните и желаните изходи.

Като критерий за спиране на обучението се използва относителната промяна на средно-квадратичната грешка (MSE), на изходите от мрежата. MSE се изчислява по следната формула: [9]

$$MSE = \sum \sum (O_{k,j} - T_{k,j})^2 ,$$

където лявата сума е по  $k$ , т.е. по всички образи в обучаващата извадка, а втората сума е по  $j$ , т.е. по всички изходи от невронната мрежа.  $O_{k,j}$  е целевият, а  $T_{k,j}$  реалния изход за  $k$ -тия образ на  $j$ -тия изходен неврон.

Именно стойностите на коефициентите  $W_{i,j}$  съдържат спектро-структурните корелации, което запазва знанията разпределени по много подобен на човешкия мозък начин.

Предимствата на невронните мрежи са, че те са: метод позволяващ дефинирането на сложни нелинейни зависимости; могат да работят с голям брой променливи; лесни са за използване; не се нуждаят от априорна информация за вида на търсената зависимост.

### 2.3. Реализация на ЛДА класификатор

За реализация на симулационен модел на класификатор работещ по метода на линейния дискриминантен анализ се използва софтуерният програмен пакет на „MATLAB”. Реализират се два много подобни програмни алгоритми с тази разлика, че единият работи със спектралните характеристики на прбите снети в УВ спектъра, а другият с тези, снети във видимия спектър. Логическият смисъл и принцип на работа на алгоритмите е обяснен в долните стъпки, а програмните кодове са разписани в Приложение 1 и Приложение 2.

*Стъпка 1.* Инициализация. Зарежда се файл (HoneyData\_UV\_New2.mat) съдържащ спектрални характеристики на абсорбция на акациев, липов и манов мед. Задават се: броят на класовете (три) на класификатора на мед; броят N на

използваните главни компоненти. Броячът на цикли “is” за изпълнение на теста “leave one out cross validation test” се поставя на 1. Нулират се броячите на успешно класифицираните мостри за всеки от трите класа.

*Стъпка 2.* Актуализира се броят на спектралните характеристики в съответствие с тествания критерий: за теста се изважда една характеристика – тази с пореден номер “is”. Останалите се използват като входни данни за синтеза (обучението) на класификатора.

*Стъпка 3.* Стартира се подпрограма “princomp” (команда на MATLAB), съдържаща метода на главната компонента. Изчисляват се: коефициентите за всяка главна компонента; входните вектори в новия базис, т.е. в пространството на главните компоненти; вариациите на главните компоненти. Избират се първите N главни компоненти.

*Стъпка 4.* Стартира се подпрограма “classify” (команда на MATLAB), съдържаща метода на линейния дискриминантен анализ. За всяка от входните данни (изразена чрез първите N главни компоненти) се предсказва линейният клас към който принадлежи.

*Стъпка 5.* Предсказаният клас за всяка мостра се съпоставя с наблюдавания: Ако съвпадат, то броячът на успешно класифицираните мостри в този клас се увеличава с 1.

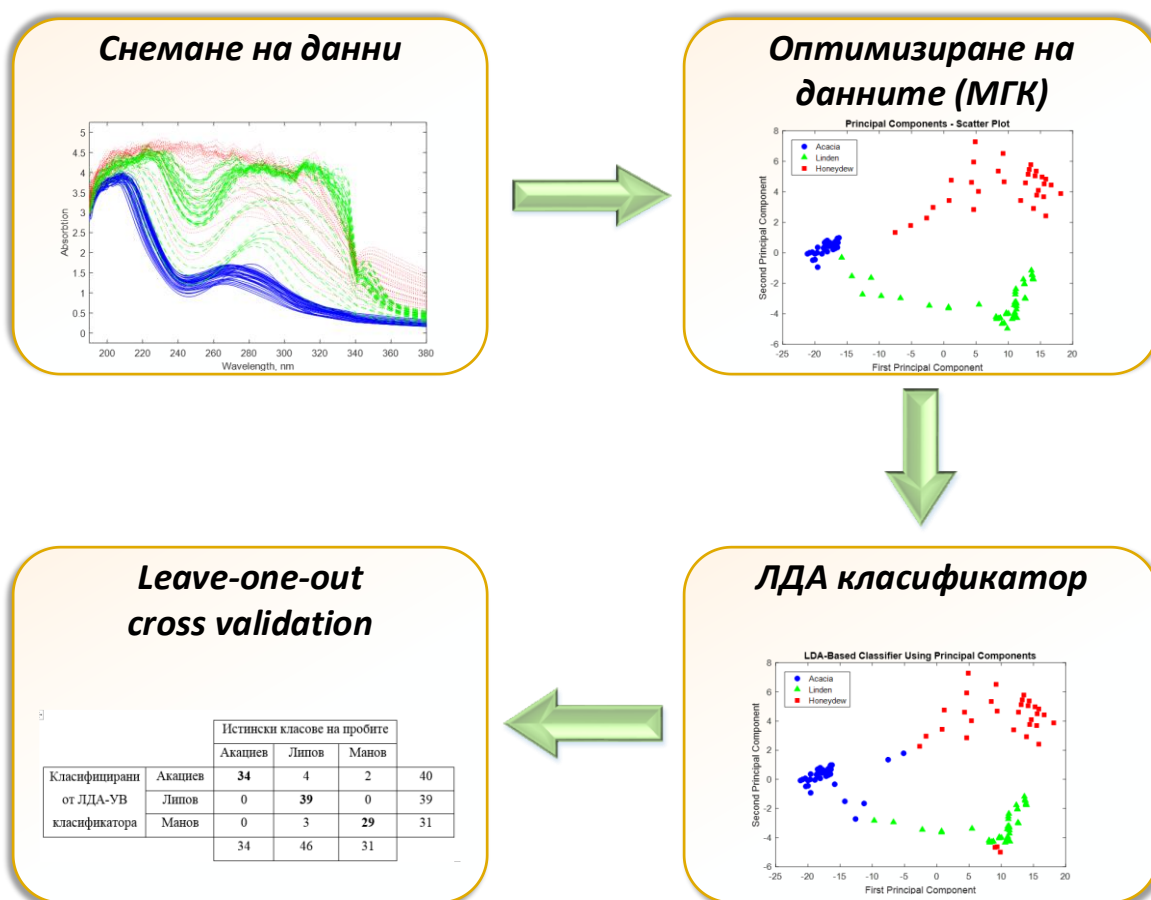
*Стъпка 6.* Броячът на цикли “is” за изпълнение на теста “leave one out cross validation test” се увеличава с 1. Проверява се дали “is” е по-малко или равно на броя на обучаващите мостри. Ако отговорът е “да”, то се отива в стъпка 2.

*Стъпка 7.* Визуализират се получените резултати: графики на спектралните характеристики, на наблюдаваните класове според анализа на



главната компонента и на предсказаните класове според линейния дискриминантен анализ; таблица за успеха на класификацията, в съответствие с предсказаните и наблюдаваните класове от мостри на пчелен мед.

На Фиг. 2.5 е представена блоковата схема на ЛДА класификатора.



Фиг. 2.5 Блок-схема на ЛДА класификатор

## **2.4. Реализация на ИНМ класификатор с права структура и Backpropagation обучаващ алгоритъм**

За реализация на симулационен модел на класификатор работещ на принципа на изкуствените невронни мрежи се използва софтуерният програмен пакет на „MATLAB”. Реализират се два много подобни програмни алгоритми с тази разлика, че единият работи със спектралните характеристики на прбите снети в УВ спектъра, а другият с тези, снети във видимия спектър. Логическият смисъл и принцип на работа на алгоритмите е обяснен в долните стъпки, а програмните кодове са разписани в Приложение 3 и Приложение 4.

*Стъпка 1.* Инициализация. Зарежда се файл (HoneyData\_UV\_New2.mat) съдържащ спектрални характеристики на абсорбция на акациев, липов и манов мед. Задават се: броят на класовете (три) на класификатора на мед; броят  $N$  на използваните главни компоненти; броят на скритите слоеве  $S$  в невронната мрежа със съответния брой неврони в тях и активационните им функции. Броячът на цикли “is” за изпълнение на теста “leave one out cross validation test” се поставя на 1. Нулират се броячите на успешно класифицираните мостри за всеки от трите класа.

*Стъпка 2.* Актуализира се броят на спектралните характеристики в съответствие с тестващия критерий: за теста се изважда една характеристика – тази с пореден номер “is”. Останалите се използват като входни данни за синтеза (обучението) на класификатора.

*Стъпка 3.* Стартира се подпрограма “princomp” (команда на MATLAB), съдържаща метода на главната компонента. Изчисляват се: коефициентите за всяка главна компонента; входните вектори в новия базис, т.е. в

пространството на главните компоненти; вариациите на главните компоненти. Избират се първите  $N$  главни компоненти.

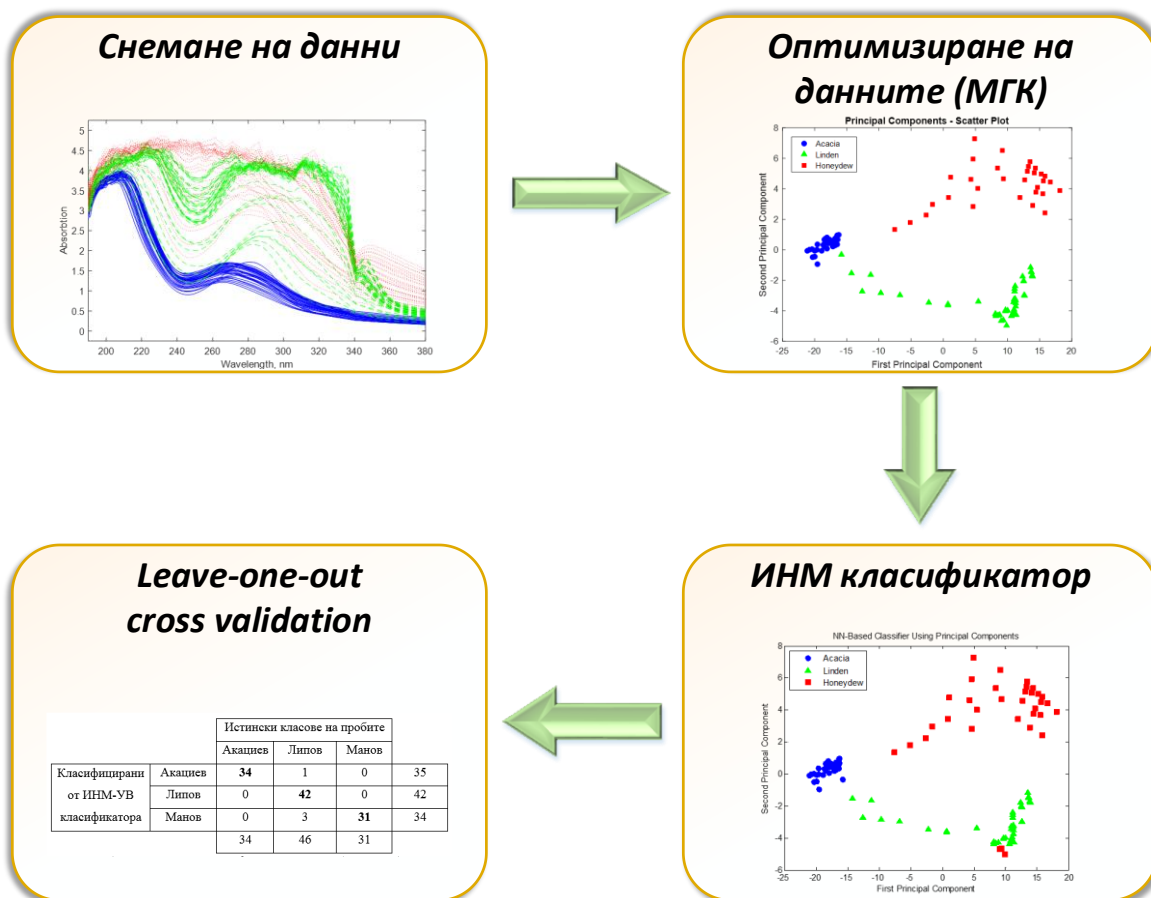
*Стъпка 4.* Инициализира се невронната мрежа. Задават се входовете и изходите на мрежата. Задават се параметрите за обучението на мрежата – скорост, коефициенти за намаляване и увеличаване на скоростта на обучение, максимален брой епохи и други. За всяка от входните данни (изразена чрез първите  $N$  главни компоненти) се предсказва линейният клас към който принадлежи.

*Стъпка 5.* Предсказаният клас за всяка мостра се съпоставя с наблюдавания: Ако съвпадат, то броячът на успешно класифицираните мостри в този клас се увеличава с 1.

*Стъпка 6.* Броячът на цикли “is” за изпълнение на теста “leave one out cross validation test” се увеличава с 1. Проверява се дали “is” е по-малко или равно на броя на обучаващите мостри. Ако отговорът е “да”, то се отива в стъпка 2.

*Стъпка 7.* Визуализират се получените резултати: графики на спектралните характеристики, на наблюдаваните класове според анализа на главната компонента и на предсказаните класове според изкуствената невронна мрежа; таблица за успеха на класификацията, в съответствие с предсказаните и наблюдаваните класове от мостри на пчелен мед.

На Фиг. 2.6 е представена блоковата схема на ИНМ класификатора.



Фиг. 2.6 Блок-схема на ИЛМ класификатор

## ГЛАВА ТРЕТА – ВАЛИДАЦИЯ И СРАВНИТЕЛЕН АНАЛИЗ НА РАЗГЛЕДАНИТЕ КЛАСИФИКАТОРИ

Валидацията е процес на оценяване на истинността на данните и формиране на становище, съобразно резултатите. В тази глава ще се извърши и сравнителният анализ на резултатите които дават разгледаните класификатори.

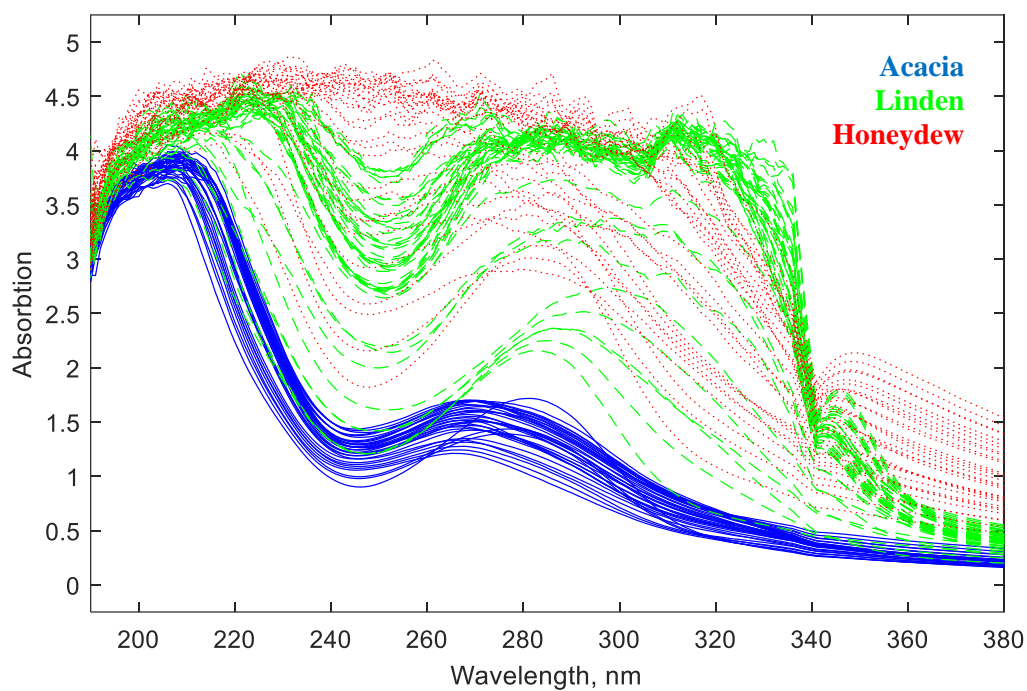
За постигането на целта на дипломната работа – синтезиране на статистически и интелигентни класификатори за разпознаване ботаническия

произход на проби мед, са разработени алгоритми(представени във втора глава от дипломната работа) и са реализирани четири програми:

- ЛДА класификатор по УВ спектър – програмният код на класификатора е представен в Приложение 1;
- ЛДА класификатор по видимия спектър – програмният код на класификатора е представен в Приложение 2;
- ИНМ класификатор с Backpropagation алгоритъм по УВ спектър - програмният код на класификатора е представен в Приложение 3;
- ИНМ класификатор с Backpropagation алгоритъм по УВ спектър - програмният код на класификатора е представен в Приложение 4.

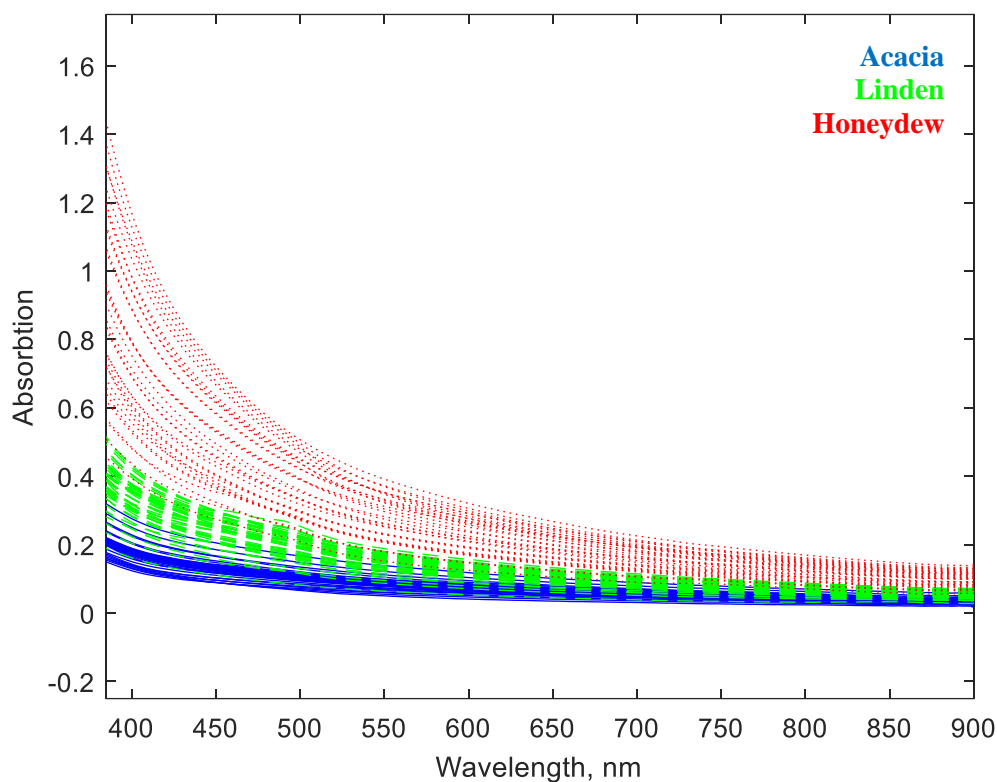
Една от целите на проекта е да се сравнят и възможностите на класификаторите в УВ и видимия спектрални диапазона. Поради това резултатите от симулациите също ще бъдат представени по отделно в тези диапазони.

На Фиг. 3.1 е представена обща абсорбционната спектрална характеристика на всички пробите мед в УВ спектъра. Най-ясно разграничаване на абсорбционните свойства на трите вида мед се наблюдава в средните диапазони (240-260nm) на спектъра, както и в крайните (360-380nm).



Фиг. 3.1 Абсорбционна спектрална характеристика в UV спектъра

На Фиг. 3.2 е представена обща абсорбционната спектрална характеристика на всички пробите мед във видимия спектър. Тук се наблюдават значително по-близки абсорбционни поведенчески качества на пробите, като най-ясно те се разграничават в диапазона (400-500nm).



Фиг. 3.2 Абсорбционна спектрална характеристика във видимия спектър

### 3.1. ЛДА класификатор по УВ спектър

В Табл. 3.1 е показана грешката при класифициране на пробите мед.

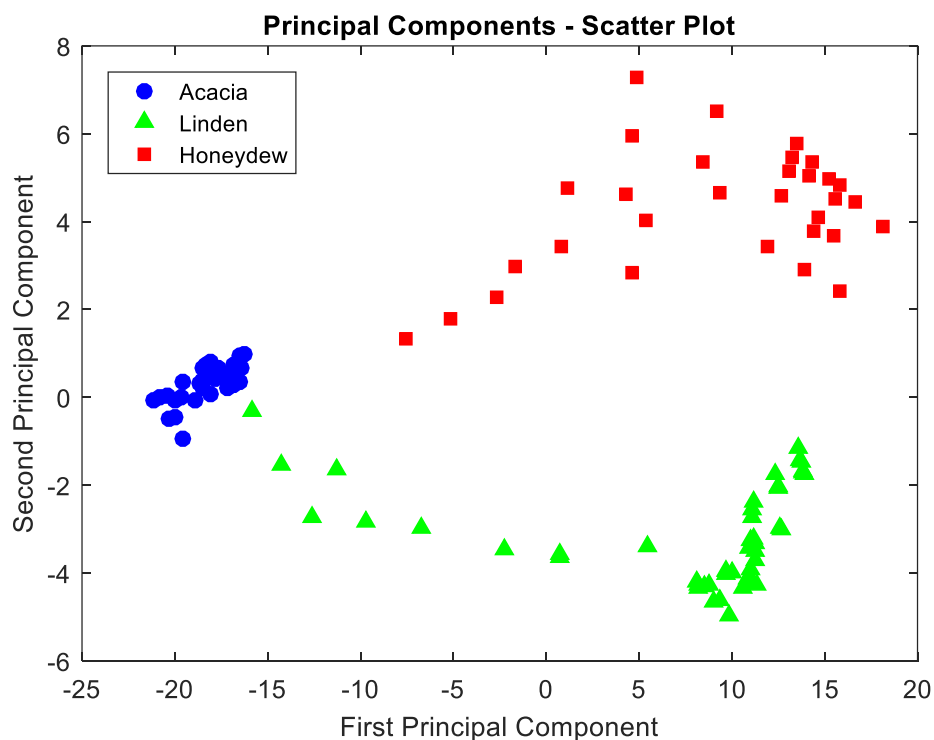
От таблицата се вижда, че класификаторът сравнително успешно класифицира пробите мед. Неуспешно са класифицирани: 4 проби липов мед като акациев; 2 проби манов мед като акациев; 3 проби липов мед като манов.

Класификаторът прави най-много грешки при класифицирането на липов мед, по-малко при класифицирането на манов мед и нито една грешка при класифицирането на акациевия мед.

		Истински класове на пробите			
		Акациев	Липов	Манов	
Класифицирани от ЛДА-УВ класификатора	Акациев	<b>34</b>	4	2	40
	Липов	0	<b>39</b>	0	39
	Манов	0	3	<b>29</b>	31
		34	46	31	

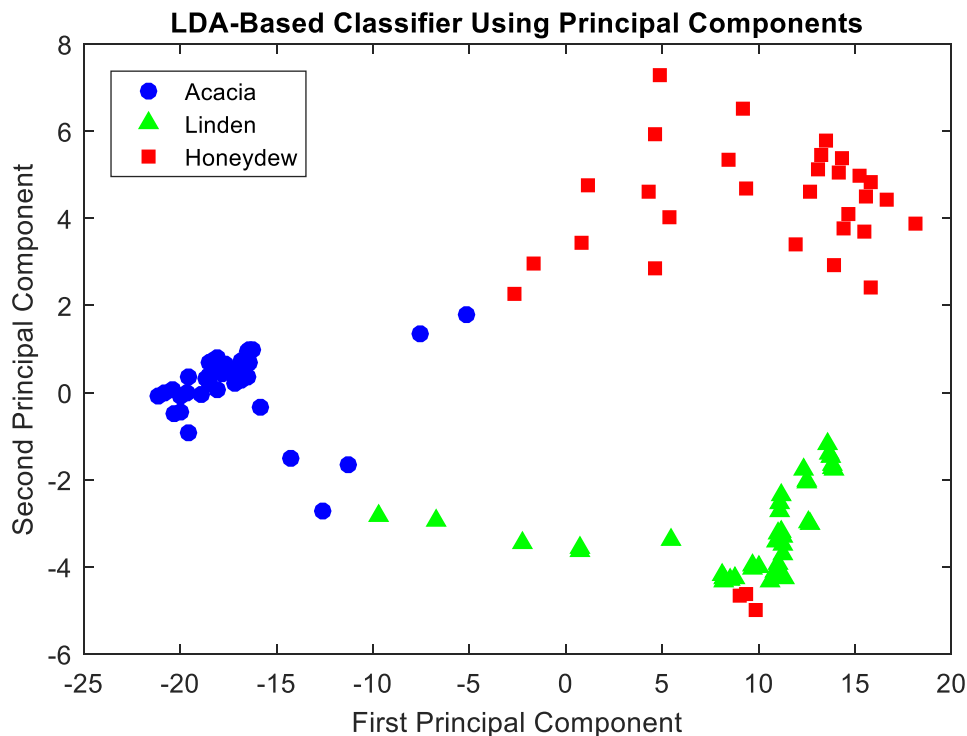
Табл.3.1 Грешка при класификацията на проби мед по ботанически произход с ЛДА класификатор по УВ спектър

На Фиг. 3.3 и Фиг. 3.4 в графичен вид по метода на главните компоненти са представени съответно реалните(истинските) клъстерни организации на пробите и изчислените от ЛДА-УВ класификатора такива. Съпоставка на двете фигури показва коят проба точно класификаторът не успява да класифицира коректно.



Фиг. 3.3 Графично представяне на реалната принадлежност на пробите мед по ботанически произход





Фиг. 3.4 Графично представяне на изчислената от ЛДА-УВ класификатора принадлежност на пробите мед по ботанически произход

Успеваемостта на представеният ЛДА класификатор по УВ спектър да разпредели пробите мед в три класа според ботаническият им произход е 91.89%, съответно грешката при разпределението е 8.11% .

### 3.2. ЛДА класификатор по видимия спектър

В Табл. 3.2 е показана грешката при класифициране на пробите мед.

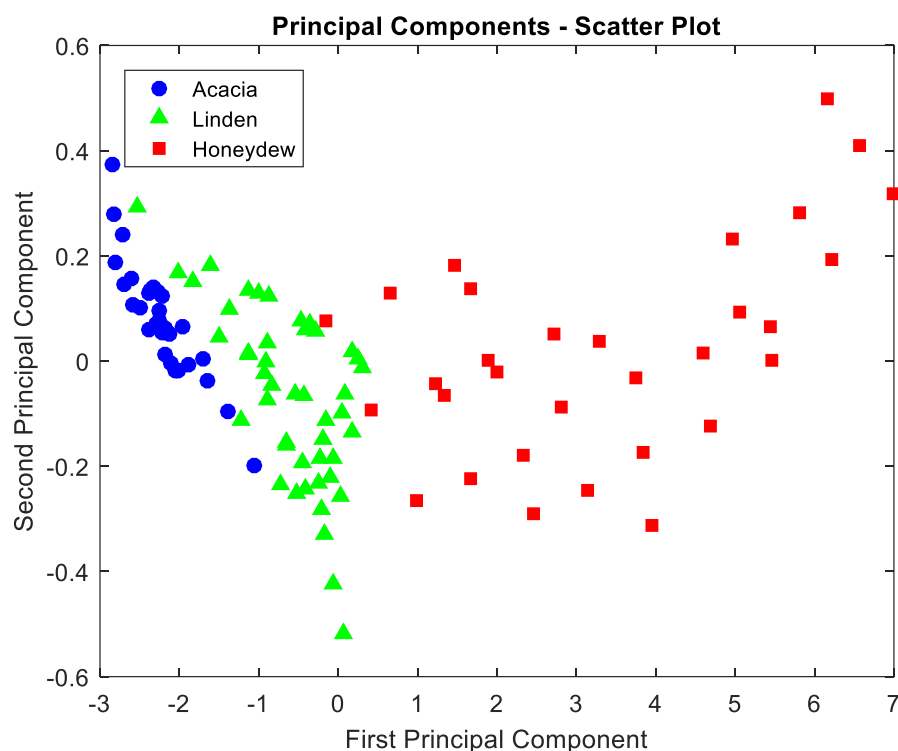
От таблицата се вижда, че класификаторът сравнително успешно класифицира пробите мед. Неуспешно са класифицирани: 9 проби липов мед като акациев; 3 проби акациев мед като липов; 6 проби манов мед като липов.

Класификаторът прави грешни класификации на проби и от трите вида мед, като най-малко са тези при акациевите проби, а най-много при липовите.

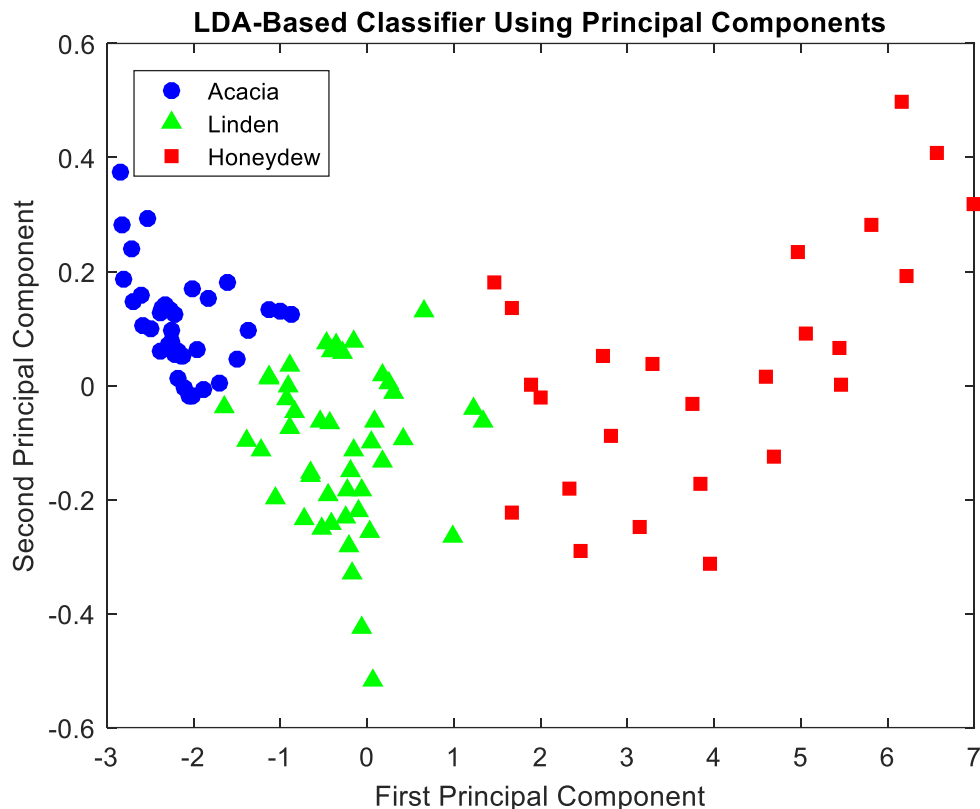
		Истински класове на пробите			
		Акациев	Липов	Манов	
Класифицирани от ЛДА-Видим класификатора	Акациев	<b>31</b>	9	0	40
	Липов	3	<b>37</b>	6	46
	Манов	0	0	<b>25</b>	25
		34	46	31	

Табл.3.2 Грешка при класификацията на проби мед по ботанически произход с ЛДА класификатор по видимия спектър

На Фиг. 3.5 и Фиг. 3.6 в графичен вид по метода на главните компоненти са представени съответно реалните(истинските) клъстерни организации на пробите и изчислените от ЛДА-Видим класификатора такива. Съпоставка на двете фигури показва коя проба точно класификаторът не успява да класифицира коректно.



Фиг. 3.5 Графично представяне на реалната принадлежност на пробите мед по ботанически произход



Фиг. 3.6 Графично представяне на изчислената от ЛДА-Видим класификатора принадлежност на пробите мед по ботанически произход

Успеваемостта на представеният ЛДА класификатор по видимия спектър да разпредели пробите мед в три класа според ботаническият им произход е 83.78%, съответно грешката при разпределението е 16.22% .

### 3.3. ИНМ класификатор с Backpropagation алгоритъм по УВ спектър

В Табл. 3.3 е показана грешката при класифициране на пробите мед.

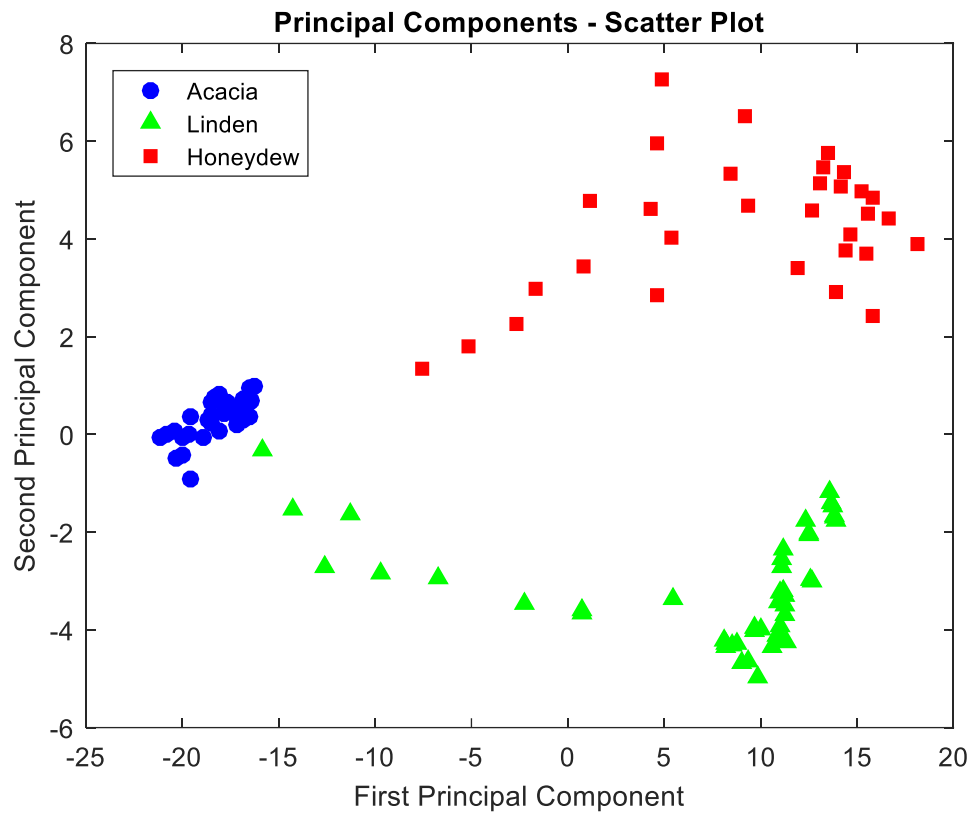
От таблицата се вижда, че класификаторът доста успешно класифицира пробите мед. Неуспешно са класифицирани: 1 проба липов мед като акациев; 3 проби липов мед като манов. Класификаторът успешно класифицира всички

проби акациев и манов мед. Отново затруднения за класификатора представляват пробите от липов мед.

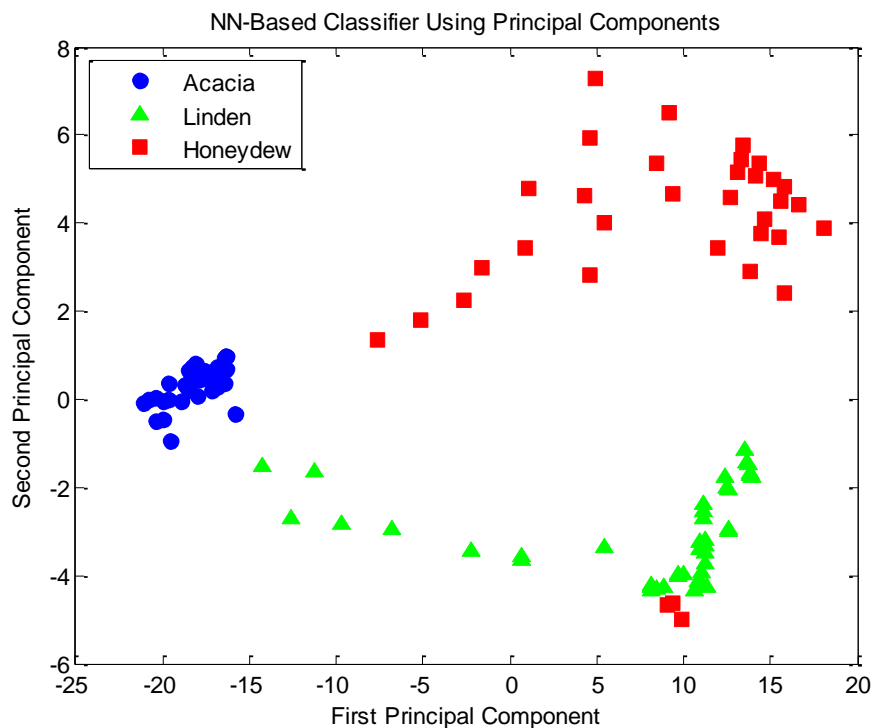
		Истински класове на пробите			
		Акациев	Липов	Манов	
Класифицирани от ИНМ-УВ класификатора	Акациев	<b>34</b>	1	0	35
	Липов	0	<b>42</b>	0	42
	Манов	0	3	<b>31</b>	34
		34	46	31	

*Табл.3.3 Грешка при класификацията на проби мед по ботанически произход с ИНМ класификатор по УВ спектър*

На Фиг. 3.7 и Фиг. 3.8 в графичен вид по метода на главните компоненти са представени съответно реалните(истинските) клъстерни организации на пробите и изчислените от ИНМ-УВ класификатора такива. Съпоставка на двете фигури показва коя проба точно класификаторът не успява да класифицира коректно.



Фиг. 3.7 Графично представяне на реалната принадлежност на пробите мед по ботанически произход



Фиг. 3.8 Графично представяне на изчислената от ИНМ-УВ класификатора принадлежност на пробите мед по ботанически произход

Успеваемостта на представеният ИНМ класификатор по УВ спектър да разпредели пробите мед в три класа според ботаническият им произход е 96.39%, съответно грешката при разпределението е 3.61% .

### 3.4. ИНМ класификатор с Backpropagation алгоритъм по видимия спектър

В Табл. 3.4 е показана грешката при класифициране на пробите мед.

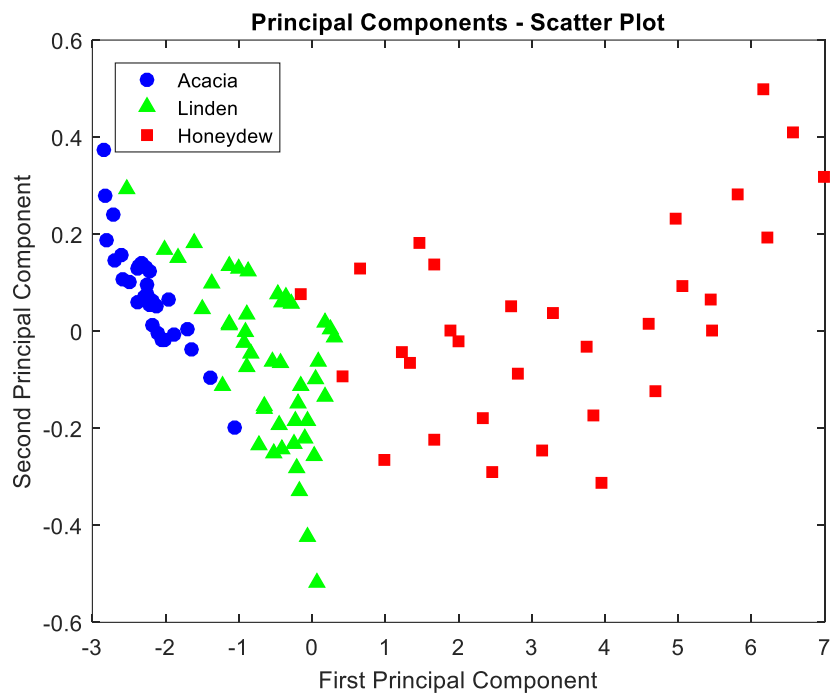
От таблицата се вижда, че класификаторът сравнително успешно класифицира пробите мед. Неуспешно са класифицирани: 2 проби липов мед като акациев; 3 проби акациев мед като липов; 2 проби манов мед като липов.

Класификаторът прави относително еднакъв брой грешни класификации на проби и от трите вида мед.

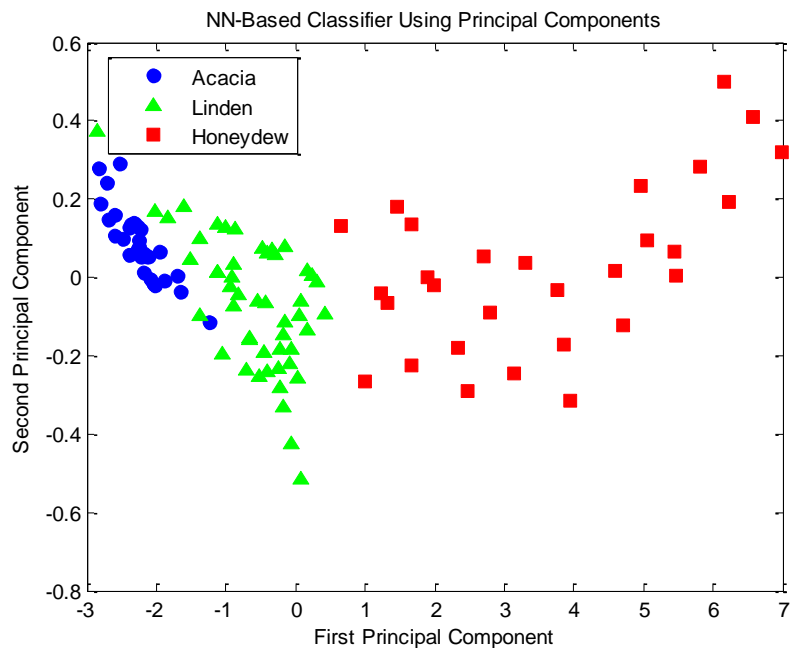
		Истински класове на пробите			
		Акациев	Липов	Манов	
Класифицирани от ИНМ-Видим класификатора	Акациев	<b>31</b>	2	0	33
	Липов	3	<b>44</b>	2	49
	Манов	0	0	<b>29</b>	29
		34	46	31	

*Табл.3.4 Грешка при класификацията на проби мед по ботанически произход с ИНМ класификатор по видимия спектър*

На Фиг. 3.9 и Фиг. 3.10 в графичен вид по метода на главните компоненти са представени съответно реалните(истинските) клъстерни организации на пробите и изчислените от ИНМ-Видим класификатора такива. Съпоставка на двете фигури показва коя проба точно класификаторът не успява да класифицира коректно.



Фиг. 3.9 Графично представяне на реалната принадлежност на пробите мед по ботанически произход



Фиг. 3.10 Графично представяне на изчислената от ИНМ-Видим класификатора принадлежност на пробите мед по ботанически произход



Успеваемостта на представеният ИНМ класификатор по видимия спектър да разпредели пробите мед в три класа според ботаническият им произход е 93.69%, съответно грешката при разпределението е 6.31% .

### **3.5. Анализ и сравняване на резултатите**

В долната Табл. 3.5 е представено сравнение на четирите разгледани класификатора по основният им критерий за надеждност – процентната успеваемост на класифициране на пробите мед.

Забелязва се, че класификаторите работещи със спектралните характеристики на пробите във ултравиолетовия спектър дават по-добри резултати. Това се дължи на по-ясно разграничените абсорбционни свойства на пробите по отношение на УВ спектъра. Най-добри резултати дава класификаторът работещ с невронна мрежа и Backpropagation алгоритъм използващ спектралните характеристики на пробите от УВ спектъра.

<b>Класификатор</b>	<b>Успеваемост, %</b>
ЛДА по УВ спектър	91.89
ЛДА по Видим спектър	83.78
ИНМ по УВ спектър	96.39
ИНМ по Видим спектър	93.69

*Табл. 3.5 Сравнение на класификаторите по успеваемост*

## ГЛАВА ЧЕТВЪРТА – ИЗВОДИ

Въпросът с фалшифицирането на храните е един от най-големите проблеми в хранителната индустрия в световен мащаб. Това налага въвеждането на все повече и по-надеждни високотехнологични решения които да си сътрудничат с установените стандарти и разпоредби за качествен контрол на храните.

В дипломната работа беше направена съпоставка на интелигентния и статистическия подходи при изграждане на технологични системи за проверка на автентичния ботанически произход на 111 проби пчелен мед от три ботанически вида(акациев, липов и манов). Също така бяха сравнени и методиките(спектрален анализ на пробите пчелен мед във видимия и в УВ спектъра) за снемане на суровите данни за класифициране с оглед на количеството информация което носят.

Класификаторът базиран на методът на Линейния дискриминантен анализ показва добри резултати при използване на спектралните данни от видимата област и още по-добри при използване на тези от УВ спектъра.

Резултатите демонстрирани от класификатора реализиран на базара на изкуствена невронна мрежа са с около 5 до 10% по-добри отново в полза на класификацията извършена по УВ спектралните данни.

Направените изследвания и симулации показват високата прецизност, надеждност на невронните мрежи и загатват потенциала на тяхното приложение в сферата на хранителната индустрия.

## ИЗПОЛЗВАНА ЛИТЕРАТУРА

- [1] – Синеков Д., Курсов проект на тема „Фотометър, фотометрични методи“, НБУ, [http://ebox.nbu.bg/stds/ne4/KURSOV%20PROEKT35\\_Denis\\_PHOTOMETER.pdf](http://ebox.nbu.bg/stds/ne4/KURSOV%20PROEKT35_Denis_PHOTOMETER.pdf)
- [2] – Андреев Г., „Молекулна спектроскопия“, Изд. ПУ “П. Хилендарски”, Пловдив, 1999г., [http://kosnos.net/issue11/absorbtion\\_law.html](http://kosnos.net/issue11/absorbtion_law.html)
- [3] – Начев В., Лекции по „Интелигентни системи“, УХТ-Пловдив, 2017г.
- [4] – Йотова Г., Автореферат „Екометрични и Екотоксикологични Методи при Оценка на Обекти от Околната Среда“, СУ „Св. Климент Охридски“, София, 2016г., [http://digilib.nalis.bg/dspviewverb/srv/image\\_singpdf/dcddde67-248a-423a-a706-4d1a26f3676d](http://digilib.nalis.bg/dspviewverb/srv/image_singpdf/dcddde67-248a-423a-a706-4d1a26f3676d)
- [6] – Нончева В., Откриване на знания в данните или Полезни статистически методи: теория, софтуер, приложения; 2010г.
- [7] – Въндев Д., Записки по приложна статистика 2, СУ Св. Климент Охридски, София, 2003г., <https://store.fmi.uni-sofia.bg/fmi/statist/personal/vandev/lectures/APPLSTAT2.pdf>
- [8] – Беров Т., Възможности за приложение на изкуствени невронни мрежи (ИНМ) при транспортното планиране, ВТУ „Т. Каблешков“, София, 2015г., [https://trans-motauto.com/sbornik/2015/3/32.POSSIBILITIES%20FOR%20APPLICATION%20OF%20ARTIFICIAL%20NEURAL%20NETWORKS%20\(ANN\)%20IN%20TRANSPORTATION%20PLANNING.pdf](https://trans-motauto.com/sbornik/2015/3/32.POSSIBILITIES%20FOR%20APPLICATION%20OF%20ARTIFICIAL%20NEURAL%20NETWORKS%20(ANN)%20IN%20TRANSPORTATION%20PLANNING.pdf)
- [9] – Пенчев П., Лекции по Хемометрия, ПУ „Паисий Хилендарски“, Пловдив, <http://web.uni-plovdiv.bg/plamenpenchev/lectures/chemos/lecture11.pdf>
- [10] – „Уикипедия“ на тема ”Изкуствена невронна мрежа“, [https://bg.wikipedia.org/wiki/%D0%98%D0%B7%D0%BA%D1%83%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%B0\\_%D0%BD%D0%B5%D0%B2%D1%80%D0%BE%D0%BD%D0%BD%D0%B0\\_%D0%BC%D1%80%D0%B5%D0%B6%D0%B0](https://bg.wikipedia.org/wiki/%D0%98%D0%B7%D0%BA%D1%83%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B2%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D1%80%D0%B5%D0%B6%D0%B0)
- [11] – Динков Д., Лекция по „Ветеринарно-Санитарна експертиза на пчелен мед“, 2013г.
- [12] – Цанкова Д., Проект по „Изследване на автентичността и качеството на пчелен мед с помощта на спектрометричен анализ и софтверни техники“, УХТ-Пловдив

- [13] – Интернет сататия на тема „Изследване на състава, физичните и органолептичните свойства на медът - част 1“, 2016г.,  
<https://kapkamed.com/pcheli/izsledvane-na-systava-fizichnite-i-organoleptichnite-svoystva-na-medyt-chast-1>
- [14] – Иванов Ц., „Стандартизация и контрол на качеството на пчелните продукти – история, настояще, бъдеще“, Институт по животновъдни науки, Костинброд, 2004г.,  
<https://sinor.bg/2399-STANDARTIZACIYA-I-KONTROL-NA-KACHESTVOTO-NA-PChELNITE-PRODUKTI---ISTORIYA-NASTOYAShE-BADEStE>
- [15] – Костадинова В., „Ядрено-магнитен резонанс разпознава на 100% истинския от фалшивия мед“, 2017г., <https://news.bg/health/yadreno-magniten-rezonans-razpoznava-na-100-istinskiya-ot-falshiviya-med.html>
- [16] - <https://www.bruker.com/products/mr/nmr-food-screening/honey-profiling-module-of-the-nmr-foodscreeener/technical-details.html>
- [17] – Oroian M., Prisacaru A., Hretcanu E., Stroe S., Leahu A., Buculei A., “Heavy Metals Profile in Honey as a Potential Indicator of Botanical and Geographical Origin”, University of Suceava, 2016,  
<https://www.tandfonline.com/doi/full/10.1080/10942912.2015.1107578>
- [18] - Fernandez-Torres R., Perez-Bernal J., Bello-Lopez M., "Mineral content and botanical origin of Spanish honeys", University of Seville, 2005,  
<https://www.sciencedirect.com/science/article/pii/S0039914004004515?via%3Dihub#!>
- [19] – Svacnjak L., Biliskov N., Bubalo D., Barisic D., "Application of Infrared Spectroscopy in Honey Analysis", University of Zagreb, 2011,  
[https://www.researchgate.net/publication/287319664\\_Application\\_of\\_Infrared\\_Spectroscopy\\_in\\_Honey\\_Analysis](https://www.researchgate.net/publication/287319664_Application_of_Infrared_Spectroscopy_in_Honey_Analysis)
- [20] – Tsankova D., Nikolova K., Evtimov T., "FLUORESCENCE SPECTROSCOPY, COLORIMETRY AND NEURAL NETWORKS IN DISTINGUISHING DIFFERENT TYPES OF HONEY", University of Food Technologies, Plovdiv, 2015,  
[https://www.researchgate.net/publication/283504156\\_FLUORESCENCE\\_SPECTROSCOPY\\_COLORIMETRY\\_AND\\_NEURAL\\_NETWORKS\\_IN\\_DISTINGUISHING\\_DIFFERENT\\_TYPES\\_OF\\_HONEY](https://www.researchgate.net/publication/283504156_FLUORESCENCE_SPECTROSCOPY_COLORIMETRY_AND_NEURAL_NETWORKS_IN_DISTINGUISHING_DIFFERENT_TYPES_OF_HONEY)

- [21] – Zhou X., Taylor M., Salouros H., Prasad S., "Authenticity and geographic origin of global honeys determined using carbon isotope ratios and trace elements", Macquarie University, Sydney, 2018, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6168535/>
- [22] – Elamine Y., Inacio P., Lyoussi B., "Insight into the sensing mechanism of an impedance based electronic tongue for honey botanic origin discrimination", University of Algarve, 2019, <https://www.sciencedirect.com/science/article/pii/S0925400519300425>
- [23] – Ribani A., Schiavo G., Bertolini F., "Application of next generation semiconductor based sequencing to detect the botanical composition of monofloral, polyfloral and honeydew honey", University of Bologna, 2017, <https://www.sciencedirect.com/science/article/pii/S0956713517305662#!>
- [24] – Anjos O., Iglesias C., Peres F., "Neural networks applied to discriminate botanical origin of honeys", 2014, <https://www.sciencedirect.com/science/article/pii/S0308814614018597>
- [25] - Nikolov DN, DD Tsankova, Features Extraction for Pollen Recognition in Honey Using Gabor Filters, Food Science and Applied Biotechnology, 1 (2), 2018, 86 – 95, <https://www.ijfsab.com/index.php/fsab/article/download/11/40/>

## Приложение 1

### Honey\_LDA\_onlyUV\_1out\_New2.m

```
clear all

% ----- Initialization -----
% From file HoneyData_UV_New2.mat:
% wLength, AbsAcacia_UV, AbsLinden_UV, AbsHoneydew_UV
load HoneyData_UV_New2

AbsAcacia_UV=AbsAcacia(521:711,:);
AbsLinden_UV=AbsLinden(521:711,:);
AbsHoneydew_UV=AbsHoneydew(521:711,:);

wLength=wLength(521:711,:);
wAcacia=wAcacia(521:711,:);
wLinden=wLinden(521:711,:);
wHoneydew=wHoneydew(521:711,:);

figure(1)
plot(wLength,AbsAcacia_UV,'b'), hold on,
plot(wLength,AbsLinden_UV,'g--'), hold on,
plot(wLength,AbsHoneydew_UV,'r:'), hold on
axis([190 380 -0.25 5.25])
xlabel('Wavelength, nm');
ylabel('Absorbtion');

nAcacia=size(AbsAcacia_UV,2); % Number of Acacia samples
nLinden=size(AbsLinden_UV,2); % Number of Linden samples
nHoneydew=size(AbsHoneydew_UV,2); % Number of Honeydew samples
nWaves=size(wLength,1); % Number of wave samples
nHoney=nAcacia+nLinden+nHoneydew; % Number of Honey samples - All

iAk=[1:nAcacia];
iLip=[1:nLinden]+nAcacia;
iMan=[1:nHoneydew]+nAcacia+nLinden;
IndexKind=[iAk iLip iMan];
B=[1.*ones(1,nAcacia) 2.*ones(1,nLinden) 3.*ones(1,nHoneydew)];
Honey=[AbsAcacia_UV AbsLinden_UV AbsHoneydew_UV];

% ----- PCA -----
AbsorbValues=(Honey)';
%mapcaplot(AbsorbValues,TitleAbs_cell11);
[pc, zscores, pcvars] = princomp(AbsorbValues);
pcvars./sum(pcvars) * 100
cumsum(pcvars./sum(pcvars) * 100)

N=2; %Number of principal components
```

```

zscoresN=zscores(:,1:N);
nMostri=nHoney;

figure(2)
xAcacia=[]; yAcacia=[]; xLinden=[]; yLinden=[]; xHoneydew=[];
yHoneydew=[];
for i=1:nMostri
    if find(IndexKind(1,i)==iAk),
        xAcacia=[xAcacia, zscores(i,1)];
        yAcacia=[yAcacia, zscores(i,2)];
    elseif find(IndexKind(1,i)==iLip),
        xLinden=[xLinden, zscores(i,1)];
        yLinden=[yLinden, zscores(i,2)];
    elseif find(IndexKind(1,i)==iMan),
        xHoneydew=[xHoneydew, zscores(i,1)];
        yHoneydew=[yHoneydew, zscores(i,2)];
    end
end
plot(xAcacia,yAcacia,'bo','MarkerFaceColor','b'), hold on,
plot(xLinden,yLinden,'g^','MarkerFaceColor','g'), hold on,
plot(xHoneydew,yHoneydew,'rs','MarkerFaceColor','r'), hold on
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('Principal Components - Scatter Plot');
legend('Acacia','Linden','Honeydew',...
        'Location','NW')

pc_arr=[]; zscores_arr=[]; pcvars_arr=[];
CurrentScore_arr=[];
DecisionThreshold=0.01;

ClassTabTest=zeros(3,3);
IndexTabTest11=[]; IndexTabTest12=[]; IndexTabTest13=[];
IndexTabTest21=[]; IndexTabTest22=[]; IndexTabTest23=[];
IndexTabTest31=[]; IndexTabTest32=[]; IndexTabTest33=[];
LowAccuracyTest=0; IndexLowAccuracyTest=[];

numberClassPredict=zeros(1,nMostri);
maxP_arr=[]; maxP_Index_arr=[]; Paposts_arr=[];

% ----- Leave 1 out cross validation test -----
nTest=1;
for is=1:nMostri
    is

    CurrentValue=AbsorbValues(is,:);
    CurrentScore=zscoresN(is,:);
    CurrentScore_arr=[CurrentScore_arr;CurrentScore];

```

```

AbsorbValues_lout=AbsorbValues; AbsorbValues_lout(is,:)=[];
[pc, zscores, pcvars] = princomp(AbsorbValues_lout);
zscoresN_lout=zscores(:,1:N);

% ----- Initialization -----
nSamples=nMostri-1;
dataClasses=zscoresN_lout; % Learning Input Data
(zscores_L(:,1:N))
numberClasses=B'; numberClasses(is)=[]; % Learning Output Data

dataClassTest=CurrentScore; % Learning Input Data;
numberClassTest=B(1,is); % Testing Output Data

dataClass1=[]; dataClass2=[]; dataClass3=[];
indexClass1=[]; indexClass2=[]; indexClass3=[]; yd_arr=[];

for i=1:nSamples
    if numberClasses(i,1)==1,
        dataClass1=[dataClass1; dataClasses(i,:)];
indexClass1=[indexClass1;i]; yd=[1 0 0];
    elseif numberClasses(i,1)==2,
        dataClass2=[dataClass2; dataClasses(i,:)];
indexClass2=[indexClass2;i]; yd=[0 1 0];
    else,
        dataClass3=[dataClass3; dataClasses(i,:)];
indexClass3=[indexClass3;i]; yd=[0 0 1];
    end
    yd_arr=[yd_arr; yd];
end
nClass1=size(dataClass1,1); nClass2=size(dataClass2,1);
nClass3=size(dataClass3,1);
nLevel=size(dataClasses,2);

% ----- LDA -----
% Linear Discriminant Analysis

LinClass =
classify(dataClassTest,dataClasses,numberClasses,'linear');
bad = ~strcmp(LinClass,numberClassTest);

% Aposteriornite veroqtnosti po otnoshenie na testvashtata
izvadka
decision=zeros(1,nTest);
if LinClass==1, Papost1=1; Papost2=0; Papost3=0;
elseif LinClass==2, Papost1=0; Papost2=1; Papost3=0;
elseif LinClass==3, Papost1=0; Papost2=0; Papost3=1;
end
Paposts=[Papost1 Papost2 Papost3];
[maxP,maxP_Index]=max(Paposts);
decision(:,1)=maxP_Index;

```



```

maxP_arr=[maxP_arr maxP];
maxP_Index_arr=[maxP_Index_arr maxP_Index];
Paposts_arr=[Paposts_arr; Paposts];

% Obshtata greshka pri klasifikaciqta za testvashtata izvadka
% --(chestotata na greshno klasificiranite obekti)----
errors=find(numberClassTest~=decision'),
%TotalError=100*(length(errors)/length(numberClassTest))

% ----- Validation -----
% ----- Verify predicted classes in test data -----

yPredictTest=Paposts_arr(is,:);
yPredictMax=maxP_arr(is);
Index_yMax=maxP_Index_arr(is);

if (Index_yMax==1)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(1,1)=ClassTabTest(1,1)+1;
        IndexTabTest11=[IndexTabTest11 is];
    elseif numberClassTest==2
        ClassTabTest(1,2)=ClassTabTest(1,2)+1;
        IndexTabTest12=[IndexTabTest12 is];
    elseif numberClassTest==3
        ClassTabTest(1,3)=ClassTabTest(1,3)+1;
        IndexTabTest13=[IndexTabTest13 is];
    else, 1
    end

elseif (Index_yMax==2)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(2,1)=ClassTabTest(2,1)+1;
        IndexTabTest21=[IndexTabTest21 is];
    elseif numberClassTest==2
        ClassTabTest(2,2)=ClassTabTest(2,2)+1;
        IndexTabTest22=[IndexTabTest22 is];
    elseif numberClassTest==3
        ClassTabTest(2,3)=ClassTabTest(2,3)+1;
        IndexTabTest23=[IndexTabTest23 is];
    else, 2
    end

elseif (Index_yMax==3)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(3,1)=ClassTabTest(3,1)+1;
        IndexTabTest31=[IndexTabTest31 is];
    elseif numberClassTest==2
        ClassTabTest(3,2)=ClassTabTest(3,2)+1;
        IndexTabTest32=[IndexTabTest32 is];
    elseif numberClassTest==3

```

```

        ClassTabTest(3,3)=ClassTabTest(3,3)+1;
        IndexTabTest33=[IndexTabTest33 is];
    else, 3
    end

    else
        LowAccuracyTest=LowAccuracyTest+1;
        IndexLowAccuracyTest=[IndexLowAccuracyTest is];
    end, % if

end

ClassTabTest
LowAccuracyTest

AllSuccess=ClassTabTest(1,1)+ClassTabTest(2,2)+ClassTabTest(3,3);
AllTest=sum(sum(ClassTabTest)); AllTest=AllTest+LowAccuracyTest;
Success=AllSuccess./AllTest.*100

figure(3)
xAcacia=[]; yAcacia=[]; xLinden=[]; yLinden=[]; xHoneydew=[];
yHoneydew=[];
for i=1:nMostri
    if maxP_Index_arr(i)==1,
        xAcacia=[xAcacia, CurrentScore_arr(i,1)];
        yAcacia=[yAcacia, CurrentScore_arr(i,2)];
    elseif maxP_Index_arr(i)==2,
        xLinden=[xLinden, CurrentScore_arr(i,1)];
        yLinden=[yLinden, CurrentScore_arr(i,2)];
    elseif maxP_Index_arr(i)==3,
        xHoneydew=[xHoneydew, CurrentScore_arr(i,1)];
        yHoneydew=[yHoneydew, CurrentScore_arr(i,2)];
    end
end
plot(xAcacia,yAcacia,'bo','MarkerFaceColor','b'), hold on,
plot(xLinden,yLinden,'g^','MarkerFaceColor','g'), hold on,
plot(xHoneydew,yHoneydew,'rs','MarkerFaceColor','r'), hold on
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('LDA-Based Classifier Using Principal Components');
legend('Acacia','Linden','Honeydew',...
        'Location','NW')

save Honey_LDA_onlyUV_New2

```

## Приложение 2

### Honey\_LDA\_delutedVis\_1out\_New2.m

```
clear all

% ----- Initialization -----
% From file HoneyData_UV_New2.mat:
% wLength, AbsAcacia_UV, AbsLinden_UV, AbsHoneydew_UV
load HoneyData_UV_New2

AbsAcacia_UV=AbsAcacia(1:520,:);
AbsLinden_UV=AbsLinden(1:520,:);
AbsHoneydew_UV=AbsHoneydew(1:520,:);

wLength=wLength(1:520,:);
wAcacia=wAcacia(1:520,:);
wLinden=wLinden(1:520,:);
wHoneydew=wHoneydew(1:520,:);

figure(1)
plot(wLength,AbsAcacia_UV,'b'), hold on,
plot(wLength,AbsLinden_UV,'g--'), hold on,
plot(wLength,AbsHoneydew_UV,'r:'), hold on
axis([385 900 -0.25 1.75])
xlabel('Wavelength, nm');
ylabel('Absorbtion');

nAcacia=size(AbsAcacia_UV,2); % Number of Acacia samples
nLinden=size(AbsLinden_UV,2); % Number of Linden samples
nHoneydew=size(AbsHoneydew_UV,2); % Number of Honeydew samples
nWaves=size(wLength,1); % Number of wave samples
nHoney=nAcacia+nLinden+nHoneydew; % Number of Honey samples - All

iAk=[1:nAcacia];
iLip=[1:nLinden]+nAcacia;
iMan=[1:nHoneydew]+nAcacia+nLinden;
IndexKind=[iAk iLip iMan];
B=[1.*ones(1,nAcacia) 2.*ones(1,nLinden) 3.*ones(1,nHoneydew)];
Honey=[AbsAcacia_UV AbsLinden_UV AbsHoneydew_UV];

% ----- PCA -----
AbsorbValues=(Honey)';
%mapcaplot(AbsorbValues,TitleAbs_cell11);
[pc, zscores, pcvars] = princomp(AbsorbValues);
pcvars./sum(pcvars) * 100
cumsum(pcvars./sum(pcvars) * 100)

N=2; %Number of principal components
```

```

zscoresN=zscores(:,1:N);
nMostri=nHoney;

figure(2)
xAcacia=[]; yAcacia=[]; xLinden=[]; yLinden=[]; xHoneydew=[];
yHoneydew=[];
for i=1:nMostri
    if find(IndexKind(1,i)==iAk),
        xAcacia=[xAcacia, zscores(i,1)];
        yAcacia=[yAcacia, zscores(i,2)];
    elseif find(IndexKind(1,i)==iLip),
        xLinden=[xLinden, zscores(i,1)];
        yLinden=[yLinden, zscores(i,2)];
    elseif find(IndexKind(1,i)==iMan),
        xHoneydew=[xHoneydew, zscores(i,1)];
        yHoneydew=[yHoneydew, zscores(i,2)];
    end
end
plot(xAcacia,yAcacia,'bo','MarkerFaceColor','b'), hold on,
plot(xLinden,yLinden,'g^','MarkerFaceColor','g'), hold on,
plot(xHoneydew,yHoneydew,'rs','MarkerFaceColor','r'), hold on
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('Principal Components - Scatter Plot');
legend('Acacia','Linden','Honeydew',...
        'Location','NW')

pc_arr=[]; zscores_arr=[]; pcvars_arr=[];
CurrentScore_arr=[];
DecisionThreshold=0.01;

ClassTabTest=zeros(3,3);
IndexTabTest11=[]; IndexTabTest12=[]; IndexTabTest13=[];
IndexTabTest21=[]; IndexTabTest22=[]; IndexTabTest23=[];
IndexTabTest31=[]; IndexTabTest32=[]; IndexTabTest33=[];
LowAccuracyTest=0; IndexLowAccuracyTest=[];

numberClassPredict=zeros(1,nMostri);
maxP_arr=[]; maxP_Index_arr=[]; Paposts_arr=[];

% ----- Leave 1 out cross validation test -----
nTest=1;
for is=1:nMostri
    is

    CurrentValue=AbsorbValues(is,:);
    CurrentScore=zscoresN(is,:);
    CurrentScore_arr=[CurrentScore_arr;CurrentScore];

```

```

AbsorbValues_lout=AbsorbValues; AbsorbValues_lout(is,:)=[];
[pc, zscores, pcvars] = princomp(AbsorbValues_lout);
zscoresN_lout=zscores(:,1:N);

% ----- Initialization -----
nSamples=nMostri-1;
dataClasses=zscoresN_lout; % Learning Input Data
(zscores_L(:,1:N))
numberClasses=B'; numberClasses(is)=[]; % Learning Output Data

dataClassTest=CurrentScore; % Learning Input Data;
numberClassTest=B(1,is); % Testing Output Data

dataClass1=[]; dataClass2=[]; dataClass3=[];
indexClass1=[]; indexClass2=[]; indexClass3=[]; yd_arr=[];

for i=1:nSamples
    if numberClasses(i,1)==1,
        dataClass1=[dataClass1; dataClasses(i,:)];
indexClass1=[indexClass1;i]; yd=[1 0 0];
    elseif numberClasses(i,1)==2,
        dataClass2=[dataClass2; dataClasses(i,:)];
indexClass2=[indexClass2;i]; yd=[0 1 0];
    else,
        dataClass3=[dataClass3; dataClasses(i,:)];
indexClass3=[indexClass3;i]; yd=[0 0 1];
    end
    yd_arr=[yd_arr; yd];
end
nClass1=size(dataClass1,1); nClass2=size(dataClass2,1);
nClass3=size(dataClass3,1);
nLevel=size(dataClasses,2);

% ----- LDA -----
% Linear Discriminant Analysis

LinClass =
classify(dataClassTest,dataClasses,numberClasses,'linear');
bad = ~strcmp(LinClass,numberClassTest);

% --- Aposteriornite veroqtnosti po otnoshenie na testvashtata
izvadka ---
decision=zeros(1,nTest);
if LinClass==1, Papost1=1; Papost2=0; Papost3=0;
elseif LinClass==2, Papost1=0; Papost2=1; Papost3=0;
elseif LinClass==3, Papost1=0; Papost2=0; Papost3=1;
end
Paposts=[Papost1 Papost2 Papost3];
[maxP,maxP_Index]=max(Paposts);
decision(:,1)=maxP_Index;

```

```

maxP_arr=[maxP_arr maxP];
maxP_Index_arr=[maxP_Index_arr maxP_Index];
Paposts_arr=[Paposts_arr; Paposts];

% Obshtata greshka pri klasifikaciqta za testvashtata izvadka
% -----(chestotata na greshno klasificiranite obekti)-----
errors=find(numberClassTest~=decision'),
%TotalError=100*(length(errors)/length(numberClassTest))

% ----- Validation -----
% ----- Verify predicted classes in test data -----

yPredictTest=Paposts_arr(is,:);
yPredictMax=maxP_arr(is);
Index_yMax=maxP_Index_arr(is);

if (Index_yMax==1)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(1,1)=ClassTabTest(1,1)+1;
        IndexTabTest11=[IndexTabTest11 is];
    elseif numberClassTest==2
        ClassTabTest(1,2)=ClassTabTest(1,2)+1;
        IndexTabTest12=[IndexTabTest12 is];
    elseif numberClassTest==3
        ClassTabTest(1,3)=ClassTabTest(1,3)+1;
        IndexTabTest13=[IndexTabTest13 is];
    else, 1
    end

elseif (Index_yMax==2)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(2,1)=ClassTabTest(2,1)+1;
        IndexTabTest21=[IndexTabTest21 is];
    elseif numberClassTest==2
        ClassTabTest(2,2)=ClassTabTest(2,2)+1;
        IndexTabTest22=[IndexTabTest22 is];
    elseif numberClassTest==3
        ClassTabTest(2,3)=ClassTabTest(2,3)+1;
        IndexTabTest23=[IndexTabTest23 is];
    else, 2
    end

elseif (Index_yMax==3)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(3,1)=ClassTabTest(3,1)+1;
        IndexTabTest31=[IndexTabTest31 is];
    elseif numberClassTest==2
        ClassTabTest(3,2)=ClassTabTest(3,2)+1;
        IndexTabTest32=[IndexTabTest32 is];
    elseif numberClassTest==3

```

```

        ClassTabTest(3,3)=ClassTabTest(3,3)+1;
        IndexTabTest33=[IndexTabTest33 is];
    else, 3
    end

    else
        LowAccuracyTest=LowAccuracyTest+1;
        IndexLowAccuracyTest=[IndexLowAccuracyTest is];
    end, % if

end

ClassTabTest
LowAccuracyTest

AllSuccess=ClassTabTest(1,1)+ClassTabTest(2,2)+ClassTabTest(3,3);
AllTest=sum(sum(ClassTabTest)); AllTest=AllTest+LowAccuracyTest;
Success=AllSuccess./AllTest.*100

figure(3)
xAcacia=[]; yAcacia=[]; xLinden=[]; yLinden=[]; xHoneydew=[];
yHoneydew=[];
for i=1:nMostri
    if maxP_Index_arr(i)==1,
        xAcacia=[xAcacia, CurrentScore_arr(i,1)];
        yAcacia=[yAcacia, CurrentScore_arr(i,2)];
    elseif maxP_Index_arr(i)==2,
        xLinden=[xLinden, CurrentScore_arr(i,1)];
        yLinden=[yLinden, CurrentScore_arr(i,2)];
    elseif maxP_Index_arr(i)==3,
        xHoneydew=[xHoneydew, CurrentScore_arr(i,1)];
        yHoneydew=[yHoneydew, CurrentScore_arr(i,2)];
    end
end
plot(xAcacia,yAcacia,'bo','MarkerFaceColor','b'), hold on,
plot(xLinden,yLinden,'g^','MarkerFaceColor','g'), hold on,
plot(xHoneydew,yHoneydew,'rs','MarkerFaceColor','r'), hold on
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('LDA-Based Classifier Using Principal Components');
legend('Acacia','Linden','Honeydew',...
        'Location','NW')

save Honey_LDA_delutedVis_New2

```

## Приложение 3

### Honey\_NN\_onlyUV\_1out\_2hl\_New2

```
clear all

% ----- Initialization -----
% From file HoneyData_UV_New2.mat:
% wLength, AbsAcacia_UV, AbsLinden_UV, AbsHoneydew_UV
load HoneyData_UV_New2

AbsAcacia_UV=AbsAcacia(521:711,:);
AbsLinden_UV=AbsLinden(521:711,:);
AbsHoneydew_UV=AbsHoneydew(521:711,:);

wLength=wLength(521:711,:);
wAcacia=wAcacia(521:711,:);
wLinden=wLinden(521:711,:);
wHoneydew=wHoneydew(521:711,:);

figure(1)
plot(wLength,AbsAcacia_UV,'b'), hold on,
plot(wLength,AbsLinden_UV,'g--'), hold on,
plot(wLength,AbsHoneydew_UV,'r:'), hold on
axis([190 380 -0.25 5.25])
xlabel('Wavelength, nm');
ylabel('Absorbtion');

nAcacia=size(AbsAcacia_UV,2); % Number of Acacia samples
nLinden=size(AbsLinden_UV,2); % Number of Linden samples
nHoneydew=size(AbsHoneydew_UV,2); % Number of Honeydew samples
nWaves=size(wLength,1); % Number of wave samples
nHoney=nAcacia+nLinden+nHoneydew; % Number of Honey samples - All

iAk=[1:nAcacia];
iLip=[1:nLinden]+nAcacia;
iMan=[1:nHoneydew]+nAcacia+nLinden;
IndexKind=[iAk iLip iMan];
B=[1.*ones(1,nAcacia) 2.*ones(1,nLinden) 3.*ones(1,nHoneydew)];
Honey=[AbsAcacia_UV AbsLinden_UV AbsHoneydew_UV];

% ----- PCA -----
AbsorbValues=(Honey)';
%mapcaplot(AbsorbValues,TitleAbs_cell1);
[pc, zscores, pcvars] = princomp(AbsorbValues);
pcvars./sum(pcvars) * 100
cumsum(pcvars./sum(pcvars) * 100)

N=2; %Number of principal components
```



```

%N=3; %Number of principal components
zscoresN=zscores(:,1:N);
nMostri=nHoney;

figure(2)
xAcacia=[]; yAcacia=[]; xLinden=[]; yLinden=[]; xHoneydew=[];
yHoneydew=[];
for i=1:nMostri
    if find(IndexKind(1,i)==iAk),
        xAcacia=[xAcacia, zscores(i,1)];
        yAcacia=[yAcacia, zscores(i,2)];
    elseif find(IndexKind(1,i)==iLip),
        xLinden=[xLinden, zscores(i,1)];
        yLinden=[yLinden, zscores(i,2)];
    elseif find(IndexKind(1,i)==iMan),
        xHoneydew=[xHoneydew, zscores(i,1)];
        yHoneydew=[yHoneydew, zscores(i,2)];
    end
end
plot(xAcacia,yAcacia,'bo','MarkerFaceColor','b'), hold on,
plot(xLinden,yLinden,'g^','MarkerFaceColor','g'), hold on,
plot(xHoneydew,yHoneydew,'rs','MarkerFaceColor','r'), hold on
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('Principal Components - Scatter Plot');
legend('Acacia','Linden','Honeydew',...
        'Location','NW')

pc_arr=[]; zscores_arr=[]; pcvars_arr=[];
CurrentScore_arr=[];
DecisionThreshold=0.01;

ClassTabTest=zeros(3,3);
IndexTabTest11=[]; IndexTabTest12=[]; IndexTabTest13=[];
IndexTabTest21=[]; IndexTabTest22=[]; IndexTabTest23=[];
IndexTabTest31=[]; IndexTabTest32=[]; IndexTabTest33=[];
LowAccuracyTest=0; IndexLowAccuracyTest=[];

numberClassPredict=zeros(1,nMostri);
maxP_arr=[]; maxP_Index_arr=[]; Paposts_arr=[];
% ----- Leave 1 out cross validation test -----
nTest=1;
for is=1:nMostri
    is

    CurrentValue=AbsorbValues(is,:);
    CurrentScore=zscoresN(is,:);
    CurrentScore_arr=[CurrentScore_arr;CurrentScore];

```

```

AbsorbValues_lout=AbsorbValues; AbsorbValues_lout(is,:)=[];
[pc, zscores, pcvars] = princomp(AbsorbValues_lout);
zscoresN_lout=zscores(:,1:N);

% ----- Initialization -----
nSamples=nMostri-1;
dataClasses=zscoresN_lout; % Learning Input Data
(zscores_L(:,1:N))
numberClasses=B'; numberClasses(is)=[]; % Learning Output Data

dataClassTest=CurrentScore; % Learning Input Data;
numberClassTest=B(1,is); % Testing Output Data

dataClass1=[]; dataClass2=[]; dataClass3=[];
indexClass1=[]; indexClass2=[]; indexClass3=[]; yd_arr=[];

for i=1:nSamples
    if numberClasses(i,1)==1,
        dataClass1=[dataClass1; dataClasses(i,:)];
indexClass1=[indexClass1;i]; yd=[1 0 0];
    elseif numberClasses(i,1)==2,
        dataClass2=[dataClass2; dataClasses(i,:)];
indexClass2=[indexClass2;i]; yd=[0 1 0];
    else,
        dataClass3=[dataClass3; dataClasses(i,:)];
indexClass3=[indexClass3;i]; yd=[0 0 1];
    end
    yd_arr=[yd_arr; yd];
end
nClass1=size(dataClass1,1); nClass2=size(dataClass2,1);
nClass3=size(dataClass3,1);
nLevel=size(dataClasses,2);

% ----- NN-BP -----
% ----- Initialization NN-BP -----
P=dataClasses'; T=yd_arr';

% Set input vector size R, layer sizes S1 & S2, batch size Q.
%[R,Q] = size(P); S1 = 100; [S2,Q] = size(T); %

[R,Q] = size(P); S1 = 250; S2 = 25; [S3,Q] = size(T); %
92.7273 (N=3, 90.9091)
%[R,Q] = size(P); S1 = 500; S2 = 250; [S3,Q] = size(T); %
92.7273 (N=3, 90.9091)
%[R,Q] = size(P); S1 = 1000; S2 = 500; [S3,Q] = size(T); %
92.7273

% ----- TRAIN THE NETWORK -----
p=P; t=T;

```

```

net=newff(minmax(p),[S1,S2,S3],{'tansig','tansig','logsig'},'train
gdx');

%net=newff(p,t,[S1,S2,S3],{'tansig','tansig','logsig'},'traingdx')
;
    %net=newff(minmax(p),[S1,S2],{'tansig','logsig'},'traingdx');
    %net.trainParam.show = 50;
    net.trainParam.show = 1000;
    net.trainParam.lr = 0.01;
    net.trainParam.lr_inc = 1.05;
    net.trainParam.lr_dec = 0.7;
    net.trainParam.mc = 0.9;
    net.trainParam.epochs = 2000;
    net.trainParam.goal = 1e-3;

    [net,tr]=train(net,p,t); hold off
    %a = sim(net,p);

    W1=net.iw{1,1}; B1=net.b{1};
    W2=net.lw{2,1}; B2=net.b{2};
    W3=net.lw{3,2}; B3=net.b{3};

    % --- Aposterioriornite verogtnosti po otnoshenie na testvashtata
    izvadka ---
    decision=zeros(1,nTest);

    x=dataClassTest(1,:); Pv1=x;
    [Rv1,Qv1] = size(Pv1);
    %Av1_1=tansig(netsum(W1*Pv1+B1*ones(1,Qv1)));
    %Av2_1=tansig(netsum(W2*Av1_1+B2*ones(1,Qv1)));
    %Av3_1=logsig(netsum(W3*Av2_1+B3*ones(1,Qv1)));

    Av1_1=tansig(netsum(W1*Pv1,B1*ones(1,Qv1)));
    Av2_1=tansig(netsum(W2*Av1_1,B2*ones(1,Qv1)));
    Av3_1=logsig(netsum(W3*Av2_1,B3*ones(1,Qv1)));
    num1=Av3_1(1);
    num2=Av3_1(2);
    num3=Av3_1(3);
    num=num1+num2+num3;
    Papost1=num1/num;
    Papost2=num2/num;
    Papost3=num3/num;
    Paposts=[Papost1 Papost2 Papost3]; %Opredeleqne na
maksimalnata i klasa
    [maxP,maxP_Index]=max(Paposts); %indeksyt e opredeleniq
klas

    decision(:,1)=maxP_Index;

```

```

maxP_arr=[maxP_arr maxP];
maxP_Index_arr=[maxP_Index_arr maxP_Index];
Paposts_arr=[Paposts_arr; Paposts];

% Obshtata greshka pri klasifikaciqta za testvashtata izvadka
% ----- (chestotata na greshno klasificiranite obekti)-----
errors=find(numberClassTest~=decision'),
%TotalError=100*(length(errors)/length(numberClassTest))

% ----- Validation -----
% ----- Verify predicted classes in test data -----

yPredictTest=Paposts_arr(is,:);
yPredictMax=maxP_arr(is);
Index_yMax=maxP_Index_arr(is);

if (Index_yMax==1)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(1,1)=ClassTabTest(1,1)+1;
        IndexTabTest11=[IndexTabTest11 is];
    elseif numberClassTest==2
        ClassTabTest(1,2)=ClassTabTest(1,2)+1;
        IndexTabTest12=[IndexTabTest12 is];
    elseif numberClassTest==3
        ClassTabTest(1,3)=ClassTabTest(1,3)+1;
        IndexTabTest13=[IndexTabTest13 is];
    else, 1
    end

elseif (Index_yMax==2)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(2,1)=ClassTabTest(2,1)+1;
        IndexTabTest21=[IndexTabTest21 is];
    elseif numberClassTest==2
        ClassTabTest(2,2)=ClassTabTest(2,2)+1;
        IndexTabTest22=[IndexTabTest22 is];
    elseif numberClassTest==3
        ClassTabTest(2,3)=ClassTabTest(2,3)+1;
        IndexTabTest23=[IndexTabTest23 is];
    else, 2
    end

elseif (Index_yMax==3)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(3,1)=ClassTabTest(3,1)+1;
        IndexTabTest31=[IndexTabTest31 is];
    elseif numberClassTest==2
        ClassTabTest(3,2)=ClassTabTest(3,2)+1;
        IndexTabTest32=[IndexTabTest32 is];
    elseif numberClassTest==3

```

```

        ClassTabTest(3,3)=ClassTabTest(3,3)+1;
        IndexTabTest33=[IndexTabTest33 is];
    else, 3
    end

    else
        LowAccuracyTest=LowAccuracyTest+1;
        IndexLowAccuracyTest=[IndexLowAccuracyTest is];
    end, % if

end

ClassTabTest
LowAccuracyTest

AllSuccess=ClassTabTest(1,1)+ClassTabTest(2,2)+ClassTabTest(3,3);
AllTest=sum(sum(ClassTabTest)); AllTest=AllTest+LowAccuracyTest;
Success=AllSuccess./AllTest.*100

figure(3)
xAcacia=[]; yAcacia=[]; xLinden=[]; yLinden=[]; xHoneydew=[];
yHoneydew=[];
for i=1:nMostri
    if maxP_Index_arr(i)==1,
        xAcacia=[xAcacia, CurrentScore_arr(i,1)];
        yAcacia=[yAcacia, CurrentScore_arr(i,2)];
    elseif maxP_Index_arr(i)==2,
        xLinden=[xLinden, CurrentScore_arr(i,1)];
        yLinden=[yLinden, CurrentScore_arr(i,2)];
    elseif maxP_Index_arr(i)==3,
        xHoneydew=[xHoneydew, CurrentScore_arr(i,1)];
        yHoneydew=[yHoneydew, CurrentScore_arr(i,2)];
    end
end
plot(xAcacia,yAcacia,'bo','MarkerFaceColor','b'), hold on,
plot(xLinden,yLinden,'g^','MarkerFaceColor','g'), hold on,
plot(xHoneydew,yHoneydew,'rs','MarkerFaceColor','r'), hold on
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('NN-Based Classifier Using Principal Components');
legend('Acacia','Linden','Honeydew',...
        'Location','NW')

save Honey_NN_onlyUV_New2

```

## Приложение 4

### Honey\_NN\_delutedVis\_1out\_2hl\_New2.m

```
clear all

% ----- Initialization -----
% From file HoneyData_UV_New2.mat:
% wLength, AbsAcacia_UV, AbsLinden_UV, AbsHoneydew_UV
load HoneyData_UV_New2

AbsAcacia_UV=AbsAcacia(1:520,:);
AbsLinden_UV=AbsLinden(1:520,:);
AbsHoneydew_UV=AbsHoneydew(1:520,:);

wLength=wLength(1:520,:);
wAcacia=wAcacia(1:520,:);
wLinden=wLinden(1:520,:);
wHoneydew=wHoneydew(1:520,:);

figure(1)
plot(wLength,AbsAcacia_UV,'b'), hold on,
plot(wLength,AbsLinden_UV,'g--'), hold on,
plot(wLength,AbsHoneydew_UV,'r:'), hold on
axis([385 900 -0.25 1.75])
xlabel('Wavelength, nm');
ylabel('Absorbtion');

nAcacia=size(AbsAcacia_UV,2); % Number of Acacia samples
nLinden=size(AbsLinden_UV,2); % Number of Linden samples
nHoneydew=size(AbsHoneydew_UV,2); % Number of Honeydew samples
nWaves=size(wLength,1); % Number of wave samples
nHoney=nAcacia+nLinden+nHoneydew; % Number of Honey samples - All

iAk=[1:nAcacia];
iLip=[1:nLinden]+nAcacia;
iMan=[1:nHoneydew]+nAcacia+nLinden;
IndexKind=[iAk iLip iMan];
B=[1.*ones(1,nAcacia) 2.*ones(1,nLinden) 3.*ones(1,nHoneydew)];
Honey=[AbsAcacia_UV AbsLinden_UV AbsHoneydew_UV];

% ----- PCA -----
AbsorbValues=(Honey)';
%mapcaplot(AbsorbValues,TitleAbs_cell1);
[pc, zscores, pcvars] = princomp(AbsorbValues);
pcvars./sum(pcvars) * 100
cumsum(pcvars./sum(pcvars) * 100)

N=2; %Number of principal components
```

```

%N=3; %Number of principal components
zscoresN=zscores(:,1:N);
nMostri=nHoney;

figure(2)
xAcacia=[]; yAcacia=[]; xLinden=[]; yLinden=[]; xHoneydew=[];
yHoneydew=[];
for i=1:nMostri
    if find(IndexKind(1,i)==iAk),
        xAcacia=[xAcacia, zscores(i,1)];
        yAcacia=[yAcacia, zscores(i,2)];
    elseif find(IndexKind(1,i)==iLip),
        xLinden=[xLinden, zscores(i,1)];
        yLinden=[yLinden, zscores(i,2)];
    elseif find(IndexKind(1,i)==iMan),
        xHoneydew=[xHoneydew, zscores(i,1)];
        yHoneydew=[yHoneydew, zscores(i,2)];
    end
end
plot(xAcacia,yAcacia,'bo','MarkerFaceColor','b'), hold on,
plot(xLinden,yLinden,'g^','MarkerFaceColor','g'), hold on,
plot(xHoneydew,yHoneydew,'rs','MarkerFaceColor','r'), hold on
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('Principal Components - Scatter Plot');
legend('Acacia','Linden','Honeydew',...
        'Location','NW')

pc_arr=[]; zscores_arr=[]; pcvars_arr=[];
CurrentScore_arr=[];
DecisionThreshold=0.01;

ClassTabTest=zeros(3,3);
IndexTabTest11=[]; IndexTabTest12=[]; IndexTabTest13=[];
IndexTabTest21=[]; IndexTabTest22=[]; IndexTabTest23=[];
IndexTabTest31=[]; IndexTabTest32=[]; IndexTabTest33=[];
LowAccuracyTest=0; IndexLowAccuracyTest=[];

numberClassPredict=zeros(1,nMostri);
maxP_arr=[]; maxP_Index_arr=[]; Paposts_arr=[];

% ----- Leave 1 out cross validation test -----
nTest=1;
for is=1:nMostri
    is

        CurrentValue=AbsorbValues(is,:);
        CurrentScore=zscoresN(is,:);
        CurrentScore_arr=[CurrentScore_arr;CurrentScore];

```

```

AbsorbValues_lout=AbsorbValues; AbsorbValues_lout(is,:)=[];
[pc, zscores, pcvars] = princomp(AbsorbValues_lout);
zscoresN_lout=zscores(:,1:N);

% ----- Initialization -----
nSamples=nMostri-1;
dataClasses=zscoresN_lout; % Learning Input Data
(zscores_L(:,1:N))
numberClasses=B'; numberClasses(is)=[]; % Learning Output Data

dataClassTest=CurrentScore; % Learning Input Data;
numberClassTest=B(1,is); % Testing Output Data

dataClass1=[]; dataClass2=[]; dataClass3=[];
indexClass1=[]; indexClass2=[]; indexClass3=[]; yd_arr=[];

for i=1:nSamples
    if numberClasses(i,1)==1,
        dataClass1=[dataClass1; dataClasses(i,:)];
indexClass1=[indexClass1;i]; yd=[1 0 0];
    elseif numberClasses(i,1)==2,
        dataClass2=[dataClass2; dataClasses(i,:)];
indexClass2=[indexClass2;i]; yd=[0 1 0];
    else,
        dataClass3=[dataClass3; dataClasses(i,:)];
indexClass3=[indexClass3;i]; yd=[0 0 1];
    end
    yd_arr=[yd_arr; yd];
end
nClass1=size(dataClass1,1); nClass2=size(dataClass2,1);
nClass3=size(dataClass3,1);
nLevel=size(dataClasses,2);

% ----- NN-BP -----
% ----- Initialization NN-BP -----
P=dataClasses'; T=yd_arr';

% Set input vector size R, layer sizes S1 & S2, batch size Q.
%[R,Q] = size(P); S1 = 100; [S2,Q] = size(T); %

[R,Q] = size(P); S1 = 250; S2 = 25; [S3,Q] = size(T); %
92.7273 (N=3, 90.9091)
%[R,Q] = size(P); S1 = 500; S2 = 250; [S3,Q] = size(T); %
92.7273 (N=3, 90.9091)
%[R,Q] = size(P); S1 = 1000; S2 = 500; [S3,Q] = size(T); %
92.7273

% ----- TRAIN THE NETWORK -----
p=P; t=T;

```



```

net=newff(minmax(p),[S1,S2,S3],{'tansig','tansig','logsig'},'train
gdx');

%net=newff(p,t,[S1,S2,S3],{'tansig','tansig','logsig'},'traingdx')
;
    %net=newff(minmax(p),[S1,S2],{'tansig','logsig'},'traingdx');
    %net.trainParam.show = 50;
    net.trainParam.show = 1000;
    net.trainParam.lr = 0.01;
    net.trainParam.lr_inc = 1.05;
    net.trainParam.lr_dec = 0.7;
    net.trainParam.mc = 0.9;
    net.trainParam.epochs = 2000;
    net.trainParam.goal = 1e-3;

    [net,tr]=train(net,p,t); hold off
    %a = sim(net,p);

    W1=net.iw{1,1}; B1=net.b{1};
    W2=net.lw{2,1}; B2=net.b{2};
    W3=net.lw{3,2}; B3=net.b{3};

    % --- Aposterioriornite veroqtnosti po otnoshenie na testvashtata
    izvadka ---
    decision=zeros(1,nTest);

    x=dataClassTest(1,:)' ; Pv1=x;
    [Rv1,Qv1] = size(Pv1);
    %Av1_1=tansig(netsum(W1*Pv1+B1*ones(1,Qv1)));
    %Av2_1=tansig(netsum(W2*Av1_1+B2*ones(1,Qv1)));
    %Av3_1=logsig(netsum(W3*Av2_1+B3*ones(1,Qv1)));

    Av1_1=tansig(netsum(W1*Pv1,B1*ones(1,Qv1)));
    Av2_1=tansig(netsum(W2*Av1_1,B2*ones(1,Qv1)));
    Av3_1=logsig(netsum(W3*Av2_1,B3*ones(1,Qv1)));
    num1=Av3_1(1);
    num2=Av3_1(2);
    num3=Av3_1(3);
    num=num1+num2+num3;
    Papost1=num1/num;
    Papost2=num2/num;
    Papost3=num3/num;
    Paposts=[Papost1 Papost2 Papost3]; %Opredeleqne na
maksimalnata i klasa
    [maxP,maxP_Index]=max(Paposts); %indeksyt e opredeleniq
klas

    decision(:,1)=maxP_Index;

```

```

maxP_arr=[maxP_arr maxP];
maxP_Index_arr=[maxP_Index_arr maxP_Index];
Paposts_arr=[Paposts_arr; Paposts];

% --- Obshtata greshka pri klasifikaciqta za testvashtata
izvadka ---
% -----(chestotata na greshno klasificiranite obekti)-----
errors=find(numberClassTest~=decision'),
%TotalError=100*(length(errors)/length(numberClassTest))

% ----- Validation -----
% ----- Verify predicted classes in test data -----

yPredictTest=Paposts_arr(is,:);
yPredictMax=maxP_arr(is);
Index_yMax=maxP_Index_arr(is);

if (Index_yMax==1)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(1,1)=ClassTabTest(1,1)+1;
        IndexTabTest11=[IndexTabTest11 is];
    elseif numberClassTest==2
        ClassTabTest(1,2)=ClassTabTest(1,2)+1;
        IndexTabTest12=[IndexTabTest12 is];
    elseif numberClassTest==3
        ClassTabTest(1,3)=ClassTabTest(1,3)+1;
        IndexTabTest13=[IndexTabTest13 is];
    else, 1
    end

elseif (Index_yMax==2)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(2,1)=ClassTabTest(2,1)+1;
        IndexTabTest21=[IndexTabTest21 is];
    elseif numberClassTest==2
        ClassTabTest(2,2)=ClassTabTest(2,2)+1;
        IndexTabTest22=[IndexTabTest22 is];
    elseif numberClassTest==3
        ClassTabTest(2,3)=ClassTabTest(2,3)+1;
        IndexTabTest23=[IndexTabTest23 is];
    else, 2
    end

elseif (Index_yMax==3)&(sum(yPredictTest)~=0)
    if numberClassTest==1
        ClassTabTest(3,1)=ClassTabTest(3,1)+1;
        IndexTabTest31=[IndexTabTest31 is];
    elseif numberClassTest==2
        ClassTabTest(3,2)=ClassTabTest(3,2)+1;
        IndexTabTest32=[IndexTabTest32 is];

```

```

elseif numberClassTest==3
    ClassTabTest(3,3)=ClassTabTest(3,3)+1;
    IndexTabTest33=[IndexTabTest33 is];
else, 3
end

else
    LowAccuracyTest=LowAccuracyTest+1;
    IndexLowAccuracyTest=[IndexLowAccuracyTest is];
end, % if

end

ClassTabTest
LowAccuracyTest

AllSuccess=ClassTabTest(1,1)+ClassTabTest(2,2)+ClassTabTest(3,3);
AllTest=sum(sum(ClassTabTest)); AllTest=AllTest+LowAccuracyTest;
Success=AllSuccess./AllTest.*100

figure(3)
xAcacia=[]; yAcacia=[]; xLinden=[]; yLinden=[]; xHoneydew=[];
yHoneydew=[];
for i=1:nMostri
    if maxP_Index_arr(i)==1,
        xAcacia=[xAcacia, CurrentScore_arr(i,1)];
        yAcacia=[yAcacia, CurrentScore_arr(i,2)];
    elseif maxP_Index_arr(i)==2,
        xLinden=[xLinden, CurrentScore_arr(i,1)];
        yLinden=[yLinden, CurrentScore_arr(i,2)];
    elseif maxP_Index_arr(i)==3,
        xHoneydew=[xHoneydew, CurrentScore_arr(i,1)];
        yHoneydew=[yHoneydew, CurrentScore_arr(i,2)];
    end
end
end
plot(xAcacia,yAcacia,'bo','MarkerFaceColor','b'), hold on,
plot(xLinden,yLinden,'g^','MarkerFaceColor','g'), hold on,
plot(xHoneydew,yHoneydew,'rs','MarkerFaceColor','r'), hold on
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('NN-Based Classifier Using Principal Components');
legend('Acacia','Linden','Honeydew',...
    'Location','NW')

save Honey_NN_delutedVis_New2

```