

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
ЧАЧАК

ПРИМЉЕНО 29. 06. 2023.			
Орг. ед.	Број	Пристај	Вредност
	1179		



УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У ЧАЧКУ

Небојша М. Андријевић

СИСТЕМ ЗА МОНИТОРИНГ И ПРЕДИКЦИЈУ
АКТИВНОСТИ МЕДОНОСНИХ ПЧЕЛА БАЗИРАН НА ИОТ
ТЕХНОЛОГИЈИ И ПРИМЕНИ ВЕШТАЧКЕ
ИНТЕЛИГЕНЦИЈЕ

ДОКТОРСКА ДИСЕРТАЦИЈА

Ментор: др Владе Урошевић, редовни професор

ЧАЧАК, 2023.



UNIVERSITY OF KRAGUJEVAC
FACULTY OF TECHNICAL SCIENCES IN ČAČAK

Nebojša M. Andrijević

**MONITORING AND PREDICTION SYSTEM OF HONEYBEE
ACTIVITY BASED ON IoT TECHNOLOGY AND
APPLICATION OF ARTIFICIAL INTELLIGENCE**

DOCTORAL DISSERTATION

Mentor: dr. Vlade Urošević, Full Professor

ČAČAK, 2023.

ИДЕНТИФИКАЦИОНА СТРАНИЦА ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

I Аутор
Име и презиме: Небојша Андријевић
Датум и место рођења: 11.10.1980. године, Призрен
Садашње запослење: Академија техничких струковних студија Београд.
II Докторска дисертација
Наслов: Систем за мониторинг и предикцију активности медоносних пчела базиран на ИоТ технологији и примени вештачке интелигенције
Број страница: 162
Број слика: 65
Број табела: 25
Број графика: 11
Број библиографских података: 105
Број прилога: 16
Установа и место где је рад израђен: Факултет техничких наука у Чачку, Универзитет у Крагујевцу
Научна област (УДК): Техничко технолошке науке, Електротехничко и рачунарско инжењерство, Примењено рачунарство
Ментор: др Владе Урошевић, редовни професор
III Оцена и одбрана
Датум пријаве теме: 10.11.2022. год.
Број одлуке и датум прихватања докторске дисертације: IV-04-309/13, 17.05.2023. год.
Комисија за оцену подобности теме и кандидата: <ol style="list-style-type: none">проф. др Небојша Митровић, редовни професор, председник Комисије Факултет техничких наука у Чачку, Универзитет у Крагујевцупроф. др Владе Урошевић, редовни професор, ментор Факултет техничких наука у Чачку, Универзитет у Крагујевцудр Дејан Стевановић, научни сарадник, члан Електронски факултет, Универзитет у Нишудоц. др Бранко Арсић, доцент, члан Природно-математички факултет, Универзитет у Крагујевцупроф. др Бранко Маркоски, редовни професор, члан Факултет техничких наука, Универзитет у Новом Саду
Комисија за оцену и одбрану докторске дисертације: <ol style="list-style-type: none">проф. др Небојша Митровић, редовни професор, председник Комисије Факултет техничких наука у Чачку, Универзитет у Крагујевцудоц. др Бранко Арсић, доцент, члан Природно-математички факултет, Универзитет у Крагујевцупроф. др Бранко Маркоски, редовни професор, члан Факултет техничких наука, Универзитет у Новом Садупроф. др Данијела Милошевић, редовни професор, члан Факултет техничких наука у Чачку, Универзитет у Крагујевцудр Дејан Стевановић, научни сарадник, члан Електронски факултет, Универзитет у Нишу
Датум одбране дисертације:

ЗАХВАЛНИЦА

Овај успех посвећујем мојој супризи, мом анђелу Биљани. Захвалан сам на свему што смо досад остварили и створили заједно, породицу и дом. Хвала јој на идеји за тему, подршци и помоћи, без ње би све ово било немогуће оствариво.

У животу емоције и љубав су ми биле највећа мотивација, а мом срцу породица је увек била највећа светиња...

Хвала мојим наследницима, мом поносу и мојој будућности Николети, Давиду, Кости и Каји, на љубави коју су ми пружили. Ово је успех свих нас заједно, синергија.

Хвала мом оцу Младену, мајци Ради, браћи Бојану и Слоби, хвала им на свему а много смо тога прошли заједно, време је да се радујемо нашим успесима и делима.

Захвалан сам највише Богу за невероватну снагу, због које сам баш овде где треба да будем. Захвалан сам свима који су ми несебично помогли и веровали у мене, ментору проф. Влади Урошевићу, проф. Небојши Митровићу, доц. Бранку Арсићу, проф. Ђорђу Херцегу, проф. Јерославу Живанићу и члановима свих Комисија током мог пута као докторанда.

Андријевић М. Небојша

РЕЗИМЕ

Свест о проблемима здравља пчела расте, док савест према нези пчела касни. Систем који је представљен у дисертацији настао је у контексту континуирано растуће свести о круцијалној улози пчела у одржавању глобалног биодиверзитета. Последњих десетак година развијена су бројна решења у виду праћења параметара који утичу на кретања и живот пчела. Хардверска решења помажу приликом прикупљања врло важних података, али подаци представљају само статистику док примена софтверских решења отвара нову димензију у манипулацији подацима. Да би дошли до изведених података и информисаних одлука, потребна је примена модела вештачке интелигенције (ВИ).

Систем који је настао као резултат дисертације представља приступ интеграције напредних технологија у управљању кошницама кроз имплементацију Internet of Things (ИоТ) технологије и вештачке интелигенције. Централни део система је ИоТ базна станица која је интегрисана са кошницом, опремљена је низом сензора који континуирано прате виталне параметре кошнице као и микроклиматске промене у екосистему кошнице.

Да би се боље разумели и предвидели обрасци понашања пчела, развијен је модел вештачке интелигенције који се базира на рекурентним неуронским мрежама (RNN). За ефикасну предикцију фреквенције изласка и уласка пчела у кошницу у односу на параметре околине, модел користи временске серије за оквир података. Систем није само у стању да идентификује потенцијална одступања, већ и да активира аларме када детектује вредности које прелазе унапред одређене сигурносне границе. Поређење развијеног модела са експерименталним подацима потврдило је високу тачност предвиђања, указујући на његову робустност и поузданост у представљању стварних процеса који се одвијају у кошници.

Сви прикупљени подаци, аналитички резултати и генерисана упозорења су приказани у реалном времену преко посебно дизајниране web апликације, пружајући пчеларима драгоцене увиде у стање кошнице и омогућавајући им да брзо реагују на потенцијалне проблеме.

Дисертација доноси значајну иновацију у области прецизног пчеларства, нудећи алате и методе који не само да пружају дубље разумевање динамике пчеларских кошница, већ и доприносе глобалним напорима за заштиту пчела и очување нашег екосистема.

Кључне речи: Интернет ствари, мониторинг, предикција, пчеле, вештачка интелигенција

ABSTRACT

The awareness of bee health issues is growing, while the conscience of their well-being lags behind. The system presented in the dissertation was developed in the context of the continuously growing awareness of the crucial role bees play in maintaining global biodiversity. Over the past decade, numerous solutions have been developed for monitoring parameters that affect bee behavior and their lives. Hardware solutions assist us in collecting vital data, but the data only represent statistical values, whereas the implementation of software solutions opens up a new dimension in data manipulation. To derive meaningful insights and informed decisions, the application of artificial intelligence (AI) models is necessary.

The system that emerged as a result of the dissertation represents an approach that integrates advanced technologies in hive management through the implementation of Internet of Things (IoT) technology and artificial intelligence. The central component of the system is an IoT base station integrated into the beehive, equipped with a range of sensors that continuously monitor vital hive parameters and microclimate changes in the hive ecosystem.

To better understand and predict bee behavior patterns, an artificial intelligence model based on recurrent neural networks (RNN) has been developed. To efficiently predict the frequency of bee entry and exit from the hive in relation to environmental parameters, the model utilizes time series as a framework for data. The system is not only capable of identifying potential deviations but it also activates alarms when it detects values that exceed pre-defined safety thresholds. Comparison of the developed model with experimental data confirmed its high accuracy in prediction, indicating its robustness and reliability in representing actual processes occurring within the hive.

All collected data, analytical results, and generated alerts are displayed in real-time through a specially designed web application, providing beekeepers with valuable insights into hive conditions and enabling them to quickly respond to potential issues.

The dissertation brings significant innovation to the field of precision beekeeping, offering tools and methods that not only provide deeper understanding of hive dynamics but also contribute to global efforts in protecting bees and preserving our ecosystems.

Key words: Internet of things, monitoring, prediction, honey bee, artificial intelligence

САДРЖАЈ

СПИСАК СКРАЋЕНИЦА	I
СПИСАК СЛИКА	III
СПИСАК ТАБЕЛА	V
СПИСАК ГРАФИКА	VI
ПРИЛОЗИ	VI
1. УВОД	1
2. МОТИВАЦИЈА И ЦИЉ РАДА	4
3. ПРЕГЛЕД И АНАЛИЗА ЛИТЕРАТУРЕ	6
4. САВРЕМЕНО ПЧЕЛАРСТВО - ДИГИТАЛНО ПЧЕЛАРСТВО	8
4.1. Проблеми и изазови савременог пчеларства	8
4.2. Дигитално пчеларство - примена информационих технологија.....	9
5. ТЕХНОЛОГИЈЕ ИНТЕРНЕТА СТВАРИ (Internet of things IoT)	12
5.1. Архитектура IoT.....	13
5.2. IoT Хардвер	15
5.3. IoT Софтвер.....	19
5.4. Оперативни системи	20
5.5. Комплексни слојеви IoT архитектуре.....	21
5.5.1. Перцептивни слој	22
5.5.2. Мрежни слој и протоколи.....	22
5.5.3. Слој средњег софтвера.....	26
5.5.4. Рачунарство у облаку - Cloud computing	28
5.5.5. Велики подаци - Big data	29
5.5.6. Слој вештачке интелигенције (ВИ) (Artificial inteligence AI)	30
5.5.7. Слој Апликације IoT.....	31
5.5.8. Безбедност података у IoT системима.....	32
5.5.9. Слој за контролу и одлучивање.....	33
6. СИСТЕМ ЗА МОНИТОРИНГ И ПРЕДИКЦИЈУ	34
7. ДИЗАЈН И АРХИТЕКТУРА ЕЛЕКТРОНИКЕ СИСТЕМА.....	36
7.1. Архитектура система за мониторинг	36
7.2. Архитектура главне јединице	38
7.3. Архитектура електронских компоненти система.....	41

7.3.1.	Микроконтролер Arduino Mega 2560.....	42
7.3.2.	Mega Sensor Shield.....	43
7.3.3.	Микроконтролер ESP32.....	44
7.3.4.	RTC DS3231 Real time clock модул.....	45
7.3.5.	SIM800L GPRS GSM модул	46
7.3.6.	DHT-22 (AM2302) сензор температуре и влажности ваздуха	46
7.3.7.	SW-420 сензор вибрација	47
7.3.8.	MHRD rain сензор кише.....	48
7.3.9.	MQ135 сензор детекције гасова и квалитета ваздуха	49
7.3.10.	BH1750 сензор дневне светлости.....	50
7.3.11.	VEML6075/6070 UV сензор.....	51
7.3.12.	SI1145 сензор светлости: дневне, инфрацрвене IC, ултраљубичасте UV	52
7.3.13.	BME280 сензор ваздушног притиска.....	53
7.3.14.	Анемометар Wind Speed Sensor	53
7.3.15.	MAX4466 и MAX9814 сензори фреквенције звука и нивоа буке.....	54
7.3.16.	Улази и излази пчела.....	56
8.	СОФТВЕР И CLOUD СИСТЕМ ПОДРШКЕ	58
8.2.	Примењене технологије.....	58
8.2.1.	PHP програмски језик	58
8.2.2.	Laravel радни оквир.....	58
8.2.3.	MVC архитектура.....	59
8.2.4.	JavaScript програмски језик	60
8.2.5.	Round-Trip и Single-Page апликације	60
8.2.6.	Vue.js радни оквир	61
8.3.	ИЗРАДА И РАЗВОЈ АПЛИКАЦИЈЕ	61
8.4.	Манипулација сензорима	64
8.5.	Манипулација кошницама.....	68
8.6.	Манипулација корисничким налозима	73
8.7.	Кориснички интерфејс апликације	74
9.	ПРИКАЗ РЕЗУЛТАТА МЕРЕЊА СЕНЗОРА И ГРАФИКА КОРЕЛАЦИЈА	81
10.	ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА.....	91
10.1.	Методe, модели и алати	93
10.2.	Примене вештачке интелигенције у ИoТ	96
10.3.	Неуронске мреже	97

10.3.1.	Рекурентне неуронске мреже (Recurrent Neural Network RNN)	102
10.3.2.	Long Short-Term Memory LSTM.....	105
10.4.	Анализа и предикција временских серија	108
10.4.1.	ARIMA	110
10.4.2.	Facebook prophet	112
10.5.	Моделовање.....	114
10.5.1.	Анализа и припрема података	114
10.5.2.	Креирање модела за предикцију у Python-у	117
10.6.	Анализа и предикција временских серија података.....	123
11.	РЕЗУЛТАТИ МОДЕЛОВАЊА.....	128
11.1.	Експериментална поставка и евалуација.....	128
11.2.	Резултати ИоТ Система	132
11.3.	Предиктивни АРІ.....	134
11.4.	Предности примена неуронских мрежа у поређењу са другим статистичким моделима	135
12.	ЗАКЉУЧАК	138
	ЛИТЕРАТУРА.....	143
	ПРИЛОЗИ	152
	БИОГРАФИЈА АУТОРА	158
	<i>ИЗЈАВА АУТОРА О ОРИГИНАЛНОСТИ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ.....</i>	<i>159</i>
	<i>ИЗЈАВА АУТОРА О ИСТОВЕТНОСТИ ШТАМПАНЕ И ЕЛЕКТРОНСКЕ ВЕРЗИЈЕ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ</i>	<i>160</i>
	<i>ИЗЈАВА АУТОРА О ИСКОРИШЋАВАЊУ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ</i>	<i>161</i>

СПИСАК СКРАЋЕНИЦА

ВИ	Вештачка интелигенција
AI	Artificial Intelligence
IoT	Internet of Things (Интернет ствари)
RNN	Recurrent Neural Networks (Рекурентне неуронске мреже)
IoT MAP	Internet of things monitoring and prediction
API	Application Programming Interface (Програмски интерфејс апликације)
MIT	Massachusetts Institute of Technology (Масачусетски институт за технологију)
M2M	Machine to Machine
Wi-Fi	Wireless Fidelity
RF	Radio frequency
RFID	Radio frequency identification (Идентификација помоћу радио-фреквенције)
CISC	Complex Instruction Set Computing (Процесор са комплексним скупом инструкција)
RISC	Reduced Instruction Set Computing (Процесор са мањим скупом инструкција)
MQTT	Message Queuing Telemetry Transport (Протокол за комуникацију)
CoAP	Constrained Application Protocol (Протокол за ограничене апликације)
HTTP	Hypertext Transfer Protocol
WSN	Wireless sensor network (Бежична сензорска мрежа)
LoRaWAN	Long-range Radio Wide Area Network (Технологија бежичне комуникације)
BLE	Bluetooth Low Energy (Технологија бежичне комуникације)
NFC	Near Field Communications (Технологија бежичне комуникације)
LPWAN	Low Power Wide Area Network (Технологија бежичне комуникације)
LTE	Long-Term Evolution (Технологија бежичне комуникације)
NB-IoT	Narrowband Internet of Things (Технологија бежичне комуникације)
IETF	Internet Engineering Task Force (Радна група за интернет инжењеринг)
IEEE	Institute of Electrical and Electronics Engineers (Институт инжењера електротехнике и електронике)
AMQP	Advanced Message Queuing Protocol (Протокол за комуникацију)

SaaS	Software as a Service (Софтвер као услуга)
PaaS	Platform as a Service (Платформа као услуга)
IaaS	Infrastructure as a Service (Инфраструктура као услуга)
IDS/IPS	Intrusion Detection System/Intrusion Prevention Systems (Систем за откривање упада у мрежу, систем за превенцију упада у мрежу)
VPN	Virtual Private Network (Виртуална приватна мрежа)
DDoS	Distributed Denial of service (Дистрибуирано одбијање услуге)
DMS	Decision Making System (Систем за доношење одлука)
IDE	Integrated development environment (Интегрисано развојно окружење)
PWM	Pulse Width Modulation (Модулација ширине импулса)
UART	Universal Asynchronous Receiver/Transmitter (Универзални асинхрони пријемник/предајник)
ICSP	In-Circuit Serial Programming
I2C	Inter-Integrated Circuit (Међусобна интегрисана веза)
SCL	Serial Clock (Серијски часовник)
SDA	Serial Data (Серијски подаци)
ADC	Analog-to-Digital Converter (Аналогно-дигитални конвертор)
DAC	Digital-to-Analog Converter (Дигитално-аналогни конвертор)
TCXO	Temperature Compensated Crystal Oscillator (Интегрисани осцилатор температурне компензације)
GNSS	Global Navigation Satellite System (Глобални систем навигације сателита)
PIC	Programmable Integrated Circuit (Програмабилно интегрисано коло)
ARM	Advanced RISC Machines (Микроконтролерски систем)
AGC	Auto Gain Control (Аутоматска контрола појачања)
SPA	Single Page Application (Једностранична апликација)
MVC	Model-View-Controller (Архитектонски образац за израду апликација)
SPA сервер	Single Page Application Server (Сервер за једностраничну апликацију)
JSON	JavaScript Object Notation (Формат)
DOM	Document Object Model
WAMP/XAMP	Windows, Apache, MySQL, PHP/Perl/Python (Софтверски пакет)
NLP	Natural Language Processing (Обрада природног језика)

GPT	Generative Pre-trained Transformer (пр. chatGPT)
BERT	Bidirectional Encoder Representations from Transformers (NLP- модел)
RoBERT	Robustly Optimized BERT (NLP- модел)
ReLU	Rectified Linear Unit
CNN	Convolutional Neural Networks (Конволуционалне неуронске мреже)
GANs	Generative Adversarial Networks (Генеративно-супарничке мреже)
SGD	Stochastic Gradient Descent (Стохастички градијентни спуст)
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Units
ARIMA	Autoregressive Integrated Moving Average (Ауторегресивни интегрисани покретни просек)
ACF	Autocorrelation Function (Аутокорелацијска функција)
PACF	Partial Autocorrelation Function (Парцијална аутокорелацијска функција)

СПИСАК СЛИКА

Слика 5.1 Примена ИОТ технологија у свим областима	12
Слика 5.2 Архитектура ИОТ-а.....	14
Слика 5.3 Проширена подела ИОТ архитектуре	15
Слика 6.1 Преглед ИОТ Система	34
Слика 7.1 Увид у шематски приказ Система	36
Слика 7.2 Експериментална поставка кошнице са сензорском електроником.....	38
Слика 7.3 Главна јединица задужена за контролу и управљање ИОТ системом	39
Слика 7.4 Шема повезивања компоненти система (fritzing [66]).	40
Слика 7.5 Arduino Mega 2560.....	42
Слика 7.6 Плоча за проширење за Arduino Mega микроконтролер (Mega Sensor Shield V2.0 sensor expansion board).....	43
Слика 7.7 ESP32 микроконтролер.....	44
Слика 7.8 RTC 3231 Real time clock модул.....	45
Слика 7.9 SIM800L GPRS GSM модул	46
Слика 7.10 Сензор DHT22 за мерење температуре и влажности ваздуха.....	47
Слика 7.11. Сензор вибрација SW420.....	48
Слика 7.12. Сензор присуства кише и воде MHRD.....	49
Слика 7.13 Сензор MQ135 за детектовање присутности гасова и мерења ppm вредности.	50
Слика 7.14 BH1750 сензор интензитета дневне светлости.....	51

Слика 7.15 VEML6075 UV сензор.....	51
Слика 7.16 SII145 сензор дневне, инфрацрвене и ултраљубичасте светлости	52
Слика 7.17 BME280 сензор	53
Слика 7.18 Анемометар.....	54
Слика 7.19 MAX4466 звучни сензор.....	55
Слика 7.20 MAX9814 звучни сензор.....	56
Слика 7.21 Уређај за детекцију смера кретања пчела	57
Слика 7.22 Пролази за улазак и излазак из кошнице	57
Слика 8.1 Садржај routes/web.php датотеке.....	62
Слика 8.2 Садржај routes/api.php датотеке	63
Слика 8.3 Шема миграције за табелу sensors	65
Слика 8.4 Класа Sensor модела.....	65
Слика 8.5 Чување података о сензору.....	66
Слика 8.6 Регистровање новог сензора back	66
Слика 8.7 Регистровање новог сензора front.....	67
Слика 8.8 Измена података о сензору	68
Слика 8.9 Шема миграције за табелу hives.....	68
Слика 8.10 Преузимање података свих кошница	69
Слика 8.11 Чување података о кошници front.....	69
Слика 8.12 Чување података о новој кошници	70
Слика 8.13 Регистровање нове кошнице	70
Слика 8.14 Подешавање појединачних параметара кошнице	71
Слика 8.15 Метода getAlerts за преузимање обавештења.....	72
Слика 8.16 Метода saveAlert за чување обавештења	72
Слика 8.17 Регистрација корисника методом registerUser.....	73
Слика 8.18 Пријава и одјава корисника помоћу login методе	74
Слика 8.19 Интерфејс ВееМар апликације.....	75
Слика 8.20 Навигациони панел ВееМар апликације.....	76
Слика 8.21 Део параметара који се прати из екосистема.....	77
Слика 8.22 Приказ статуса кошнице са предиктивним и аналитичким алармом.....	78
Слика 8.23 График кретања за изабрани параметар.....	78
Слика 8.24 Приказ логова мерних сензора.....	79
Слика 10.1 Однос вештачке интелигенције, машинског учења и дубоког учења	92
Слика 10.2 Методе примене вештачке интелигенције	94
Слика 10.3 Једнослојна неуронска мрежа	98
Слика 10.4 Вишеслојне неуронске мреже	99
Слика 10.5 Принцип рада RNN мреже.....	100
Слика 10.6 Мрежне структуре RNN, LSTM, и GRU.	104
Слика 10.7 LSTM ћелија.....	105
Слика 10.8 Облик седам временских серија које одговарају излазу и улазима.....	116

Слика 10.9 Приказ правих и трансформисаних података о броју излазака пчела по сату за 20 дана.....	117
Слика 10.10 Резултати фазе тренинга Facebook Prophet алгоритма за променљиву BEE_OUT.....	126
Слика 10.11 Резултати фазе тренинга Facebook Prophet алгоритма за BEE_IN променљиву.....	126
Слика 11.1 Предвиђање за целу временску серију (20 дана) BEE_OUT и BEE_IN.....	129
Слика 11.2 Вее OUT фаза тренинга.....	130
Слика 11.3 Вее IN фаза тренинга.....	130
Слика 11.4 Плава линија - Оригинални сегмент временске серије BEE_OUT; Црвена линија- предвиђање за свако преклапање (петострука подела временске серије).....	131

СПИСАК ТАБЕЛА

Табела 5.1 Перформансе и техничке спецификације LoRaWAN технологије.....	23
Табела 5.2 Перформансе и техничке спецификације WiFi технологије.....	23
Табела 5.3 Перформансе и техничке спецификације мобилне технологије.....	23
Табела 5.4 Перформансе и техничке спецификације Thread технологије.....	24
Табела 5.5 Перформансе и техничке спецификације Bluetooth и BLE технологије.....	24
Табела 5.6 Перформансе и техничке спецификације NFC технологије.....	25
Табела 5.7 Перформансе и техничке спецификације ZigBee технологије.....	25
Табела 5.8 Перформансе и техничке спецификације NB-IoT технологије.....	26
Табела 9.1 Корелација температурних вредности у односу на кретање пчела IN/OUT.....	82
Табела 9.2 Корелација вредности влажности ваздуха у односу на кретање пчела IN/OUT.....	83
Табела 9.3 Корелација вредности присуства кише и ваздушног притиска у односу на кретање пчела IN/OUT.....	85
Табела 9.4 Корелација вредности квалитета ваздуха у односу на кретање пчела IN/OUT.....	86
Табела 9.5 Корелација вредности присуства вибрација у односу на кретање пчела IN/OUT.....	87
Табела 9.6 Корелација вредности дневне светлости у односу на кретање пчела IN/OUT.....	87
Табела 9.7 Корелација вредности UV индекса у односу на кретање пчела IN/OUT.....	88
Табела 9.8 Корелација вредности IC зрачења и интензитета дневне светлости у односу на кретање пчела IN/OUT.....	89
Табела 10.1 Сензори, контролни сензори и опис мерења.....	114
Табела 10.2 Резултати Dickey-Fuller теста.....	124
Табела 11.1 Резултати коришћењем рекурентне неуронске мреже.....	132
Табела 11.2 Поређење сумираних оптималних параметара за све испитиване методе.....	132
Табела 11.3 Приказ резултата модела за BEE_messageIN.....	135
Табела 11.4 Значајност статистичких модела за BEE_messageIN.....	136
Табела 11.5 Приказ резултата модела за BEE_messageOUT.....	136

Табела 11.6 Значајност статистичких модела BEECNT_messageOUT.....	137
Табела 11.7 Резултати свих модела.....	137

СПИСАК ГРАФИКА

График 7.1 Улазак у кошницу и излазак пчела из кошнице за 24 сата.....	56
График 9.1 Упоредни тренд кретања пчела у односу на промене температуре (DHT22) за период од 20 дана.....	81
График 9.2 Упоредни тренд кретања пчела у односу на промене влажности ваздуха (DHT22) за период од 20 дана.....	82
График 9.3 Упоредни тренд кретања пчела у односу на промене присуства кише и ваздушног притиска (MHRD) за период од 20 дана.....	83
График 9.3.1 Упоредни тренд кретања пчела у односу на промене присуства кише и ваздушног притиска на дневном нивоу.....	84
График 9.4 Упоредни тренд кретања пчела у односу на промене квалитета ваздуха (MQ135PPM) за период од 20 дана.....	85
График 9.5 Упоредни тренд кретања пчела у односу на присуство вибрација (SW420) за период од 20 дана.....	86
График 9.6 Упоредни тренд кретања пчела у односу на промене вредности дневне светлости (BH1750) за период од 20 дана.....	87
График 9.6.1 Упоредни тренд кретања пчела у односу на промене вредности дневне светлости на дневном нивоу.....	88
График 9.7 Упоредни тренд кретања пчела у односу на промене UV индекса (VEMML6075) пчела за период од 20 дана.....	89
График 9.8. Упоредни тренд кретања пчела у односу на промене IR зрачења и интензитета дневне светлости (SI1145) за период од 20 дана.....	90

ПРИЛОЗИ

Прилог 7.1 Техничка спецификација микроконтролера Arduino Mega 2560.....	152
Прилог 7.2 Техничка спецификација за Mega Sensor Shield V2.0.....	152
Прилог 7.3 Техничка спецификација ESP32 микроконтролера.....	152
Прилог 7.4 Техничка спецификација за RTC 3231 модул реалног времена (Real-Time Clock).....	153
Прилог 7.5 Техничка спецификација за SIM800L GPRS GSM модул.....	153
Прилог 7.6 Техничка спецификација за DHT-22 (AM2302) модул.....	154
Прилог 7.7 Техничка спецификација за сензор вибрације SW-420.....	154
Прилог 7.8 Техничка спецификација за сензор MHRD rain.....	154

Прилог 7.9 Техничка спецификација за сензор MQ135.....	155
Прилог 7.10 Техничка спецификација за сензор BH1750.....	155
Прилог 7.11 Техничка спецификација за сензор VEML6075.....	155
Прилог 7.12 Техничка спецификација за сензор SI1145.....	156
Прилог 7.13 Техничка спецификација за сензор VME280.....	156
Прилог 7.14 Техничка спецификација за сензор Анемометер.....	156
Прилог 7.15 Техничка спецификација за сензор MAX4466.....	157
Прилог 7.16 Техничка спецификација за сензор MAX9814.....	157

1. УВОД

Интензиван развој савременог друштва, нарочито у технолошком сегменту са приметним експоненцијалним растом из године у годину, неизбежно је утицао на животну средину. Без обзира на интенције човека да модерним технологијама влада и помаже себи у напорима у освајању природе и процеса производње добара за егзистенцијалне потребе, немарно и без намере човек неминовно мења природу и екосистем који га окружује, утичући тако на природне процесе. Природа има своје токове, и промене које човек направи у смеру контроле природних процеса, ремете савршени баланс природне равнотеже. Детерминисан је правац у ком се технологија креће, а то је имплементација у свим сегментима човековог утицаја, живота и рада. Пољопривреда је грана на коју тренутно има највећи импакт. Интеграција најновијих решења из области Интернета ствари (Internet of things ИоТ) у пољопривреди значајно је померила границе ка смарт пољопривреди.

Модерна пољопривреда се ослања на примену савремених приступа у третирању пољопривреде, коришћење разних алата и помоћне технологије, инжењеринг у развоју и употреби пестицида и хербицида, опрашивање дреновима различитим хемијским агенсима, технике вештачког опрашивања. Наведено је само део који човек имплементира а због чега природа има различите манифестације као одговор и реакцију. Цветање природе у вештачким условима и периодима кад за то није време, а свако неприродно буђење природе изазива поремећаје и у животима пчелињих друштава. Поред природних предатора, болести и инфекција као што су *Varroa destructor*, нових сојева вируса који се брзо шире, недостатка хране у окружењу, губитка матице, проблеми са тровањем пчела, су последњих година утицали на значајан нестанак и изумирање огромног дела пчелињих друштава. Такође и утицај поремећених метеоролошких прилика како у пролећу са великим кишама, летњим и јесењим сушним периодима, тако и у зимском периоду са топлијом климом и малим падавинама. Све ово утиче на нормалан живот и свакодневне рутине пчелињег друштва, како у периодима када се пчела храни тако и када је активна у прикупљању полена и нектара за зимски период.

Пчелари се сусрећу са бројним проблемима, поморима читавих пчелињих друштава и огромним штетама које се манифестују кроз смањење броја кошница, финансијским притисцима, што их приморава да имплементирају савремене приступе у пчеларству. Можда се не може утицати на поједине догађаје и дисбалансе који настају у природи, али се могу разумети. Већина наведених фактора се може пратити, снимати и могу се прикупљати подаци о њима, применом ИоТ-а и интеграцијом сензора у екосистем најужег пчелињег окружења.

Изазов је био створити унапређени систем који обрађује податке алгоритмима који се користе у напредној анализи података, користећи искуство стечено из сличних претходних студија. Иако се не може утицати на догађаје у пчелињем окружењу који се јављају природним путем, могу се пратити, мерити и прикупљати подаци о њима. Методолошком применом софтверских алгоритама можемо створити системе за предикцију и учити њихове обрасце. Догађаји и проблеми који су иницирани од стране

човека, такође се могу пратити, мерити и прикупљати подаци о њима, и на њих се може утицати како би се контролисао исход и спречила штета коју човекова интервенција може изазвати.

Овакви и слични проблеми се изучавају годинама, а методолошки приступи су се мењали током тог периода. Развојем прецизне микроелектронике и применом напредних методолошких приступа, омогућен је нов начин мерења података и њихова анализа. Примена оваквих система омогућава мапирање узрока проблема, као и приступе за спречавање и решавање истих. На тај начин се приближавамо решавању проблема апикултуре и савременог пчелара.

Систем развијен у оквиру ове докторске дисертације, неопходан је корак ка разумевању процеса који се одвијају у кошници. Свака кошница ће бити јединствена, јер њен екосистем зависи од микроклиматских параметара, и осталих фактора који су специфични за конкретну локацију и окружење. Следећи корак након разумевања процеса је управљање кошницом како би се услови у њој држали у оквирима који погодују пчелама.

Активности пчела, попут излазака и улазака, формирају низ података који се секвенцијално појављују током одређеног временског периода, омогућавајући на тај начин примену алгоритама за предвиђање у облику временских серија. Анализа временских серија је поступак обраде података кроз статистику и моделирање, са циљем стварања предвиђања и информисања стратегијског доношења одлука. Рекурентне неуронске мреже су постале често коришћен алат за ову врсту анализе, јер су се показале веома ефикасне у решавању задатака повезаних са временским серијама. У овом истраживању, тестирани су различити приступи, од најновијих до најсофистициранијих и најкомплекснијих.

Подаци се прикупљају путем IoT уређаја, док се компонента вештачке интелигенције примењује на прикупљене податке. Софтверска решења, попут апликација које користе вештачку интелигенцију, треба да прате хардверске компоненте, што представља кључни аспект система за управљање који је представљен у овом раду. Кроз истраживање се дошло до закључка да је управљање кошницама следећи корак у прецизном пчеларству, а имплементација IoT MAP (Internet of things monitoring and prediction) система за надзор и контролу кошница је неизбежна за архитектуру овог решења. Свеобухватно решење проблематике које је представљено у овој дисертацији састоји се из више независних система, од којих сваки доприноси на свој начин.

Приказана решења направљена су да служе пчеларима и биолозима за праћење и анализу понашања пчела, узимајући у обзир различите факторе као што су метеоролошки услови, годишње доба, неочекиване промене у понашању или параметрима мерења, итд. Уколико се детектују одступања у параметрима, деформације у понашању или промене у стању, пчелари имају могућност да искористе аутоматски систем за затварање кошница, проветравање кошница, премештање кошница, активирање аутоматске хранилице или да предузму друге адекватне интервенције.

Докторска дисертација је организована по поглављима на следећи начин: У поглављу 2. направљен је преглед и урађена кратка анализа тренутног стања у овој области

која је била основ за научно полазиште ове дисертације. Поглавља 3. и 4. описују проблеме и изазове са којима се суочава савремено пчеларство имплементацијом савремених технолошких решења стварајући нову област – дигитално и прецизно пчеларство. У поглављу 5. опширно је описана област ИоТ-а, његова архитектура, поделе, хардвер и софтвер. Дат је детаљан опис свих технологија и слојева. Поглавље 6. даје увид у архитектуру система за мониторинг који је резултат ове дисертације, а поглавље 7. детаљно описује шему електронских компоненти, сензоре и микроконтролере који су коришћени. У поглављу 8. приказан је увид у примену вештачке интелигенције и неуронских мрежа у области ИоТ-а, док поглавље 9. описује моделовање и примену модела рекурентних неуронских мрежа за предикцију кретања пчела, анализу података и дизајн модела. Софтверски део, апликативно решење и cloud систем приказан је детаљно у поглављу 10. Поглавље 11. описује закључак и будућа истраживања која се надовезују на досадашња решења, а у поглављу 12. дат је комплетан списак коришћене литературе.

2. МОТИВАЦИЈА И ЦИЉ РАДА

Проблем који се истражује и решава је круцијални и планетарни проблем, са умирањем и нестајањем пчела неизбежне су катастрофалне последице за глобални екосистем. Напредна индустрија, лош однос према природи и окружењу, савремена пољопривреда и агротехничке мере применом пестицида и хербицида, затим климатске промене итд., довеле су до озбиљних негативних промена у пчелињем свету. Долази до наглих помора читавих друштава, миграција из кошница, тровања приликом прикупљања полена, хране и меда и на тај начин мења се локацијски екосистем јер пчеле представљају најбитнију карику у опрашивању биљака.

Циљ дисертације је примена Internet of Things (ИоТ) технологија за развој, конструисање и изградњу целокупног система интегрисане кошнице са сензорима и микроконтролерима. Систем пружа увид у здравље пчелиње заједнице и омогућава пчеларима да добију упозорења на нагла одступања вредности параметара у односу на номиналне вредности. Систем би на основу дубинске анализе требало да предвиђа промене параметара и указује на тренутно стање у кошници, што представља директну предност у односу на ручни преглед кошнице који може изазвати стрес код пчелиње заједнице. Систем најпре прати улазно-излазне фреквенције, док се модул за дубинску анализу података користи за предвиђање кретања пчела уз помоћ фактора са највећим утицајем. Web оријентисана апликација омогућава контролу и управљање кошницама. Манипулација и приказ података у реалном времену се спроводи преко интерфејса који заокружује ИоТ систем за праћење и предвиђање са опцијама за активирање аларма.

Примена паметних сензорских модула на перцептивном ИоТ слоју за прикупљање података из микроекосистема пчелиње заједнице и интеграција сензора за прикупљање амбијенталних и метеоролошких промена обезбеђују квалитативне и квантитативне улазне податке. Дизајнирање пролазних капија на улазу у кошницу, са паром фото-рефлектујућих отпорника за активирање сензора и детекцију смера кретања, омогућава прикупљање података о улазно-излазној фреквенцији кретања пчела. Примена аналитичких модела на скуп ових података омогућава откривање окидача за кретање пчела. С обзиром на специфичности екосистема кошнице, сензорски систем је потребно инсталирати тако да не ремети свакодневне активности пчела. Сензори постављени на прототипу кошнице омогућавају аквизицију изворних података који се потом интегришу у микроконтролерима, одакле се шаљу у базу података.

Анализа података прикупљених кроз сензорски и перцептивни слој, њихово укрштање са улазно-излазним фреквенцијама пчела, омогућава пчеларима да идентификују потенцијалне промене или ризике пре него што дођу до тачке где би могли да изазову озбиљну штету или губитак пчелиње заједнице. Кроз ову анализу, пчелари могу пратити здравље своје пчелиње заједнице, са упозорењима на било какве наглашене промене у понашању или активности које би могле указивати на проблем.

Увођење модела вештачке интелигенције унапређује капацитет овог система. Узорци понашања, одступања вредности у односу на номиналне и предвиђање трендова

кретања параметара се могу детектовати употребом аналитичких модула за обраду велике количине података. Ови подаци даље служе за тренирање модела вештачке интелигенције који могу пружити још прецизније и корисне информације за пчеларе.

Апликација је пројектована да приказује податке у реалном времену, са административним подешавањима за манипулацију подацима. То пружа пчеларима једноставан, интуитиван начин за праћење својих кошница, без обзира на то где се налазе. Поред тога, апликација пружа online сервис, надзор и управљање кошницама са уграђеним модулом за аналитику и предикцију за праћење кретања података према границама.

Овакав интегрисани приступ, који користи технологије ИоТ, аналитику података и вештачку интелигенцију, пружа свеобухватно решење за пчеларе, помажући им да боље разумеју, прате и управљају својим пчелињим друштвима. Поред тога, овакав систем такође има потенцијал да помогне у идентификацији и разумевању ширег утицаја промена у окружењу на пчеле, што би могло имати значајне импликације за очување пчела и њихову улогу у екосистему.

3. ПРЕГЛЕД И АНАЛИЗА ЛИТЕРАТУРЕ

Анализа литературе наглашава проблеме са угинућем пчела, колапса читавих колонија [1][2] и њихове везе са климатским променама и човековим деловањем [3][4], недостатак хране у непосредном окружењу. Пољопривредници имају негативан утицај на екосистем пчела немарном употребом пестицида и хербицида, као и неусаглашеним терминима третмана заштите [5]. Истраживања указују на потребу за интелигентним праћењем пчелињих станишта и микро-екосистема, како би боље разумели и управљали животном средином пчела. Претходни радови пружају увод у ову проблематику и истичу значај различитих зависних и независних фактора који утичу на кретање и живот пчела [6]-[9].

Циљ ових истраживања је да се пружи дубље разумевање пчеларства и да се пчеларима омогући боље неговање и надзор пчела. Константан мониторинг, примена вештачке интелигенције и увид у стање пчела у кошници омогућавају брзу реакцију на промене и потенцијалне проблеме. Ова истраживања обухватају широк спектар фактора и аспеката који утичу на живот пчела, што је од великог значаја за одржавање здравих пчелињих заједница и одржавање природног процеса опрашивања биљака.

Ова анализа додатно указује на потребу за интеграцијом дигиталних алата и технологија у пчеларство, како би се унапредио ниво надзора и неге пчела, са циљем одржавања њиховог здравља и повећања продуктивности.

Прецизно пчеларство, које се у последње време активно истражује, се фокусира на индивидуални приступ пчелама и њиховом окружењу [10]-[16]. Online алати и ИоТ технологије омогућавају даљинско континуирано праћење и контролу понашања пчела, без излагања пчела додатном стресу и непродуктивним активностима [17]-[24]. Ови приступи се фокусирају на праћење појединачних пчелињих друштава и њиховог окружења, што омогућава дубље разумевање фактора који утичу на њихову производност и здравље.

Анализирани радови указују на важност утицаја неких од фактора, као што су температура и влажност ваздуха [25]-[28], присуство гасова [29][30], вибрације, присуство буке и звучне фреквенције [31]-[35], временски услови као што су присуство ветра и кише [36], UV и IC зрачење, интензитет дневне осветљености, итд. Ова истраживања представљају интеграцију свих ових фактора заједно. Прикупљање података се врши кроз временске серије у сатним или дневним интервалима. У претходним радовима, коришћени су различити приступи и анализе података, док се истраживања ове дисертације која следе ослањају на напредне методе и анализу података за добијање тачних резултата.

Константан мониторинг и мерење параметара у кошници и околини има велики значај за праћење утицаја појединачних и условних фактора на живот пчела [37]-[46]. Многи радови су истакли значај ИоТ технологија у дизајнирању система за овај вид пчеларства [47]-[52], наглашавајући значај прецизног пчеларства и индивидуалног приступа пчелињег друштва. Укључивање различитих фактора и метода анализе података

помаже у добијању комплексних и корисних информација. Коришћење ИоТ технологија и напредних метода анализе података омогућава прецизније и ефикасније пчеларство.

У неким од радова, примењени су алгоритми стабла одлучивања и дубоке неуронске мреже за класификацију стања кошнице и активности пчелињег роја [53]-[55]. У овом истраживању, примењују се методе регресије са циљем да се на основу активности пчела процени статус кошнице, односно да ли су услови у кошници здрави или нездрави.

Праћени су различити параметри како унутар тако и ван кошнице, са циљем истраживања да се проучи утицај различитих фактора и параметара, њихов интензитет и међусобна зависност. Укључивање свих ових параметара и њихова анализа помажу у разумевању и процени стања кошнице и пчела, као и у откривању најмањих одступања и поремећаја у њиховом функционисању у односу на природну средину [56]-[59].

4. САВРЕМЕНО ПЧЕЛАРСТВО - ДИГИТАЛНО ПЧЕЛАРСТВО

4.1. Проблеми и изазови савременог пчеларства

На основу досад изведеног, проблем који се решава у овом истраживању је унапређење и оптимизација пчеларства кроз примену дигиталних технологија и вештачке интелигенције с циљем побољшања здравља кошнице (друштва), продуктивности и одрживости пчела (пчеларства). Главни циљеви истраживања су праћење и анализа активности пчела унутар и око кошнице, предвиђање понашања пчела на основу прикупљених података, идентификација кључних фактора који утичу на живот и кретање пчела, као и развој система за управљање кошницама путем дигиталних алата.

Проблем је дефинисан кроз изазове с којима се пчелари суочавају, укључујући поред свега претходно наведеног: масовно угинуће пчела, миграцију пчела услед климатских промена, утицај временских услова на цветање биљака, потребу за праћењем здравља и активности пчела, ради правовременог препознавања и адекватног реаговања на евентуалне проблеме. Коришћење дигиталних технологија, као што су ИоТ уређаји, сензори за мерење различитих параметара екосистема кошнице и околине, примена вештачке интелигенције за анализу прикупљених података, омогућава пчеларима да добију прецизне информације о стању својих кошница у реалном времену, предвиде потенцијалне проблеме и предузму одговарајуће мере.

Овакав приступ дефинисан као „дигитално пчеларство“ (дигитализација пчеларства) има за циљ побољшање здравља пчелињег друштва у кошници, продуктивности пчела, очување локалних екосистема и опрашивање биљака, што има значајан утицај на биодиверзитет и одржавање равнотеже у природи. Истовремено, омогућава пчеларима да ефикасније управљају својим кошницама, и доносе информисане одлуке засноване на детаљној анализи података, чиме се унапређује целокупна пракса пчеларства.

Изазови у примени дигитализације кошница се могу посматрати са више аспеката од којих су најзначајнији:

- **Технички изазови.** Имплементација дигиталних технологија у пчеларству захтева одговарајућу инфраструктуру, сензоре, ИоТ уређаје и софтверске алате. Интеграција ових технологија и њихово правилно функционисање у реалном окружењу представљају техничке изазове са којима се пчелари могу суочити. Обезбеђивање услова за правилан рад електронских компоненти, како би резултати мерења или контроле уређаја били исправни.
- **Анализа великог броја података.** Прикупљањем података о активностима пчела, условима из окружења и другим релевантним факторима за живот и функционисање пчела, генеришу се велике количине података. Ефикасна анализа и интерпретација овако добијених података, захтева примену напредних аналитичких метода, као што

је примена метода вештачке интелигенције, ради доношења информисаних одлука на основу обраде података.

- **Едукација корисника (пчелара).** Увођење дигиталних технологија у пчеларство захтева и едукацију корисника о њиховој адекватној примени, употреби и интерпретацији резултата, како би се постигао максималан ефекат. Разумевање метода, и обученост пчелара у коришћењу дигиталних алата и технологија битни су за успешну имплементацију и савремено управљање кошницама.

Све наведено представља изазове савременог приступа дигиталног пчеларства, чијом применом пчелари имају могућност да унапреде своју праксу, повећају ефикасност пчеларства, сачувају здравље пчела и допринесу очувању пчелињих заједница, и животне средине.

4.2. Дигитално пчеларство - примена информационих технологија

Дигитално пчеларство представља примену напредне технологије и дигиталних алата за управљање и узгајање пчелињих заједница. Коришћењем софтвера за праћење и анализу података о стању кошнице и продуктивности пчела, као и ИоТ уређаја за праћење и контролу пчелињих заједница на даљину, ова област отвара нове могућности за пчеларе у побољшању квалитета и ефикасности пчеларства.

Један од главних изазова у пчеларству је управо праћење фреквенције кретања пчела. Захваљујући софтверима за праћење, пчелари могу пратити параметре кретања, микроклиматске услове, као и друге информације о активностима пчела унутар кошнице, храњење, размножавање итд. Применом напредне аналитичке методе и вештачке интелигенције за анализу зависних и независних параметара могу се предвидети њихови трендови. Ово омогућава да се открију корелације и утицаји фактора као што су амбијентални услови, расположивост хране и метеоролошке промене на производњу меда и општег стања пчелињег друштва.

Испитивање и развој дигиталног пчеларства има значајну улогу у одржавању локалних екосистема и опрашивању природе. Због своје важности, ова област захтева сарадњу између различитих актера, укључујући научнике, пчеларе, технолошке стручњаке, пољопривреднике, локалну заједницу итд. Дигитално пчеларство представља иновативан приступ који има потенцијал да унапреди пчеларство, што је важно за одрживост пчеларства, заштиту пчела, као и подршку у производњи меда и производа од меда.

Бенефити од имплементације дигиталних технологија у пчеларству су вишеструки, ова решења између осталог омогућавају:

1. **Праћење здравља пчела.** Употребом софтвера и сензора за праћење здравља пчела, пчелари могу да идентификују потенцијалне болести и штеточине на време, што им омогућава да правовремено предузму одговарајуће мере за заштиту пчела и спречавање ширења болести, или потенцијалног проблема.

2. **Предвиђање кретања пчела.** Анализом података о активностима пчела, околним климатским условима и другим узрочним факторима кретања пчела, могуће је користити алгоритме машинског учења и предиктивне моделе, како би се предвидело кретање пчела и њихово понашање у односу на те факторе. Ово омогућава пчеларима да прилагоде своје активности и ефикасније управљају кошницама.
3. **Мониторинг услова околине.** Коришћење сензора за мерење метеоролошких параметара температуре, влажности, квалитета ваздуха, зрачења, присуства магле, ветра, омогућава пчеларима да прате и разумеју утицај временских услова на пчеле и њихову активност. На основу тих информација, пчелари могу предузети одговарајуће мере за обезбеђивање оптималних услова за пчеле.
4. **Интеграција ИоТ уређаја.** Комбиновање ИоТ уређаја, као што су сензори, актуатори, микроконтролери, са софтверским решењима омогућава пчеларима да даљински прате стање својих кошница и примају информације у реалном времену. Ово пружа могућност да се правовремено реагује на промене и потенцијалне проблеме.
5. **Аналитика и вештачка интелигенција.** Комбинација аналитике података и вештачке интелигенције омогућава пчеларима да добију информисан увид из прикупљених података, и да тек онда донесу одлуке. Напредне технике анализе података могу идентификовати обрасце, трендове и узрочно-последичне везе које могу бити корисне пчеларима за оптимизацију услова.
6. **Едукација и размена знања.** Дигитална решења омогућавају пчеларима да се повежу, деле искуства и размењују знање путем интернет платформи, форума и апликација. Овакав приступ одржава континуирано образовање и подршку међу пчеларима, што може довести до побољшања целокупне праксе пчеларства.
7. **Систем за упозоравање и аларме.** Интеграција ИоТ уређаја са системима за упозоравање омогућава пчеларима да добију обавештења и аларме о одређеним променама или проблемима у пчелињим заједницама. На пример, детекција пада температуре испод одређеног нивоа или присуство штетних супстанци може аутоматски генерисати упозорење.

Узимајући у обзир све ове факторе, дигитална решења имају потенцијал да донесу значајна унапређења и бенефите у пчеларству. Међутим, како би се успешно имплементирала дигитална решења у пчеларство, потребно је превазићи претходно наведене препреке и изазове, као што су имплементација технике и технологије, обука пчелара за коришћење технологија и тумачење резултата. Са развојем ИТ инфраструктуре и обуке пчелара, могуће је остварити пун потенцијал дигиталног пчеларства.

Поред хардверских компоненти, софтвер има важну улогу у анализи и интерпретацији прикупљених података. Специјализовани софтверски алати омогућавају праћење здравља пчела, анализу понашања, предвиђање промена и идентификацију

фактора који утичу на продуктивност. Софтверски алати пружају пчеларима могућност да генеришу детаљне извештаје, графове и препоруке на основу прикупљених података. На пример, софтвер може пружити пчеларима информације о потенцијалним болестима или штеточинама на основу анализе података о активностима пчела и услова окружења. Такође, софтвер омогућава аутоматизацију појединих процеса, као што је аутоматско управљање унутрашњом климом кошнице (температура, влажност) или планирање активности у складу са природним циклусима пчела.

5. ТЕХНОЛОГИЈЕ ИНТЕРНЕТА СТВАРИ (Internet of things IoT)

Интернет ствари (ИоТ) представља револуционарну технологију која свет информационах технологија уводи у еру web 3.0. Ова технологија мења начин на који се уређаји али и људи повезују, комуницирају и функционишу у свакодневном животу. Ова технологија је резултат напретка у области хардвера, софтвера, сензора, бежичне комуникације и cloud технологија. ИоТ омогућава да физички објекти, као што су smart уређаји, електронски апарати из нашег окружења, возила, кућни апарати, и слична инфраструктура, буду повезани и комуницирају међусобно путем интернета или директном бежичном везом (Слика 5.1).



Слика 5.1 Примена ИоТ технологија у свим областима

ИоТ, дакле представља свеprisутни технолошки феномен који се инфилтрирао у све аспекте нашег функционисања, превазилазећи његову основну дефиницију, што указује да је његов значај и потенцијал непрекидно растући.

Дефиниција ИоТ-а обухвата систем повезаних физичких уређаја који чине: крајњи уређаји за прикупљање података, уређаји за обраду и презентацију података, комуникационе уређаје за жичну и бежичну комуникацију, софтвер који обухвата

оперативне системе, апликације и вештачку интелигенцију са интелигентним системима за одлучивање или управљање. Интернет ствари, са својим развијеним инфраструктурним системом, омогућава напредне услуге повезивањем физичких и виртуелних ствари на основу постојећих и интероперабилних информационих и комуникационих технологија у развоју.

Од почетне идеје, IoT је пролазио кроз еволуцију паралелно са развојем технологије и друштва. Постоји више индикација о настанку самог термина IoT-а, а једна од најчешће помињаних референцира професора Кевина Аштона са Масачусетског института за технологију (MIT). У његовој почетној визији, IoT се односио на надоградњу рачунара сензорима, што би омогућило ефикасније прикупљање података без претеране зависности од људског учешћа. Од својих почетака као концепт, IoT је пролазио кроз зрелост технологије и добио примену у облику у коме га препознајемо данас. Идеја IoT-а се развијала и променила ток са појавом нових технологија и практичне примене. Данас, IoT системи укључују велики број уређаја, услуга и апликација који превазилазе саму комуникацију између машина (M2M). IoT уређаји данас имају своје IP URL адресе, а повезивање система се јавља кроз различите протоколе, домене и технологије. Идеја IoT-а базирана на M2M комуникацији и рачунарству у cloud-у омогућава ефикасније процесе, аутоматизацију и анализу података у реалном времену. Главни циљ IoT технологија је омогућити праћење, управљање и контролу уређаја на даљину, што има значајан утицај на повећање ефикасности, продуктивности и смањењу трошкова.

IoT је постављен као носилац четврте индустријске револуције, која је покретач данашње технологије и развоја. Кроз IoT, дигитални свет и физички свет се све више прожимају, стварајући нове могућности за примену, оптимизацију, аутоматизацију и побољшање ефикасности у многим секторима, укључујући пољопривреду, здравство очување животне средине као приоритне. Префикс "смарт" се данас примењује у свим аспектима IoT система, док рачунарство у cloud-у додаје нову димензију укључивањем услуга у cloud-у. Потенцијал IoT-а је неограничен, а интеграција са вештачком интелигенцијом и машинским учењем је допринела унапређењу овог феномена. Кроз прикупљање и анализу података са свих повезаних перцептивних уређаја, IoT има потенцијал да донесе нова сазнања и подигне квалитет живота у многим аспектима.

5.1. Архитектура IoT

Архитектура IoT-а представља основни оквир који омогућава комуникацију, интеракцију и интеграцију између свих компоненти у IoT систему. У ужој подели, IoT архитектура се састоји од неколико основних компоненти (Слика 5.2). Прва компонента је хардвер, који обухвата све физичке уређаје и сензоре који се користе за прикупљање података. Овај хардвер може бити уграђен у различите објекте, уређаје или окружења.



Слика 5.2 Архитектура ИIoT-а

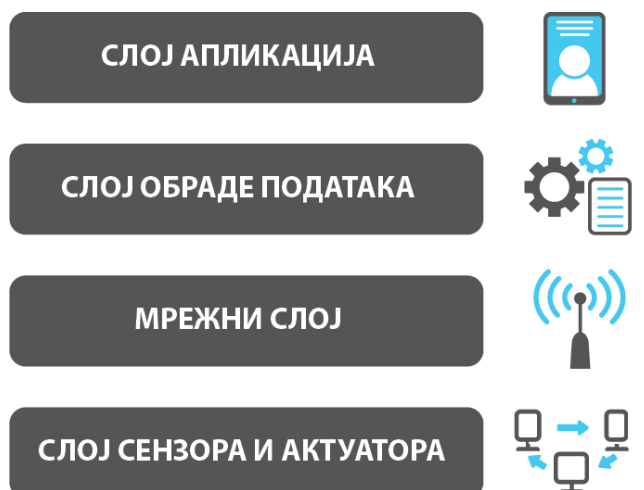
Друга компонента је инфраструктура, која се односи на мрежну и комуникациону инфраструктуру која омогућава померање података између различитих уређаја и система. Ово укључује различите протоколе, мрежне везе и сервере који обрађују податке, складиште и обезбеђују приступ подацима.

Трећа компонента је софтвер, који обухвата оперативне системе, апликације и програмске интерфејсе који омогућавају комуникацију и обраду података. Овај софтвер је одговоран за управљање и контролу уређаја, превођење и интерпретирање података, као и обезбеђивање корисничког интерфејса за приступ и контролу ИIoT система.

Четврта компонента је сервис, који обезбеђује окружење у којем ИIoT решење пружа одређене услуге и функционалности корисницима. Ово могу бити различите апликације, платформе, пословни модели који користе податке и могућности ИIoT система да пруже корисне информације, управљају ресурсима и на тај начин унапреде корисничко искуство.

Постоји и алтернативна подела ИIoT архитектуре (Слика 5.3) која се фокусира на основне компоненте. Она обухвата сензоре и актуаторе, мрежну инфраструктуру, затим слој обраде података и апликациони слој. Физички уређаји, сензори и актуатори овде се односе на хардвер, док се софтвер овде дели на обраду података и апликативни слој, а мрежни слој представља у ствари инфраструктуру комуникације. Технологија се односи на различите технолошке стандарде и принципе који подржавају ИIoT, док протоколи представљају начин комуникације и преноса података у ИIoT систему.

Обе поделе су основне и имају заједничку перцепцију есенцијалних компоненти ИIoT-а и омогућавају целовиту и комплексну функционалност. Одабир архитектуре зависи од конкретних потреба и захтева пројекта, а правилна архитектура је од кључног значаја за успешну имплементацију и оптимално функционисање ИIoT система.



Слика 5.3 Проширена подела ИIoT архитектуре

Иако обе поделе имају исту перцепцију есенцијалних компоненти, у ширем смислу ИIoT данас представља много комплекснији систем, који се константно унапређује. Проширена подела обухвата још неколико важних компоненти ИIoT архитектуре међу којима посебно место заузима мрежна инфраструктура. Она обухвата различите уређаје и технологије које омогућавају повезивање ИIoT уређаја у јединствену мрежу. То укључује рутере, свичеве, gateway-е, бежичне технологије, као и различите протоколе за пренос података. Мрежна инфраструктура пружа основу за комуникацију између ИIoT уређаја и омогућава размену података са удаљеним серверима и cloud платформама.

Уз то, неопходно је да ИIoT архитектура обезбеди безбедност и заштиту података. С обзиром на бројне повезане уређаје и проток података, безбедност је такође важна компонента. То подразумева употребу енкрипције података, аутентификацију уређаја, заштиту од напада и сигурносне механизме за очување приватности корисника. Такође, важно је да ИIoT систем буде отпоран на различите врсте напада и да има механизме за детекцију и реаговање на потенцијалне безбедносне претње.

Још један аспект ИIoT архитектуре који је битан је аналитика података. Велике количине података прикупљених од стране ИIoT уређаја могу бити искоришћене за добијање вредних информација и увида. Аналитика података омогућава препознавање узорака, идентификацију трендова, предвиђање будућих догађаја и доношење информисаних одлука. Комбинација машинског учења, вештачке интелигенције и аналитике података омогућава да ИIoT систем постане интелигентан и да пружи додатну вредност корисницима.

5.2. ИIoT Хардвер

Интернет ствари (ИIoT) је данас подржан широким спектром електронских компоненти које чине хардверски систем и део инфраструктуре. Овај систем обухвата сензоре, моторе, релеје, уређаје за конекцију, рутере, бежичне уређаје, RF уређаје, сервере, паметне уређаје, таблете, рачунаре и многе друге. Захваљујући константном развоју и

напретку технологије, свака од ових хардверских компоненти може бити повезана на cloud или online систем, као и на било коју мрежу. Повезани на овај начин IoT уређаји добијају функције smart технологије. Ови уређаји су скалабилни и имају вишенаменску сврху, неки служе за контролу других уређаја, неки за прикупљање података, а неки за презентацију. Оно што је најважније јесте да су конфигурабилни и модуларни, те могу чинити једноставне склопове, али исто тако могу бити комбиновани у веома сложене системе састављене од више мањих IoT система.

Најважније хардверске компоненте IoT-а су:

- сензори (акцелерометри, температурни сензори, гасни RFID сензори...),
- актуатори (за контролу климе, осветљења, даљински управљане уређаје...),
- микроконтролери (Espressif Adafruit, Teensy, Intel Galileo, Arduino...)
- микрорачунари (ARM Cortex, SimpleLink, NVIDIA Jetson, Raspberry Pi...),
- мрежна инфраструктура (рутери, свичеви, gateway, Wi-Fi, Ethernet...).

Сензори су од виталне важности за операције унутар IoT окружења, где служе за снимање, праћење и прикупљање података. Ови уређаји, који се крећу од крајње једноставних сензора до комплексних модула, опремљени су различитим компонентама које проширују њихову функционалност. Сложенији сензорски модули могу садржати контролну јединицу, напајање, модул за управљање, модул за предобраду сигнала, као и модул за комуникацију. Комуникацијски модули могу користити различите технологије као што су Wi-Fi, ZigBee, Bluetooth, RF модул, радио пријемник и друге, омогућавајући сензорима да преносе прикупљене податке на друге IoT уређаје или на cloud.

Сензори могу такође бити програмирани да извршавају одређене задатке или да се аутоматизују применом софистицираних софтверских система за доношење одлука (машинско учење, вештачка интелигенција на нижим слојевима). Ово значи да сензори могу динамички да се прилагоде променама у окружењу, побољшавајући тиме ефикасност и функционалност IoT система. Нова генерација сензора је често минијатурних димензија, што омогућава њихову интеграцију у различите уређаје, укључујући оне које носимо на себи, попут наочара, паметних сатова и других сличних уређаја. Ово проширује домен примене IoT-а, чинећи технологију доступном у свакодневном животу. На фундаменталном нивоу, сензори представљају тачку сусрета између физичког и дигиталног света. Они преводе реалне, физичке феномене у дигиталне податке који се могу обрадити, анализирати и користити за доношење информисаних одлука. У том смислу, сензори су суштинска веза између реалног света и дигиталних IoT система.

Актуатори представљају незаобилазну компоненту сваког IoT система, служећи као уређаји који конвертују електричне сигнале у физичке акције. Њихова примарна улога је да извршавају команде примљене преко мреже, што се може манифестовати различитим физичким акцијама у зависности од специфичних захтева система. Актуатори су нарочито важни за аутоматизацију и контролу система у реалном времену. На пример, актуатору

може бити наређено да промени стање или положај неког објекта, да покрене одређени механички процес, или да се активира на промене физичких параметара окружења, као што су температура или светлост. Примена ових уређаја је широка и обухвата различите индустријске секторе, укључујући производњу, енергетику, пољопривреду, саобраћај и многе друге.

Актуатори се данас најчешће користе у комбинацији са сензорима у IoT окружењу. Сензори прикупљају податке о окружењу и шаљу их на централни IoT систем, који затим анализира ове податке и шаље одговарајуће команде актуаторима. Актуатори онда извршавају ове команде, мењајући физичко стање система у складу са добијеним инструкцијама. Ова синергија између сензора и актуатора омогућава стварање интелигентних IoT система способних за аутоматизацију комплексних задатака у реалном времену.

Микроконтролери су компактни електронски уређаји који се састоје од једног интегрисаног склопа (чипа), који обухвата микропроцесор, меморију (RAM, ROM, EPROM) и различите периферне интерфејсе. Због своје мале величине, микроконтролери су посебно погодни за употребу у уграђеним системима, где служе као централна компонента која контролише рад других уређаја или система, укључујући сензоре, актуаторе и преноснике података. Иако су микроконтролери генерално слабији у поређењу са микропроцесорима опште намене, они су оптимизовани за ефикасно управљање код ограничених хардверских ресурса и могу да обављају бројне задатке у реалном времену. Због ове флексибилности, микроконтролери су постали важна компонента у IoT уређајима, где покривају неопходну улогу за интелигентно управљање и контролу различитих функција уређаја.

Популарни примери микроконтролера укључују Arduino микроконтролере (као што су Uno, Mega, и друге верзије), као и Adafruit Feather микроконтролере. Ови уређаји се састоје од минималног сета компоненти потребних за извршавање једноставних логичких операција, укључујући микропроцесор (CPU), меморију (RAM), складиште (ROM), улазно/излазну јединицу, магистрале и напајање. Микроконтролери се класификују према различитим критеријумима, укључујући врсту меморије (уграђени или екстерни), број битова у архитектури (8-битни, 16-битни, 32-битни), и скуп инструкција (CISC - Complex Instruction Set Computing, RISC - Reduced Instruction Set Computing). Ове разлике у карактеристикама дају различите нивое перформанси и ефикасности, што омогућава да се при дизајну архитектуре IoT система бирају одговарајући микроконтролери у складу са специфичним захтевима.

Микрорачунари су електронски уређаји који се састоје од микропроцесора или микроконтролера који делују као централна компонента свеобухватног рачунарског система. Ови уређаји комбинују микропроцесор, RAM и ROM меморију, централну процесорску јединицу (CPU), напајање, као и различите периферне интерфејсе, укључујући тастатуру, екран, уређаје за складиштење података и мрежне функције. Овако интегрисана архитектура омогућава микрорачунарима да функционишу аутономно или као интегрални део већег система.

Микропроцесор је суштински део рачунарског система који обавља рачунске операције, док микроконтролер интегрише микропроцесор са додатним компонентама за управљање специфичним уређајима или системима. Микрорачунари су компактни, једноставни за коришћење, релативно ниских трошкова и дизајнирани за одређене намене.

Неки од примера су популарни микрорачунари попут Raspberry Pi, NVIDIA Jetson или ARM Cortex, који пружају снагу и флексибилност потребну за реализацију сложених IoT пројеката. Они се често користе у домену аутоматизације и контроле система, где могу да обрађују податке, складиште информације и извршавају специфичне програме. Њихова велика предност лежи у томе што пружају више процесорске снаге и више могућности за конективност него стандардни микроконтролери, чинећи их идеалним за развој напредних IoT апликација. Такође, микрорачунари често подржавају различите оперативне системе и програмске језике, пружајући тако већу флексибилност у развоју IoT система.

Како би задовољили изазове у оквиру Интернета ствари, микроконтролери и микрорачунари морају испуњавати неколико услова:

- **Интеграција.** IoT је хетероген екосистем, што значи да мора да подржава различите уређаје, платформе и технологије. Микроконтролери и микрорачунари морају бити способни да се интегришу са различитим типовима уређаја и технологијама.
- **Скалабилност.** IoT екосистеми су обично велики и комплексни, са стотинама или чак хиљадама уређаја. Микроконтролери и микрорачунари морају бити скалабилни како би подржали овакве велике системе.
- **Семантичка интероперабилност.** У IoT екосистему, уређаји често морају да комуницирају са другим уређајима. Да би то постигли, морају да користе заједничке стандарде и протоколе.
- **Енергетска ефикасност.** IoT уређаје често покрећу батерије и раде на местима са ограниченим изворима напајања. Због тога, микроконтролери и микрорачунари морају бити енергетски ефикасни.
- **Сигурност.** IoT уређаји прикупљају и обрађују осетљиве информације, што их чини атрактивним метама за нападе. Микроконтролери и микрорачунари морају имплементирати сигурносне механизме како би заштитили те податке.

Уз растућу популарност IoT-а, мобилни уређаји као што су паметни телефони, таблети и паметни сатови постају неизоставни део овога концепта. Они представљају микрорачунарске системе вишеструке намене, који омогућавају корисницима да комуницирају и обављају интеракцију са IoT системима. Ови уређаји садрже интерфејсе за управљање IoT апликацијама и пружају командне могућности за контролу актуатора. Такође, долазе са интегрисаним комуникационим системом, који им омогућава више начина повезивања са осталим уређајима у IoT мрежи.

5.3. IoT Софтвер

Софтвер је такође једна од најзначајнијих компоненти у IoT системима и обухвата неколико компоненти које омогућавају рад и функционисање IoT система. Најважније компоненте софтвера у контексту примене у IoT технологији:

1. **Оперативни системи.** IoT уређаји обично раде на мањим оперативним системима оптимизованим за окружења са ограниченим ресурсима. Ови оперативни системи, омогућавају управљање ресурсима, извршавање програма и комуникацију са другим уређајима.
2. **Средњи софтвер.** Средњи софтвер служи као мост између оперативног система и апликационог слоја. Он обезбеђује услуге као што су управљање уређајима, обрада и анализа података, сигурност података и управљање комуникацијама.
3. **Мрежни протоколи.** IoT уређаји комуницирају једни са другима, као и са cloud-ом користећи различите мрежне протоколе. Неки од најчешће коришћених протокола у IoT-у су: MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), HTTP (Hypertext Transfer Protocol) и други. Ови протоколи омогућавају пренос података између уређаја и cloud-а, као и комуникацију између самих уређаја.
4. **Платформе у cloud-у.** Cloud системи обезбеђују инфраструктуру за складиштење, анализу и управљање великим количинама података које генеришу IoT уређаји. Неке од познатијих платформи у cloud-у су: AWS IoT, Microsoft Azure IoT Hub, Google Cloud IoT и IBM Watson IoT, оне омогућавају пренос и складиштење података, обраду података у реалном времену, анализу података и управљање IoT уређајима.
5. **Апликациони слој.** Апликациони слој је одговоран за обезбеђивање интерфејса крајњег корисника и дефинисање логике којом ће уређај комуницирати са другим уређајима и cloud-ом. Обухвата различите апликације и сервисе који омогућавају контролу, управљање и анализу података добијених са IoT уређаја.
6. **Безбедност.** Безбедност је критичан аспект IoT система. Изазови безбедности укључују заштиту уређаја од неовлашћеног приступа, шифровање комуникације, аутентификацију и ауторизацију уређаја и корисника, заштиту приватности и заштиту од напада. Безбедносни механизми и протоколи се примењују на свим нивоима IoT система како би се ови изазови савладали.

Софтверске компоненте IoT-а омогућавају управљање комуникацијама, обраду и анализу података, као и контролу уређаја у Интернету ствари. Константан напредак у овој области омогућава развој нових и иновативних IoT решења која имају значајан утицај на различите индустрије и секторе и као рефлексција на све аспекте свакодневног живота.

5.4. Оперативни системи

Оперативни системи за IoT уређаје, микроконтролере и микрорачунаре су намењени за рад са ограниченим хардверским ресурсима. У питању су мањи кодови попут *firmware*-а, дизајнирани да раде са процесорима мање снаге и мањом количином оперативне меморије. Интегрисане функције ових оперативних система обезбеђују потребне функционалности и захтеве за оперативност у реалном времену.

Битни захтеви у контексту IoT система су хетерогена повезаност, умрежавање, интероперабилност и мобилност. Будући да IoT уређаји прикупљају велике количине података који се морају пренети на друге уређаје или сервисе путем комуникационе инфраструктуре, управљање подацима је један од кључних захтева. Такође, важна је и повезаност кроз широк спектар комуникационих протокола и стандарда за међусобну комуникацију уређаја (Wi-Fi, Bluetooth, RF...).

Иако су оперативни системи за IoT уређаје компактни, они су потпуно функционални и способни за покретање више апликација истовремено, уз ефикасно управљање ресурсима и напајањем. Ово је битно због управљања процесорском снагом, повезивања, умрежавања, управљања удаљеним уређајима, складиштења, безбедности и за остале потребе IoT система. Поузданост је једна од најважнијих карактеристика оваквих система, заједно са њиховом вишеструком функционалношћу. Већина IoT оперативних система је *open-source*, са великом подршком и ангажовањем *open-source online* заједнице.

Неки од најчешће присутних оперативних система за IoT су:

- **FreeRTOS** је *open source* оперативни систем, развијен од стране Amazona који је једноставан и флексибилан за употребу у IoT апликацијама. Он пружа низ алата и функција за управљање процесима, меморијом, и комуникацијом.
- **Zephyr** је оперативни систем за уграђене (*embedded*) уређаје, који је пројектован да буде компатибилан са различитим архитектурама микропроцесора и микроконтролера. Нуди широк спектар функционалности за IoT уређаје.
- **Contiki** је *open source* оперативни систем који има мале системске захтеве. Специјализован је за примену код малих IoT уређаја, обезбеђујући им алате за комуникацију, обраду податке и повезивање на интернет.
- **RIOT** је оперативни систем отвореног кода за IoT уређаје који пружа низ услуга и функционалности за све интернет стандарде и мрежне технологије. Има подршку за безбедоносне механизме.
- **TinyOS** је *open source* оперативни систем за IoT уређаје који је пројектован и оптимизован за уграђене системе са малим меморијским капацитетом.
- **ARM Mbed OS** је оперативни систем за IoT уређаје са ARM архитектуром, који је намењен за повезивање IoT уређаја и апликација кроз различите врсте бежичних

мрежа. Пружа низ универзалних библиотека и услуга за управљање уређајима и комуникацијом. Има подршку за вишеслојну сигурност за безбедну комуникацију.

- **Windows IoT** је оперативни систем компаније Microsoft који је пројектован за развој ИоТ апликација на бази Windows платформе. Омогућава развој и управљање ИоТ уређајима користећи познате Windows развојне алате.
- **Linux IoT** је варијанта оперативног система Linux која је оптимизирана за ИоТ уређаје. Омогућава развој и управљање ИоТ уређајима користећи богате функционалности и услуге Linux платформе.
- **Raspberry Pi OS** је оперативни систем који је карактеристичан за Raspberry Pi ИоТ уређаје, чије је језгро базирано на Debian окружењу. Овај ОС има свој графички интерфејс што омогућава широку примену оваквих уређаја, а најчешће се користи код едукационих ИоТ система.
- **Tizen** је оперативни систем базиран на Linux kernelu који је развијен од водећих ИТ компанија на челу са Samsung-ом. Овај оперативни систем се користи за потребе крајњих уређаја, као што су телевизори, паметни сатови, мобилни телефони, аутомобили, којима у свом окружењу омогућава смарт функције. ОС ради на уређајима са x86 и ARM архитектуром.
- **Android Things** је ИоТ платформа развијена од стране Google-a, која користи Андроид оперативни систем као своју основу.

5.5. Комплексни слојеви ИоТ архитектуре

Водећи се наведеним, и тренду комплексности развоја технологија, долазимо до проширеног појма погледа на архитектуру ИоТ-а, где имамо више слојева:

- Слој перцепције
- Слој мрежа и протокола
- Слој средњег софтвера
- Слој обраде података
- Cloud системи и сервиси
- Слој апликације
- Слој примене вештачке интелигенције
- Систем за контролу и одлучивање
- Слој безбедности

Наведена архитектура је проширена јер није лако и једноставно свести је на само поједине делове система. Слојеви су међусобно зависни и преплићу своје функционалности.

5.5.1. Перцептивни слој

Перцептивни слој је слој уређаја улазних података који служе за прикупљање података и препознавање истих. У нашем случају то су подаци са претходно описаних сензора, актуатора и електронике за детекцију. Сензори у овом слоју откривају и прикупљају податке из свог окружења, претварајући физичке параметре као што су температура, светлост, звук, притисак итд. у дигитални формат. Овај процес превођења или 'транслације' је најбитнији за функционисање ИоТ система, јер омогућава подацима да буду обрађени, анализирани и интерпретирани на вишим слојевима архитектуре.

Важно је напоменути да слој перцепције не служи само за детекцију и прикупљање података, већ и за њихову предобраду и филтрирање. Компоненте у овом слоју често користе разне технике као што су обрада сигнала и детектовање карактеристичних појава да би уклониле грешке, или истакле специфичне информације пре него што су подаци послати на следећи слој за даљу обраду.

5.5.2. Мрежни слој и протоколи

Мрежни слој представља читав инфраструктурни сет кога поред протокола чине уређаји и технологије за међусобну комуникацију уређаја, као и пренос сигнала и прикупљених података. Подаци се шаљу до система за складиштење (локална меморија, сервер или cloud), или се шаљу другим уређајима коришћењем различитих бежичних сензорских мрежа, или других технологија. Транспортни слој архитектуре ИоТ система користи мрежни хардвер, комуникационе модуле, gateway, рутере, различите типове медија за пренос података. У систему из дисертације, то је микропроцесорски систем који представља компоненту мрежног слоја која обрађује и смешта податке у локалну меморију, а затим их шаље на cloud систем путем SIM модула. Исто тако, преко мрежног слоја шаљу се и команде наредбе за контролу модула на кошници.

Сегмент мрежне архитектуре задужен је и адресирањем (управљање адресама) и управљањем уређаја (Internet of Things Device Management). Овај сегмент обезбеђује праћење, идентификацију, аутентификацију, конфигурацију и управљање проширеног низа ИоТ уређаја у мрежи, што омогућава ефикасно управљање ресурсима и осигурава континуирану функционалност система.

У делу који следи биће детаљније описане неке од најзаступљенијих бежичних технологија које користе ИоТ уређаји, као и протоколи за комуникацију у мрежном слоју.

LoRaWAN (Long-range Radio Wide Area Network) је бежична мрежна технологија дизајнирана за дуже домете и покривање широког подручја. Пружа могућност повезивања великог броја уређаја у окружењу. Карактерише је изузетно мала енергетска потрошња, двосмерна комуникација (M2M) и могућност примене безбедоносних слојева комуникације. Покрива комуникацију у широком распону користећи star топологију, а подаци се од LoRaWAN gateway-а преносе на cloud или сервер преко стандардног IP протокола (Табела 5.1).

Табела 5.1 Перформансе и техничке спецификације LoRaWAN технологије.

Перформансе	Техничка спецификација
Домет сигнала	До 15 km (директна видљивост и преко)
Брзина преноса података	До 50 kbps
Фреквенција рада	Од 863 MHz до 873 MHz (лиценца за Европу)
Енергетска ефикасност	Ниска потрошња енергије
Сигурносни протоколи	Шифровање података (енкрипција)

WiFi (Wireless Fidelity) је једна од најчешћих технологија за повезивање уређаја и комуникацију средњег домета око 100 метара, стандарди су IEEE 802.b/a/g/n/ac за Wi-Fi 5 генерације са брзином преноса података од 433 Mbps до 6.9 Gbps. Шеста генерација доноси стандард 802.11ax и брзине преноса података од 574 Mbps до 9.6 Gbps. Карактерише је star топологија, а радне фреквенције су 2.4 GHz, 5 GHz и 6 GHz. Нуди широку палету подршке различитим уређајима, користи се за размену великих података и пружа поуздану комуникацију за уређаје у блиском окружењу. Ради на принципу IP протокола (Табела 5.2).

Табела 5.2 Перформансе и техничке спецификације WiFi технологије.

Перформансе	Техничка спецификација
Домет сигнала	Варира (неколико стотина метара за покривеност рутера на отвореном, јачим спољашним антенама и дуже)
Брзина преноса података	Wi-Fi 5 генерација, стандард 802.11ac, 433 Mbps - 6.9 Gbps, Wi-Fi 6 генерација, стандард 802.11ax 574 Mbps – 9.6 Gbps
Фреквенција рада	2.4 GHz, 5 GHz и 6 GHz
Енергетска ефикасност	Захтева релативно високу енергију
Сигурносни протоколи	WPA и шифровање података (енкрипција)

Мобилна технологија (Mobile Technology) се односи на технологију повезивања уређаја која омогућава пренос велике количине података и комуникацију путем мобилних протокола. Ове технологије су доступне у широком спектру различитих стандарда, укључујући 2G, 3G, 4G, 5G и 6 генерација која је у тест фази. Мобилна технологија је део ИоТ технологије, због своје широке заступљености, као и брзине преноса података и поузданости у раду. Омогућава брз пренос велике количине података и заштићену комуникацију између уређаја, и врло је заступљена у реализацији многих ИоТ апликација. У систему који је део ове дисертације, користи се ова технологија за слање података и примање команди (Табела 5.3).

Табела 5.3 Перформансе и техничке спецификације мобилне технологије.

Перформансе	Техничка спецификација
Домет сигнала	Варира (Зависи од конкретне технологије, до неколико километара на отвореном простору и неколико стотина метара у затвореном простору.)
Брзина преноса података	5G до 10 Gbps, 4G до 100 Mbps,

Фреквенција рада	5G од 600 MHz до 100 GHz, 4G око 2 GHz
Енергетска ефикасност	Захтева релативно високу енергију (зависи од уређаја)
Сигурносни протоколи	Шифровање података (енкрипција), аутентификација, ауторизација

Thread је бежична мрежна технологија за повезивање уређаја у IoT, заснована на IEEE 802.15.4 бежичном стандарду. Thread користи 6LoWPAN, и mesh топологија за повезивање уређаја. Има могућност за повезивање са великим бројем сензорских модула и уређаја, као и скалабилност у различитим врстама IoT апликација. Базиран је на IPv6 протоколу и AES енкрипцији, што га чини применљивим у IoT системима, са широким спектром интероперабилности. Thread технологију карактерише сигурност, флексибилност и једноставност коришћења у повезивању и раду IoT уређаја (Табела 5.4).

Табела 5.4 Перформансе и техничке спецификације Thread технологије.

Перформансе	Техничка спецификација
Домет сигнала	До 120 m (на отвореном простору)
Брзина преноса података	До 250 kbps
Фреквенција рада	2.4 GHz
Енергетска ефикасност	Ниска потрошња енергије (висока ефикасност због подршке за sleep mode)
Сигурносни протоколи	Шифровање података (енкрипција), аутентификација, ауторизација

Bluetooth, и његова напреднија верзија Bluetooth Low Energy (BLE), је широко распрострањена технологија која се углавном користи за персоналне уређаје кратког домета, до 10 m. Ради на фреквенцији од 2.4 GHz и користи UHFV радио таласе, карактерише га сигурност у преносу података, ниже брзине преноса и енергетска ефикасност. Конекција уређаја са овом технологијом је базирана на peer 2 peer и mesh топологији. Најчешће се имплементира код мањих уређаја, чији су захтеви краћи домет сигнала уз примену апликација које нису захтевне, са мањим брзинама преноса података (Табела 5.5).

Табела 5.5 Перформансе и техничке спецификације Bluetooth и BLE технологије.

Перформансе	Техничка спецификација
Домет сигнала	Bluetooth до 100 m (на отвореном простору) Bluetooth Low Energy (BLE) 200 m (на отвореном простору)
Брзина преноса података	Bluetooth до 3 Mbps Bluetooth Low Energy (BLE) до 1 Mbps
Фреквенција рада	2.4 GHz
Енергетска ефикасност	Bluetooth висока потрошња енергије (због већег протока података и сложенијих функционалности) Bluetooth Low Energy (BLE) ниска потрошња енергије (висока ефикасност због подршке за sleep mode)

Сигурносни протоколи	Шифровање података (енкрипција), аутентификација, ауторизација
-----------------------------	--

NFC (Near Field Communications) је бежична технологија и омогућава комуникацију између уређаја на веома краткој удаљености до 10 cm. Карактеристике је брзи и сигурни бесконтактни пренос података између два уређаја који се налазе у непосредној близини. Ова технологија је најчешће заступљена код мобилних уређаја или картица за плаћење где се захтевају кратке трансакције мањих података. Имплементација ове технологије обезбеђује сигурност везе и података, јер NFC поседује подршку за сигурносне протоколе и стандарде (Табела 5.6).

Табела 5.6 Перформансе и техничке спецификације NFC технологије.

Перформансе	Техничка спецификација
Домет сигнала	До 10 cm
Брзина преноса података	До 424 kbps
Фреквенција рада	13.56 MHz
Енергетска ефикасност	Ниска потрошња енергије (пасивни режим рада)
Сигурносни протоколи	Шифровање података (енкрипција), NFC сигурносни протоколи, аутентификација

ZigBee представља нову генерацију напредне бежичне комуникације, дефинисан је стандардом IEEE 802.15.4 који покрива средишњи домет и најчешће ради на фреквенцији од 2.4 GHz. За повезивање уређаја користи mesh топологију, и може да повеже велики број уређаја најчешће за М2М комуникацију. ZigBee се најчешће користи у апликацијама ИОТ-а које захтевају енергетску ефикасност тј. ниску потрошњу, велику брзину комуникације и сигурносне протоколе при преносу података. Технологија је заступљена код многих ИОТ уређаја, јер иза ове технологије стоји више од 300 водећих светских ИТ компанија (Табела 5.7).

Табела 5.7 Перформансе и техничке спецификације ZigBee технологије.

Перформансе	Техничка спецификација
Домет сигнала	Од 10 m до 100 m
Брзина преноса података	До 250 kbps
Фреквенција рада	2,4 GHz (најчешће, зависи од региона)
Енергетска ефикасност	Ниска потрошња енергије (оптимизирана за ИОТ уређаје)
Сигурносни протоколи	Шифровање података (AES енкрипција), аутентификација

NB-IoT (Narrowband Internet of Things) је стандард за ускопојасне бежичне мреже, намењен је за повезивање ИОТ уређаја и апликација са ниским захтевима. Нејзаступљенији је када су у питању велике брзине преноса података и велики домет преноса, као што су мерење потрошње енергије, мониторинг уређаја у пољопривреди или праћење стања машина. У питању је Low Power Wide Area Network (LPWAN) мрежна технологија, и дизајниран је за потребе уређаја и сервиса који користе мобилне мреже. Уређаји који

користе ову технологију немају константну комуникацију већ се користе за периодична слања података, већином времена су у sleep моду (Табела 5.8).

Табела 5.8 Перформансе и техничке спецификације NB-IoT технологије.

Перформансе	Техничка спецификација
Домет сигнала	До неколико километара
Брзина преноса података	До 250 kbps
Фреквенција рада	Опсег од 700 MHz, 800 MHz, 900 MHz, 1800 MHz, 2100 MHz (зависи од региона)
Енергетска ефикасност	Ниска потрошња енергије (оптимизирана за IoT уређаје, подршка за sleep мод)
Сигурносни протоколи	Шифровање података (енкрипција), аутентификација

Протоколи за комуникацију у мрежном слоју имају функцију да податке које добију са уређаја из перцептивног слоја пренесу на друге уређаје или сервисе. Док стандардни комуникациони протоколи имају опсежне захтеве у погледу ресурса и често нису најбоље решење за IoT окружење, IoT протоколи су дизајнирани да буду адекватни за специјализоване уређаје са ограниченим ресурсима који користе најчешће бежичну или жичану комуникацију за повезивање два или више IoT уређаја. Мрежни слој IoT-а је уско повезан са апликативним слојем, јер је одговоран за проток података до интерфејса крајњих уређаја, што значи да је одговоран за комуникацију у оба смера, као и за манипулацију, форматирање и презентацију података. То значи да је IoT мрежни слој, односно слој протокола задужен за комуникацију у свим правцима са једне стране и манипулацију односно форматирање и презентацију података са друге стране.

Избор технологије зависи од инсталиране опреме и подршке, услова примене одређене технологије, захтева за потрошњом енергије итд. У систему из дисертације, било је потребно повезати уређаје на специфичан начин тако да биолошке потребе и свакодневне активности пчела не буду угрожене.

Протоколи у принципу представљају скуп правила за комуникацију, а с обзиром на то да се информације шаљу на друге уређаје, интернет и cloud, они осигуравају да размењени подаци буду разумљиви у смислу хетерогености уређаја и података. Широки спектар IoT уређаја захтева дизајн различитих протокола и њихову оптимизацију за различите сценарије и намене. У контексту протокола за комуникацију у мрежном слоју најбитније је обезбедити сигурност и приватност података. У складу с тим, различити механизми могу бити имплементирани како би се заштитили уређаји и подаци од потенцијалних претњи. Ови механизми обухватају шифровање података, аутентификацију и ауторизацију како би се осигурали интегритет и поузданост података у IoT екосистему.

5.5.3. Слој средњег софтвера

Слој средњег софтвера и слој обраде података представљају две зависне компоненте неодвојиве од захтева напредног IoT система, служећи као стратешка веза између слоја физичких уређаја и апликационог слоја. Овој слој је заслужан за пружање апстракције

хардвера, што омогућава апликацијама интеракцију са хардвером без потребе за директним знањем о специфичностима самог хардвера. То значи да апликације могу да комуницирају са уређајима на унифицирани начин, независно од конкретних техничких детаља уређаја.

Middle software layer (слој средњег софтвера) и cloud processing layer (слој обраде података) чине две међусобно повезане компоненте које служе за ефикасно функционисање било ког савременог система Интернета ствари. Ови слојеви, познатији као "Edge computing", омогућавају ефикасну обраду, управљање и дистрибуцију великог броја података генерисаних у оквиру ИоТ система.

Обрада података у слоју Edge computing одвија се у три фазе:

1. **Прикупљање података.** У овој фази се идентификују типови података и структурирају се на начин који омогућава њихову даљу обраду. Ова фаза се одиграва на самој ивици система где се подаци прикупљају са различитих извора и преусмеравају вишем слоју.
2. **Апстракција података.** Ова фаза се односи на форматирање и апстракцију података. Подаци се конвертују у одговарајући формат који је погодан за даљу обраду и анализу. Ово укључује форматирање података тако да их је лако интерпретирати и обрадити.
3. **Анализа података.** У овој фази се изводи анализа података како би се открили обрасци и корисне информације. Ово може укључивати рано откривање аномалија или карактеристичних појава применом алгоритама машинског учења. Анализа се обавља на ивици система, што омогућава брзу реакцију и дејство без претераног оптерећења главног система.

ИоТ систем генерише велику количину података са слоја перцепције, где исти морају бити обрађени и прослеђени наредном процесу. У тим процесима може доћи до кашњења и отказивања перформанси ИоТ система, и као последица прекид рада или гушење система. Подаци се морају брзо анализирати и обрадити, конвертовати у формате који су захтевани за даље прослеђивање. Овај изазов решавају ова два посредничка слоја, спуштајући процесну обраду на саму ивицу система штедећи тако читав систем. Подаци се на овај начин одмах обрађују, где се у раној фази могу детектовати аномалије, грешке и одмах реаговати без оптерећења главног система, зато се и уводи управљање подацима на edge слоју.

Важан елемент у овом слоју је и подршка за ИоТ стандарде и протоколе као што су MQTT, CoAP, AMQP и други, који су специјализовани за ниско-латентну и енергетски ефикасну комуникацију у оквиру ИоТ окружења. Овај међуслој има веома важну улогу и у управљању услугама, конекцији са другим уређајима као и слање података за смештање у базу података (сервер, cloud), у реалном времену обезбеђујући притом QoS (quality of service) и доступност података.

5.5.4. Рачунарство у облаку - *Cloud computing*

Cloud computing или рачунарство у облаку заједно са Big data и вештачком интелигенцијом дају ИоТ свету нову димензију проширујући је у Интернет свега (Internet of everything), наравно на крилима хетерогености технологије крајњих уређаја. Кренућемо поново од ИоТ и генерисања велике количине података који могу бити смештени на локалним меморијама или серверима, али то носи са собом ризик доступности података, ресурса и сервиса.

Задаци овог слоја варирају, од складиштења великог обима података, презентације истих у реалном времену, машинског учења све до аналитике података. Cloud системи и сервиси могу да обезбеде неопходне ресурсе и инфраструктуру који подржавају интензивне задатке обраде података који су често неопходни у ИоТ апликацијама. Овај слој такође обезбеђује значајан ниво прилагодљивости и скалабилности, који омогућавају системима да се прилагоде променљивим захтевима за обрадом података.

Cloud computing у својој бити је online скуп сервиса који обезбеђује доступност истих без обзира на хетерогеност технологије који приступа том скупу. Од почетне идеје виртуелизације и оптимизоване понуде ресурса рачунарства до данас, рачунарство у облаку нуди широку палету ресурса, услуга и сервиса. Ограниченост ресурса локалног карактера условило је рапидан раст ових услуга, тако да данас концепт информатичке услуге засноване на облаку даје могућност смештања велике количине података које су увек доступне, наравно подразумева се интернет приступ. Поред базе података на серверима у cloud системима, ту су: хардверски ресурси, рачунарски ресурси, виртуелизација, апликативна решења, услуге заштите података, имплементацију сервиса вештачке интелигенције за анализу и обраду података, Big data анализу, серверске апликације са интерфејсом за презентацију података и аналитику, сервиси за контролу и управљање података. Web оријентисана технологија је данас нешто што се подразумева управо због једноставног приступа, услуге и сервиса, и то је оно што ИоТ чини комплетним јер захтева приступ у реалном времену. ИоТ функционалност се брзо шири са новим захтевима и парадигмама, што даје предност примене cloud технологија које подразумевају удруживање ресурса и сервиса, повезивање различитих технологија, аутоматизацију и примену edge технологије за предобраду података и њихово конвертовање, могућност коришћења web апликација за манипулацију над подацима.

Сервиси рачунарства у облаку могу се поделити на неколико категорија:

1. **Софтвер као услуга** (Software as a Service - SaaS). Ова категорија обухвата апликације као услугу за управљање које раде у облаку. Корисници им могу приступити преко web претраживача без потребе за инсталацијом или одржавањем локалних апликација.
2. **Платформа као услуга** (Platform as a Service - PaaS). Ова категорија обухвата скуп сервисних ресурса потребних за изградњу и испоруку апликација крајњим корисницима. Платформа као услуга омогућава корисницима да развијају, тестирају

и користе апликације без неопходности за инфраструктуром и конфигурацијом хардвера.

3. **Инфраструктура као услуга (Infrastructure as a Service - IaaS).** Ова категорија обухвата комплетну хардверску инфраструктуру неопходну за рад апликација. Инфраструктура као услуга подразумева пружање база за складиштење података, сервера, мрежне инфраструктуре, рутера, чворишта за обраду података и других хардверских компоненти.

Cloud computing представља савремени приступ услуга и технологију која се успоставља као стандард у IoT свету. Омогућава доступност и управљање подацима, анализу и обраду великих количина података, аутоматизацију и управљање сервисима. Осим што обезбеђује комплексне ресурсе за обраду података, cloud слој такође игра веома битну улогу у безбедности IoT-а. Примена безбедносних протокола и стандарда, као и поштовање правила о заштити података, у овом контексту су од есенцијалног значаја.

5.5.5. Велики подаци - Big data

IoT технологија и систем великих података (Big data) имају симбиотску везу, јер је једна од примарних функција IoT система да генерише велике количине података. Захтеви које Big data треба испунити као део IoT-а су: складиштење података великог обима, њихова обрада, управљање подацима и аналитички рад у реалном времену. Овакви захтеви превазилазе могућности аналитике стандардних софтверских алата за манипулацију над подацима. Додатни проблем је различитост улазних података са слоја перцепције, јер се користи широк спектар сензора и технологија дизајнираних за детекцију и прикупљање типова података. Управљање овако комплексним сетом података, различог формата, где подаци могу бити структурирани или неструктурирани подразумева примену специјализованих процеса и алата како би се у реалном времену одговорило на захтеве IoT система. Систем се дизајнира тако да се избегне гушење подацима и тако подаци учине недоступним, док је систему потребна анализа података ради филтрирања и конвертовања података за похрањивање у наредне слојеве.

Примена Big data аналитичког система у IoT-у доноси низ новина и подиже ниво обраде података. Ове новине укључују:

- **Дубинску анализу података.** Користећи напредне аналитичке методе, могуће је извући значајне закључке из великог скупа података, што може помоћи у доношењу информисаних одлука.
- **Откривање корелација.** Идентификација повезаности међу различитим подацима може открити трендове и обрасце који могу бити кључни за унапређење модела или разумевање узрочних фактора.
- **Откривање трендова.** Анализом великих података могу се идентификовати дугорочни трендови или промене које могу утицати на стратешко планирање.

- **Откривање скривених образаца.** Big data анализа може открити скривене obrasce и везе које се не могу лако препознати кроз стандардне аналитичке приступе и статистичке методе.
- **Извлачење нових информација.** Кроз процес дубинске анализе, могуће је открити нове, неочекиване информације које могу бити корисне за доношење одлука или разумевање одређених феномена.

Big data анализа, уз употребу алата за машинско учење и вештачку интелигенцију, омогућава ефикасну трансформацију огромне количине података у корисне информације. Модели машинског учења могу идентификовати трендове и obrasce, што омогућава прецизну предикцију, превентивно деловање, аутоматизацију процеса и информисано доношење одлука. Big data је једна од најзначајнијих компоненти која IoT екосистем чини заиста интелигентним.

5.5.6. Слој вештачке интелигенције (ВИ) (*Artificial intelligence AI*)

Конвергенција интернета ствари (IoT), великих података (Big data) и аналитике вођене вештачком интелигенцијом (ВИ) откључава потенцијале IoT технологије. Генерисање велике количине прикупљених података у систему интернета ствари, захтева примену машинског учења и вештачке интелигенције за приступ обраде и анализе истих. Подаци су као што је поменуто доста често у различитим форматима, са различитих извора, а пут података до презентационог или слоја вишег нивоа захтева одређену обраду и анализу података. Узимајући у обзир нове изазове IoT-а, и захтеве у реалном времену, може доћи до отказивања приступа из више разлога: лоших улазних података, недовољних информација, отказивање сензора јер читав систем зависи од овог слоја.

С обзиром на нове изазове IoT система и потребу за обрадом у реалном времену, вештачка интелигенција се примењује у више праваца и нивоа, где је први у edge слоју на самом улазу података у систем. Систем вештачке интелигенције има способност да научи из историјских података, открива obrasce и тенденције, а затим примени стечено знање у реалном времену. Она може да пружи предиктивне анализе и увиде, помажући у предвиђању будућих догађаја и предлагању оптималних акција. На тај начин осигурава стабилност система, подаци са обраде из ивичних слојева се у зависности од дизајна система даље конвертују или шаљу у базу или вишим слојевима за шта је задужен систем за одлучивање који прати obrasce и открива информације. Други ниво примене вештачке интелигенције је у вишим слојевима на сервисима у облаку о ком ће у наредним поглављима бити детаљнији опис, где се подаци користе за тренинг система вишег нивоа, који има задатак да помаже у одлучивању. На овом нивоу широки је спектар алата за примену вештачке интелигенције који даје неограничен сет могућности, од система за аутоматско одлучивање и реакцију, до система за комплексне операције, предикције, дубоку обраду података и генерисање знања.

У контексту ове дисертације, ВИ се користи за предиктивне моделе у понашању пчела. На основу прикупљених података и идентификованих образаца, креиран је модел

који може предвидети кретање пчела. Овај систем за предикцију, као и резултати које је постигао, биће детаљно описани у наредним поглављима.

5.5.7. Слој Апликације ИоТ

Апликацијски слој у контексту ИоТ екосистема представља карику која омогућава ефикасну интеракцију између корисника и ИоТ система. Примарни задатак овог слоја је да пренесе информације добијене из различитих делова ИоТ система кориснику на једноставан и разумљив начин. Овај слој се може поделити на два главна сегмента: сегмент анализе и процесирања података и сегмент презентације података. Слој апликације обједињује више слојева јер је везан за све слојеве од програмирања микроконтролера, сензора, конверзија података, аналитике и обраде, до презентације података на различитим врстама уређаја за приказ као што су апликације за крајње кориснике.

Процесирање података је комплексан процес који обухвата низ алгоритама и техника за анализу и интерпретацију података. Ова анализа се обавља уз помоћ технологија као што су машинско учење и вештачка интелигенција, које омогућавају ефикасну екстракцију корисних информација из великог броја сакупљених података. Из овог процеса се генеришу подаци и информације које су спремне за даљу употребу на вишим слојевима апликације, али и за презентацију крајњим корисницима.

Сегмент презентације података се бави приказом обрађених података на разумљив и приступачан начин. Користи различите технике визуализације, попут графикона, дијаграма и табела, како би приказао обрађене податке корисницима на начин који могу лако разумети. На овом слоју, подаци се форматирају и презентују кориснику путем различитих врста корисничких интерфејса, укључујући мобилне апликације и web платформе. Презентација података је значајна за доношење информисаних одлука у ИоТ систему. С обзиром да ИоТ систем функционише као један живи организам, где се подаци континуирано прикупљају, анализирају и ажурирају у реалном времену, апликацијски слој омогућава корисницима инстант увид у стање система.

Када се подаци пренесу на web оријентисане сервисе, они се даље могу користити за мониторинг, контролу и управљање различитим ИоТ уређајима. Све ово омогућава широку примену ИоТ технологије у различитим доменима, попут паметних кућа и градова, индустријских система, здравствене неге и многих других. Као што је већ речено, сваки ИоТ систем је дизајниран са специфичним циљевима у виду и сваки апликацијски слој у ИоТ систему има своје специфичности и функционалности, у складу са захтевима система. То значи да се сложеност и специфичност апликацијског слоја могу значајно разликовати у зависности од природе ИоТ система. Апликацијски слој представља мост између сложених технолошких процеса у ИоТ систему и крајњих корисника. Учествује у анализи и обради података, али и презентацији тих података корисницима на начин који могу лако разумети, омогућавајући им да доносе информисане одлуке. Кроз све ове функционалности, апликацијски слој омогућава ефикасну интеракцију између корисника и ИоТ система.

Синергија ова три слоја даје за резултат подизање функционалности ИоТ технологија, а то су:

- Хетерогеност уређаја за прикупљање података
- Скалабилне инфраструктурне капацитете на захтев корисника
- Повећана ефикасност свих сегмената у свакодневним задацима
- Брзина, реакција и широка дистрибуција апликација за приступ на сваком месту
- Дубока анализа података и преглед статуса повезаних IoT уређаја
- Ефикасност система на свим нивоима
- Управљање ризиком

Апликације могу варирати од једноставних мобилних апликација за праћење стања уређаја до сложених система за управљање и анализу података у реалном времену. Слој апликација такође игра веома важну улогу у безбедности IoT система. Једна од важних одлика овог слоја је способност да обезбеди аутентификацију и ауторизацију корисника, што је кључно за спречавање неауторизованог приступа и злоупотребе система. Ово је слој у коме се сусрећу корисник и IoT систем. Успешна апликација ће омогућити кориснику да максимално искористи предности и могућности које IoT пружа, без потребе да разуме сву сложеност која стоји иза тога.

5.5.8. Безбедност података у IoT системима

Безбедност података представља један од највећих изазова у савременом свету дигиталне технологије, посебно узимајући у обзир комплексност инфраструктуре Интернета ствари. Велики број повезаних уређаја у међусобној интеракцији, вишеслојна архитектура, интелигентна обрада података, све то компликује императив заштите података. Безбедносни слој у IoT екосистему представља скуп различитих мера и техника дизајнираних за обезбеђивање сигурности података и уређаја који чине овај систем. Ово подразумева кораке као што су енкрипција података, аутентификација уређаја и корисника, заштита од сајбер напада, контрола приступа и многе друге.

На сваком нивоу IoT система, од уређаја и сензора преко мреже и cloud-a, све до презентације на апликацијском слоју, потребно је имплементирати одговарајуће сигурносне мере. Најважнији аспекти безбедности које треба узети у обзир:

- **Безбедност опреме и уређаја**, то подразумева заштиту самих уређаја, сензора, микроконтролера. IoT уређаји су често најслабија тачка у ланцу безбедности, јер се могу физички манипулисати или хаковати. Заштита уређаја може укључивати енкрипцију, сигурне boot механизме, редовне безбедносне надоградње и друго.
- **Безбедност података**, обзиром на количину података које IoT уређаји генеришу, заштита тих података постаје императив. То може укључивати енкрипцију података у транзиту и на мировању, као и употребу сигурних протокола за пренос података.
- **Безбедност мреже**, IoT уређаји често комуницирају путем мреже, било да се ради о локалној мрежи или интернету. Заштита мреже може укључивати употребу

безбедносних протокола и шифровање података, firewall-a, IDS/IPS система или сигурних VPN веза.

- **Безбедност у cloud-у (Cloud Security).** Већина ИоТ система се ослања на cloud платформе за складиштење и обраду података. То значи да безбедност cloud система, подразумева примену заштите од свих врста напада, енкрипцију и сигурне механизме за аутентификацију.
- **Безбедност идентитета,** се односи на спречавање неовлашћених приступа, где је важно користити снажне механизме за аутентификацију и ауторизацију.

Слој безбедности у ИоТ-у системима је један од највећих изазова и веома важан, јер ИоТ уређаји често раде са осетљивим подацима и комуницирају у мрежи и са другим уређајима, што их чини потенцијалним метама.

5.5.9. Слој за контролу и одлучивање

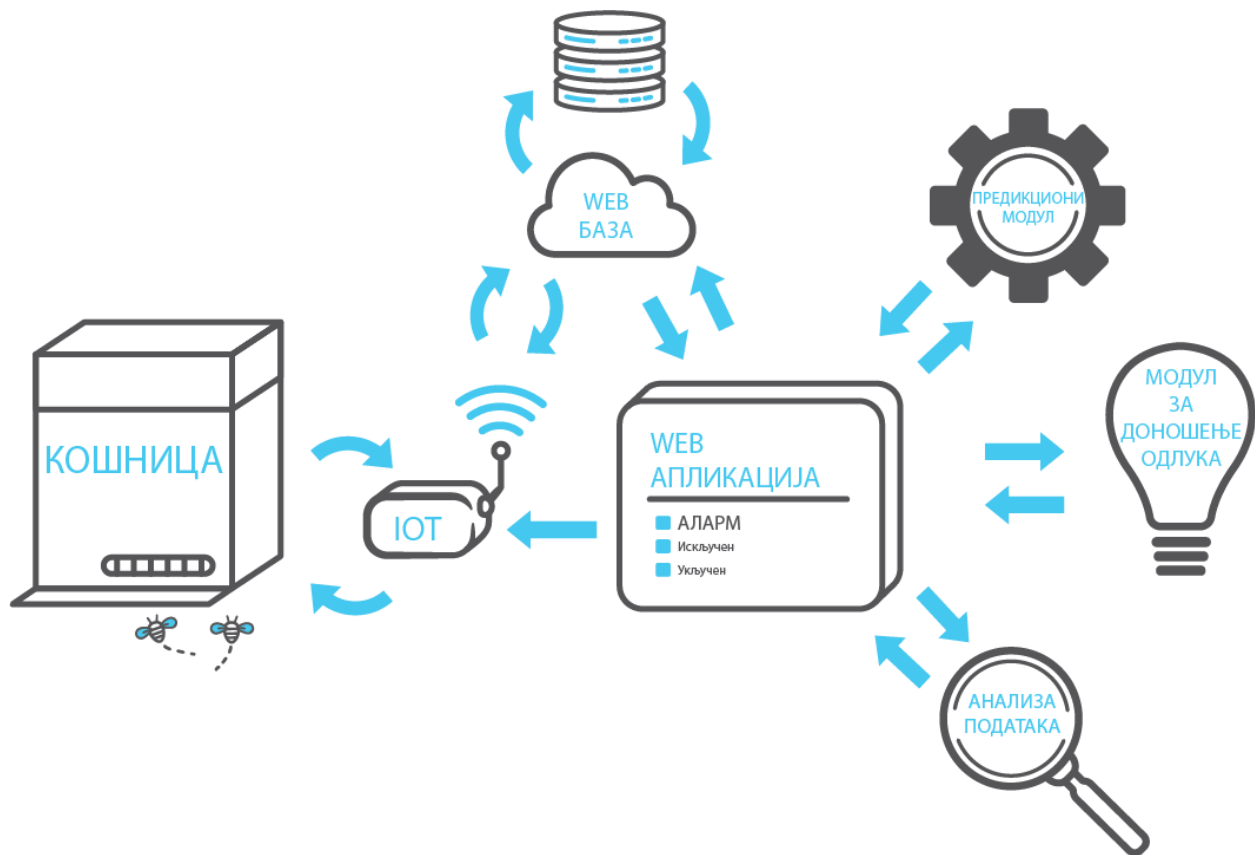
Следећи слој у архитектури ИоТ-а је систем за контролу и одлучивање који управља целокупним ИоТ системом. Овај слој обезбеђује координацију и регулисање интеракција између свих претходно наведених слојева архитектуре. Систем за контролу и одлучивање има за циљ да синхронизује рад различитих компоненти ИоТ система и управља њиховим функционалностима на основу добијених података из интелигентних алгоритама. Са овог слоја примењују се одговарајуће методе управљања, где овај слој обезбеђује ефикасну и координирану радну околину.

Улога система за контролу и одлучивање обухвата надгледање перформанси, управљање ресурсима, праћење стања уређаја, управљање енергијом, оптимизацију функционалности и пружање одговарајућих одлука. У овом слоју се анализирају и усмеравају подаци добијени из различитих слојева архитектуре, и на основу тога се доносе одговарајуће одлуке за управљање системом. Систем за контролу и одлучивање може да се користи и за имплементацију алгоритама самоадаптивности и оптимизације, где систем научи да се прилагоди околностима и да донесе најбоље одлуке у реалном времену. Ово доприноси ефикасности, оптималном искоришћењу ресурса, аутоматизацији и побољшаној перформанси ИоТ система.

Као интегрални део архитектуре ИоТ, систем за контролу и одлучивање обезбеђује стабилност, координира рад уређаја и модула, управља решавањем конфликта и пружа синхронизовано и ефикасно функционисање целокупног система. Један је од најважнијих елемената за управљање сложеним и динамичним окружењем ИоТ система, и обезбеђује његову оптималну радну перформансу и исправност.

6. СИСТЕМ ЗА МОНИТОРИНГ И ПРЕДИКЦИЈУ

Систем прецизног пчеларства је сложен склоп који се састоји од бројних хардверских и софтверских компоненти (Слика 6.1). Главна IoT јединица, смештена унутар кошнице, задужена је за прикупљање информација са низа сензора распоређених унутар и око кошнице, као и са бројача пчела постављеног на улазу кошнице. Ова централна јединица заснована је на микроконтролерским плочама Arduino Mega 2560 и Espressif ESP32. Сви прикупљени подаци, означени су одговарајућим временским печатом, преносе се у базу података у cloud-у путем мобилног модема. Како би се спречио евентуални губитак података, исти се додатно архивирају на локалној меморијској картици.



Слика 6.1 Преглед IoT Система

Развој овог софистицираног система захтевао је пажљив и детаљан приступ са више аспеката. Приликом дизајнирања концепта, приоритет је био постављен на безбедност пчела, тако да су дизајн система, имплементација сензора и опреме морали бити конструисани на начин који не би реметио дневне активности, кретање и нормално функционисање пчела. Познато је да пчеле могу прекрити воском било који објект унутар кошнице који перципирају као претњу. Систем је структуриран као комбинација више мањих подсистема који прикупљају различите типове података, и централног система који интегрише све прикупљене информације.

Још један аспект који је био од кључне важности је имплементација широког спектра сензора, са циљем прикупљања великог броја прецизних података укључујући више фактора утицаја. Овде је било неопходно водити рачуна о врсти и типу сензора, техничким спецификацијама дефинисаним од стране произвођача, условима под којима сензор треба да прикупља податке, као и о могућој интерференцији са осталим хардвером. Будући да сензорски модули долазе у различитим напонским варијантама, било је потребно пажљиво планирати напонску секцију, што је кључни фактор при одабиру одређених модула. Осцилације напона могу изазвати грешке у мерењима, стога је било неопходно обезбедити константан напонски извор.

При избору сензора, осим техничких карактеристика, узети су у обзир и широк спектар покривености података, као и претходна темељна анализа одређених параметара. Закључено је да сваки од изабраних фактора има утицај на живот пчела, али са различитим степеном интензитета, односно, неки фактори имају већи директни утицај, док се други испољавају само у комбинацији са осталим параметрима.

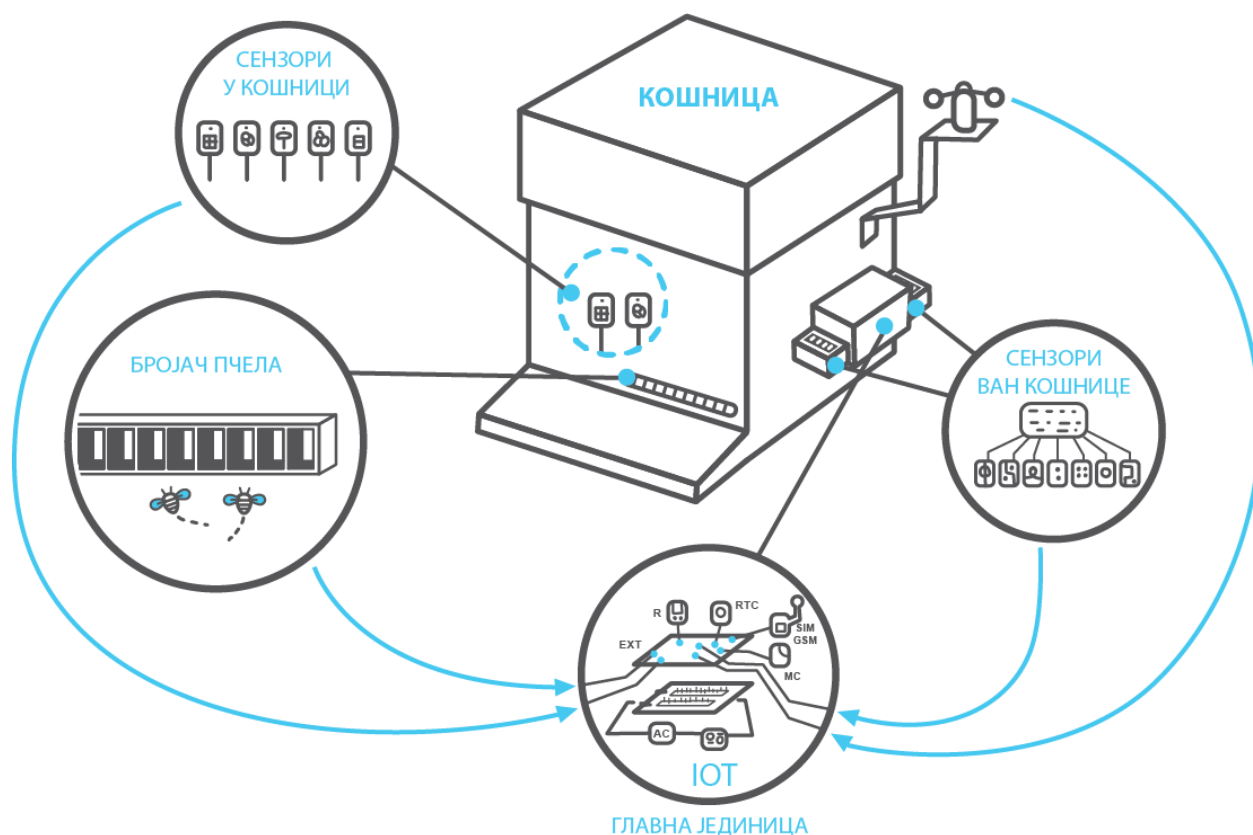
Осим сензора локације, пажљиво је разматрана и сама инсталација сензора, каблова и микроконтролера. Сензори могу бити осетљиви на спољашње промене, метеоролошке услове, директну изложеност сунцу и друге факторе који могу утицати на њихов рад и квалитет прикупљених података. Зато су сензори постављени и распоређени тако да могу несметано прикупљати податке, а да су истовремено заштићени од утицаја спољашњих фактора.

Приликом дизајна система посебно се водило рачуна о томе да се што мање користе бежични сензори, посебно унутар кошнице, како би се избегле потенцијалне интерференције које би могле утицати на пчелиње активности.

7. ДИЗАЈН И АРХИТЕКТУРА ЕЛЕКТРОНИКЕ СИСТЕМА

7.1. Архитектура система за мониторинг

Систем је конципиран као скуп хардверских и софтверских слојева. Хардверску архитектуру система чине различите одвојене целине (Слика 7.1), укључујући главну микропроцесорску јединицу (Main unit), базу станицу са сензорима (Base sensor unit), као и улазне капије за пчеле са бројачем (Bee counting array). Софтверска архитектура система је подељена на средњи слој задужен за прикупљање података, cloud систем за складиштење података, интегрисани API са примењеним системом вештачке интелигенције за анализу и предвиђање, као и апликативни слој за манипулацију подацима, смештен на серверу cloud система.



Слика 7.1 Увид у шематски приказ Система

Слој перцепције, инсталиран у кошници и околини, чини скуп сензора распоређених на кључним подручјима. Сензори који прате параметре стања у кошници инсталирани су да прате структуру оквира рамова (због потребе за вађење рамова) и ергономију кошница, док су неки постављени на дну исте кошнице. Да би се прикупили информације о стању у кошници, сензори су пратили следеће параметре: температуру ваздуха, релативну влажност ваздуха, квалитет ваздуха тј. детекцију различитих гасова, присуство дима, угљен диоксида, вибрације, ваздушни притисак, спектар звучних фреквенција, присуство и ниво буке.

Комуникација са главном микропроцесорском јединицом омогућена је кроз посебне отворе, што омогућава несметано вршење инспекције кошнице.

Други сет сензора смештен је у непосредном окружењу, у најужем делу екосистема кошнице. За смештај ових сензора израђен је посебан ормарић од тврде UV стабилизирајуће постојане пластике са отворима за вентилацију. Надограђена верзија садржи напредну конструкцију са додатним елементима и електроником, укључујући дисплеј за увид у податке и систем, без потребе за отварање ормарића (на слици је стари модел). Овај дизајн омогућио је сензорима да функционишу и прикупљају податке без сметњи од стране метеоролошких услова чије податке делом такође прикупља, док истовремено спречава директан контакт пчела са сензорима инсталираним на спољашњој страни. Неопходност оваквог приступа била је условљена избегавањем утицаја спољних фактора на прецизност сензора који се може десити на овај начин.

Ормарић се састоји из централног дела и два бочна додатна дела, са посебним отворима. У једном делу су смештени сензори са бочним и доњим отвором за несметано снимање података о температури ваздуха, релативној влажности, присуству воде и кише, спектру звучне фреквенције, присуству и нивоу буке, ваздушном притиску са спољашне стране. У другом делу кућишта је направљен провидни део заштићен UV неутралним стаклом где су смештени сензори за прикупљање података о: интензитету дневне светлости, UV зрачењу, ИС зрачењу, присутности амбијенталне видљивости. На спољној страни, изнад кошнице, постављен је стандардни анемометар са војним сертификатом издржљивости (због изложености спољашним временским условима) за детекцију интензитета и снаге ветра.

На Слици 7.2 приказан је први модел активне кошнице са које су прикупљани подаци коришћени у истраживању и дисертацији. На слици се види потпуно функционисање система као и кошнице, где пчеле несметано обављају дневне активности не долазећи у контакт са опремом система. Улаз у кошницу је такође дизајниран на начин да пчеле могу прво слетети па ући у кошницу. Циљ оваквог дизајна је био да примена електронских компоненти не омета пчелиње перцепторе, из тог разлога је примена бежичних технологија сведена на минимум.

Опис сваког сензора са спецификацијама је дат у следећем поглављу, у ком је дат и опис микроконтролера и комплетне електронике и модула који су коришћени.



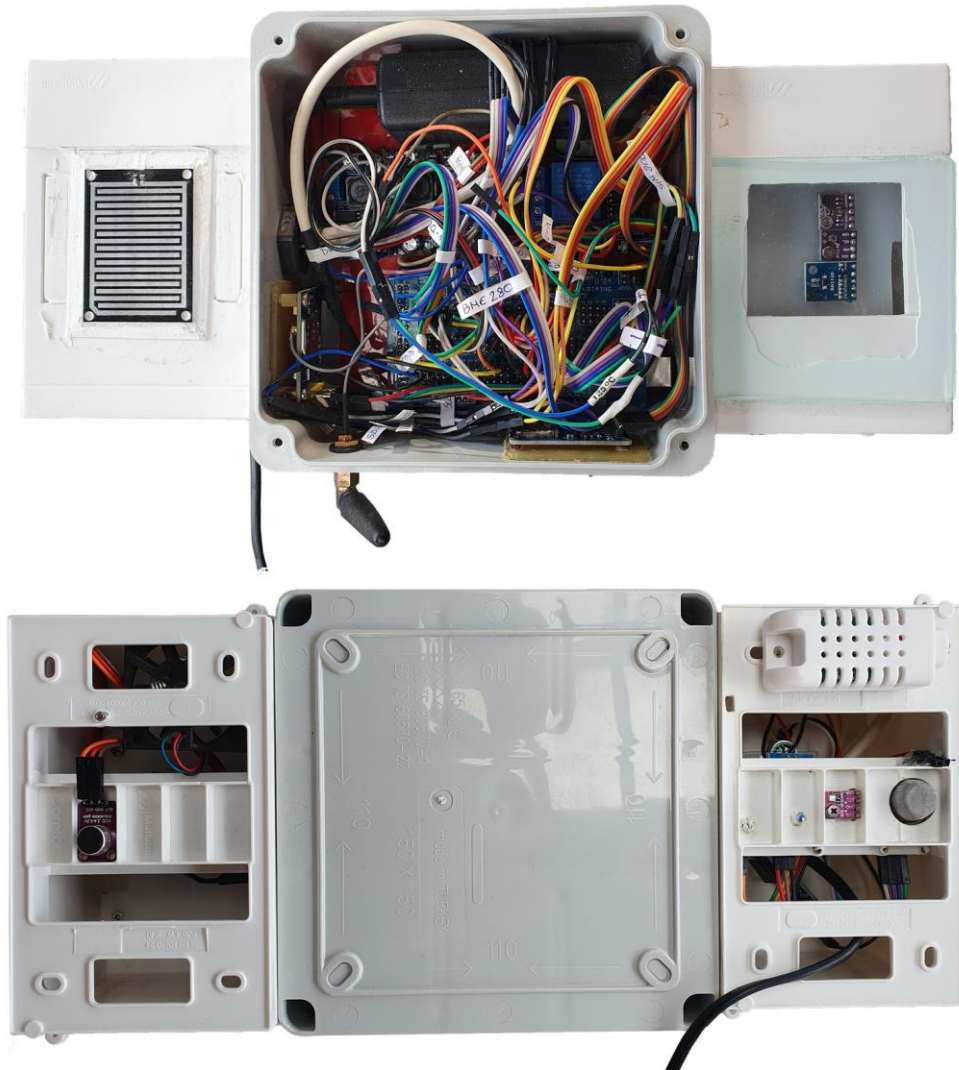
Слика 7.2 Експериментална поставка кошнице са сензорском електроником

7.2. Архитектура главне јединице

Главна јединица је задужена за контролу и управљање целокупним ИоТ системом (Слика 7.3). Дизајнирана је тако да се може напајати из различитих извора како би се избегли прекиди у раду. Напонска секција на контролној јединици регулише подизање и спуштање напона путем напонских додатака за регулацију, с обзиром да поједине електронске компоненте имају различите струјне захтеве. Главна јединица је повезана и са микропроцесором и са сензорском капијом за бројање пчела. Читав систем има мале енергетске захтеве, изузетак је само SIM модул који је мало захтевнији, али само приликом комуникације тј. слања података.

Област за улазак пчела у кошницу са капијицама за пролаз налази се на самом улазу у кошницу. Систем поседује од 12 до 24 пролаза за пчеле како би се избегло загушење на улазу и омогућило већем броју пчела несметано кретање. Пролази су довољно велики за кретање пчела, али недовољни за нежељене госте попут оса или стршљена. На горњем елементу сваког пролаза налази се сет рефлектујућих фото-отпорника за детекцију смера кретања, који се окида у одређеним смеровима. Пчеле се детектују прекидом фоторефлексије. Ако се активирају фото-ћелије у смеру бр. 1 па бр. 2, детектовано је да

пчела излази из кошнице, док у обрнутом смеру пчела улази у кошницу. Важно је напоменути да оба отпорника морају бити активирани и то у једном смеру како би се пролаз забележио. На тај начин се избегава дуплирање уноса уколико би се пчеле кретале из оба правца. Микропроцесор задужен за детекцију активности и бројање шаље прикупљене податке у временским серијама главној јединици. Подаци се пре слања означавају временским жигом, како би се имао увид у пролаз пчела по јединици времена.

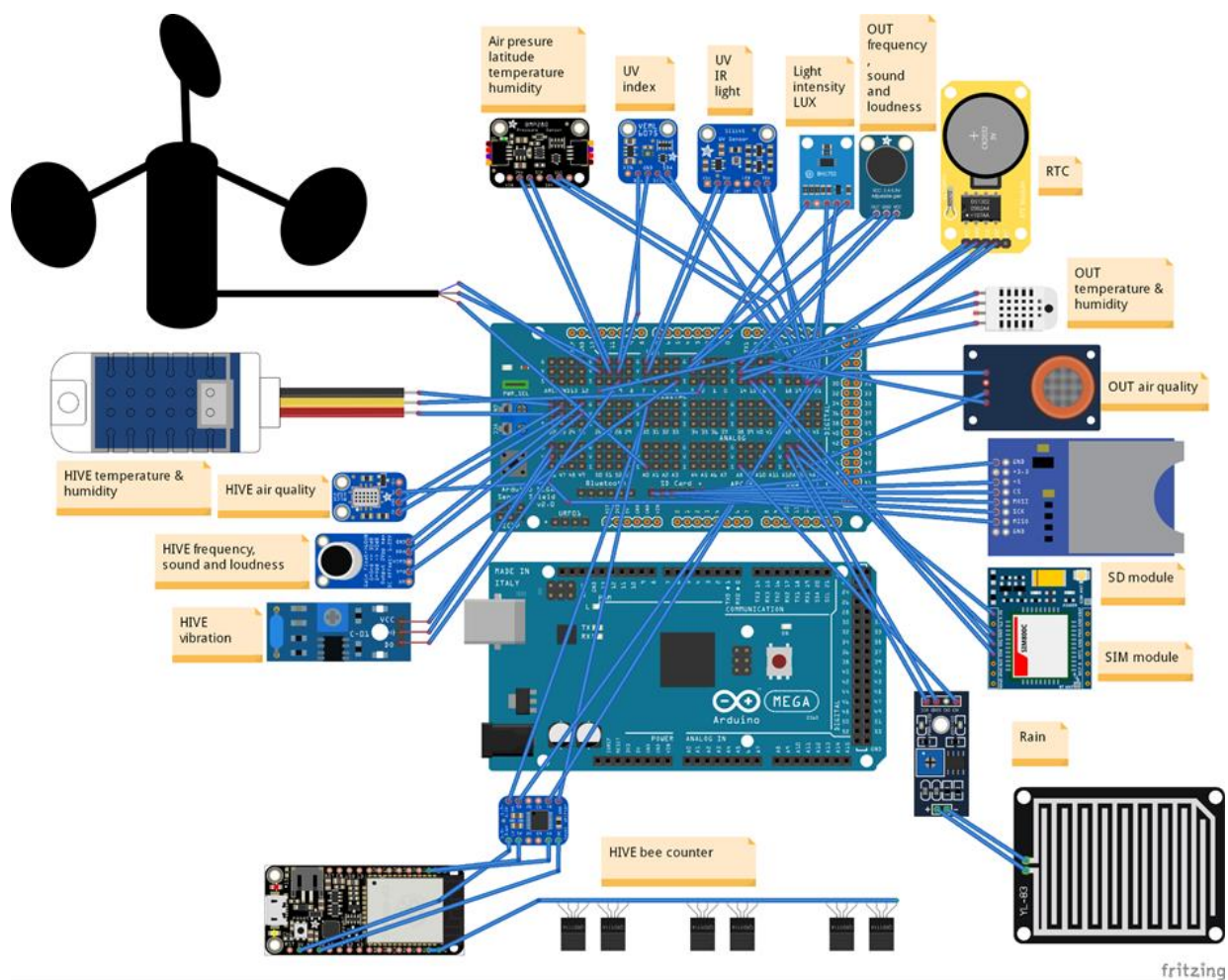


Слика 7.3 Главна јединица задужена за контролу и управљање ИоТ системом

Постоје бројни уређаји и апликације задужени за детектовање кретања пчела и бројање улазака и излазака из кошница, неки су приказани у радовима [60][61], а неки су комерцијални међу којима су познатији Arnia [62], Beecheck [63], Bee counter [64], Honey bee counter [65]. Методе детекције кретања се разликују у зависности од решења. Неки од њих су засновани на систему кондензатора за детекцију кретања пчела, неки су засновани на примени полупроводника, неки на оптичким сензорима, итд. Свако од ових решења има своје предности.

Као што је већ поменуто, за контролу читавог система задужена је главна микропроцесорска јединица коју чини микроконтролер Arduino Mega. Због великог броја сензора, модула и електронских компоненти за контролу и управљање системом процесорски модул је проширен екстензионом плочом. Ова допунска плоча подмирује читаве потребе система. Главна јединица је подељена у неколико сегмената, програмираних за различите фазе и задатке. За рад већине сензора потребан је индустријски стандард SPI и I2C магистрале, који се дефинише различитим адресама у регистру.

У програмском пакету Fritzing направљен је нацрт шеме главних јединица ИоТ базе станице, са конекцијом већег дела електронских компоненти које су коришћене у дизајнираном систему дисертације. На слици 7.4 приказана је централна компонента - Arduino Mega микроконтролер који повезује помоћни модул, ESP32 микропроцесор и периферне сензорске модуле.



Слика 7.4 Шема повезивања компоненти система (fritzing [66]).

Подаци са свих перцепцијских слојева и микроконтролера прикупљају се у главној јединици пре него што се пошаљу на сервер у cloud-у. Измерене параметре са сензора, микропроцесор уписује и у локалну меморију која је део главне јединице, како би се

спречила могућност губитка података у случају квара или прекида система. Примљени подаци се конвертују и прави се јединствени запис који се преко SIM модула шаље у web базу података на cloud-у. За везивање података са временским вредностима задужен је RTC модул, тако да сви подаци имају временску линију.

За обраду података и анализу, имплементиран је модул вештачке интелигенције на серверској страни, заједно са аналитичким алатима који прате кретање података у реалном времену. Модул вештачке интелигенције (<https://github.com/andrijevicnebojsa/BeeMAP> [67]), који се може наћи на датој адреси, задужен је за предиктивни аспект кретања параметара. Трениран је на моделу података који се такође могу наћи на адреси. Аналитички део надзире измерене параметре, и у случају кретања ка граничним вредностима критичних параметара, покреће аларм. Вредности параметара су дефинисане у апликативном делу презентационог слоја.

Презентациони слој апликације одговоран је за приказ података, манипулацију подацима, контролу и управљање системом. Апликација која је направљена за управљање системом носи назив BeeMAP (monitoring and prediction), оријентисана је ка web-у, тако да су сви сервиси и подаци на серверу у cloud-у. Управљање апликацијом се одвија преко контролног панела са администраторског корисничког налога. Систем поседује могућност подешавања аутоматизације, преко система за доношење одлука (DMS) који се налази на серверској страни и део је аналитике система. Ова компонента је способна да детектује одступања од номиналних параметара и на основу тога активира аларме. Детаљни опис модула са сервиса, као и web апликације BeeMAP биће представљен у наредним поглављима.

Две кључне компоненте су описане: хардверске компоненте, као и сложени систем за аналитику и препознавање образаца понашања како би се параметри повезали и откриле зависне и независне променљиве за одређивање интензитета утицаја на промене у кретањима пчела.

Важно је нагласити да прецизна мерења кретања пчела пружају основу за доношење закључака о стању кошнице, јер су везана за све активности, кретања и свакодневно функционисање пчела. Међутим, за добијање прецизних резултата кретања, подаци о кретању морају се повезати са зависним и условним параметрима.

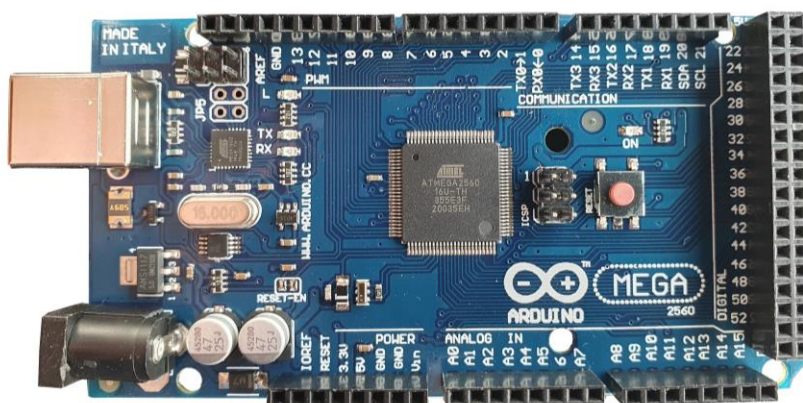
7.3. Архитектура електронских компоненти система

У овом делу биће детаљно описан комплетан хардвер ИоТ система, уз кратак опис спецификација које су важне за разумевање комплексности саме архитектуре. Важно је напоменути да су у међувремену извршене надоградње и додате нове функционалности које нису обрађене у овој дисертацији, али ће бити поменуте. Детаљније информације о овим надоградњама биће представљене у завршном одељку, пошто се проширења и надоградње тренутно тестирају, а резултати ће бити анализирани и представљени у будућим радовима који ће бити објављени.

Када је реч о коришћеним подацима за техничке спецификације, подаци су преузети са званичних сајтова произвођача или комерцијалних сајтова.

7.3.1. Микроконтролер Arduino Mega 2560

Главни и централни део IoT система у коме се интегришу подаци чини **Arduino Mega 2560** (Слика 7.5, детаљна спецификација *Прилог 7.1*) микроконтролер који је изабран због економичности потрошње енергије, једноставног повезивања и уградње додатних модула, могућности хардверског проширења плоче и компатибилности са другим произвођачима микропроцесорских система [68]. Arduino Mega се показао као поуздан избор, због довољног меморијског простора који је неопходан за писање самог кода у IDE радном окружењу.



Слика 7.5 Arduino Mega 2560

Arduino плоче су микроконтролери отвореног кода, чију основу чини Atmega 2560 чипсет. Њихова једноставна и ефикасна имплементација значајно је утицала на еволуцију индустрије аутоматизације и примене Интернета ствари. Имају сопствено развојно IDE окружење и компатибилни су са скоро свим програмским језицима, што им омогућава широки спектар примене. Када су у питању техничке спецификације које треба истаћи, ATmega2560 микроконтролер поседује архитектуру AVR (Advanced Virtual RISC), која обезбеђује ефикасно извршавање инструкција, што доприноси оптималним перформансама система. Радна фреквенција је 16 MHz, која омогућава брз и ефикасан пренос података са сензорских модула. Arduino Mega је опремљен са 54 дигиталних и аналогних улазно/излазних пинова. Од њих се 15 могу користити као PWM (Pulse Width Modulation) излази, а постоји 16 аналогних улаза, 4 UART (Universal Asynchronous Receiver/Transmitter) хардверска серијска порта, кристални осцилатор на 16 MHz, USB конекција, порт за напајање, ICSP (In-Circuit Serial Programming) заглавље и дугме за ресетовање. Поред тога, има и I2C (Inter-Integrated Circuit) пинове (SCL и SDA), који омогућавају повезивање са великим бројем дигиталних сензора.

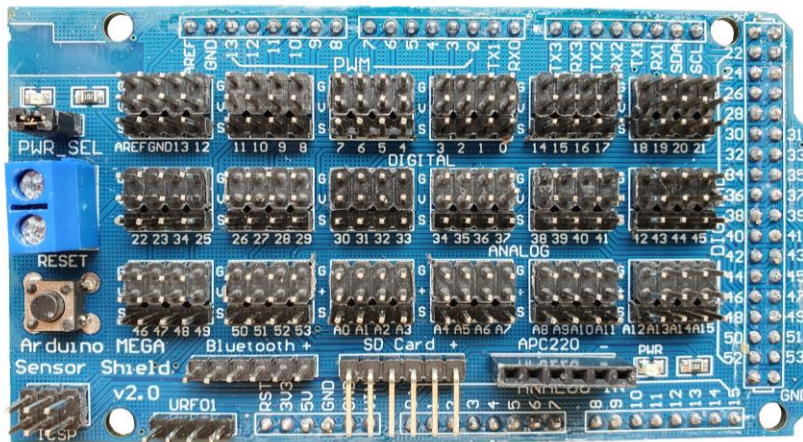
Сваки од ових пинова на Arduino Mega може се користити као улаз или излаз, користећи функције `pinMode()`, `digitalWrite()` и `digitalRead()`. Пинови користе напон од 5 V, а сваки пин може да напаја 20 mA. Поседује унутрашњи pull-up отпорник (подразумевано

о немогућен) од 20-50 k Ω . Максимална вредност струје је 40 mA, која се не сме прекорачити како се микроконтролер не би трајно оштетио.

Arduino Mega 2560 има 256 KB флеш меморије за складиштење кода, 8 KB SRAM меморије за привремено складиштење и обраду података и 4 KB EEPROM меморије која служи за дуготрајно чување података.

7.3.2. Mega Sensor Shield

Један од кључних фактора који је утицао на избор микроконтролера је компатибилност са екстензијама - шилдовима (Shields). Шилдови омогућавају једноставно повезивање и комбиновање сензорских модула са Arduino платформом, олакшавајући проширење функционалности система.



Слика 7.6 Плоча за проширење за Arduino Mega микроконтролер (Mega Sensor Shield V2.0 sensor expansion board)

Mega Sensor Shield V2.0 је наменска плоча за Arduino Mega микроконтролер која је коришћена као базна станица за проширења, и конектовања већег броја сензора и додатних модула на главни IoT микроконтролер (Слика 7.6 детаљна спецификација *Прилог 7.2*). Физички, Mega Sensor Shield се поставља директно на врх Arduino Mega 2560 плоче, повезујући се са свим пиновима микроконтролера и пружајући једноставан начин за додавање и уклањање различитих компоненти. Овај шилд проширује све дигиталне I/O пине, аналогне улазне пине и пине за напајање Arduino Mega 2560, чиме се омогућава широк спектар могућности за повезивање и коришћење различитих компоненти.

Плоча поред великог броја проширених аналогних и дигиталних пинова поседује директна повезивања за Bluetooth модул, SD меморијску картицу и APC220 бежични RF модул. Може се повезати и са LCD екраном за приказ података са спољашње стране уређаја (део који је додат на новом моделу за приказ основних података на спољној страни кошнице), серво мотором повезаним преко релеја као и пумпом за аутоматску хранилицу за пчеле (део проширења након тестираног модела система који није приказан на сликама, више о томе је описано у завршном делу).

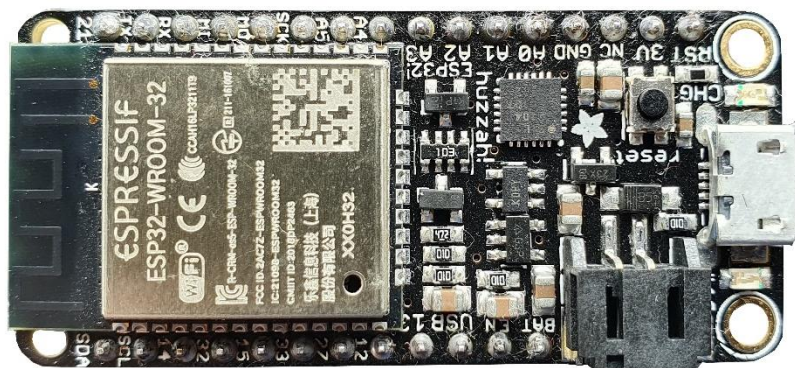
Архитектура пинова је дизајнирана кроз електронске блокове за конектовање, на које се лако повезују Arduino компатибилне плоче са различитим дигиталним, аналогним и I2C/UART интерфејсима. Сваки терминални модул има порт са VCC, GND и излазом, који има одговарајући порт на сензорској плочи, који се повезују кабловима преко пинова што олакшава и убрзава повезивање.

7.3.3. Микроконтролер ESP32

За прикупљање података са улаза у кошницу коришћен је уређај са пролазним капијицама којим се управља преко ESP32 микроконтролера, који је описан у наставку.

ESP32 је базиран на Tensilica Xtensa LX6 микропроцесору који ради на фреквенцији до 240 MHz и може се наћи у варијантама са једним или два језгра [69] (Слика 7.7, детаљна спецификација *Прилог 7.3*). Овај микроконтролер је развио Espressif Systems и карактерише га изузетна флексибилност и разноврсност функција. Микроконтролер долази са 520 KB RAM-а, који се може проширити на 4 MB помоћу спољашње меморије (SPI-Flash). Осим тога, ESP32 има уграђен Bluetooth 4.2 и Wi-Fi 802.11 b/g/n модул, што омогућава бежично повезивање и комуникацију са другим уређајима. Произведен је на бази TSMC 40nm технологије са ниском потрошњом. Дизајниран је да постигне најбоље RF (радио фреквенцијске) перформансе са највећом ефикасношћу потрошње енергије, показујући робусност и поузданост у широком спектру примене различитих апликација.

Једна од кључних специфичности ESP32 је његова способност да ради као Wi-Fi приступна тачка (Access Point) или као станица (Station), као и да подржава истовремено повезивање са више уређаја. Поред тога, микроконтролер подржава Bluetooth Low Energy (BLE) технологију. ESP32 има аналогно-дигитални (ADC) и дигитално-аналогни (DAC) конвертор, што омогућава прихватање аналогних сигнала и генерисање аналогних излаза. Микроконтролер подржава и touch-sensitive пинове, што омогућава развој уређаја са капацитивним додирним екранима или другим тач интерфејсима.



Слика 7.7 ESP32 микроконтролер

Дизајниран је тако да се може користити за мобилну, преносиву електронику и апликације ИоТ-а. Садржи све најсавременије карактеристике чипова мале потрошње,

вишеструке режиме напајања и динамичко скалирање потрошње енергије. ESP32 код захтева минималну потрошњу енергије, јер има могућност да се „буди“ повремено само када се открије одређено стање. Циклус ниског оптерећења се користи за минимизирање количине енергије коју чип троши. Напонски излаз појачивача снаге је такође подесив, чиме се доприноси оптималном компромису између домета комуникације, брзине преноса података и потрошње енергије. ESP32 је одлично интегрисано решење за све IoT апликације са захтевима за Wi-Fi и Bluetooth, са око 20 спољних компоненти.

7.3.4. RTC DS3231 Real time clock модул

За бележење времена записа коришћен је сензор RTC DS3231 чије су спецификације и опис дате у наставку. RTC DS3231 Real time clock модул (Слика 7.8, детаљна спецификација *Прилог 7.4*) је изузетно прецизан сат који се користи за праћење реалног времена и датума у електронским системима. Такође, RTC DS3231 има уграђен алармни систем који може генерисати прекиде на микроконтролеру или другом електронском уређају. RTC DS3231 користи интегрисани осцилатор температурне компензације (TCXO) како би пружио високу прецизност и стабилност времена.

Real time clock чип DS3231 је један модул са две функције, има батерију и EEPROM AT24C32 на истој плочици, што га чини идеалним за микроконтролере, Arduino и сличне развојне системе. RTC даје информације о секундама, минутама, сатима, дану, датуму, месецу и години, и врши аутоматске исправке за преступну годину. Сат ради у 24-часовном или AM/PM 12-часовном формату, нуди два подесива будилника и календар који се такође може подесити. Адреса и подаци се преносе серијски преко I2C протокола, који служи и за комуникацију са микроконтролером или другим уређајем. Поред тога, RTC DS3231 има и додатне пинове за напајање (VCC и GND) и за контролу алармног система (INT/SQW).

Једна од специфичности RTC DS3231 је његова способност чувања података о времену и датуму чак и када је напајање искључено. Модул има интегрисану батерију (CR2032) која обезбеђује резервно напајање и омогућава очување података и времена. Ово је корисно у случају нестанка струје или када је систем искључен.



Слика 7.8 RTC 3231 Real time clock модул

7.3.5. SIM800L GPRS GSM модул

За слање података на cloud систем коришћен је мобилни модем SIM800L GPRS GSM (Слика 7.9, детаљна спецификација *Прилог 7.5*). SIM800L је компактни GSM модул који омогућава бежичну комуникацију путем GPRS и GSM мреже. Овај модул је популаран због своје једноставне употребе, пузданости и широког спектра подржаних функција. SIM800L GPRS GSM модул има уграђен SIM слот који омогућава повезивање са GSM мрежом. Модул подржава 2G мрежу и ради на фреквенцијама 850/900/1800/1900 MHz.



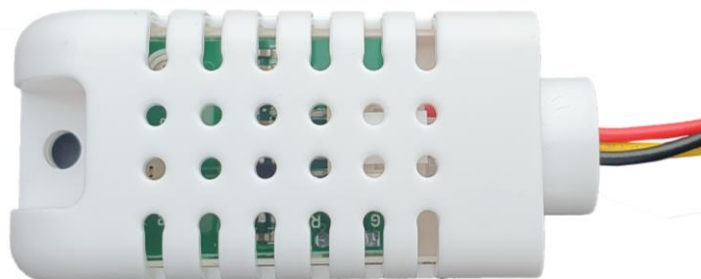
Слика 7.9 SIM800L GPRS GSM модул

Повезивање са SIM800L модулом се врши путем UART интерфејса, где се подаци шаљу и примају преко серијске комуникације. Такође, модул подржава AT команде за једноставно управљање и конфигурацију. SIM800L модул има уграђене функције за бежичну комуникацију као што су SMS слање и примање, гласовни позиви и GPRS подаци. Модул подржава TCP/UDP протоколе за слање података преко GPRS везе. Поред тога, SIM800L има и аналого-дигитални (ADC) конвертор и дигитално-аналогни (DAC) конвертор за прихватање аналогних сигнала. Једна од специфичности SIM800L модула је његова подршка за GNSS (Global Navigation Satellite System) као што је GPS. Ово омогућава модулу да добије информације о локацији, брзини и времену, што је корисно у апликацијама које захтевају праћење возила или локације уређаја.

Сензори који су коришћени за прикупљање података биће описани у наредном делу.

7.3.6. DHT-22 (AM2302) сензор температуре и влажности ваздуха

За прикупљање података о температури и влажности ваздуха коришћен је сензор DHT22 (Слика 7.10, детаљна спецификација *Прилог 7.6*). Иако је тестирано више врста (DHT, SHT, VME, LM, TMP...) и типова сензора различитих спецификација и произвођача, као најпозданији сензор показао се DHT22. У питању је била стабилност у појављивањима грешака при мерењу где су други сензори давали сличне или податке са мањом тачношћу и већим коефицијентом грешака. Такође, цена је била један од услова узимања у разматрање.



Слика 7.10 Сензор DHT22 за мерење температуре и влажности ваздуха

DHT-22 такође познат и као AM2302 је дигитални сензор температуре и влажности ваздуха, који се истиче својом прецизношћу, пузданошћу и једноставним коришћењем. Састоји се од сензора влажности ваздуха, термистора за мерење температуре и електронике за комуникацију са микроконтролером. Његов претходник је популарни DHT11. Фабрички је калибрисан, па не захтева додатна подешавања и може се одмах користити. Најчешће се користи са микроконтролерским системима Arduino, PIC, ARM, итд. Долази у пластичном кућишту израђеном од чврсте ABS пластике, одличних механичких карактеристика, отпорним на топлоту, те је погодан и за спољну монтажу, а при врху се налази и отвор за монтажу.

За комуникацију са микроконтролером, сензор користи једноставан дигитални протокол. Обично се повезује са једним дигиталним пином и користи се библиотека за читање података са сензора. DHT22 сензор има тачност од ± 0.5 °C за мерење температуре и $\pm 2\%$ за мерење влажности. Он такође аутоматски компензује влажност у зависности од температуре ваздуха. Сензор има широк опсег мерења температуре од -40 °C до $+80$ °C, што га чини прикладним за примене у различитим температурним условима. Такође, има опсег мерења влажности од 0% до 100%. AM2302 модул има уграђен филтерски кондензатор и pull-up отпорник, што му даје већи опсег мерења и бољу тачност.

7.3.7. SW-420 сензор вибрација

За детекцију вибрација у кошници коришћен је сензор вибрације SW-420 (Слика 7.11, детаљна спецификација *Прилог 7.7*).

Модул SW-420 садржи високоосетљив сензор вибрација, подесиви регулатор и LM393 компаратор чип. Сензор има три основна пина за повезивање. Повезивање сензора је једноставно - GND пин се повезује на заједничко уземљење, VCC пин се повезује на напајање од 3.3 V до 5 V, а OUT пин се повезује на дигитални пин микроконтролера или другог електронског уређаја ради читавања излазног сигнала сензора.



Слика 7.11. Сензор вибрација SW420

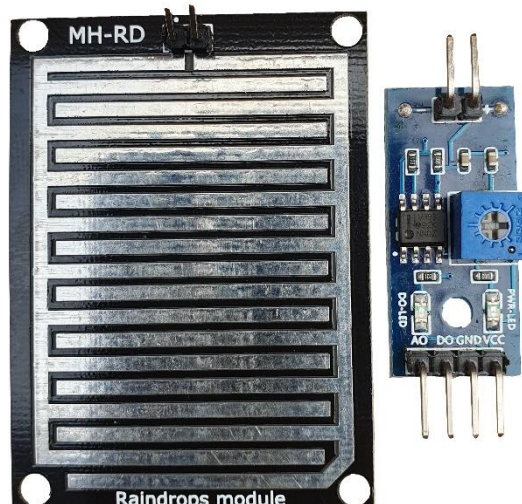
Специфичност овог сензора је његова висока осетљивост на вибрације. Сензор реагује на промене у вибрацијама окружења и генерише дигитални сигнал на излазу када је вибрација препозната. Осетљивост сензора може се подешавати помоћу потенциометра који се налази на самом сензору. На сензору су такође постављена два индикатора - један за приказ статуса напајања, а други за приказ статуса очитавања сензора (тригер). Модул сензора вибрација обезбеђује излазе на логичком нивоу на основу вибрација и спољних импулса који делују на њега. То значи да када нема вибрација, његов излаз остаје у логички ниском (Logic Low) стању. Али током детекције вибрација, стање излаза се мења на ниво логички високог (Logic High) стања.

Сензор ради на принципу вибрационог прекидача, који реагује на амплитуде вибрација којима је изложен. Ова реакција може се манифестовати кроз електрично затварање или отварање контакта. Ови контакти могу бити електромеханички релеји или полупроводнички уређаји.

7.3.8. MHRD rain сензор кише

У наставку ће бити описан сензор **MHRD** који је коришћен за детекцију падавина (Слика 7.12, детаљна спецификација Прилог 7.8). Састоји се из два модула, плоче са слојевима никла која детектује кишу и контролног модула, који упоређује аналогну вредност и претвара је у дигиталну вредност. Осетљива сензорска плочица детектује било какву воду присутну на плочи док контролна плоча чита ове сигнале. MHRD Rain sensor ради на принципу детекције електричне проводљивости воде.

Сензор присуства воде је у основи плоча на којој је никл обложен у облику линија, и ради на принципу промене електричне отпорности. Када на плочици нема воде или капи кише, отпорност је висока тако да се добија висока вредност напона према ($U=R \cdot I$). Када је присутна вода или киша, она смањује отпорност јер је вода проводник и присуство воде повезује водове никла паралелно, тако да се смањује отпорност што доводи до смањења напона.



Слика 7.12. Сензор присуства кише и воде MHRD

Овај сензор за кишу користи компаратор широког напона LM393 да упореди референтни напон (који се поставља потенциометром) са улазним напонем (који варира у зависности од количине воде на плочици). Ако улазни напон премашује референтни напон, компаратор даје високи излазни сигнал. С друге стране, ако је референтни напон већи од улазног напона, компаратор даје ниски излазни сигнал. Модул има и LED сигнале за напајање и LED за статус. LED за напајање ће се упалити када се модул напаја док статусна LED светли и показује статус дигиталног излазног пина.

7.3.9. MQ135 сензор детекције гасова и квалитета ваздуха

Када је у питању мерење квалитета ваздуха или детекција концентрације гасова у ваздуху, гасни сензори серије MQ су најповољнији и најчешће коришћени. У наставку ће бити описан један од сензора MQ135 (Слика 7.13, детаљна спецификација Прилог 7.9), помоћу кога се детектују гасови који се најчешће појављују и могу израчунати квалитет ваздуха представљен у ppm вредностима. За надоградњу система у смислу детекције више различитих типова агенаса, поред овог сензора коришћени су и много скупљи и квалитетнији сензори MICS6814 и MICS5524, са већим спектром мерења, који ће у овом делу бити само поменути.

MQ135 сензор ради на принципу електрохемијске детекције гасова. Узорак ваздуха улази у сензор, а гасови реагују са електрохемијским материјалом унутар сензора, што доводи до промене електричне проводљивости. Сензор детектује промене проводљивости и генерише аналогни излазни сигнал који се може очитати и обрадити помоћу микроконтролера. Има интегрисан компаратор, аналогни излаз се користи ради мерења концентрације гаса, а дигитални излаз се користи за детекцију присуства гаса преко дигиталног улаза. Поседује способност детекције различитих гасова, укључујући амонијак (NH_3), NO_x , алкохол, бензен, дим и CO_2 . Сензорски модул MQ135 се може користити за детекцију само једне врсте одређеног гаса, али може да мери и више врста гасова и тако одређује ppm вредности. Има уграђен потенциометар за fino подешавање осетљивости.



Слика 7.13 Сензор MQ135 за детектовање присутности гасова и мерења ppm вредности

Модул користи напајање од 5 V тако да се може користити са већином уобичајених микроконтролера. Поседује LED индикацију за напајање и детекцију гаса на модулу. Сходно томе, LED лампице ће светлети у присуству гаса и рада самог модула. Принцип рада је следећи: сваки пут када сензор дође у контакт са гасом у одређеној концентрацији, дигитални пин ће бити висок (5 V), иначе ће остати низак (0 V). Сензор примењује SnO_2 који има већи отпор у чистом ваздуху као материјал који детектује гас. Када дође до повећања загађујућих гасова, електрична отпорност гасног сензора се смањује.

7.3.10. BH1750 сензор дневне светлости

За мерење интензитета дневне светлости коришћен је сензор BH1750 (Слика 7.14, детаљна спецификација *Прилог 7.10*), чије су измерене вредности за потребе дисертације комбиноване са другим сензорима како би се генерисале нове вредности за проналажење везе са зависним факторима кретања пчела.

BH1750 сензор је дигитални сензор осветљености и ради на принципу мерења светлости помоћу фотоелектричног ефекта. Сензор има интегрисану фотодиоду која претвара светлосну енергију у електрични сигнал. Сензор може читавати широк спектар светлости, укључујући видљиву и инфрацрвену светлост. Излазни сигнал је дигитални, а сензор подржава I2C интерфејс за комуникацију са микроконтролером или другим уређајем. Сензор прецизно мери интензитет светлости у распону од 1 до 65535 lx.

Користи I2C комуникациони протокол што га чини веома лаким за коришћење са свим типовима микроконтролера. SCL и SDA пинови су намењени за I2C конекцију. За мерење вредности интензитета светлости није потребна калкулација јер сензор директно даје вредности, мери интензитет према количини светлости која пада на њега. Ради на напонском опсегу од 3.3 V до 5 V и троши веома малу струју од 0,12 mA. Резултати мерења сензора не зависе од извора светлости који се користи и утицај инфрацрвеног зрачења је занемарљив, са веома малим процентом грешака у мерењу.



Слика 7.14 BH1750 сензор интензитета дневне светлости

Хардвер који чини BH1750 је веома једноставан, главна компонента је BH1750FVI IC. Вредности осветљења се добијају преко I2C магистрале, док компоненте ADC конвертују аналогну осветљеност у дигиталну вредност. Користи фотодиоду која даје реакцију еквивалентну људском оку.

7.3.11. VEML6075/6070 UV сензор

За мерење утицаја UV индекса коришћен је сензор високе прецизности VEML6075/6070 који је показао одличне резултате у мерењу (Слика 7.15, детаљна спецификација *Прилог 7.11*). Сензори који ће бити представљени у наставку такође имају могућност мерења утицаја UV зрачења па смо податке искористили за упоредно тестирање прецизности. VEML6075 је напредни сензор ултраљубичастог UV светла са I2C интерфејсом за комуникацију, и истиче се високом прецизношћу. Овај сензор је конструисан да детектује интензитет ултраљубичастог светла и претвара га у дигиталне податке.



Слика 7.15 VEML6075 UV сензор

Главне компоненте VEML6075 UV сензора (интегрисане у један чип) су фотодиода, појачивач, аналогна и дигитална кола за конвертовање сигнала. Захваљујући Filtron™ UV технологији, VEML6075 пружа врхунску спектралну осетљивост за детекцију UV зрака.

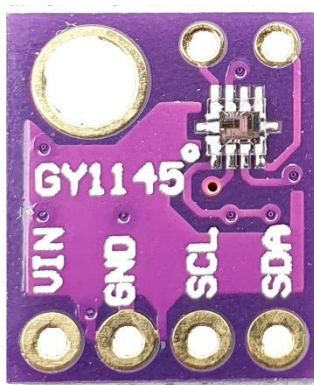
Има изванредну температурну компензацију и лако се подешава уз помоћ спољашњег отпорника.

VEML6075 сензор ради на принципу мерења UV зрачења помоћу фотодиода. Сензор има две фотодиоде - једну за UV-A зрачење и другу за UV-B зрачење. Фотодиода реагује на UV зрачење и генерише струјни сигнал који је пропорционалан интензитету зрачења. Сензор има интегрисани AD конвертор који претвара аналогни сигнал у дигитални формат. Један од важних аспеката овог сензора је могућност подешавања брзине освежавања без потребе за спољним RC нископропусним филтером. Такође, пружа софтверски мод за искључивање, који може смањити потрошњу струје на мање од 1 μ A. Радни напон VEML6075 се креће од 3.3 V до 5 V.

7.3.12. SI1145 сензор светлости: дневне, инфрацрвене IC, ултраљубичасте UV

Велика предност модерних сензора је да често долазе са интегрисаним решењима за вишеструка мерења, што даје шири спектар примене. Сензор SI1145 (Слика 7.16, детаљна спецификација *Прилог 7.12*) поседује уграђене компоненте за мерење инфрацрвеног зрачења, дневне светлости, UV зрачења па смо те вредности користили као упоредне или контролне.

SI1145 је дигитални сензор који комбинује мерење UV индекса, видљиве светлости и инфрацрвеног зрачења. Ради на принципу мерења светлости помоћу фотоелектричних ћелија, има три фотодиоде - једну за UV зрачење, другу за дневну светлост и трећу за инфрацрвено зрачење. Фотодиода реагује на светлост и генерише струјни сигнал који је пропорционалан интензитету светлости. Сензор има интегрисани аналогно-дигитални конвертор који претвара аналогни сигнал у дигитални формат. Сензор може читавати светлост у опсезима од 400 nm до 1000 nm за дневну светлост, и од 200 nm до 370 nm за UV зрачење. Повезивање се врши преко I2C дигиталног интерфејса, а има могућност програмабилног излаза за прекид догађаја. Карактерише га ниска потрошња енергије.



Слика 7.16 SI1145 сензор дневне, инфрацрвене и ултраљубичасте светлости

Нуди одличне перформансе са широким динамичким опсегом и различитим изворима светлости укључујући директну сунчеву светлост. Реакција фотодиода и повезаних кола за дигиталну конверзију пружају одличну отпорност на шум треперења

природног светла и шум треперења вештачког светла, може да ради и под тамним стаклом. Поседује калибрисани алгоритам за детекцију светлости који рачуна UV индекс.

7.3.13. BME280 сензор ваздушног притиска

За мерење ваздушног притиска коришћен је сензор новије генерације који има интегрисан шири спектар мерења, тако да мери температуру, релативну влажност ваздуха и алтитуду тј. висину терена.

BME280 (Слика 7.17, детаљна спецификација *Прилог 7.13*) је дигитални сензор који мери температуру, влажност и атмосферски притисак, долази као калибрисан и не захтева додатна подешавања. Главни део модула је дигитални сензор (чип) последње генерације реномираног произвођача Bosch. Наследник је веома квалитетних сензора као што су BMP180, BMP085 и BMP183.

BME280 сензор користи комбинацију термистора за мерење температуре, капацитивног сензора за мерење влажности и барометарског сензора за мерење атмосферског притиска. Сензор користи дигиталне компензационе технике за побољшање прецизности и стабилности мерења. Овај прецизни сензор може да мери релативну влажност од 0 до 100% са тачношћу од $\pm 3\%$, барометарски притисак од 300 hPa до 1100 hPa са апсолутном тачношћу од ± 1 hPa и температуру од -40 °C до 85 °C са тачношћу од $\pm 1,0$ °C. Мерења су толико прецизна, да се може користити чак и као висиномер са тачношћу од ± 1 m.

Модул долази са уграђеним LM6206 3.3V регулатором, тако да се може користити напајање са 3.3 V или 5 V, и I2C интерфејсом. BME280 троши мање од 1 mA током мерења и само 5 μ A током мировања. Ова ниска потрошња енергије омогућава имплементацију у уређаје на батерије као што су слушалице, GPS модули или паметни сатови.



Слика 7.17 BME280 сензор

7.3.14. Анемометар Wind Speed Sensor

За мерење утицаја ветра кроз брзину и снагу коришћен је анемометар (Слика 7.18, детаљна спецификација *Прилог 7.14*), који је постављен на врху кошнице, помоћу којег су рачунати снага и брзина ветра.

Сензор мерења брзине ветра анемометар Wind Speed Sensor мери брзину ветра на основу ротације елисе. Када ветар дува, елиса се окреће и сензор региструје ту ротацију. На основу броја обртаја у јединици времена, сензор израчунава брзину ветра. Постоје различити типови анемометара, укључујући механичке и ултразвучне верзије. Повезивање се врши преко пинова, при чему се сензор напаја из напона од 5 V (за мерење 9-24 V). Излазни сигнал може бити аналогни или дигитални, у зависности од модела сензора.

Сензор може мерити брзину ветра у распону од неколико метара у секунди до неколико десетина метара у секунди, зависно од модела. Тачност сензора може бити унапређена калибрацијом и подешавањем.



Слика 7.18 Анемометар

Намењен је дуготрајном раду са великом прецизношћу и отпорности на радне температуре, може нормално да ради на $-20\text{ }^{\circ}\text{C}$ до $+50\text{ }^{\circ}\text{C}$, и влажности ваздуха од 35% до 85%, што га чини погодним за широку примену.

Принцип рада је следећи: брзина окретања лопатица од стране ветра је процењена усвајањем сигнала излазног напона (0-5 V), па се нивои брзине ветра израчунавају помоћу кода. Снага и подаци за анемометар се преносе преко трожилног кабла, максимална излазна струја је 20 mA (напајање 9 V – 24 V). Анемометар је способан да мери максималну брзину ветра до 70 m/s.

7.3.15. МАХ4466 и МАХ9814 сензори фреквенције звука и нивоа буке

За снимање података кретања фреквенцијског спектра звука коришћен је микрофонски сензор МАХ9814, као и ниво и степен буке за који је коришћен микрофонски сензор МАХ4466. Подаци су прикупљани са оба сензора и показали су високе прецизности за измерене вредности.

МАХ4466 (Слика 7.19, детаљна спецификација *Прилог 7.15*) је висококвалитетни микрофон са појачивачем и ниском дисторзијом и широким опсегом фреквенција. Овај сензор се истиче својом поузданошћу, ниским шумом и једноставним коришћењем. МАХ4466 се често користи у аудио апликацијама које захтевају висококвалитетно снимање звука, као што су бежични микрофони, аудио снимачи, аудио системи за надзор и друге сличне примене. МАХ4466 сензор ради на принципу појачавања аудио сигнала. Сензор има

интегрисани микрофонски појачивач ниског шума који повећава амплитуду аудио сигнала са микрофона. Појачани сигнал се може даље обрадити или снимити помоћу микроконтролера или аудио уређаја.



Слика 7.19 MAX4466 звучни сензор

Повезивање MAX4466 сензора обавља се преко три пина - VCC (напајање), GND (земља) и OUT (излаз), при чему се сензор напаја напоном од 2.4 V до 5.5 V. Излазни сигнал је аналогни и може се директно повезати на улаз микроконтролера или аудио уређаја. Специфичност MAX4466 сензора је у његовој високој пузданости, ниском нивоу шума, ниској дисторзији, што омогућава прецизно снимање звука чак и у бучним окружењима. Такође, сензор има широк опсег фреквенција, обухватајући аудио спектар од ниских до високих фреквенција. Излазни пин није дизајниран да покреће звучнике или било шта осим најмањих слушалица у уху.

MAX9814 (Слика 7.20, детаљна спецификација *Прилог 7.16*) је такође висококвалитетни микрофон са аутоматском контролом појачавања (Auto Gain Control - AGC) и интегрисаним микрофонским предпојачивачем. Сензор има интегрисан AGC који аутоматски прилагођава појачање сигнала како би одржао константан ниво излазног сигнала, без обзира на промене јачине улазног звука. Ово омогућава јасно снимање звука чак и у ситуацијама са великим динамичким распонем.

Повезивање се врши преко пинова, при чему се сензор напаја из напона од 2.7 V до 5.5 V. Излазни сигнал је аналогни и може се директно повезати на улаз микроконтролера или аудио уређаја. Специфичност MAX9814 сензора је у његовој високој пузданости, ниском шуму и AGC функцији, чиме се постиже конзистентан излазни сигнал без преоптерећења или дисторзије, што омогућава снимање чистог звука чак и у бучним окружењима.



Слика 7.20 MAX9814 звучни сензор

7.3.16. Улази и излази пчела

За снимање кретања пчела коришћен је микроконтролерски модул претходно описан ESP32, који је задужен за рад пролазних капија кроз које се крећу пчеле. Фреквенција кретања пчела је кључни фактор као што је већ поменуто у уводном делу, због извођења нових вредности, повезивања фактора утицаја, одређивања граничних вредности. Најважније је праћење узрочника, односно зависних и мање зависних фактора кретања пчела. На графику 7.1 можемо видети тренд кретања пчела, уласка у кошницу и излазак пчела из кошнице, за један дан.

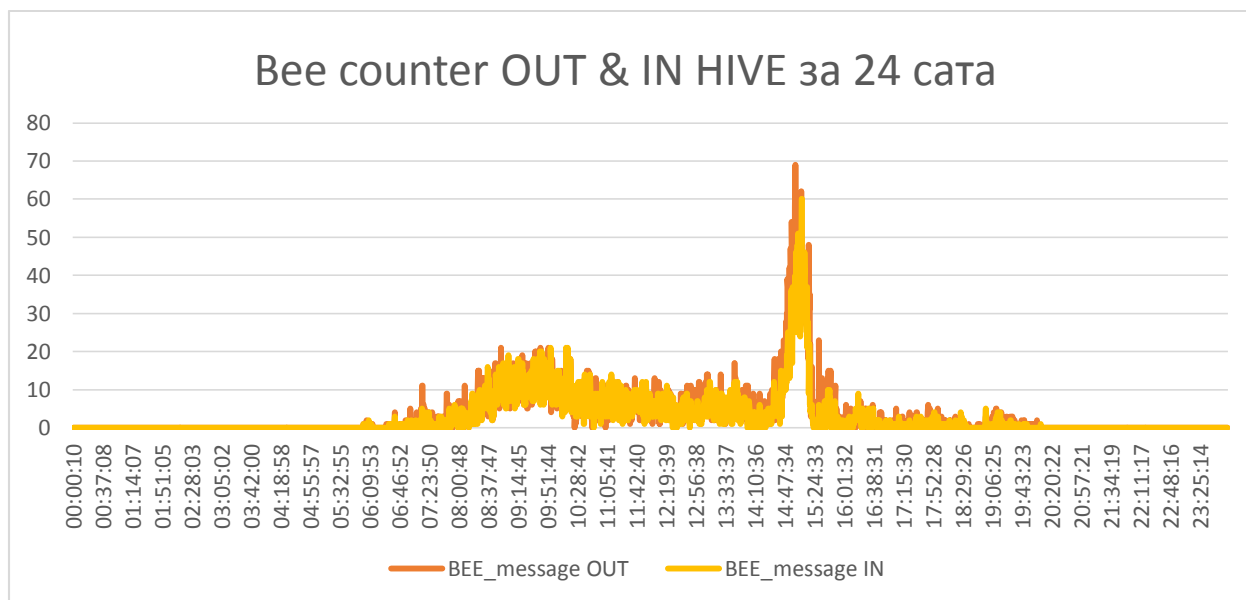
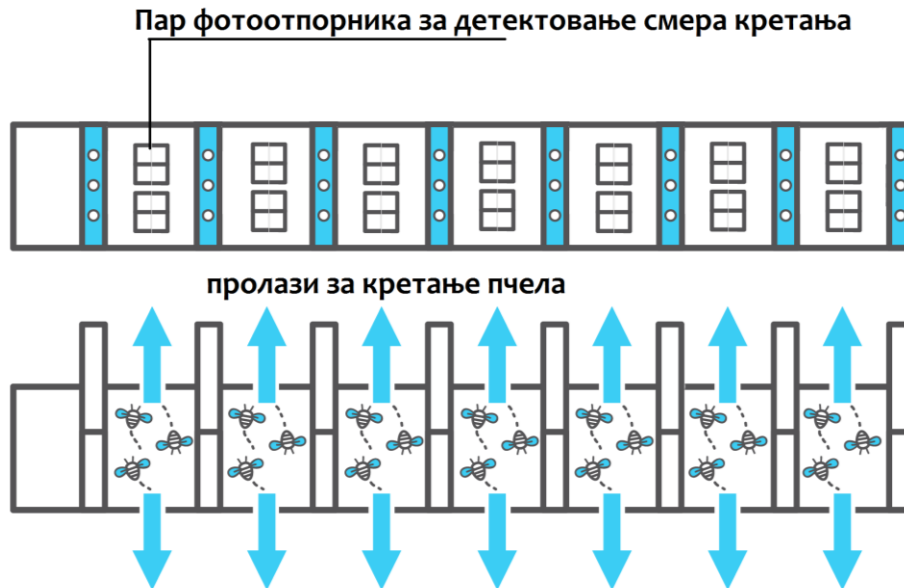


График 7.1 Улазак у кошницу и излазак пчела из кошнице за 24 сата.

Уређај за детекцију смера (Слика 7.21 и Слика 7.22) кретања пчела и бројање има више улаза (од 12 до 24 у зависности од модела кошнице). Тестирани модел је стандарна кошница па је могао да користи 24 улаза. Систем рада бројача са улазима за пчеле је

претходно појашњен, пчеле пролазе несметано кроз кратке ходнике изнад којих се налази сет фотоотпорника. Прекид рефлексије оба отпорника уписује као пролаз у зависности од смера кретања, упис је улаз или излаз. Прекид само једног не активира упис, ходници за пролаз су кратки па пчеле не могу да се сретну на пролазу и на тај начин нема лажних тригерована уписа.



Слика 7.21 Уређај за детекцију смера кретања пчела

Електроника пролаза је подељена на сегменте, за сваки сегмент је задужен по један регистар са отпорником и мосфетом. Овакви модели бројача су бољи и напреднији, јер не остављају пуно простора за грешке. Спецификација пролаза одговара висином и ширином величини стандардне пчеле које узгајају пчелари. Величина отвора не одговара нежељеним гостима попут стршљена или осе. Електроника је смештена у адекватном кућишту, и на тај начин потпуно заштићена и безбедна.



Слика 7.22 Пролази за улазак и излазак из кошнице

8. СОФТВЕР И CLOUD СИСТЕМ ПОДРШКЕ

Комплетирајући доследно систем „Интернета ствари“, целина софтверско-хардверске архитектуре ове дисертације употпуњена је и web апликацијом *BeeMAP*. Апликација је постављена на идејним основама микросервисне архитектуре, што значи да се састоји од више повезаних сервиса који се могу изоловано контролисати, модификовати и повезивати чиме се остављају широке могућности за проширење система изградом и повезивањем нових сервиса.

Тренутни web систем чине сервер апликација кодирана у *Laravel* радном оквиру који је изграђен у *PHP* програмском језику и клијентска web апликација кодирана у *Vue.js* радном оквиру који припада домену *JavaScript* радних оквира. Сервер апликација је *API* систем који испоставља готове *URL* путање за прикључивање уређаја и манипулацију подацима из уређаја паметне кошнице, док је клијентска апликација *SPA* (*single page app* – једностранична апликација). Клијентска апликација садржи интерактивни кориснички интерфејс на коме се приказују подаци из базе, који представљају мерења добијена са сензора из уређаја, као и обавештења о граничним вредностима.

Помоћу апликације корисник може управљати подешавањима сензора, дефинисати сензоре чија се мерења прате и одредити жељене горње и доње граничне вредности за које ће добијати нотификације. Поред тога корисник може интерактивно управљати уређајем на даљину, укључивати и искључивати вентилатор и хранилицу и дефинисати интервале њиховог аутоматског рада (што ће бити део наредних унапређења и проширивање система).

8.2. Примењене технологије

8.2.1. *PHP* програмски језик

Програмски језик *PHP* је један од најпознатијих језика за развој web страница, који омогућава развој web апликација са подршком за базе података. Неки од најпознатијих сајтова користе *PHP*, нпр. Facebook, Yahoo, Wikipedia, WordPress итд. *PHP* је скрипт језик проширен могућностима које укључују унапређену манипулацију ресурсима, руковођење *HTTP* сесијама, енкрипцију података, објектно оријентисано програмирање и друго. Последња и тренутно актуелна верзија овог програмског језика је *PHP* 8 [70].

Објектно-оријентисано пројектовање софтверских апликација (*object-oriented design*) је метод за пројектовање софтвера који моделира карактеристике апстрактних или реалних објеката користећи класе и објекте [71]. На објектно оријентисаном *PHP*-у заснован је популарни *Laravel* радни оквир који је употребљен у развоју пројекта, и који је у наставку рада описан.

8.2.2. *Laravel* радни оквир

Laravel представља радни оквир за развој web апликација базираних на *PHP* програмском језику. Овај радни оквир следи *MVC* (*model-view-controller*) архитектонски образац за изграду апликација. *Laravel* је постао један од најбрже растућих *PHP* радних

оквира икада. Са својом експресивном синтаксом и одличном документацијом, веома је лако добити потпуно функционалну web апликацију за кратко време [72].

Laravel има уграђену подршку за аутентификацију и ауторизацију корисника, служи као радни оквир који пружа ту могућност у PHP програмском језику. Има употпуњену MVC архитектуру и систем за шаблонизовање (templating system) који се назива Blade, и друга унапређења као и командни интерфејс назван Artisan. Последња верзија је (Laravel 9) названа “Illuminate”, code потпуно рефакторисан и испоручиван у одвојеним пакетима помоћу Composer система за управљање пакетима, има мноштво додатних унапређења и проширења [73].

Поред споменуте подршке за аутентификацију и ауторизацију корисника, Laravel између осталог укључује и следеће функционалности и карактеристике [74]:

- Осигуран систем миграција
- Подршка за MVC архитектуру
- Уграђен систем сигурности
- Систем за шаблонизирање
- Систем за тестирање и друго

8.2.3. MVC архитектура

MVC је скраћеница састављена од почетних слова речи Model, View и Controller. MVC представља софтверски дизајн образац а основна је идеја да се њиме постиже један од императива у програмирању који доприносе квалитетнијем организовању кода и тај принцип се назива принципом раздвајања задужења (енгл. SOC – separation of concerns). Код којим је изграђена апликација се помоћу овог принципа раздваја на три засебне целине – моделе, приказе и контролере [75].

Модел (*model*) – представља модел података. Модел се дефинише класама чији атрибути осликавају структуру самих података, односно табела у бази. У Laravel радном оквиру помоћу Artisan командног интерфејса се на основу дефинисане класе модела помоћу поступка такозване миграције може креирати табела у бази [76]. Тиме се читава база, и у смислу креирања и у смислу свих измена, обрађује на нивоу модела.

Приказ (*view*) - представља слој апликације који је доступан кориснику. На овом нивоу се употребом језика за маркирање (HTML и CSS) садржај из базе испоставља на web страници у одређеном графичком решењу. Laravel помоћу система за шаблонизовање Blade, омогућава генерисање готових приказа и графичких решења, који се даље могу прилагодити специфичним потребама апликације.

Контролер (*controller*) – овај слој апликације омогућава комуникацију између претходна два слоја. Класе контролера имају приступ класама модела и помоћу њих преузимају податке из базе. Затим се на нивоу самих метода унутар класе контролера могу

дефинисати додатне програмске операције и мутације, филтрације података на жељени начин пре него што се из контролера подаци испоруче самом приказу (view). Одатле су подаци доступни на нивоу прегледа и могу се одређеном логиком излиставати и приказивати кориснику. Поред тога, контролери обављају комуникацију и у супротном смеру. Форме за унос и друге интерактивне компоненте на нивоу самог приказа, а након корисникове потврде, податке са приказа шаљу контролерима. Поново се на нивоу контролера ти подаци могу модификовати, валидирати или на било који други начин обрнути пре него што се преко модела сачувају измене у бази.

Имајући све то у виду јасно је зашто се каже да MVC архитектура следи SOC принцип. То у пракси значи да се могу вршити измене на било ком слоју апликације а да се не поремети значајно неки други слој. Ми можемо вршити сваку врсту измене дизајна приказа и података који се приказују, чиме се апликација може радикално трансформисати у визуелном смислу а да се и даље користе исти модели и исти контролери.

8.2.4. JavaScript програмски језик

JavaScript је динамички програмски језик. Малог је обима и најчешће се користи као део web страница чија имплементација дозвољава скриптама на клијентској страни да врше интеракције са корисником и направе web странице динамичким. То је један од интерпретираних програмских језика који има објектно-оријентисане способности [77].

Изворно JavaScript је развијен како би се користио само на клијентској страни и учинио странице интерактивним, касније је овај језик добио употребу и на серверској страни појавом Node.js платформе. Node.js је асинхрон, догађајима вођен систем који извршава JavaScript у току рада (runtime), и дизајниран је да омогући изградњу скалабилних мрежних апликација [78].

Временом је развијен велики број радних оквира и библиотека у JavaScript-у, са различитом наменом. Неки од њих намењени су серверској страни и олакшавају рад у Node.js-у, као што је случај са Express.js-ом, док су други намењени клијентској страни. Међу њима најпознатија библиотека је jQuery, а међу радним оквирима окренутим кориснику издвајају се Angular, Vue и React.

8.2.5. Round-Trip и Single-Page апликације

Радни оквири на клијентској страни су добили на значају и популарности услед све већег продора „апликација на једној страни“ (енгл. single page app). Овакве апликације су популарне због свих побољшаних перформанси. Док се у класичној архитектури свака страница на сајту припрема на сервер страни и затим испоручује као готов HTML са пратећим JavaScript и CSS садржајем, у случају SPA сервер шаље само један HTML фајл, који затим повлачи и преостали JavaScript и CSS садржај.

Апликације са класичном архитектуром се понекад називају и Round-trip апликацијама (апликације засноване на "путу у круг"). Свака акција од стране корисника као што је клик на линк или потврда форме доводе до тога да web читач добија од сервера

комплетно нов HTML документ. У оваквим врстама апликација, читач је суштински само машина за приказивање HTML садржаја, а сва логика и обрада података почива на серверу. Претраживач прави серије HTTP захтева које сервер обрађује генеришући HTML документе динамички.

Главни недостаци round-trip апликација јесу у томе што корисник чека док се следећи HTML документ припреми на серверу. При промени странице на сајту, тј. путање унутар URL-а нема поновног слања читаве странице са сервера као што је то случај у класичној архитектури, већ се само динамички, на основу JavaScript кода мења приказани HTML садржај. Са сервера се, у том случају, једино добијају нови подаци у JSON формату који се затим приказују на одговарајућем месту на страници, али при томе свакако нема поновног читавања DOM-а, нити испоруке HTML-а са сервера.

8.2.6. *Vue.js* радни оквир

Vue.js је флексибилан и моћан радни оквир отвореног кода (open source) намењен развоју клијентских апликација, који преузима дизајн принципе из развоја на сервер страни и примењује их на HTML елементе [79].

Vue.js је један од три најпопуларнија радна оквира која су развијена у JavaScript-у, а намењена клијентској страни апликације. За разлику од React-а и Angular-а који су развијени у великим компанијама Google и Facebook, Vue.js је развио појединац Evan You при чему је за инспирацију користио оба радна оквира. Због тога Vue.js спаја одређене елементе приступа и филозофије рада од оба наведена радна оквира.

8.3. ИЗРАДА И РАЗВОЈ АПЛИКАЦИЈЕ

У наставку је изложен сам процес израде и развоја web апликације. Експликацију пројектне апликације која је изложена кроз сам поступак израде сматрамо најпогоднијим начином да се елаборира сам пројекат, јер поред тога што ће се тиме добити приказ и објашњење самог пројекта, уједно се добија и својеврсно техничко упутство за креирање једног таквог (истог или сличног) пројекта чиме се, сматрамо, додатно повећава употребна вредност рада.

Подешавање радног окружења

При изради апликације први корак подразумева подешавање радног окружења. Апликација је најпре развијена у локалном радном окружењу а затим постављена на хостинг. Радно окружење најпре подразумева инсталацију WAMP или XAMP софтверског пакета који укључује Apache сервер, MySQL базу података и PHP програмски језик. То нам омогућава употребу Laravel радног оквира на сервер страни апликације. Затим, потребно је инсталирати и Node.js програм како би се користио Vue.js радни оквир на клијент страни апликације.

Инсталација WAMP развојног окружења

WAMP у потпуности спада у домен отвореног кода (енгл. open source) и инсталациони пакет се може преузети са следеће адресе: <https://www.wampserver.com/en/> [80]. Пре инсталације самог WAMP пакета, потребно је инсталирати и *Visual C++ Redistributable for Visual Studio 2012*, који се такође може бесплатно преузети и инсталирати са следеће адресе: <https://www.microsoft.com/en-us/download/details.aspx?id=30679> [81].

Креирање Laravel апликације

Како би могли да користимо Laravel радни оквир унутар WAMP развојног окружења неопходно је инсталирати и Composer менаџер пакета. Инсталација Composer-а се може преузети са адресе: <https://getcomposer.org/download/> [82]. Када је та инсталација завршена сви предуслови су испуњени за израду апликације, односно радно окружење је припремљено и може се започети са креирањем Laravel пројекта.

Креирање Vue.js апликације

Када је Laravel пројекат креиран и припремљен потребно је припремити и радно окружење са клијентске стране инсталацијом Vue.js радног оквира. Предуслов је, као што је раније речено, инсталиран Node.js програм. Инсталација се може бесплатно преузети са следеће адресе: <https://nodejs.org/en/download/> [83], при чему је потребно одабрати последњу стабилну верзију за онај оперативни систем који користимо.

Рутирање

Систем рутирања организован је тако да се рутирање врши и на сервер страни и на клијентској страни система, одн. и унутар Laravel API-ја и унутар Vue.js апликације (Слика 8.1 и Слика 8.2). Рутирање на серверској страни има задатак да дефинише различите путање које омогућавају манипулацију подацима у бази, док рутирање на клијентској страни дефинише изглед и садржај корисничког интерфејса.

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('home');
});

Route::get('{any}', function () {
    return view('home');
})->where('any', '.*');
```

Слика 8.1 Садржај routes/web.php датотеке

```

<?php
use Illuminate\Support\Facades\Route;

Route::post('login', 'AuthController@login');
Route::post('logout', 'AuthController@logout');

Route::middleware('auth:sanctum')->group( function () {
    Route::get('authenticated', function(){ return true; });
    Route::get('readLogs/{id}', 'LogsController@readLogs');
    Route::post('logsFromDevice', 'LogsController@logsFromDevice');

    Route::get('getHives', 'HivesController@getHives');
    Route::get('getSingleHive/{id}', 'HivesController@getSingleHive');
    Route::post('saveHive', 'HivesController@saveHive');
    Route::put('editHive/{id}', 'HivesController@editHive');

    Route::post('saveSensor', 'SensorsController@saveSensor');
    Route::put('editSensor/{id}', 'SensorsController@editSensor');
    Route::delete('deleteSensor/{id}', 'SensorsController@deleteSensor');

    Route::post('saveAlert', 'AlertsController@saveAlert');
    Route::get('getAlerts', 'AlertsController@getAlerts');
    Route::delete('deleteAlert/{id}', 'AlertsController@deleteAlert');

    Route::get('getUsers', 'UsersController@getUsers');
    Route::post('registerUser', 'UsersController@registerUser');

    Route::get('getHomepageStatistics', 'HomeController@getHomepageStatistics');
    Route::get('getWeather', 'WeatherController@getWeather');
});

```

Слика 8.2 Садржај routes/api.php датотеке

Рутирање на страни сервера

На серверској апликацији правила рутирања се дефинишу у оквиру фолдера *app/routes*, овде се могу пронаћи неколико РНР датотека од којих су нама од значаја *web.php*, и *api.php*. Датотека *routes/web.php* дефинише путање које користи веб интерфејс. Основно место у коме ће се тражити метода за задату путању јесте *web.php* фајл. Међутим, будући да наша сервер апликација има искључиво улогу АРІ система и не треба да шаље различити HTML садржај на основу различитих путања, сва логика у тој датотеци постављена је тако да се за било коју путању враћа home view. Унутар HTML-а, тог home приказа иницијализује се Vue.js апликација која ће заправо преузети параметре руте и приказати одговарајући HTML садржај у зависности од тога која је путања у питању.

Ако размотримо следеће три путање: <https://www.iobeers.rs/dashboard> (почетна страница апликације) <https://www.iobeers.rs/hives> (приказ свих кошница) и <https://www.iobeers.rs/users> (приказ свих корисника), или било коју другу путању која даје различити приказ на клијентској апликацији, за сваку од ових путања са сервера се враћа

исти приказ, односно `home view`. Да је архитектура била таква да се уместо SPA користи сервер апликација, онда би се за сваку од тих путања враћао посебан приказ тј. користио посебан `view` фајл на серверу. Садржај `web.php` датотеке приказан је на слици 8.2. Ту можемо видети два различита позива статичког `get` метода `Route` класе и сваки од њих има задатак да преусмерава на `home view`, без обзира каква је путања и њени параметри.

Рутирање на клијент страни

Путање на клијент страни дефинишу изглед и садржај посебних страница на `web` апликацији. Оне су дефинисане унутар `resources/js/router/index.js` датотеке. Унутар ње се најпре врши увоз саме `Vue` библиотеке, па затим `VueRouter` модула који се ставља у употребу унутар апликације помоћу методе `use` којој се прослеђује као улазни параметар а која се позива над увезеном `Vue` класом.

Све посебне странице јесу компоненте које су углавном постављене унутар `resources/js/views` фолдера. Свака компонента за коју се жели дефинисати посебна рута и која ће се на тој специфичној страници приказати мора најпре бити увезена помоћу `import` команде.

8.4. Манипулација сензорима

Креирање модела и табеле у бази података

Податке о сензорима чувамо у табели под називом `sensors`. Ова табела садржи следећа поља: Назив сензора (`name`), пин код (`pin`), назив сензора као уређаја (`sensor_device_name`), опис сензора (`description`), доња гранична вредност упозорења (`bottom_warning`), доња гранична вредност за аларм (`bottom_danger`), горња гранична вредност упозорења (`top_warning`), горња гранична вредност за аларм (`top_danger`), јединицу мере која одговара резултату који сензор мери (`sensor_unit`), спољни кључ за кошницу како би се знало којој кошници сензор припада (`hive_id`), податак када је сензор креиран (`created_at`) и када су подаци о сензору последњи пут измењени (`updated_at`).

Покретањем команде `php artisan make: migration create_sensors_table` у терминалу добијамо генерисану датотеку `create_sensors_table.php` на локацији `app/database/migrations` унутар које се налази нова класа миграције (Слика 8.3). Ова датотека садржи класу `CreateSensorsTable` која наслеђује класу `Migration`. То важи за сваку класу миграција које ћемо у наставку израде пројекта креирати. Унутар метода `up()` постављамо код приказан на следећој слици.

Након завршетка уређивања шеме за нову табелу, покрећемо миграцију командом: **`php artisan migrate`** - којом се креира нова табела у бази на основу шеме.


```

public function up()
{
    Schema::create('sensors', function (Blueprint $table) {
        $table->id();
        $table->string("name")->nullable();
        $table->string("pin")->nullable();
        $table->string("sensor_device_name")->nullable();
        $table->string("description")->nullable();
        $table->double("bottom_warning")->nullable();
        $table->double("bottom_danger")->nullable();
        $table->double("top_warning")->nullable();
        $table->double("top_danger")->nullable();
        $table->integer("hive_id");
        $table->timestamps();
    });
}

```

Слика 8.3 Шема миграције за табелу sensors

У следећем кораку креирамо класу модела која ће комуницирати непосредно са самом табелом у бази и посредством које контролер може добијати податке из базе. Класа модела носи назив Sensor и садржи код приказан на слици 8.4.

```

namespace App;
use Illuminate\Database\Eloquent\Model;

class Sensor extends Model
{
    public $table = 'sensors';
}

```

Слика 8.4 Класа Sensor модела

Класа Sensor модела самим тим што наслеђује класу Model наслеђује и приступ бази података, довољно је да се само \$table атрибут дефинише и да се њему додели вредност која одговара називу табеле у бази податка (sensors) како би се сама класа модела повезала са раније креираном табелом. Након тог корака, класа модела је доступна за употребу на контролер слоју архитектуре.

Израда контролера

Унутар фолдера App/Http/Controllers креирамо нову php класу под називом SensorsController, која наслеђује класу Controller. Ова класа садржи методе за креирање новог сензора (Слика 8.6 и Слика 8.7), измену сензора (Слика 8.8) и брисање сензора. Када је преглед унетих сензора у питању, ова класа не садржи методу за то. Будући да сензори припадају кошницама, када се преузимају подаци о кошницама уједно се преузимају и подаци о сензорима који су у вези са том кошницом. Та метода је обрађена у следећем поглављу које се бави манипулацијом кошницама.

Чување података о појединачној кошници се обавља помоћу метода `saveHive()` који је приказан на слици 8.5.

Чување података о сензору

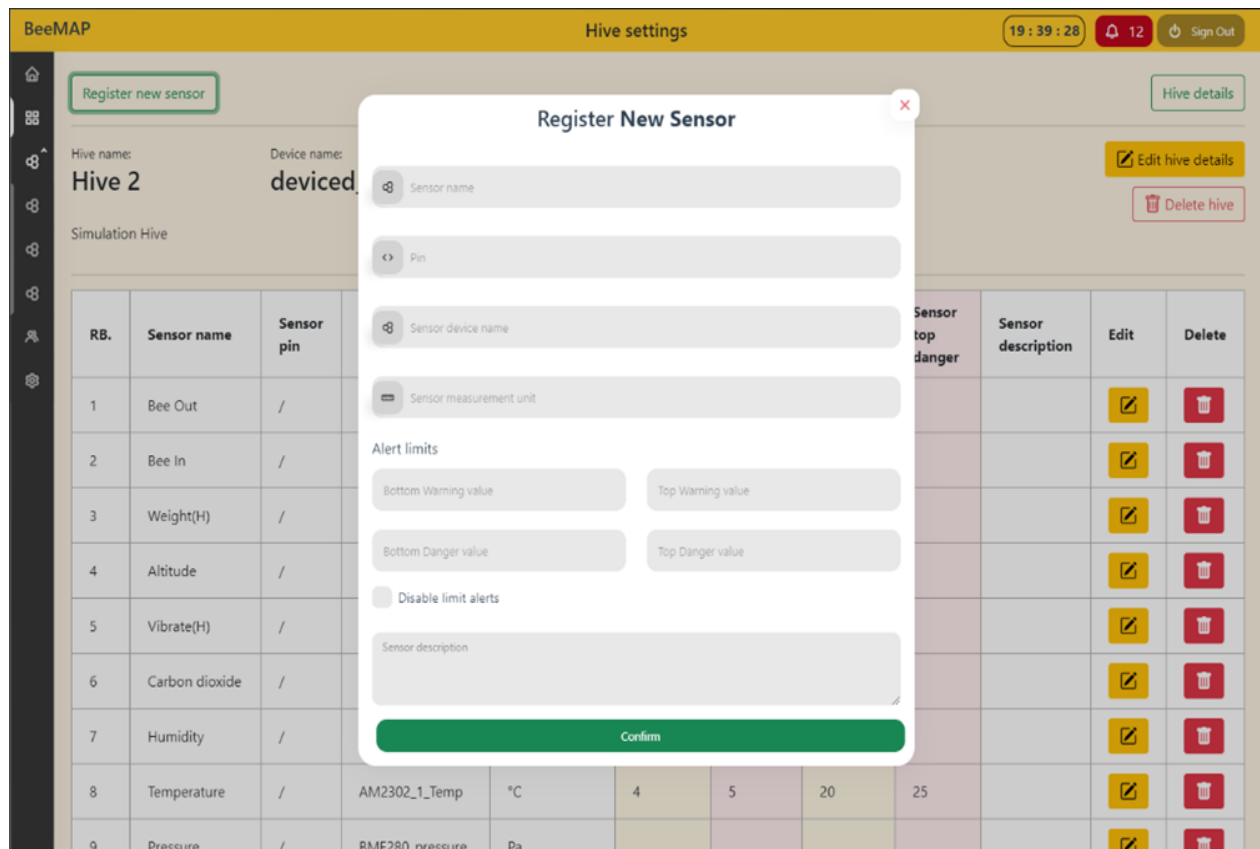
```
public function saveSensor(Request $request) {
    $sensor = new Sensor;
    $sensor->name = $request->name;
    $sensor->pin = $request->pin;
    $sensor->sensor_device_name = $request->sensor_device_name;
    $sensor->bottom_warning = $request->bottom_warning;
    $sensor->bottom_danger = $request->bottom_danger;
    $sensor->top_warning = $request->top_warning;
    $sensor->top_danger = $request->top_danger;
    $sensor->sensor_unit = $request->sensor_unit;
    $sensor->description = $request->description;
    $sensor->hive_id = $request->hive_id;

    if($sensor->save()) {
        return response()->json(['status' => true,
            'message' => 'Sensor Added Successfully']);
    } else {
        return response()->json(['status' => false,
            'message' => 'SENSOR NOT ADDED! There is some problem']);
    }
}
```

Слика 8.5 Чување података о сензору

```
<template>
  <vs-dialog v-model="active" @close="close" prevent-close width="640px">
    <template #header>
      <h4 class="not-margin"> Register <b>New Sensor</b> </h4>
    </template>
    <div class="con-form">
      <div class="input-field">
        <vs-input v-model="formData.name" @focus="errors.name.status = false"
          :state="errors.name.status?'danger': 'dark'" label-placeholder="Sensor name">
          <template #icon>
            <i class='bx bx-hive'></i>
          </template>
          <template #message-danger v-if="errors.name.status">
            {{errors.name.message}}
          </template>
        </vs-input>
      </div>
    </div>
  </vs-dialog>
</template>
```

Слика 8.6 Регистровање новог сензора *back*



Слика 8.7 Регистравање новог сензора front

Измена података о сензору врши се на готово идентичан начин као и код креирања новог сензора у бази. Разлика је само у томе што се уместо позивања конструктора за креирање новог објекта помоћу статичке методе `find()`, а преко класе `Sensor`, проналази већ постојећи сензор. Том објекту се затим додељују вредности из `request` улазног параметра. На крају, поново се позива метода `save()` као и код креирања новог сензора. Друга разлика је што ова метода има поред објекта класе `Request` још један улазни параметар којим се добија примарни кључ сензора, који се потом прослеђује `find` методи. Приказ кода методе `editHive()` којом се обавља измена података о кошници унутар контролера `HivesController` дата је на слици 8.8.

Слика 8.8 Измена података о сензору

8.5. Манипулација кошницама

Креирање модела и табеле у бази података

Подаци о кошницама се памте у посебној табели под називом *hives*. Ова табела садржи поља за унос назива кошнице (*name*), описа (*description*), кода (*code*) и временских ознака када је кошница креирана (*created_at*) и када је последњи пут измењена (*updated_at*).

```
public function up()
{
    Schema::create('hives', function (Blueprint $table) {
        $table->id();
        $table->string("name")->nullable();
        $table->string("description")->nullable();
        $table->string("code")->nullable();
        $table->timestamps();
    });
}
```

Слика 8.9 Шема миграције за табелу *hives*

Покретањем команде `php artisan make: migration create_hives_table` у терминалу добијамо генерисану датотеку `create_hives_table.php` на локацији `app/database/migrations` унутар које се налази нова класа миграције под називом `CreateHivesTable`. Унутар метода `up()` постављамо код приказан на слици 8.9.

Израда контролера

Унутар фолдера App/Http/Controllers креирамо нову php класу под називом HivesController, која наслеђује класу Controller. Ова класа садржи методе за креирање нове кошнице, преузимање података о свим кошницама, преузимању података о појединачној кошници, измени података постојеће кошнице и брисању записа о кошници из базе.

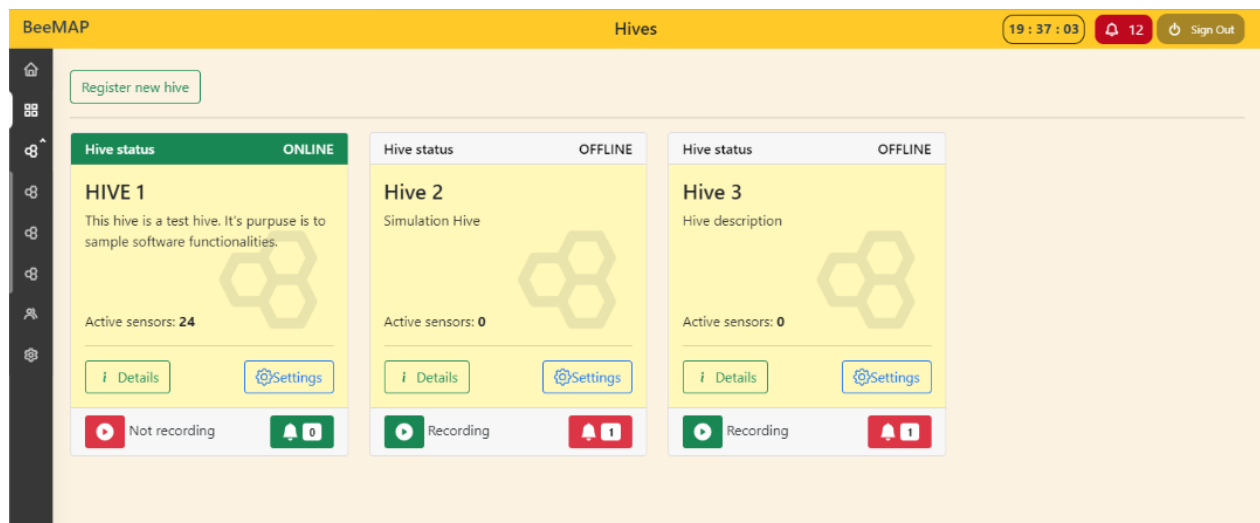
Свака од ових функција представља једну од „крајњих тачака“ API система које нам омогућавају потпуну манипулацију над подацима о кошницама у бази.

Преузимање података о свим кошницама

```
public function getHives()
{
    $hives = DB::table('hives')->get();
    return $hives;
}
```

Слика 8.10 Преузимање података свих кошница

Сви подаци о свим кошницама који су унети у базу могу се добити помоћу методе getHives(), која је приказана на слици 8.10. Ова метода позива статичку методу table() класе DB прослеђујући јој као улазни параметар назив табеле у којој се чувају подаци о кошницама (hives), што успоставља везу са том табелом, а затим се посредством метода get() добијају сви записи из те табеле. Тако преузети сви редови табеле постају повратна вредност функције.



Слика 8.11 Чување података о кошници front

Чување података о појединачној кошници се обавља помоћу метода saveHive() који је приказан на слици 8.11 (Слика 8.12 back).

```

public function saveHive(Request $request) {
    $hive = new Hive;
    $hive->name = $request->name;
    $hive->code = $request->code;
    $hive->description = $request->description;

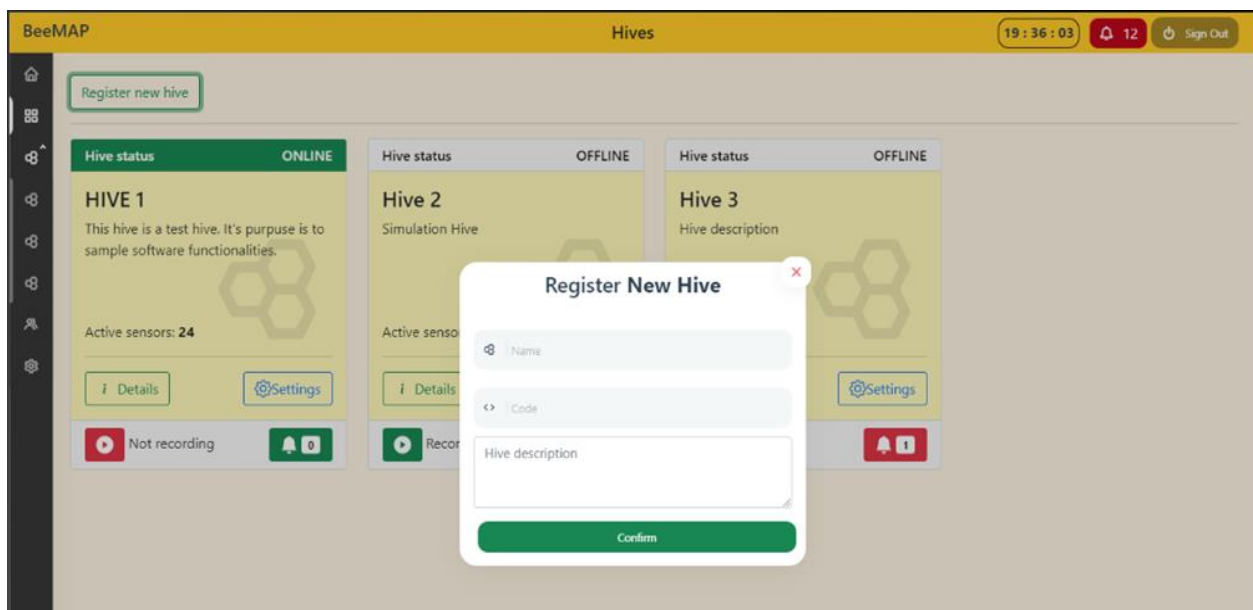
    if($hive->save()) {
        return response()->json(['status' => true,
                                'message' => 'Hive Added Successfully']);
    } else {
        return response()->json(['status' => false,
                                'message' => 'HIVE NOT ADDED! There is some problem']);
    }
}

```

Слика 8.12 Чување података о новој кошници

Ова метода у својој почетној верзији прихвата објекат Request класе на основу захтева који је упућен преко HTTP POST методе. Затим се у телу методе најпре креира нови објекат класе Hive, као вредности атрибута name, code и description постављају се вредности атрибута request објекта. Затим се над тим објектом позива функција save(), и уколико она има позитиван резултат извршења методе, враћа JSON објекат са потврдним статусом и поруком о успешној операцији, а у супротном поруку о неуспеху. У даљем унапређењу кода ова метода може садржати и неку врсту провера приспелих података, како би се евентуално извршила контрола и осујетили покушаји уноса неправилних података.

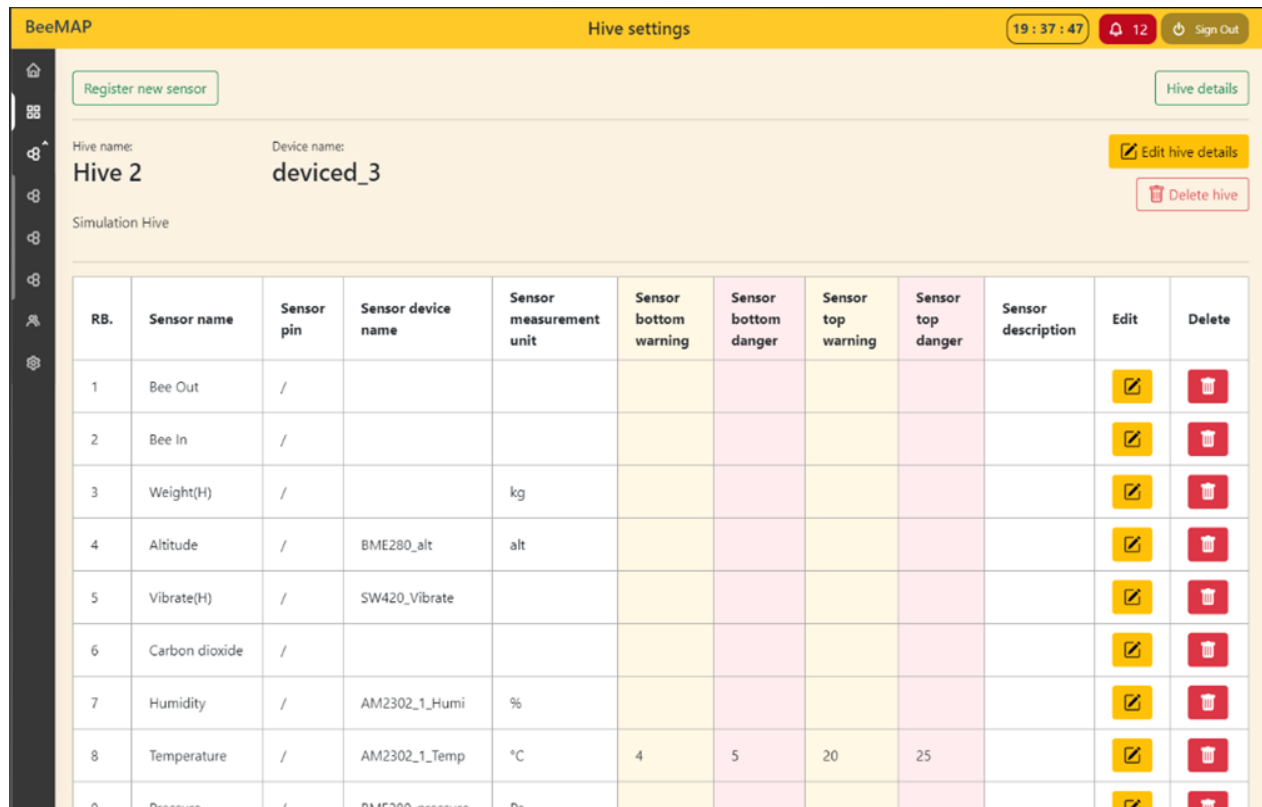
На слици 8.13 приказана је регистрација нове кошнице, која се иницира на контролном панелу где су приказане све регистроване кошнице.



Слика 8.13 Регистравање нове кошнице

При креирању нове кошнице, преузимање података о појединачној кошници се обавља помоћу `getSingleHive` методе унутар `HivesController` контролера. Метода као улазни параметар има вредност примарног кључа појединачне кошнице чији се подаци претражују. Унутар методе селекује се, на основу примарног кључа, појединачна кошница али исто тако и сви сензори који припадају тој кошници из табеле са сензорима. Будући да се селекција врши из две различите табеле покреће се метода `beginTransaction()` из `DB` класе, што омогућава да се упит у базу оптимизује и ефикасније изврши, на крају се позива метода `commit()` како би се означило да је трансакција завршена. Резултати се у низу као повратна вредност враћају на крају извршења функције. Повратни асоцијативни низ садржи под индексом `hive` податке о кошници, а под индексом `sensors` низ са подацима о сензорима.

Након регистравања нове кошнице следи подешавање појединачних параметара и вредности за сваки сензор уколико је потребно нешто изменити, у супротном могу се оставити предефинисане поставке (слика 8.14).



RB.	Sensor name	Sensor pin	Sensor device name	Sensor measurement unit	Sensor bottom warning	Sensor bottom danger	Sensor top warning	Sensor top danger	Sensor description	Edit	Delete
1	Bee Out	/									
2	Bee In	/									
3	Weight(H)	/		kg							
4	Altitude	/	BME280_alt	alt							
5	Vibrate(H)	/	SW420_Vibrate								
6	Carbon dioxide	/									
7	Humidity	/	AM2302_1_Humi	%							
8	Temperature	/	AM2302_1_Temp	°C	4	5	20	25			
9	Pressure	/	BME280_pressure	Pa							

Слика 8.14 Подешавање појединачних параметара кошнице

Манипулација обавештењима

Читање, чување и брисање нотификација се обавља у класи `AlertsController` која је написана унутар истоименог `php` фајла.

Преузимање обавештења

Обавештења се преузимају помоћу методе `getAlerts` (Слика 8.15). Ова метода има исти код као и метода за преузимање података о свим кошницама. Користи се `DB` класа и њена статичка метода `table` па метода `get`. Метода враћа низ са свим нотификацијама.

```
public function getAlerts()
{
    $alerts = DB::table('alerts')->get();
    return $alerts;
}
```

Слика 8.15 Метода `getAlerts` за преузимање обавештења

Чување обавештења

Обавештења се чувају помоћу `saveAlert` методе (Слика 8.16). Унутар ње се, као и у претходним примерима где су се креирали нови уноси у базу, позива конструктор одговарајуће класе модела. У овом случају то је `Alert` класа. Креираном објекту се затим задају вредности атрибута који се односе на назив кошнице, назив сензора, измерену вредност, граничну вредност која је прекорачена као и податак о томе да ли је прекорачена горња или доња гранична вредност (овоме служи атрибут `top`, то је заправо вредност логичког типа, уколико је `true` прескочена је горња гранична вредност, у супротном прескочена је доња гранична вредност). Као и у претходним примерима позива се затим функција `save` како би се сачувало обавештење.

```
public function saveAlert(Request $request) {
    $alert = new Alert;
    $alert->hive_name = $request->hive_name;
    $alert->limit_value = $request->limit_value;
    $alert->sensor_name = $request->sensor_name;
    $alert->top = $request->top;
    $alert->value = $request->value;

    if($alert->save()) {
        return response()->json(['status' => true,
                                'message' => 'Alert Added Successfully']);
    } else {
        return response()->json(['status' => false,
                                'message' => 'ALERT NOT ADDED! There is some problem']);
    }
}
```

Слика 8.16 Метода `saveAlert` за чување обавештења

8.6. Манипулација корисничким налозима

Израда модела

Израда контролера

За потребе манипулације корисничким налозима креиран је фајл `UserController.php` са класом `UserController`. Ова класа садржи методу за преузимање листе свих постојећих корисника и методу за регистрацију новог. Будући да је у тренутној верзији апликације само администраторима омогућено да креирају нове налоге и да та опција није доступна обичном посетиоцу метода за регистрацију корисника и јесте смештена у овом контролеру. У супротном била би смештена у `AuthController`-у.

Регистрација корисника

Регистрација корисника се обавља помоћу методе `registerUser` (Слика 8.17). Ова метода добија као улазни параметар објекат класе `Request`. Та класа под атрибутима треба да садржи податке о кориснику кога треба регистровати на систем. Подаци који су обавезни за унос су име корисника, email адреса и лозинка. Поред тога, врши се и додатна валидација email адресе како би се проверило да ли је адреса добијена у правилном формату, као и то да ли је јединствена за све кориснике. Ово се обавља помоћу `validate` методе која се позива над добијеним објектом `Request` класе. Уколико је валидација успешна, наставља се извршење кода у методи. Уколико валидација није успешна порука о грешци се одмах враћа кориснику са свим детаљима о изузетку који се десио.

```
public function registerUser(Request $request)
{
    $request->validate([
        'name' => ['required'],
        'email' => ['required', 'email', 'unique:users'],
        'password' => ['required']
    ]);

    $newUser = Users::create([
        'name' => $request['name'],
        'email' => $request['email'],
        'role' => $request['role'],
        'password' => Hash::make($request["password"]),
        'first_login' => $request['first_login'],
        'remember_token' => Str::random(80),
    ]);

    return $newUser;
}
```

Слика 8.17 Регистрација корисника методом `registerUser`

Након успешне валидације позивом методе `create` креира се нови корисник чије се име додељује на основу атрибута `name`, email адреса на основу атрибута `email` и улога на основу атрибута `role`. Лозинка се додељује позивом статичке методе `make` из класе `Hash`

како би се енкриптовала унета шифра. Атрибут `first_login` дефинише да ли ће се од корисника захтевати да рестартује лозинку одмах након прве пријаве. На крају се креира и токен за корисника у форми насумичног стринга.

Аутентификација

Пријава корисника

Пријава корисника се врши помоћу `login` методе унутар класе `AuthController` (Слика 8.18.). Као улазни параметар добија се објекат класе `Request` са подацима унетим путем пријаве. Приликом пријаве се уносе само `email` и лозинка. Помоћу `validate` методе се проверава да ли су добијена оба податка, као и да ли је `email` у правилном формату. Уколико јесте, онда се помоћу класе `User` покушава претрагом добити корисник са том `email` адресом. Уколико претрага није успела и вредност променљиве није постављена то значи да уопште не постоји корисник са таквом `email` адресом у бази. Другим речима, унета је погрешна `email` адреса, у том случају се таква порука враћа кориснику. Уколико је корисник пронађен, помоћу класе `Auth` и статичке методе `attempt` покушава се аутентификација корисника са унетим `email`-ом и унетом лозинком. Уколико је тај корак успешан враћају се подаци о пријављеном кориснику како би се користили на клијентској апликацији. У супротном, уколико аутентификација није успела, то значи да лозинка није правилно унета и кориснику се враћа порука о томе.



Слика 8.18 Пријава и одјава корисника помоћу `login` методе

8.7. Кориснички интерфејс апликације

Развијена је веб апликација која има вишеструку улогу, позиционирана је на cloud систему где се налази и база података снимљених са слоја перцепције (откључана демо

кошница [84]). Апликација за приказ података на интерфејсу, учитава податке из базе података и приказује тренутна читавања сензора из активне кошнице. Интерфејс апликације (Слика 8.19) дизајниран је у духу пчелињег екосистема, и има могућност вишеструке презентације података, информација и командног и управљачког блока. Као што је наведено у претходном делу сама апликација се налази на cloud страни, што даје низ предности и могућности. Web оријентисане апликације су у предности над другим типовима (локалне апликације) из више разлога, од којих је најбитнији самоодржив аутоматизовани систем. ИoT доноси новине на овом пољу, потискујући израду архитектуре и дизајн ка cloud сервисима.



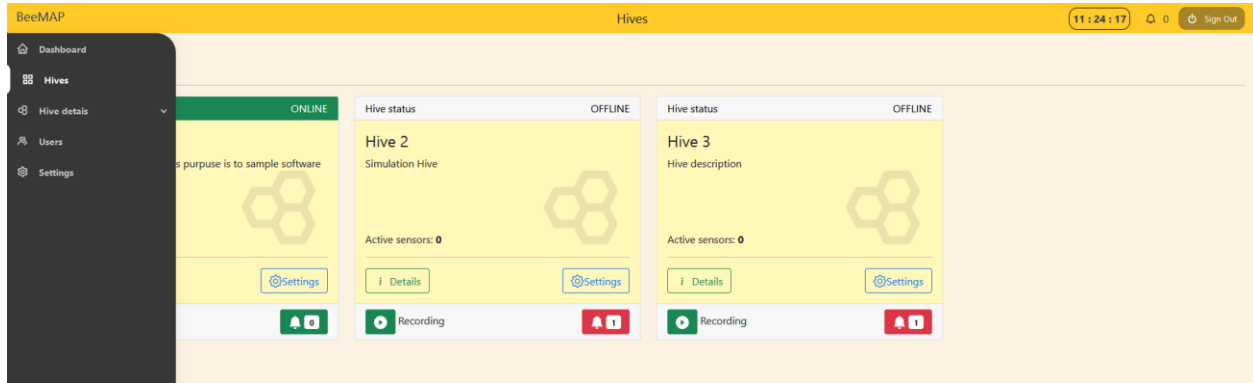
Слика 8.19 Интерфејс ВееМар апликације

У поглављу 3. приказани су примери решења системског приступа мониторинга кошница који прате појединачне параметре као што су: температура, термичке промене, аудио сигнале, медоносни принос, од којих већина користе ИoT технологије за прикупљање и слање података у централну базну станицу. Апликативна решења за манипулацију и приказ података део су система за мониторинг пчела [85]-[89], а неке од њих примењују методе вештачке интелигенције за извођење закључака и предикције.

Већ смо дефинисали важност параметара за праћење, као и сет параметара које прати ИoT систем који је настао као резултат ове дисертације [90], а то су следећи параметри: температура, влажност, UV и IC зрачење, вибрација, бука, количина и интензитет дневне светлости, киша, брзина и снага ветра, ваздушни притисак, квалитет ваздуха, присутност гасова као и улазак и излазак пчела из кошнице.

Ови подаци се након снимања шаљу серверском делу на упис у *web* базу података која се налази на *cloud* систему, са ког се фронтални приказ за потребе презентације вредности похрањује подацима. Апликација представљена у овом раду прати тренд савремених технологија, респонзивна је што олакшава приступ са било ког уређаја (мобилног телефона, таблета, лаптоп или десктоп рачунара) који има приступ интернету. Тренд апликација је да буду *web* оријентисане, јер доноси предности које смо већ поменули у уводу, као и лакшу надоградњу софтверских система, олакшаног одржавања саме апликације, као и сервиса за заштиту и филтрирање приступа.

Управљачки део апликације задужен за презентациони слој. Састоји се из више целина којима приступамо преко навигационог панела (dashboard), где имамо избор за манипулацију подацима и кошницама (Слика 8.20). Начин рада овог дела апликације је претходно појашњен у серверском делу описа апликације.

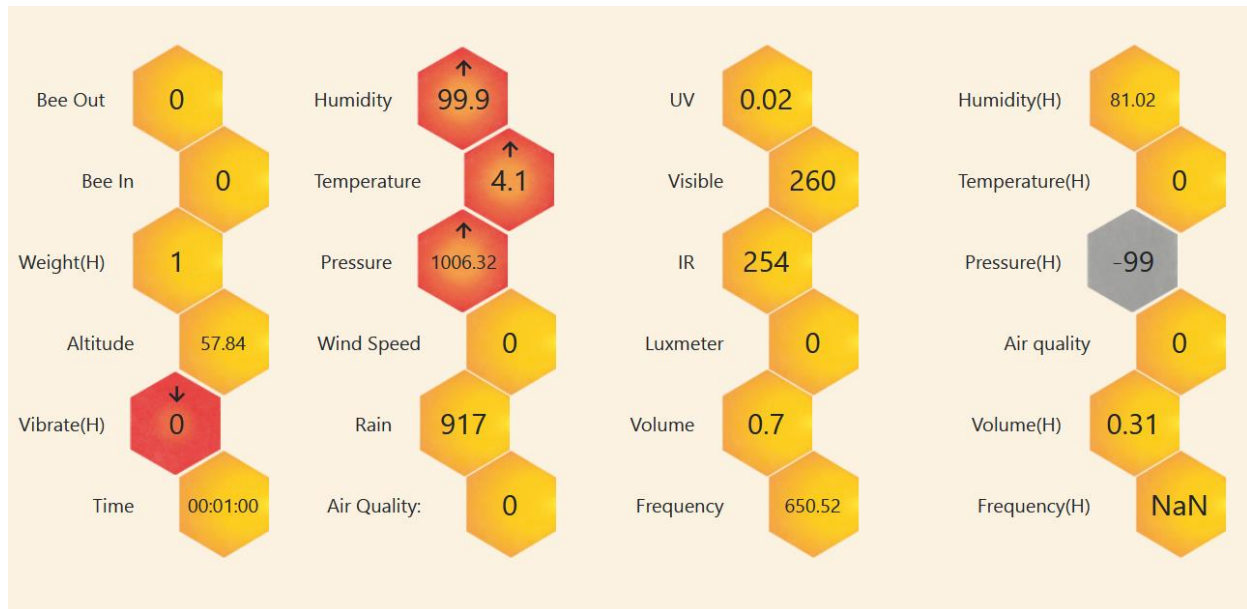


Слика 8.20 Навигациони панел ВееМар апликације

Главни мени нам омогућава манипулацију кошницама и ћелијама за приказ података са сензора. Панел нуди приказ активних кошница, где можемо додавати нове кошнице или обрисати кошнице уколико су неактивне из неког разлога. За сваку кошницу можемо засебно бирати које параметре желимо пратити и на тај начин активирамо само ћелије за приказ параметара које желимо да пратимо, параметре и касније можемо додавати или деактивирати.

Речено је да овај део апликације користи податке у реалном времену и прати кретања параметара анализирајући их. У односу на поставке које можемо дефинисати за сваки параметар који се прати посебно (Слика 8.20), а којим дефинишемо локална кретања микроклиме и фреквенцију кретања пчела, можемо дефинисати подешавања критичних вредности само за кључне параметре као што су температура, влажност или дневна светлост, или нека сезонска кретања вредности. Такође можемо дефинисати праћење укупног броја улазака и излазака пчела из кошнице у току дана. Аналитички део апликације прати кретања ових параметара у односу на граничне вредности уоквирене локалним климатским условима или укупним бројем фреквенције кретања пчела. Ако се појаве одступања од граничних вредности активира се аналитички аларм који обавештава пчелара о променама стања у кошници. Ћелије које приказују вредности у апликацији су динамичке и мењају боје у односу на приказане вредности.

Аналитички аларм се активира када вредност мерења дође до границе критичне вредности. Прво ћелија мења боју а затим се активира аларм. Ове вредности се дефинишу појединачно за сваки параметар посебно у опцији за дефинисање приказа параметара. Пример је критична вредност када је висока температура и повишена влажност која утиче на нормално функционисање пчела. Када вредност температуре унутар кошнице пређе 35 °C [91], или се релативна влажност приближи вредности 90% активира се аларм. Шаље се *pushup* нотификација, која обавештава о променама у кошници.

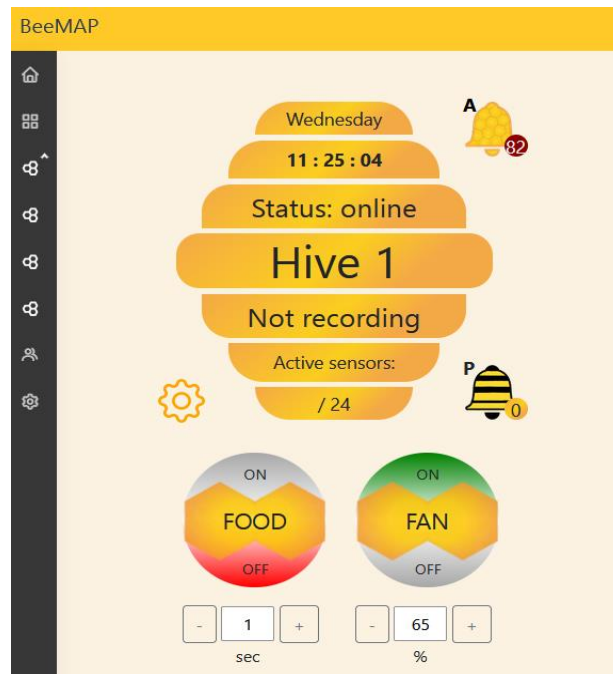


Слика 8.21 Део параметара који се прати из екосистема

Предиктивни и аналитички аларми приказани су на Слици 8.22 на којој можемо поред статуса кошнице видети и број активних сензора, видети и део апликације за управљање кошницом путем ИоТ. Управљање кошницом на основу промена стања омогућава независна архитектура *cloud* система. Сама апликација има контролу укључивања вентилације у кошници, као и укључивање аутоматске хранилице која се налази у кровном делу кошнице. Систем за регулацију пролаза пчела на основу детекције гасова, дима или других агенаса је изостављен у дисертацији (више о томе у поглављу закључак).

ИоТ архитектура омогућава слање командних наредби за активацију регулације квалитета ваздуха у кошници и регулацију влажности. Кошница поседује два вентилатора на супротним странама без директног усмерања ваздуха како се не би ометале активности пчела. Вентилатори су мањих габарита 80x80, нечујни са пригушеним радом и могућношћу регулације брзине обртаја.

Систем оваквим приступом уочава обрасце понашања и кретања пчела и везује их за параметре правећи зависну релацију и на тај се начин похрањује модул вештачке интелигенције, на основу кога се врши предикција.



Слика 8.22 Приказ статуса кошнице са предиктивним и аналитичким алармом

Web интерфејс садржи поред аналитичког и индикаторе за предиктивни аларм. За активирање предиктивног аларма задужен је интегрисани API, који је део модела предиктивног алгоритма на основу кретања пчела (детаљније у наредном поглављу који описује моделовање предикције). Кретање пчела је узроковано најчешће хранењем, метеоролошким факторима, дневним активностима у односу на исте факторе или људским деловањем. На тај начин формулишемо директно и индиректно зависне факторе и њихову међусобну зависност. Овакав приступ изградње апликација олакшава примену модела вештачке интелигенције за анализу података.

Тренд активности пчела можемо пратити преко графикана за приказ података (Слика 8.23), за сваки параметар засебно. Можемо изабрати и преглед статистичких података о историји кретања параметара. Можемо селектовати параметре које желимо видети засебно, а можемо сродне параметре пратити истовремено. Овде постоје ограничења због вредности самих параметара.



Слика 8.23 График кретања за изабрани параметар

Графички приказ даје визуелни увид у аналитички део статистике, на основу којих можемо пратити поједине параметре и њихов тренд кретања. Посебна секција у овом делу је приказ логова (Слика 8.24). Овде можемо пратити све сензоре који су у активном статусу али и да видимо оне који су из неког разлога престали да мере или имају проблема са мерењем. На тај начин имамо увид у стабилност сензора и мерних електронских компоненти који служе за прикупљање података и њихово слање у базу. Анализом логова можемо утврдити да ли треба неку компоненту заменити и да ли су прикупљени подаци валидни.

Most recent logs

Bee Out	Bee In	Weight H	Altitude H	Vibrate H	Time	Humidity	Temperature	Pressure	Wind Speed	Rain	Air Quality	Uv	Visible	Ir	Luxmeter	Volume	Frequency	Humidity H	Temperature H	Pressure H	Air Quality H	Volume H	Frequency H
0	0	1	58.92	0	00:46:02	99.9	4.8	1006.19	0	852	0	0.02	261	253	0	0.76	636.55	79.49	0	-99	0	0.49	NaN
0	0	1	58.81	0	00:45:42	-99	-99	1006.21	0	855	0	0.01	259	253	0	0.7	362.22	79.59	0	-99	0	0.43	NaN
0	0	1	58.73	0	00:45:22	-99	-99	1006.22	0	858	0	0.02	260	255	0	0.68	241.58	79.51	0	-99	0	0.32	NaN
0	0	1	58.88	0	00:45:01	-99	-99	1006.2	0	858	0	0.03	262	253	0	0.8	371.1	79.57	0	-99	0	0.35	NaN
0	0	1	58.98	0	00:44:41	-99	-99	1006.19	0	858	0	0.02	260	254	0	0.73	635.98	79.51	0	-99	0	0.33	NaN

Слика 8.24 Приказ логова мерних сензора

Оријентација архитектуре апликакативног слоја ка *web* и *cloud* системима даје велику предност у виду развоја алата са константним приступом систему. Примена IoT технологија омогућила је имплементацију микроелектронских сензора и компонената за прикупљање података са различитих удаљених локација. Податке које смо прикупили шаљемо преко GPRS SIM модула, или других бежичних технологија које смо обрадили у претходном делу, и смештамо са било које локације у *web* базу која је смештена на *cloud* систем. Захваљујући *open source* приступу и дељењу података и решења, развој система за апикултуру, и не само апикултуру већ и читаву пољопривреду, расте.

Као што је већ наведено, сензори снимају податке у кратким интервалима од 20 секунди, док се записи праве на сваких 5 минута а генерисање излазних података у сатним интервалима. Тако да се база освежава новим подацима у назначеном периоду. Као и у сваком систему који користи мноштво сензора може доћи до отказивања рада сензора или до учитавања погрешних параметара. У том случају подаци се морају проверити и исправити пре уписа у базу. За први део задужена је сама скрипта која генерише слање података са *Edge* слоја која проверава упис података и уклања погрешну вредност уколико се јави. Уколико се појави неправилност у подацима и као такви прођу процес слања у базу, модул за прихватање и упис података у базу врши проверу вредности за упис у базу.

Севрисни део апликације који је од виталног значаја за рад читавог апликативног слоја је увек приступачан захваљујући примени *cloud* архитектуре. Овакав приступ омогућава праћење рада свих компоненти система, прати се рад сензора који на интерфејсу

има своју презентацију вредности. Као што је већ појашњено уколико дође до отказивања или нерегуларности у раду иницира се само презентационо поље што имплицира на рад сензора. На тај начин смо обавештени о раду сензора па можемо проверити узроке. Уколико се ради о само неисправном читању или делимичном отказивању, систем се може ресетовати јер опрема има могућност хардверског ресета. Ово су велике предности јер опрема и инсталације често нису близу и то отежава сервисни део одржавања читавог система.

Друга велика предност је приступ подацима и контролним функцијама, уколико се сниме одступања, можемо проверити да ли се одступање десило случајном грешком мерења или се вредности понављају. Уколико је случајна грешка система можемо утврдити употребом графика кретања тока података или на други начин. Једна од управљачких команди, а то је хардверски ресет, је веома значајна за поновно покретање система без одласка до кошнице. Постоји могућност режимског програмирања ресета, јер постоје годишњи периоди са великим осцилацијама у влажности ваздуха што може ометати рад опреме, па се опрема може ресетовати сама на свака 24 сата.

9. ПРИКАЗ РЕЗУЛТАТА МЕРЕЊА СЕНЗОРА И ГРАФИКА КОРЕЛАЦИЈА

Следи опис и анализа резултата мерења у циљу приказа кретања појединачних параметара, независних променљивих у односу на кретање пчела за период од 20 дана (01.10.2021-21.10.2021.год). Пажљива анализа упоредних кретања вредности омогућава увид у поклапању тренда вредности појединих параметара са кретањем пчела, што указује на зависне или узрочне факторе кретања пчела. Поједини фактори немају директан утицај и поклапање тренда вредности али се могу везати за друге факторе како би се прецизније утврдио тренд. Све вредности снимљене са сензора ИоТ система, анализирани су на дневном нивоу, тако да графици садрже и дневна запажања. За лакшу визуелизацију и графички приказ, користе се упросечене сатне вредности параметара за наведени период.

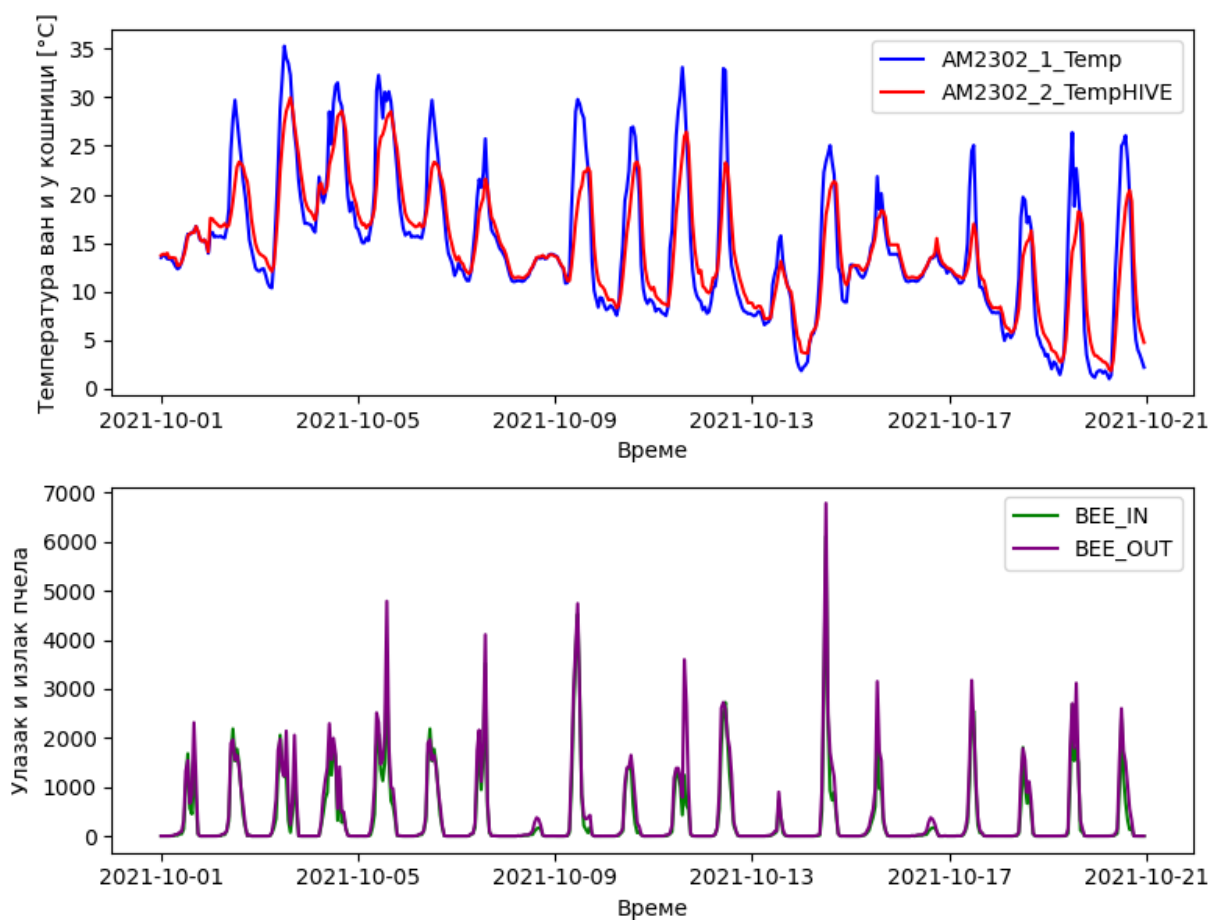


График 9.1 Упоредни тренд кретања пчела у односу на промене температуре (DHT22) за период од 20 дана

Иако се температуре измерене у кошници и ван ње разликују у измереним дневним вредностима, јасно се види поклапање са трендом кретања пчела. Преклапање само потврђује претходне наводе из истраживачког дела, да је параметар температуре један од кључних узрочних фактора кретања. Примећујемо да је промена у току дана и до 25 °C

степени, са великим осцилацијама дневних и ноћних температура (График 9.1). У табели 9.1 приказане су појединачне корелације температурних вредности са кретањима пчела.

Корелације приказане у наредним табелама урађене су у програмском језику Python коришћењем библиотеке `pandas` за математичке методе, и библиотека `matplotlib` за исцртавање графика.

Табела 9.1 Корелација температурних вредности у односу на кретање пчела IN/OUT.

Корелација променљивих у односу на уласке и изласке пчела	Вредност корелације
Корелација температуре у кошници и улазак пчела:	0.654
Корелација температуре ван кошнице и улазак пчела:	0.434
Корелација температуре у кошници и излазак пчела:	0.683
Корелација температуре ван кошнице и излазак пчела:	0.474

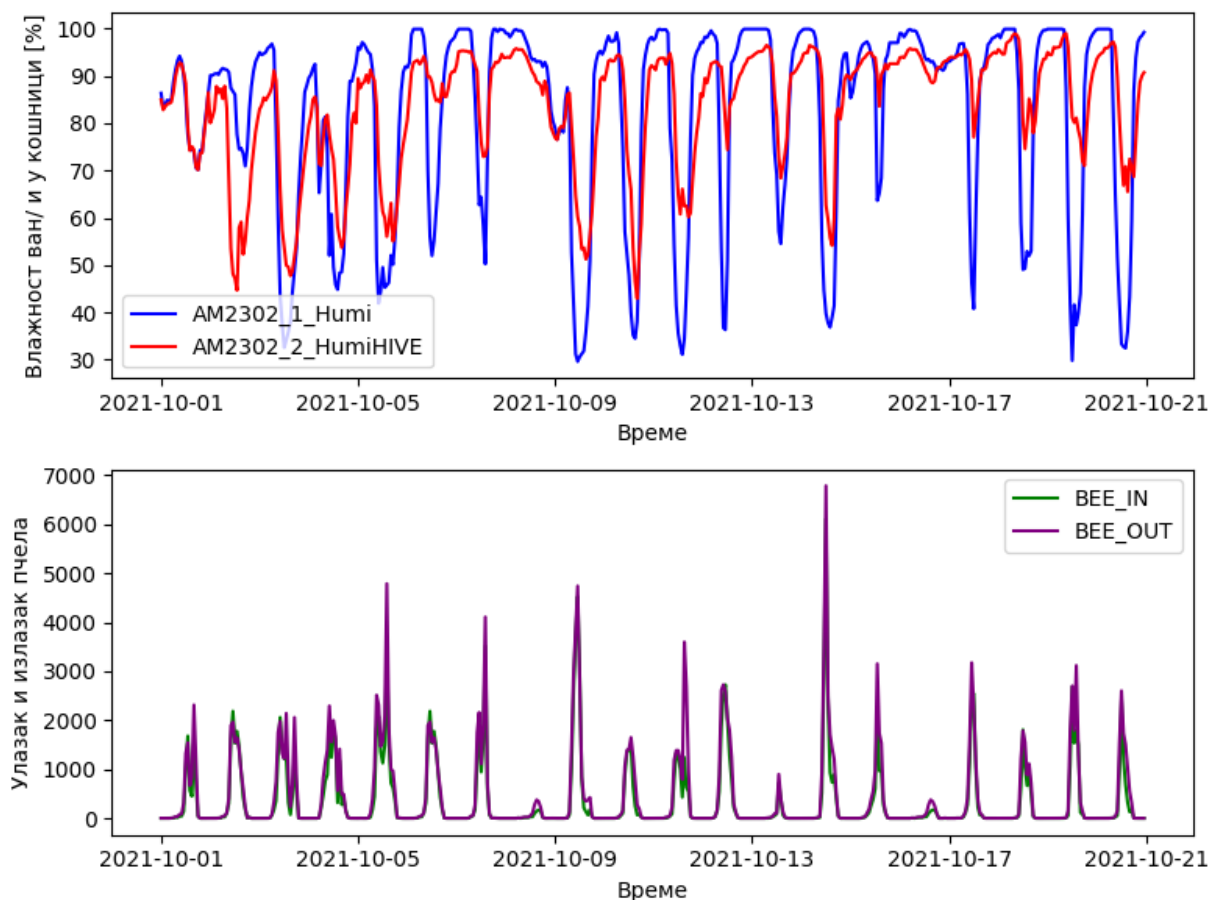


График 9.2 Упоредни тренд кретања пчела у односу на промене влажности ваздуха (DHT22) за период од 20 дана

График влажности ваздуха (График 9.2) такође јасно указује преклапање трендова промена овог параметра са кретањем пчела. Ово додаје параметар влажности у табелу параметара који су повезани са кретањем пчела. Примећујемо да је релативна влажност

висока у граничним деловима дана, са трендом опадања вредности током дневног периода и сунчаним активностима. Релативна влажност ваздуха је у тим периодима дана достигала вредност од преко 95%. У табели 9.2 приказане су појединачне корелације вредности влажности ваздуха са кретањима пчела.

Табела 9.2 Корелација вредности влажности ваздуха у односу на кретање пчела IN/OUT.

Корелација променљивих у односу на уласке и изласке пчела	Вредност корелације
Корелација влажности ваздуха у кошници и улазак пчела:	-0.687
Корелација влажности ваздуха ван кошнице и улазак пчела:	-0.421
Корелација влажности ваздуха у кошници и излазак пчела:	-0.732
Корелација влажности ваздуха ван кошнице и излазак пчела:	-0.452

Следећи график приказује вредности присуства кише и ваздушног притиска (График 9.3). Овде примећујемо веома слабо присуство кише за приказани период иако постоје забележене вредности, међутим ове вредности треба повезати са вредностима влажности ваздуха који су иницијатори активације сензора. У октобру месецу сензори су детектовали високу влажност што овај сензор и потврђује, а имамо приметну активност кретања пчела.

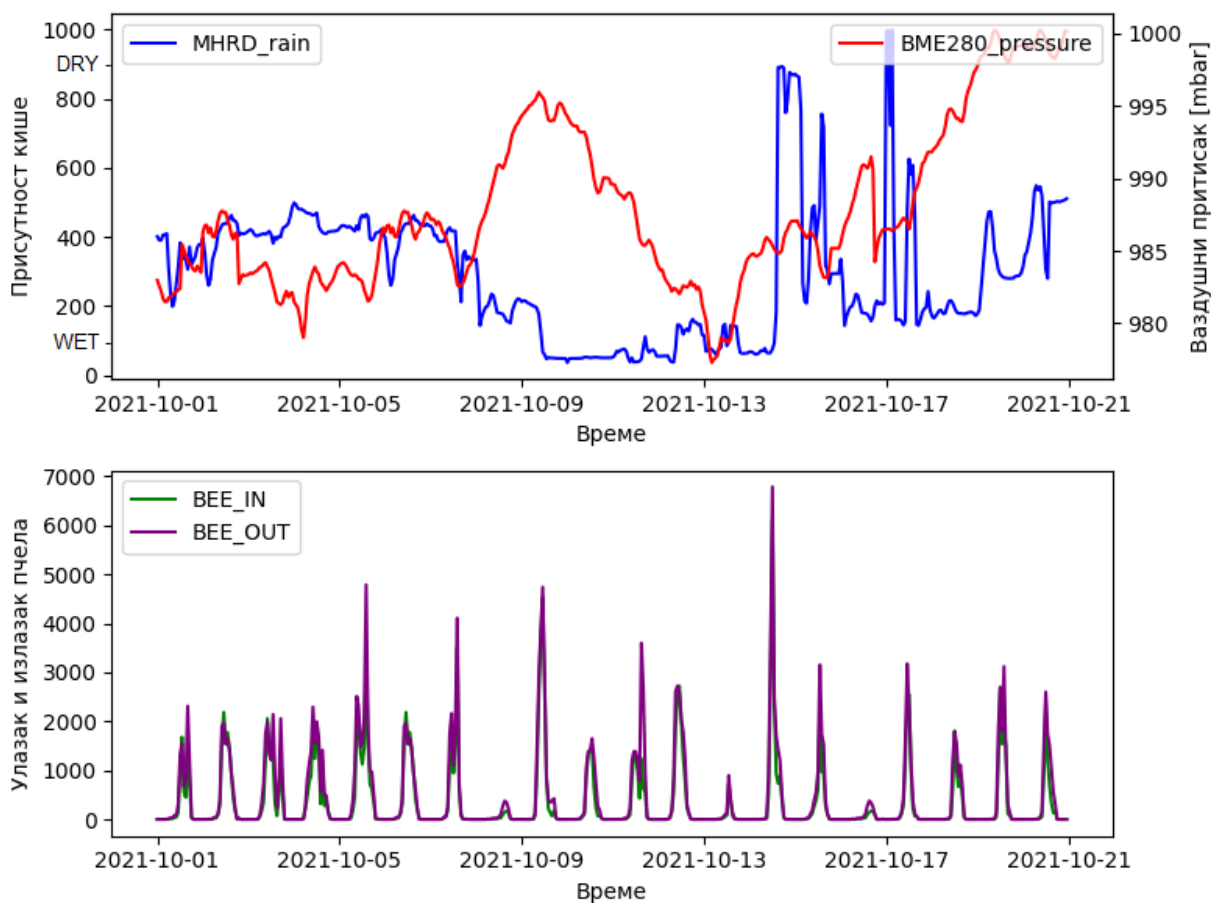


График 9.3 Упоредни тренд кретања пчела у односу на промене присуства кише и ваздушног притиска (MHRD) за период од 20 дана

Ове вредности указују на то да је висока прустност влаге утицала на сензор који мери присутност кише, да високу влажност препозна као кишу, што потврђују вредности кретања влажности у деловима дана када пчеле нису активне, тј. касним вечерњим и раним јутарњим часовима. У току дана када су пчеле активне вредност апсолутне влажности опада, због повећане дневне температуре. Забележени су кратки и делимични кишни периоди, што није довољно за квалитетнију анализу корелације, већ се морају комбиновати фактори.

На Графику 9.3.1 могу се прецизније видети промене на дневном нивоу (за дане 12-15.10.2021).

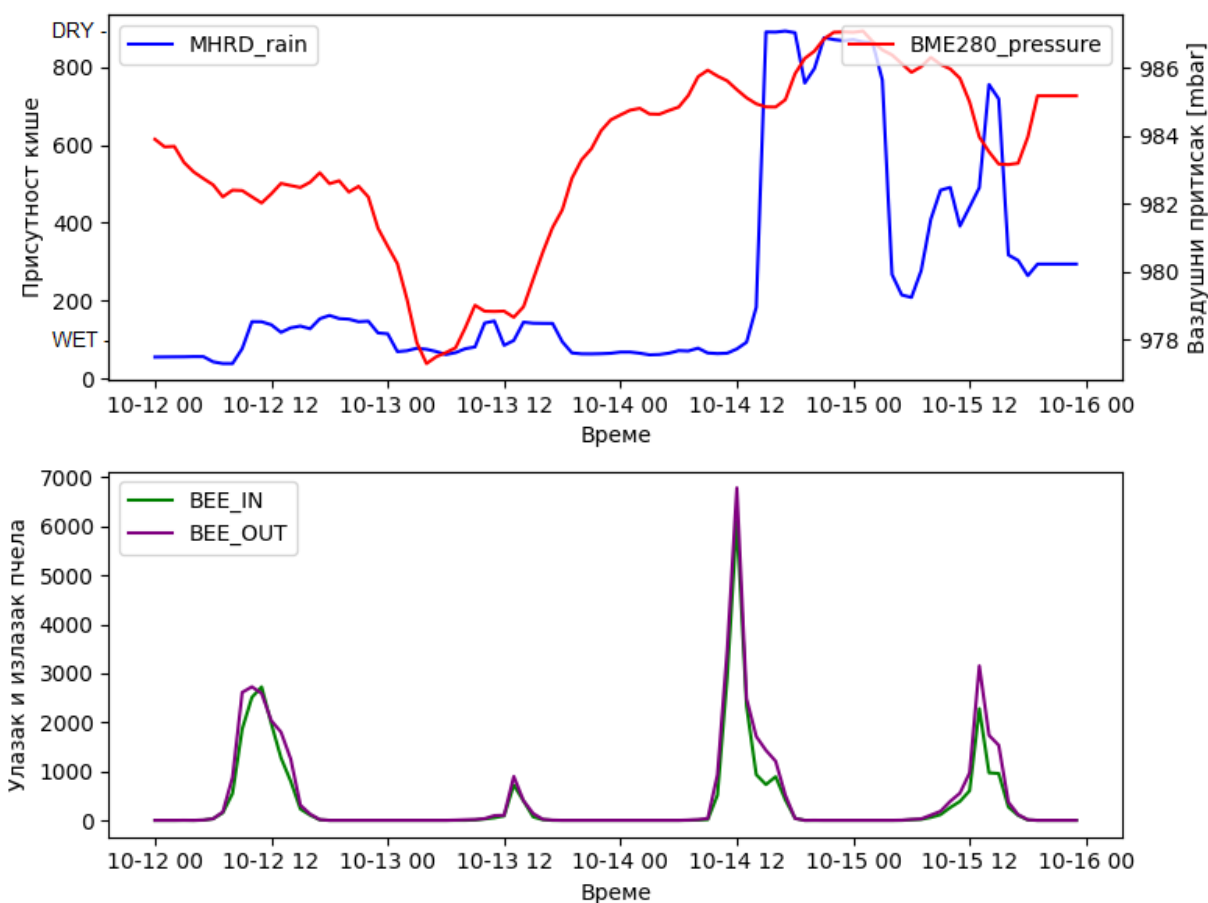


График 9.3.1 Упоредни тренд кретања пчела у односу на промене присуства кише и ваздушног притиска на дневном нивоу

Међутим, ваздушни притисак је последица тренда промена временских услова, стога се вредности ових параметара укрштају са другим параметрима ради откривања заједничке корелације које треба узети у обзир. Овај тип података захтева другачију методу за анализу, која укључује релације више узајамних параметара. Табела корелације приказана у наставку показује тренд измерених вредности корелација над подацима (9.3).

Табела 9.3 Корелација вредности присуства кише и ваздушног притиска у односу на кретање пчела IN/OUT

Корелација променљивих у односу на уласке и изласке пчела	Вредност корелације
Корелација присуства кише и уласка пчела:	0.037
Корелација ваздушног притиска и уласка пчела:	-0.027
Корелација присуства кише и изласка пчела:	0.039
Корелација ваздушног притиска и изласка пчела:	-0.035

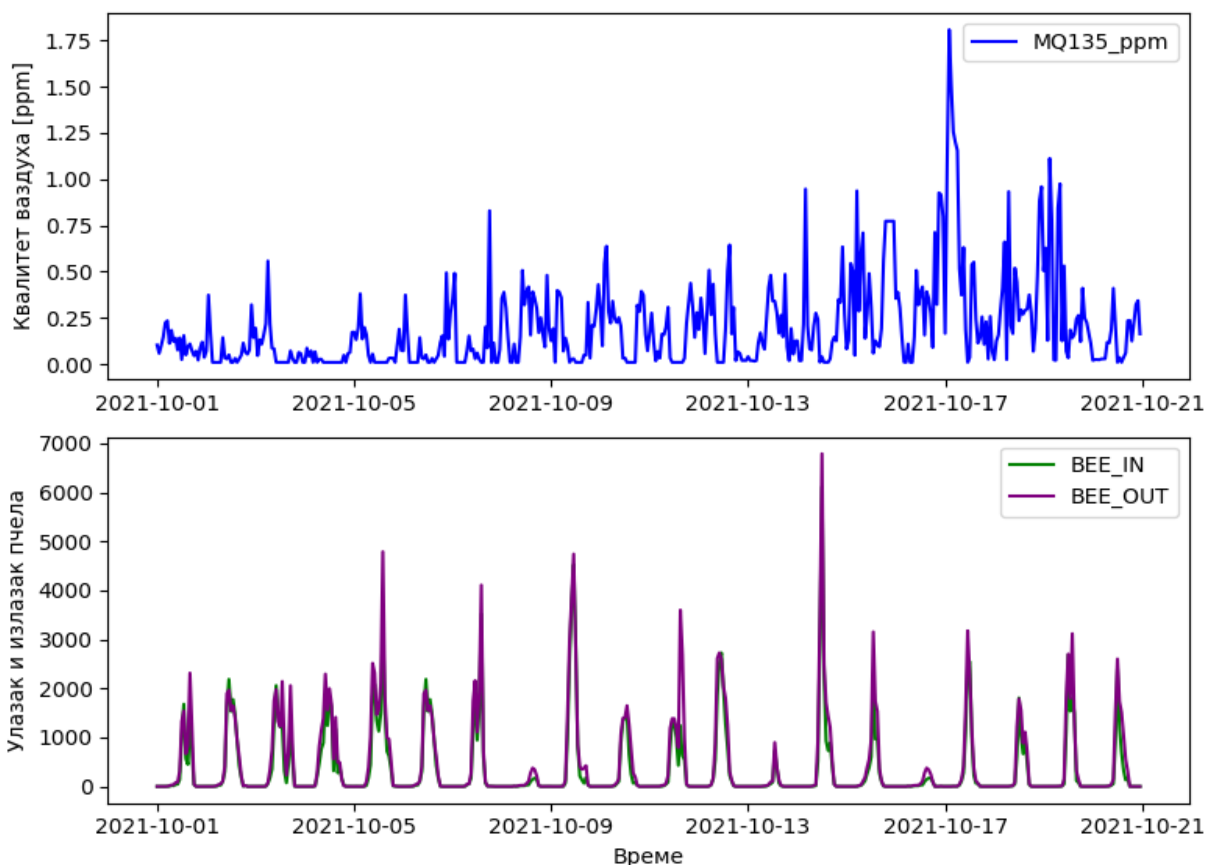


График 9.4 Упоредни тренд кретања пчела у односу на промене квалитета ваздуха (MQ135PPM) за период од 20 дана

График 9.4 показује измерене промене квалитета ваздуха, ове вредности треба повезати са присуством повећане влажности и осцилаторним променама температуре. У овом периоду се јављају ране јутарње магле, које утичу на кретање пчела што се може приметити у поређењу са данима без присуства магле (дани када су ниске вредности присуства кише и влажности ваздуха Графици 9.3 и 9.2). График тренда за 20 дана јасно потврђује тренд активности пчела у периоду када је параметар који описује присуство контаминације ваздуха у опадању. У овом случају присутност магле и велике влажности утиче на лошији квалитет ваздуха, тј. долази до повећања вредности параметра који указује на квалитет ваздуха. Корелације квалитета ваздуха са кретањима пчела приказана је у наставку (Табела 9.4).

Табела 9.4 Корелација вредности квалитета ваздуха у односу на кретање пчела IN/OUT

Корелација променљивих у односу на уласке и изласке пчела	Вредност корелације
Корелација квалитета ваздуха и уласка пчела:	-0.264
Корелација квалитета ваздуха и изласка пчела:	-0.274

На графику који приказује присуство вибрација (График 9.5) запажа се да у овом периоду нема значајнијих забележених вредности вибрација осим у неколико кратких момената.

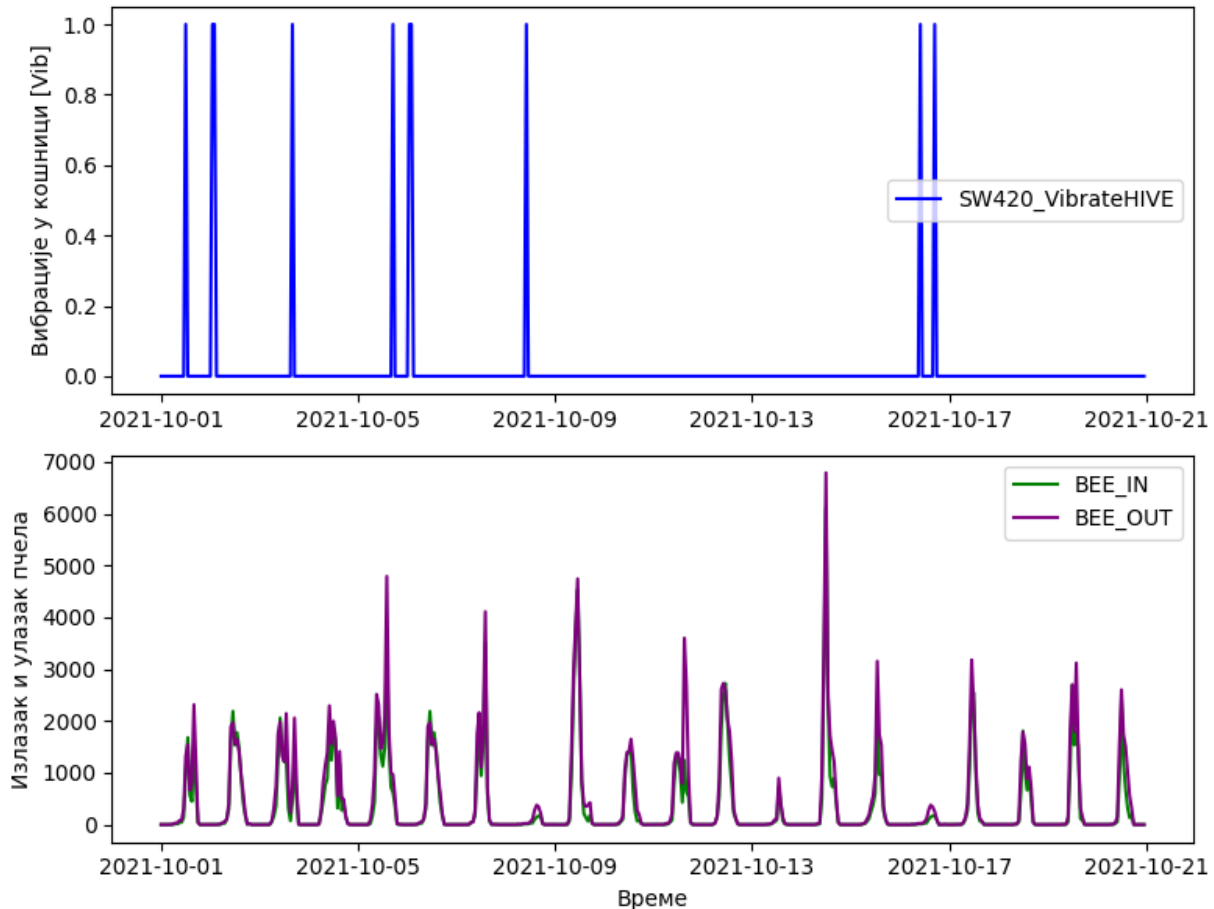


График 9.5 Упоредни тренд кретања пчела у односу на присуство вибрација (SW420) за период од 20 дана

Присуство вибрација за приказани период не даје пуно информација о корелацији са активностима пчела, јер су појаве измерених вредности периодичне. Овде треба узети у обзир разматрања кретања тренда за дуже временске периоде и повезати са кретањима вредности нивоа буке, звучне фреквенције или присуства кише. Јаке кише могу изазвати појаву вибрација, стога је потребна адекватнија анализа података.

Корелација ових параметара приказана је у следећој табели (Табела 9.5).

Табела 9.5 Корелација вредности присуства вибрација у односу на кретање пчела IN/OUT

Корелација променљивих у односу на уласке и изласке пчела	Вредност корелације
Корелација вибрација и уласка пчела:	0.001
Корелација вибрација и изласка пчела:	0.006

График који приказује вредности дневне светлости (График 9.6), указује на преклапање периода активности кретања вредности са пчелињим активностима. Наравно у питању је дневна светлост а пчеле су искључиво активне по дану. Вредности измерене дневне светлости, температуре и влажности ваздуха представљају важне узрочне факторе активности пчела, и могу указивати на обрасце кретања.

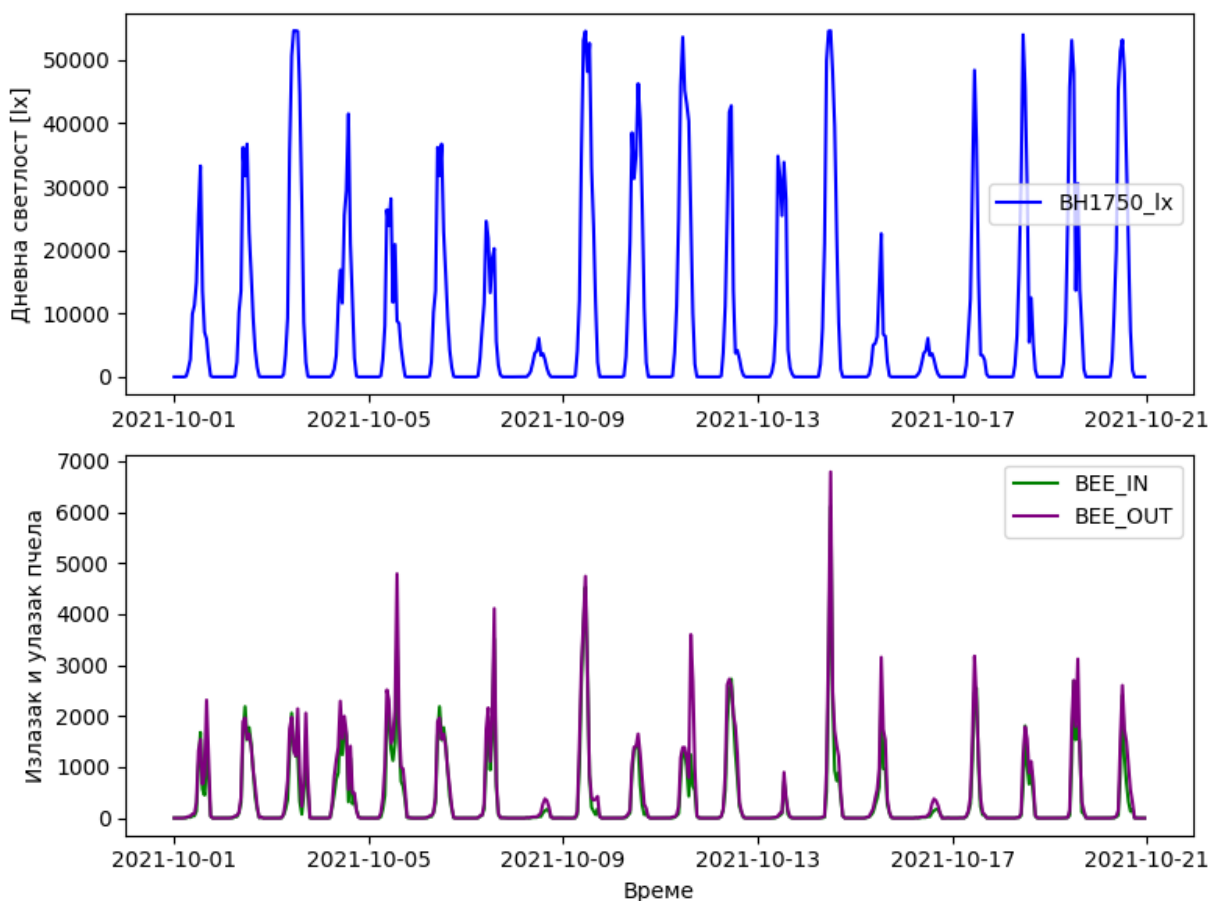


График 9.6 Упоредни тренд кретања пчела у односу на промене вредности дневне светлости (BH1750) за период од 20 дана

Корелација ових параметара приказана је у следећој табели (Табела 9.6).

Табела 9.6 Корелација вредности дневне светлости у односу на кретање пчела IN/OUT

Корелација променљивих у односу на уласке и изласке пчела	Вредност корелације
Корелација дневне светлости и уласка пчела:	0.722
Корелација дневне светлости и изласка пчела:	0.719

На Графику 9.6.1 могу се прецизније видети промене на дневном нивоу (за дане 08-09.10.2021).

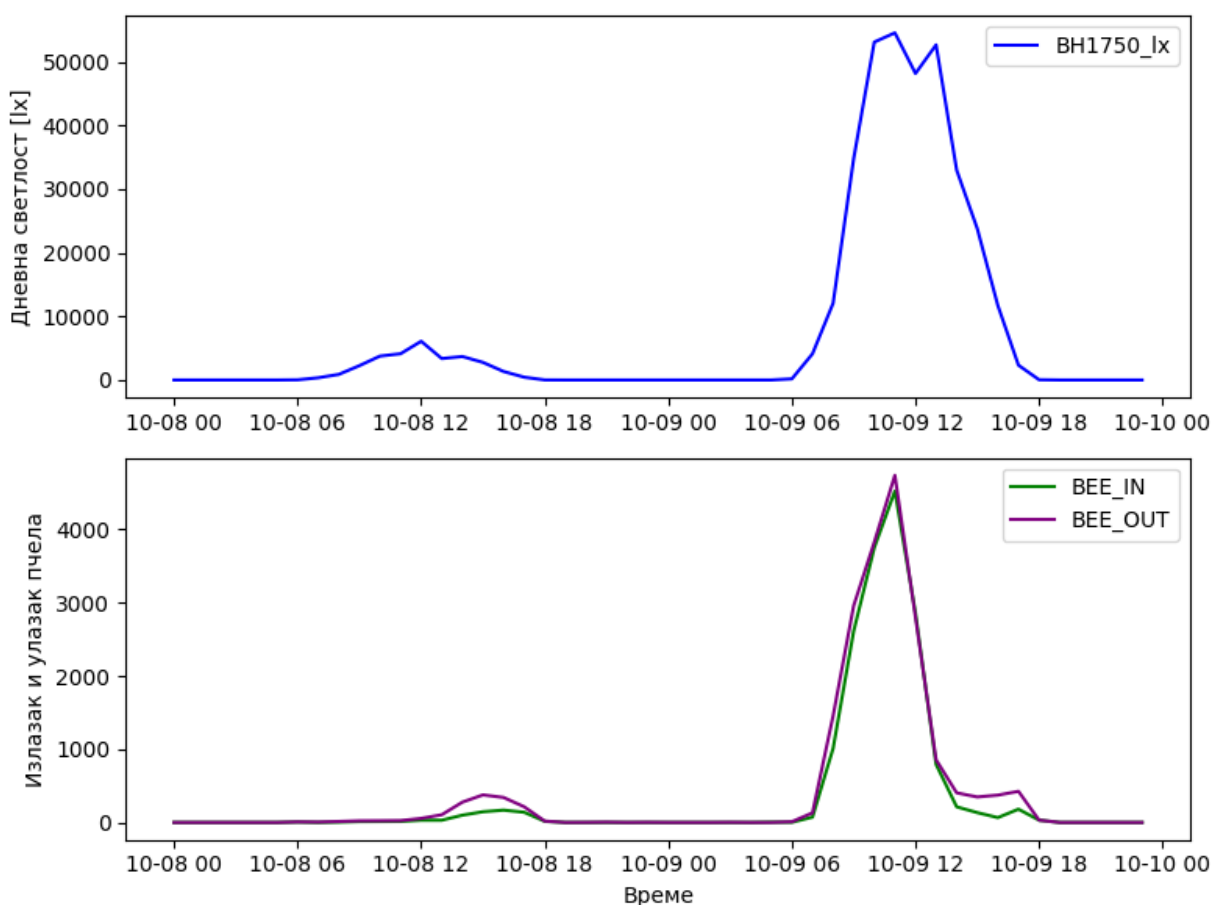


График 9.6.1 Упоредни тренд кретања пчела у односу на промене вредности дневне светлости на дневном нивоу

Када су у питању промене UV индекса (График 9.7), тренд ових промена се поклапа са периодом активности пчела. Уколико повежемо ове вредности са интензитетом дневне светлости приметимо еквивалентан тренд. Ове вредности нам такође могу указати на корелацију фактора и утицај на кретање пчела, тј. узрочне факторе.

Корелација ових параметара приказана је у Табели 9.7.

Табела 9.7 Корелација вредности UV индекса у односу на кретање пчела IN/OUT

Корелација променљивих у односу на уласке и изласке пчела	Вредност корелације
Корелација UV индекса и уласка пчела:	0.697
Корелација UV индекса и изласка пчела:	0.706

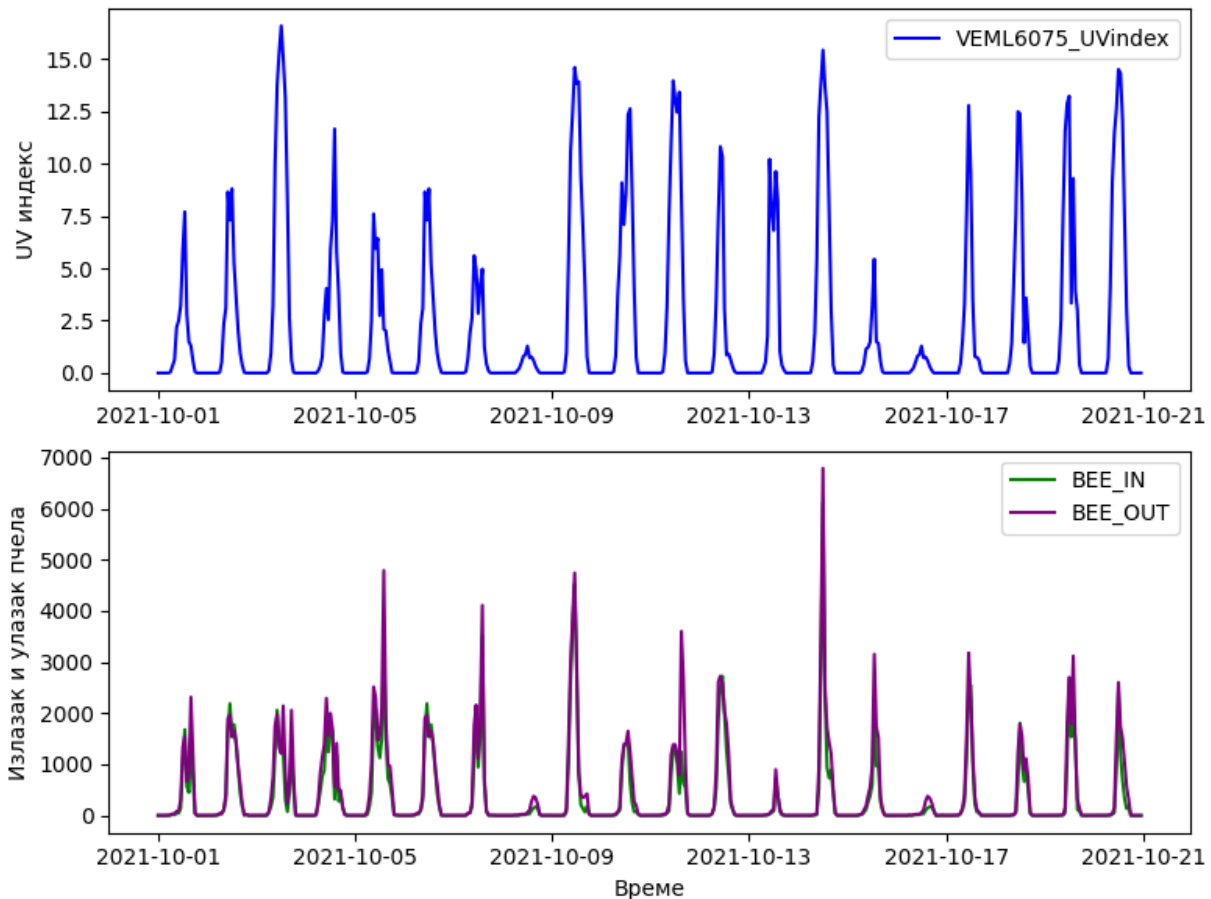


График 9.7 Упоредни тренд кретања пчела у односу на промене UV индекса (VEMML6075) пчела за период од 20 дана

Вредности инфрацрвеног зрачења и јачине дневне светлости (График 9.8) приказане на графику, такође указују на преклапање периода промена вредности са активностима пчела. Све вредности са сличним трендовима коришћене су при моделовању предиктивног алгоритма, што је појашњено у претходном сегменту. Овде су дата два параметра који имају различите интензитете дневних вредности али одлична поклапања са активностима пчела у току дана.

Корелација ових параметара приказана је у следећој табели (Табела 9.8).

Табела 9.8 Корелација вредности ИС зрачења и интензитета дневне светлости у односу на кретање пчела IN/OUT

Корелација променљивих у односу на уласке и изласке пчела	Вредност корелације
Корелација ИС зрачења и уласка пчела:	0.701
Корелација ИС зрачења и изласка пчела:	0.704
Корелација интензитета дневне светлости и уласка пчела:	0.709
Корелација интензитета дневне светлости и изласка пчела:	0.708

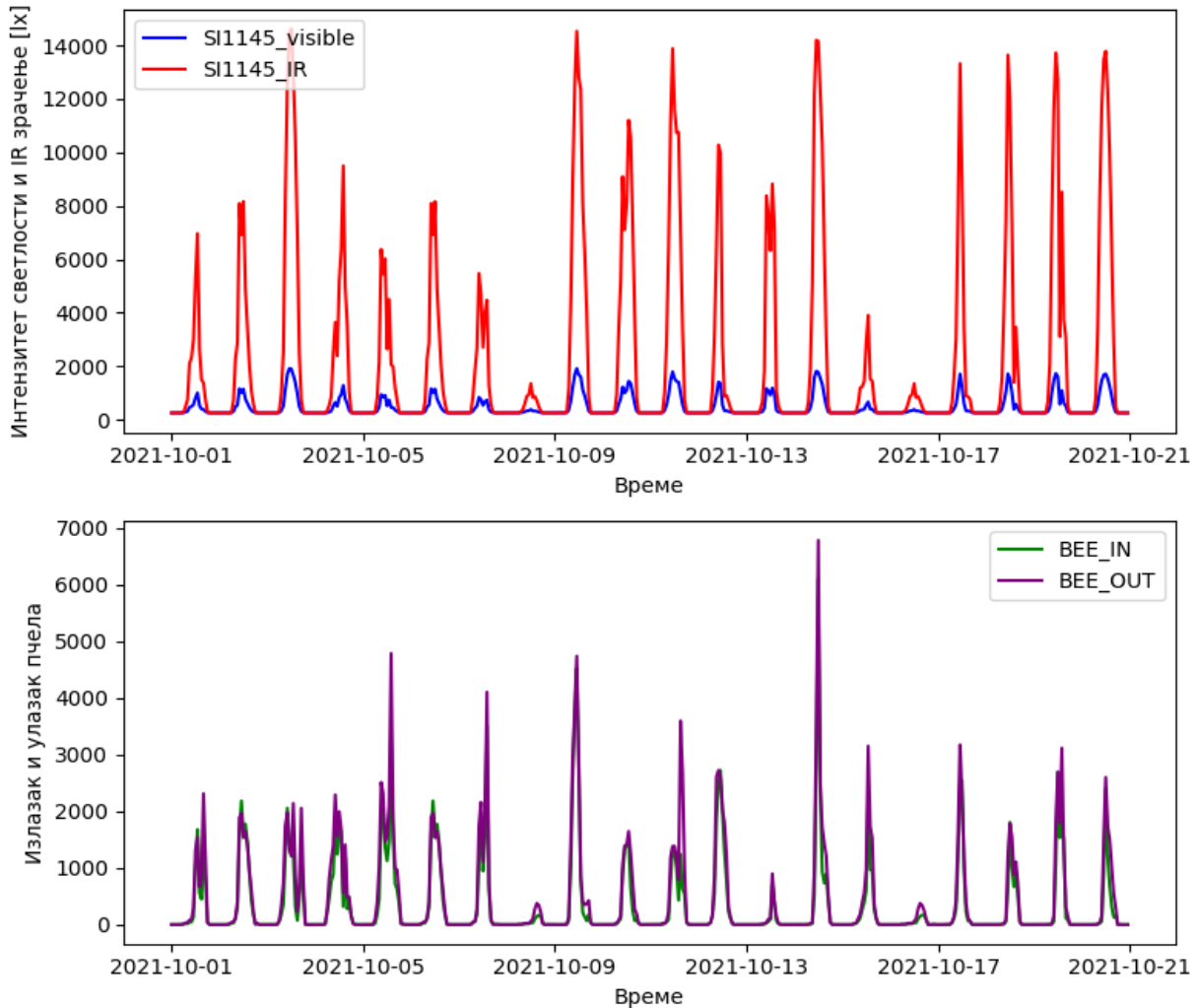


График 9.8. Упоредни тренд кретања пчела у односу на промене IR зрачења и интензитета дневне светлости (SI1145) за период од 20 дана

Промене брзине и јачине ветра за избрани период захтевају конверзију измерених вредности у вредности које се користе у метеорологији (брзина ветра приказана у односу m/sec и вредности Beaufortове скале за приказ јачине ветра). Обзиром да се за предикцију у овом случају користе просечне сатне вредности, упросечене вредности за брзину и јачину ветра не приказују верно утицај на кретање пчела. Анализа и интеграција ових вредности у формулу предикције из дисертације захтева посебан приступ, што је остављено за даља истраживања.

Звучна фреквенција и ниво буке такође захтевају другачије методе и приступ обраде и анализе, због специфичне структуре самих података. Свакако то спада у један део наставка тренутних и будућих истраживања. У овој секцији су само анализирани вредности које су узете за предикцију, кроз графике паралелним приказом за дневне активности и периоде од 20 дана, измерене са сензора ИоТ Система.

10. ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА

Захтеви савремених рачунарских система су брза обрада велике количине података, њихова анализа и изведене информације како би имали сврху у виду система за доношење одлука. Наведене технологије омогућавају прикупљање велике количине података у кратком периоду, њихову анализу, изведено стање, како би се на основу добијених резултата могли разумети обрасци понашања, симулација, предикција, трендови кретања итд. *Data Science* представља најбитнију мултидисциплинарну област која се фокусира на екстракцију, анализу и интерпретацију корисних информација из података. Ова грана науке комбинује принципе математике, статистике, програмирања и доменског знања како би се разумели сложени проблеми засновани на подацима и извукла практична сазнања. Вештачка интелигенција је круна рачунарских система последњих година и део свих система човековог деловања и окружења.

Вештачка интелигенција (ВИ), из угла комплексности могла би да се дефинише као интердисциплинарни научни приступ посвећен креирању алгоритама и система за обраду информација, укључујући читаву *Data Science* област: истраживачку анализу података (Exploratory Data Analysis), статистичку анализу, дубоку анализу, визуелизацију података, теорију рачунарства, машинско учење, контролу и управљање. Циљ вештачке интелигенције је да креира рачунарске системе који могу да искажу интелигенцију и капацитет за доношење одлука сличних онима које човек доноси. Интелигентно понашање се односи на способност рачунарског система да учи, закључује, прилагођава се, решава проблеме и евентуално показује креативност.

На почетку је битно направити разлику између вештачке интелигенције, машинског учења и дубоког учења (Слика 10.1). Укратко, вештачка интелигенција се односи на шире поље које има за циљ стварање интелигентних рачунарских система. Машинско учење је техника у оквиру вештачке интелигенције која омогућава рачунарима да уче из података. Дубоко учење је специфична техника машинског учења која се ослања на вештачке неуронске мреже и користи се за обраду сложених података.

У ширем смислу, вештачка интелигенција се односи на област рачунарства која се бави развојем система и алгоритама који имају способност да симулирају интелигентно понашање. Циљ вештачке интелигенције је да омогући машинама да обављају задатке који обично захтевају људску интелигенцију, као што су препознавање слика, разумевање природног језика, доношење одлука и решавање проблема.

Машинско учење је грана вештачке интелигенције која се фокусира на развој алгоритама и техника које омогућавају рачунарима да уче из података, без експлицитног програмирања. Ова техника омогућава рачунарима да идентификују обрасце у подацима и да предвиђају или доносе одлуке на основу тих образаца. Машинско учење се дели на учење под надзором, учење без надзора и појачано учење, у зависности од врсте података и циља алгоритама.

Дубоко учење је подграна машинског учења која се заснива на вештачким неуронским мрежама. Ове мреже имају сложену структуру инспирисану неуронским везама у људском мозгу. Дубоко учење је посебно ефикасно у анализи великих количина података, као што су слике, звукови или текстови. Дубоко учење се користи за препознавање објеката, превод језика, генерисање текста, препоруке производа и многе друге задатке. Карактерише га способност да се науче репрезентације података без експлицитног програмирања специфичних карактеристика.



Слика 10.1 Однос вештачке интелигенције, машинског учења и дубоког учења

Неизоставност вештачке интелигенције у ИТ-у долази из њених способности да аутоматизује рутинске и напорне процесе, као и да пружи тачније и брже резултате. Вештачка интелигенција такође помаже у повећању продуктивности, смањењу трошкова и побољшању квалитета услуга. Примењује се у разним областима, укључујући препознавање говора, анализу слика, аутономност вожње, аутоматизоване системе трговине, персонализоване препоруке итд. ВИ се користи за решавање проблема који се обично сматрају доменом људске интелигенције, као што су разумевање природног језика, препознавање узорака или стратегијско планирање. Такође, ВИ решава и многе оперативне проблеме који су изван домашаја људских способности, као што је обрада великих количина података или извршавање задатака у милисекундама. Због тога је вештачка интелигенција постала део свакодневне ИТ културе и њен значај ће само расти у будућности. Најбољи пример је примена напредних NLP модела chat botova, искорак који су направили у примени вештачке интелигенције. Предводник је chatGPT који је унео велике промене у ИТ свет.

ChatGPT је развијен од стране организације OpenAI, представља иновативни језички модел заснован на архитектури познатој као Generative Pre-trained Transformer (GPT). Као генеративни модел, GPT користи transformer архитектуру и трениран је на опсежној бази текстуалних података како би боље разумео и генерисао текст на природном језику. ChatGPT, као специјализована верзија овог модела, нарочито се истиче када је реч о генерисању природног језика у формату конверзације. Остали кључни модели за обраду природног језика (NLP) које је развила организација OpenAI укључују Llama, T5 (Text-to-Text Transfer Transformer) и Falcom.

Llama, у суштини, представља NLP модел намењен за разумевање и генерисање језика. Издвојен је својом способношћу да пружи корисницима већи ниво контроле над генерисаним одговорима, омогућавајући им да дефинишу тон и стил комуникације. T5 (Text-to-Text Transfer Transformer), с друге стране, представља свеобухватни модел за обраду текста способан да се суочи са различитим NLP задацима. Захваљујући тренингу на огромном скупу података обухватајући различите задатке, овај модел је оспособљен за извршавање разноврсних задатака попут превођења, сумирања текста, одговарања на питања и слично. Falcom је NLP модел који се такође користи за разумевање и генерисање језика. Својим јединственим карактеристикама и применама, Falcom је дизајниран да пружи супериорне функционалности у контексту обраде природног језика.

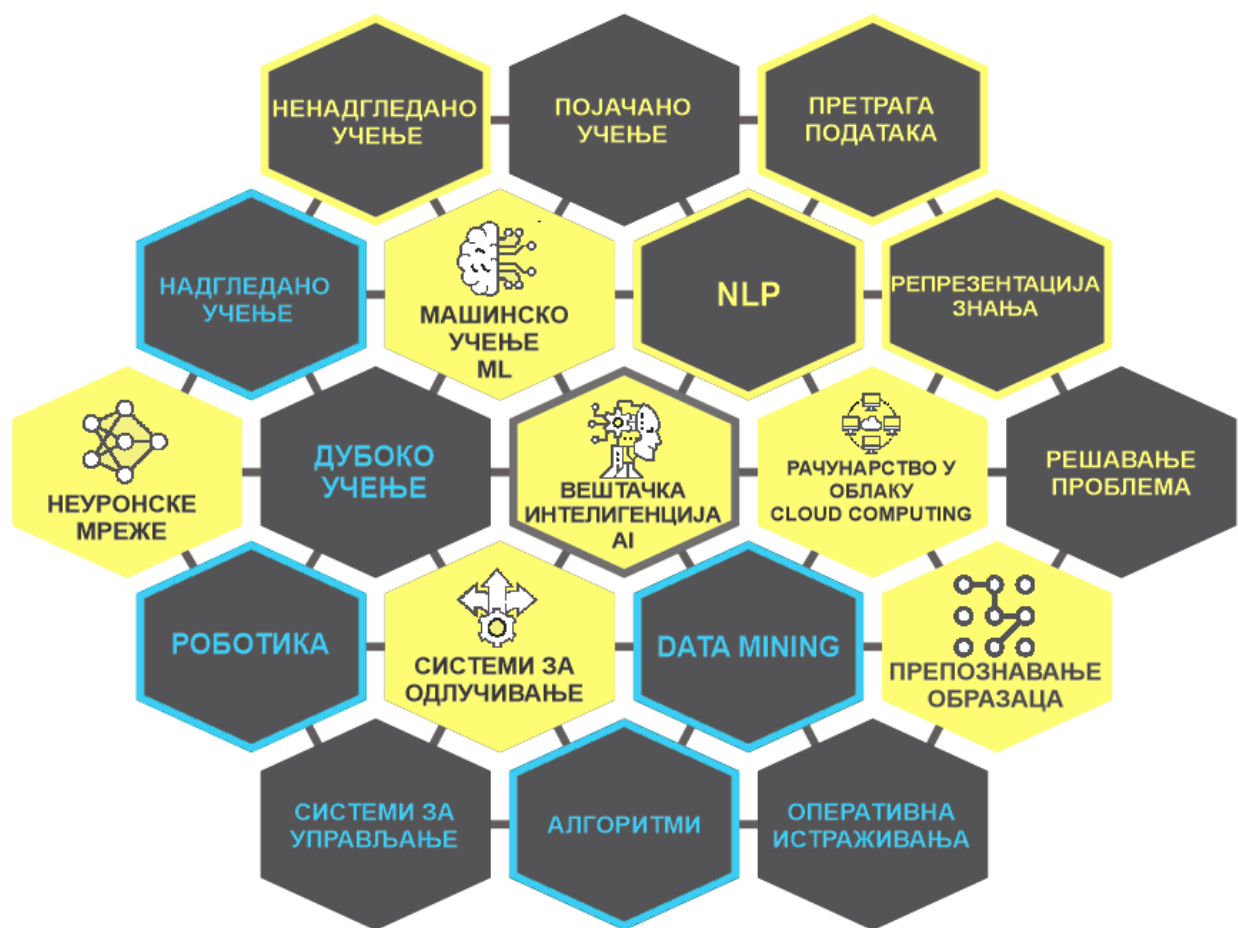
Ово су само неки од модела које је OpenAI развио као део својих истраживања у области NLP-а. Постоје и друге компаније које су развиле своје моделе као одговор, неки од њих су:

- BERT (Bidirectional Encoder Representations from Transformers) је један од најутицајнијих модела у NLP-у, који је развила компанија Google. BERT користи двосмерно учење како би боље разумео контекст речи у реченици.
- RoBERTa је модел који је Facebook AI развио на основу BERT-а, с циљем да унапреди његове перформансе.
- ELECTRA је још један модел базиран на BERT-у, развијен од стране Google Research-а. Овај модел примењује алтернативни приступ тренирању, који га чини ефикаснијим у погледу ресурса.
- XLNet је модел који комбинује предности GPT-а и BERT-а. За разлику од BERT-а, који кодира контекст у два смера, XLNet кодира контекст у свим могућим пермутацијама, омогућавајући му да боље разуме језик.

10.1. Методе, модели и алати

Метода, модел, и алат су термини који се често користе у контексту вештачке интелигенције. Зашто је овакав преглед битан, највише због појашњења приступа у зависности од проблема и прикупљених података. Многи савремени приступи вештачке интелигенције комбинују различите методе како би се постигло оптимално решење за одређени проблем.

Метода представља алгоритам или процес који се користе за учење, закључивање и предвиђање. У контексту вештачке интелигенције, методе обично представљају технике приступа и типове алгоритама који се користе за тренирање модела, класификацију података или извођење других функционалности. На пример, методе дубоког учења користе неуронске мреже са много слојева (deep neural networks) за учење из великих количина података. Вештачка интелигенција је богато поље истраживања које обухвата много метода, алгоритама и приступа. У начелу, методе вештачке интелигенције могу се груписати у неколико категорија: симболичке методе, методе засноване на знању, еволуционарне методе, методе машинског учења и методе засноване на вероватноћи (Слика 10.2).



Слика 10.2 Методе примене вештачке интелигенције

Као што је речено, методе вештачке интелигенције подразумевају различите приступе који укључују примену неуронских мрежа, генетског програмирања или логичког резонавања. Осим ових метода, постоје и друге технике у вештачкој интелигенцији као што су fuzzy логика, еволуирајући алгоритми, и дубоко учење. Свака од ових метода има своје карактеристике, предности и ограничења, и њихов избор зависи од специфичног проблема који треба решити.

Модели су конкретна имплементација ових метода, прилагођена специфичном проблему. На пример, модел машинског учења може бити конфигуриран да предвиђа цену станова на основу података о величини, локацији, години изградње итд. Модел представља математичку репрезентацију неког система, процеса или феномена, односно алгоритамску имплементацију неке интелигентне функционалности, као што је класификација, регресија, анализа садржаја или препознавање облика. Модели се тренирају коришћењем великих скупова података да би се научили функционисати у одређеним задацима.

Један од најважнијих приступа у вештачкој интелигенцији су неуронске мреже. Неуронске мреже су инспирисане структуром мозга и састоје се од великог броја међусобно повезаних јединица, названих неурони, који обрађују и преносе информације. Неуронске мреже могу бити једноставне, сложене као што су рекурентне неуронске мреже и конволутивне неуронске мреже. Ове мреже се обучавају користећи различите технике учења, као што су надгледано учење, ненадгледано учење и дубоко учење, како би постигле жељене резултате.

Алат представља програмски пакет, софтвер или платформу који се користи за имплементацију модела вештачке интелигенције или извођење одређених функционалности. Алати се користе за тренирање модела, имплементацију функционалности вештачке интелигенције, или интеграцију модела у остале системе.

То у пракси захтева примену различитих софтверских платформи и алата. Неки од најпознатијих алата су TensorFlow и PyTorch, две снажне библиотеке за машинско учење које су широко прихваћене у индустрији и академској заједници. Обе библиотеке нуде подршку за широк спектар алгоритама машинског учења, укључујући дубоко учење, и имају јаке могућности за манипулацију подацима и визуелизацију. Keras је још једна библиотека која се користи за рад са неуронским мрежама. Она је интегрисана у TensorFlow и представља API интерфејс високог нивоа за TensorFlow, чинећи развој и тренинг модела дубоког учења интуитивнијим и лакшим за оне који су нови у овој области. За обраду и анализу података, Pandas и NumPy су кључни алати у екосистему Python за науку о подацима. Они омогућавају ефикасну манипулацију подацима и математичким операцијама на високом нивоу. Scikit-learn је још један користан алат, који пружа једноставне и ефикасне библиотеке за машинско учење, укључујући класификацију, регресију, груписање и редукцију димензионалности.

Када говоримо о платформама у облацима Google Cloud AI, Amazon SageMaker и Microsoft Azure Machine Learning су међу најпознатијима. Ове платформе нуде низ услуга које олакшавају развој, тренинг и имплементацију модела вештачке интелигенције на великом броју података.

Важно је нагласити да избор алата и метода зависи од специфичних захтева пројекта, укључујући природу проблема, величину и врсту података.

10.2. Примене вештачке интелигенције у ИоТ

Вештачка интелигенција (ВИ) и Интернет ствари (ИоТ) су два технолошка концепта која се комбинују и допуњују, како би се остварила smart функционалност, ефикасна повезаност и аутоматизација различитих уређаја и система. Примена вештачке интелигенције у ИоТ-у пружа могућности за анализу, интерпретацију и доношење одлука на основу великих количина података генерисаних од стране ИоТ уређаја. Ова синергија омогућава стварање интелигентних система који могу функционисати потпуно аутономно, и као таква могу пружити персонализована искуства корисницима.

Примењена област вештачке интелигенције у ИоТ-у је аналитика података. ИоТ уређаји генеришу велике количине података у реалном времену, коришћење техника машинског учења, дубоког учења и обраде природног језика (NLP) омогућава ефикасну анализу тих података. Алгоритми машинског учења могу идентификовати обрасце, израчунавати предикције, откривати аномалије и доносити одлуке на основу података прикупљених са ИоТ уређаја.

У системима интернета интелигентних уређаја (ИоТ), примена вештачке интелигенције се може поделити на три главне категорије, приступајући различитим слојевима система и обухватајући различите аспекте анализе и одлучивања.

1. **Примена ВИ на нижим слојевима.** У овом делу система ИоТ, вештачка интелигенција се примењује за обраду и анализу података које генеришу сензори и уређаји у Edge окружењу. Овде се користе различите технике машинског учења и дубоког учења за препознавање шаблона, класификацију, предвиђање и детекцију аномалија, анализу сигнала, препознавање објеката и претраживање података. Ово је слој за скупљање и обраду података пре него што се примени ВИ на вишим слојевима.
2. **Примена ВИ на средњим слојевима.** На овим слојевима система ИоТ, вештачка интелигенција се користи за различите апликације и намене. Ова категорија укључује комбинацију примене ВИ на вишим и нижим слојевима. Примери примене у овом слоју укључују препознавање говора и обраду природног језика, компјутерски вид препознавања и емоционалну интелигенцију. ВИ у овим слојевима помаже у стварању смисленог интерфејса између људи и ИоТ уређаја, пружајући могућност за комуникацију на природном језику и интеракцију на бази гласа.
3. **Примена ВИ у вишим слојевима.** На вишим слојевима система ИоТ-а, вештачка интелигенција се примењује за стварање система за доношење одлука који су способни за адаптацију у реалном времену. Овде се користе алгоритми појачаног (reinforcement learning) учења, еволуциони алгоритми за оптимизацију и предиктивну анализу. Ово укључује развој паметних агената који могу аутономно обављати праћење и анализу података, учење из искуства и доношење одлука на основу тренутног контекста и циља. На пример, ВИ може бити коришћена за развој

паметних система за праћење здравља, који пружају персонализовану медицинску негу и предлажу превентивне мере на основу података о здрављу.

Укључивање вештачке интелигенције у системе интернета интелигентних уређаја омогућава напредак у различитим аспектима, укључујући системе за мониторинг, бољу обраду података, унапређење комуникације са корисницима, предвиђање и оптимизацију. Ово отвара нове могућности за паметну аналитику, аутономне системе и унапређење корисничког искуства у различитим областима.

Примена вештачке интелигенције у ИоТ-у представља широк спектар бенефита који се постижу имплементацијом комплексних техника и алгоритама. Значај ВИ у ИоТ системима за доношење одлука, обраду и анализу великих података (Big Data) је изузетно важан и обухвата неколико кључних аспеката:

- **Аутоматизацију.** Примена ВИ у ИоТ системима омогућава аутоматизацију процеса анализе и обраде података. То доводи до бржег и тачнијег доношења одлука. Коришћење алгоритама машинског учења и дубоког учења омогућава системима да сами уче и прилагођавају се променљивим условима.
- **Предиктивно управљање.** Једна од кључних предности ВИ у ИоТ системима је могућност предвиђања будућих догађаја и трендова. Комбиновање података снимљених са ИоТ уређаја са алгоритмима за предиктивну анализу омогућава системима да идентификују обрасце и дају предвиђања о будућим догађајима. На основу ових анализа, систем може доносити информисане одлуке и предлагати оптималне акције.
- **Повећање ефикасности.** Примена ВИ у ИоТ системима омогућава брже и ефикасније разумевање и обраду података. Алгоритми машинског учења и дубоког учења могу ефикасно анализирати велике количине података и открити скривене обрасце и везе међу подацима.

Генерално гледано, ово имплицира да вештачка интелигенција чини неизоставан део у ИоТ системима чија је улога обрада и анализа великих података, омогућавајући тиме прецизније доношење одлука и повећање ефикасности у различитим инфраструктурама и системима.

10.3. Неуронске мреже

Неуронске мреже су један од основних модела у вештачкој интелигенцији и машинском учењу [92]. Као што је већ поменуто, основна јединица неуронске мреже је неурон. Сваки неурон прима улазе, множи их са одговарајућим тежинским коефицијентом, затим сумира све улазе и примењује активациону функцију на резултат. Активациона функција, попут сигмоидне, тангенс хиперболичне или ReLU функције, користи се за додавање нелинеарности модела, омогућавајући му да научи сложеније обрасце.

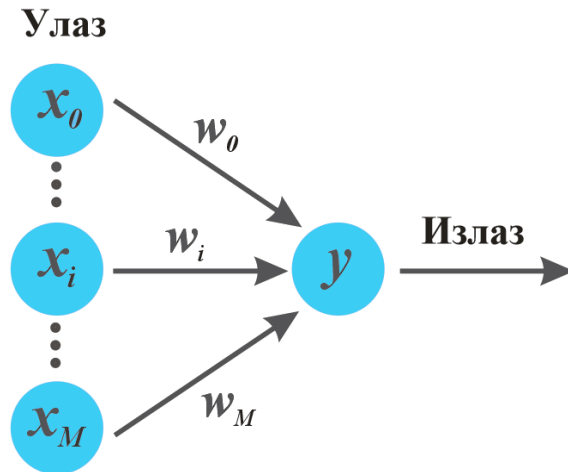
Неуронске мреже се могу класификовати на основу различитих критеријума, укључујући број слојева, правца преноса информација, повезаност неурона и друге.

Према броју слојева, неуронске мреже можемо поделити на једнослојне и вишеслојне:

- **Једнослојне неуронске мреже** (Слика 10.3), су једноставан облик неуронских мрежа који се састоји само од једног слоја неурона. Ови модели су погодни за решавање једноставних проблема који не захтевају сложене зависности између улаза и излаза. Математички, најједноставније можемо представити једнослојне неуронске мреже као:

$$y = f(W \cdot x + b) \quad (10.1)$$

где је x улазни вектор, W матрица тежинског коефицијента, b вектор одступања (bias), f активациона функција, а y излазни вектор. Једнослојне мреже су ефикасне у ситуацијама када је улазни простор једноставан и постоји линеарна зависност између улаза и излаза. Ове мреже се често користе за једноставне класификацијске проблеме и регресију.

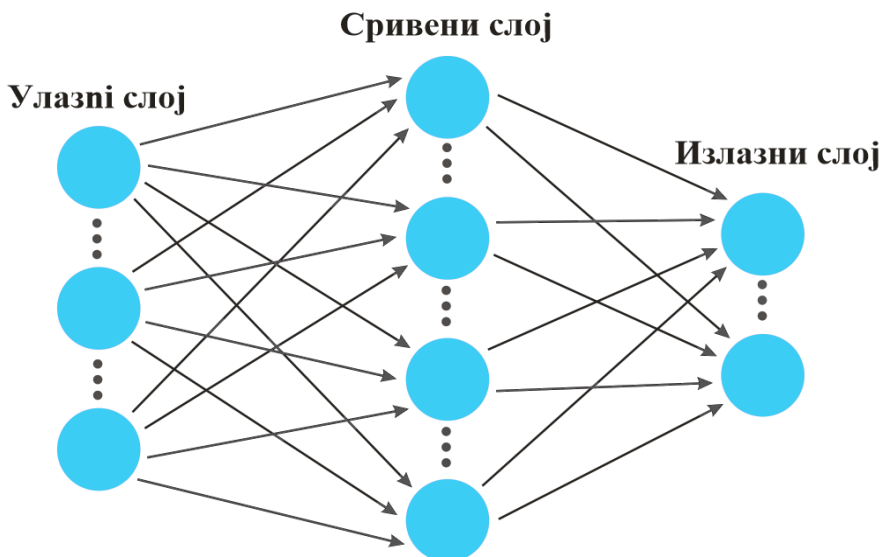


Слика 10.3 Једнослојна неуронска мрежа

- **Вишеслојне неуронске мреже** (Слика 10.4), су сложенији модели који се састоје од више слојева неурона, укључујући улазни, скривене и излазни слој. Ове мреже су флексибилне и способне да моделирају комплексне зависности између улаза и излаза. Вишеслојне мреже могу моделовати нелинеарне зависности и постићи висок степен прецизности у разним задацима као што су препознавање облика, обрада природног језика, препознавање објеката итд.

Разлог за примену вишеслојних неуронских мрежа је њихова способност да се прилагоде сложеним и неравномерним подацима. Скривени слојеви у овим мрежама омогућавају моделирање сложених структура у подацима. Такође, вишеслојне мреже користе поступак учења са пропацијом грешке уназад (backpropagation), који омогућава прилагођавање тежинског коефицијента како би се минимизовала грешка између

предвиђања и стварних вредности. Ова метода омогућава мрежи да научи репрезентације и карактеристике из података.



Слика 10.4 Вишеслојне неуронске мреже

У смислу правца преноса информација, постоје feedforward и рекурентне неуронске мреже:

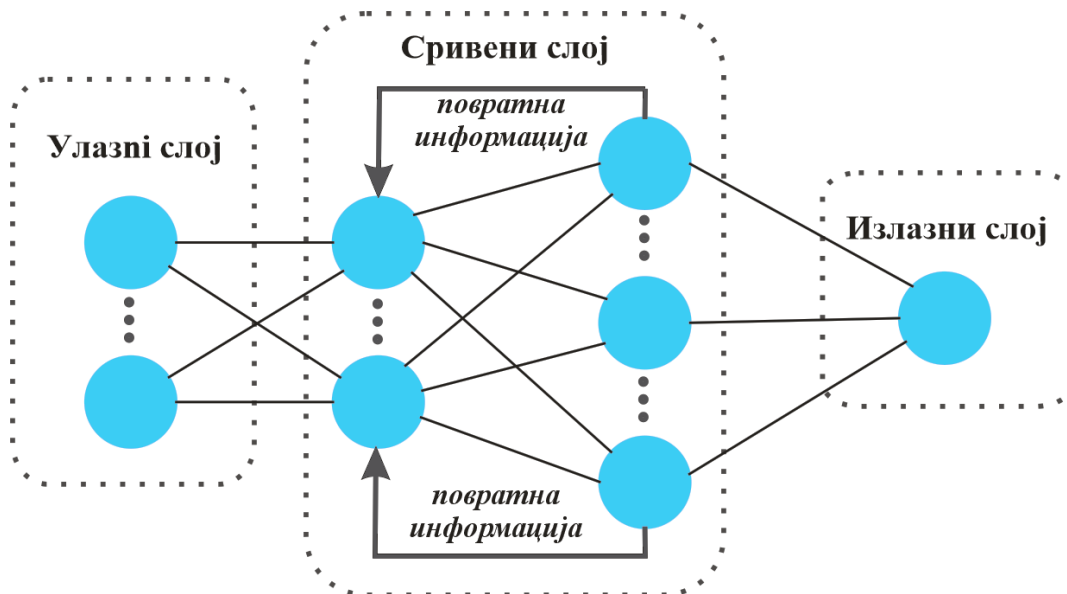
- **Мреже са пропагацијом унапред** (feedforward) су најосновнији облик неуронских мрежа, пренесе информације само унапред, од улазних до излазних слојева, без петљи. Ове мреже се често користе за задатке као што су класификација, регресија и препознавање облика. Оне су погодне када је улазни простор фиксне димензије и не постоји потреба за повратном везом.
- **Рекурентне неуронске мреже** (Слика 10.5) укључују петље које дају могућност да се информације преносе уназад, омогућавајући моделу да запамти претходне информације, приликом обраде тренутног корака. То им омогућава да узимају у обзир контекст и зависности које се јављају током времена. Ова особина чини рекурентне мреже погодним за обраду секвенцијалних података, попут временских серија или природног језика. Ове мреже могу учити сложене обрасце и узајамне везе између елемената секвенце, чиме су посебно ефикасне у задацима где је контекст кључан фактор. Математички, то се може представити као:

$$h_t = f(W_{hx} \cdot x_t + W_{hh} \cdot h_{t-1} + b) \quad (10.2)$$

$$y_t = g(W_{hy} \cdot h_t + b_y) \quad (10.3)$$

где је h_t скривено стање у тренутку времена t , f активациона функција за скривено стање, W_{hx} матрица тежинског коефицијента између улазног слоја и скривеног слоја,

W_{hh} матрица тежинског коефицијента између претходног скривеног стања и тренутног скривеног стања, W_{hy} матрица тежинског коефицијента између скривеног слоја и излазног слоја, x_t улазни податак у тренутку t , b , b_y и c вектори одступања (bias), а g активациона функција за излаз.



Слика 10.5 Принцип рада RNN мреже

Према врсти података поред рекурентних неуронских мрежа, постоје и конволуционе неуронске мреже (engl. Convolutional Neural Networks, скраћено CNN). Оне су посебна врста неуронских мрежа дизајниране за обраду података који имају просторну структуру, као што су слике или видео снимци. Оне су један од најефикаснијих алата за анализу визуелних података и постигле су изванредне резултате у областима као што су препознавање објеката, сегментација слика, детекција и класификација. Основни елементи конволутивних неуронских мрежа су конволуциони слојеви (eng. convolutional layers), који користе операцију конволуције како би извршили локално дељење тежинских фактора. Конволуција у контексту CNN-а је математичка операција која комбинује улазне податке са сетом филтера како би извршила просторну конволуцију. Филтери (такође названи језгра, филтери сазнања или кернели) су матрице параметара које се примењују на локалне регионе улазних података како би се израчунала конволуција.

Конволуциони слојеви омогућавају да CNN научи локалне обрасце и карактеристике у подацима, без обзира на њихову тачну позицију у простору. Ови слојеви често користе активационе функције попут ReLU (Rectified Linear Unit) за додавање нелинеарности у мрежу. Након конволуције, обично се примењује слој за агрегацију (eng. pooling layer) који смањује димензионалност и просторну резолуцију података, чиме се постиже инваријантност на translацију у улазним подацима. Осим конволуционих слојева, CNN-ови често садрже потпуно повезане слојеве (eng. fully connected layers) који комбинују информације из претходних слојева и извршавају класификацију или регресију. Потпуно

повезани слојеви се обично користе у завршном делу CNN-а, где се израчунава излазна класификација или континуална вредност.

Генеративно супарничке мреже (Generative Adversarial Networks GANs) су напредни модел у којем се две неуронске мреже - генератор и дискриминатор - такмиче једна против друге. Генератор је неуронска мрежа која има задатак да генерише нове податке. На почетку тренинга, генератор обично генерише случајне податке који не изгледају реалистично. Током тренинга, циљ генератора је да научи како да генерише податке који су што сличнији стварним подацима из скупа за тренинг. Генератор прима неки вектор случајних бројева, који се назива латентни простор или шум, и користи га за генерисање података. Дискриминатор, с друге стране, је неуронска мрежа која има задатак да разликује између стварних података из скупа за тренинг и података које генерише генератор. Дискриминатор добија као улаз податке, било стварне или генерисане од стране генератора, и врши класификацију да ли су подаци стварни или вештачки. Циљ дискриминатора је да што тачније разликује између ова два типа података. Током тренинга, генератор и дискриминатор се међусобно надмећу. Генератор настоји да генерише податке који требају што боље преварити дискриминатор, док дискриминатор настоји да их што прецизније класификује између стварних и генерисаних података. Овај процес се одвија кроз итерације, где се генератор и дискриминатор тренирају заједно, а њихове перформансе се побољшавају током времена.

Овај приступ је користан за генерисање нових података који личе на оригинални скуп података, попут генерисања нових слика.

Неуронске мреже показују значајан потенцијал у многим областима, њихова примена захтева пажљиву анализу података. Избор архитектуре, број слојева, број неурона у сваком слоју, активационе функције, методе оптимизације и други хиперпараметри су кључни фактори који утичу на перформансе модела. Поред тога, неуронске мреже захтевају велике количине података за учење, и могу бити хардверски захтевне.

Основна структура неуронске мреже укључује три типа слојева:

- **Улазни слој** је први слој у неуронској мрежи и његова улога је да прима податке. Сваки елемент у улазном слоју представља један атрибут или карактеристику података које желимо да обрадимо или анализирамо. На пример, у задатку препознавања слике, сваки елемент улазног слоја може представљати пиксел слике. Улазни слој не врши никакву обраду или трансформацију података, он само прослеђује податке скривеним слојевима на даљу обраду. Број елемената у улазном слоју одређен је димензијама улазних података.
- **Скривени слојеви** су слојеви између улазног и излазног слоја. Ови слојеви обављају обраду и трансформацију података како би научили карактеристике из података. Број скривених слојева и број неурона у сваком слоју могу се прилагодити специфичном задатку и сложености проблема. Дубоке неуронске мреже, које имају више скривених слојева, користе се у дубоком учењу за моделовање сложених функција и представљања података. Сваки неурон у скривеном слоју прима улазе од

свих неурона у претходном слоју, комбинује те улазе користећи тежински коефицијент и примењује функцију активације да генерише излаз. Овај излаз се затим прослеђује на следећи слој.

- **Излазни слој** је последњи слој у неуронској мрежи и даје коначно предвиђање или одлуку мреже. Број неурона у излазном слоју зависи од броја класа у задатку, класификације или од димензионалности излазних података у другим задацима. Сваки неурон излазног слоја прима улазе од свих неурона у последњем скривеном слоју, комбинује те улазе користећи тежински коефицијент и примењује одговарајућу функцију активације да генерише коначни излаз.

Основна структура неуронске мреже са овим слојевима омогућава ефикасну обраду података и израчунавања предвиђања или доношење одлука. Подешавање броја скривених слојева и броја неурона у сваком слоју омогућава прилагодљивост неуронске мреже за различите проблеме и сложеност.

Функционисање неуронских мрежа има следећи ток: када подаци улазе у мрежу, они пролазе кроз сваки слој, од улазног до излазног слоја. У сваком неурону, улаз се множи тежинским коефицијентом везе и затим се све те вредности сумирају. Ова сума се затим прослеђује кроз активациону функцију, која одређује излаз неурона. Овај процес се понавља кроз све слојеве мреже. На крају, мрежа даје крајње предвиђање на основу вредности неурона у излазном слоју.

Да би неуронска мрежа могла да даје тачне предикције, она мора да се обучи. Обука се састоји од прилагођавања тежинског фактора веза између неурона на основу разлике између стварног и предвиђеног излаза (познатог као "грешка"). Ово се обично постиже кроз алгоритам назван пропација уназад (backpropagation) и оптимизацију која користи методе као што је стохастички градијентни спуст (Stochastic Gradient Descent, SGD). У пропацији уназад, грешка се прво израчунава на излазном слоју, а затим се "пропагира" уназад кроз мрежу, прилагођавајући тежински коефицијент на основу тога колико су допринеле грешци. Оптимизација користи ове грешке да ажурира тежине веза у мрежи, с циљем минимизације укупне грешке. SGD је један од најчешће коришћених алгоритама за оптимизацију, мада постоје и друге методе, као што су Adam и RMSprop.

Споменуте су активационе функције раније. Оне су кључне за рад неуронских мрежа, јер оне додају нелинеарност у мрежу, што омогућава мрежи да научи сложеније обрасце. Најчешће коришћене активационе функције су: ReLU (Rectified Linear Unit), сигмоидна функција и тангенс функција.

10.3.1. Рекурентне неуронске мреже (Recurrent Neural Network RNN)

Рекурентна неуронска мрежа (Recurrent Neural Network RNN) је тип неуронске мреже дизајнирана за обраду података који имају временску или секвенцијалну структуру, као што су временске серије, природни језик, говорни сигнали и остали контекстуални подаци [93]. RNN користи рекурзију да би претходне информације учиниле доступне у тренутном стању.

Основни елемент рекурентне неуронске мреже је рекурентни слој (engl. recurrent layer), који има способност да одржава стање (eng. hidden state) и преноси информације из претходних корака на следеће. Рекурентни слој користи повратне везе (eng. feedback connections) како би омогућио мрежи да има меморијску компоненту, чиме се моделује зависност између података у секвенцијалном контексту.

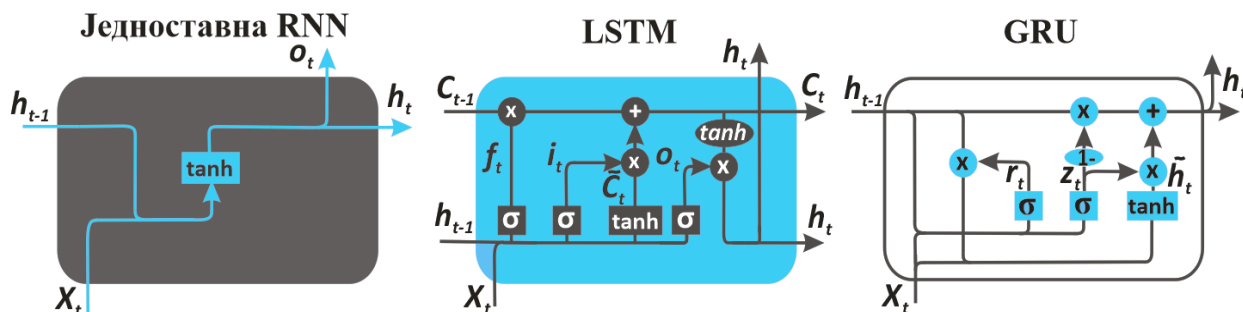
Неуронске мреже су структуриране тако да врше обраду података унапред, без референци на претходни улаз или будуће излазе. Међутим, у многим практичним применама, као што су обрада природног језика, временске серије или било која врста података где је временска зависност важна, овај приступ није адекватан. Стога су RNN конструисане да задрже "меморију" претходних улаза у своју унутрашњу структуру, омогућавајући им да моделују временске зависности. RNN су такође добре за обраду података који имају дугачке зависности, као што су текстови са дугачким реченицама и временским серијама са дуготрајним трендовима.

Сваки неурон у рекурентној неуронској мрежи прима улазне податке из тренутног корака и информације из претходног корака. То омогућава да се претходна стања мреже памте и користе за боље разумевање тренутног корака у секвенци. Једна од карактеристика RNN-а је њихова способност да запамте информације из претходних елемената секвенце и користе их за обраду тренутног елемента секвенце. RNN се састоје од једног или више слојева. Сваки слој има више неурона, који обрађују улазне податке и преносе их на следећи слој. Јединственост RNN лежи у томе што поседују интерне петље, што омогућава да се информације преносе кроз време, што је изразито корисно код обраде секвенцијалних података.

RNN се обучавају путем уназадне пропагације грешке и алгоритама као што су градијентни спуст или његове варијације. Међутим, овај приступ има своје слабости. Један од главних проблема с којима се RNN суочава је нестајући или експлодирајући градијенти током повратног ширења кроз време, што отежава учење дугорочних зависности. То је ситуација када информације које се преносе кроз мрежу постају или премале и нестају, или превелике и узрокују нестабилност. Овај проблем је посебно изражен код дугих секвенци података. Градијент у неуронским мрежама представљају вредности које се користе за оптимизацију и прилагођавање мрежних параметара током процеса учења. Неуронске мреже користе алгоритме градијентног спуштања да би минимизирале грешку између стварних и предвиђених вредности. Градијент пружа информације у ком правцу и колико треба ажурирати параметре да би се грешка свела на минимум.

Како би се превазишли ови изазови, развијени су различити типови RNN, укључујући Long Short-Term Memory (LSTM) и Gated Recurrent Units (GRU). LSTM мреже користе специјалне врсте ћелија уместо стандардних неуронских јединица, које омогућавају мрежи да ефикасно учи дугорочне зависности. Слично томе, GRU ћелије користе поједностављен механизам капије за контролу протока информација, што олакшава учење и израчунавање.

Дакле, као што је наведено, постоје три главна типа рекурентних неуронских мрежа: једноставна (simple RNN), затворена рекурентна јединица (Gated Recurrent Units GRU) и Long Short-Term Memory (LSTM). Разлика међу њима приказана је на слици 10.6.



Слика 10.6 Мрежне структуре RNN, LSTM, и GRU.

Три главна типа рекурентних неуронских мрежа су:

1. **Vanilla RNN**, је најосновнији облик RNN-а, у којем излаз из претходног корака улази у следећи корак, познатије као једноставне рекурентне неуронске мреже (Simple RNN).
2. **Long Short-Term Memory (LSTM)** је врста RNN-а који укључује меморијске ћелије које могу задржати информације дуже време, што их чини посебно кориснима за обраду дугих секвенци података.
3. **Gated Recurrent Units (GRU)** је врста RNN-а слична LSTM-у, али има једноставнију структуру, што може убрзати учење и обраду.

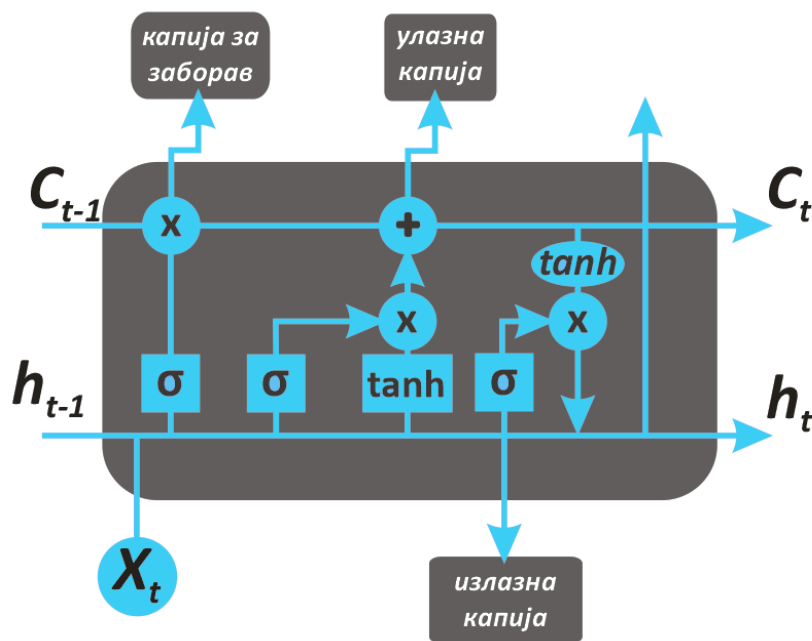
Једноставне рекурентне неуронске мреже (Simple RNN) су прве рекурентне мреже које су се користиле за обраду секвенцијалних података. Међутим, проблем са једноставним RNN је да имају потешкоће у памћењу дугорочних зависности између елемената секвенце. Рекурентне неуронске мреже са дугорочном зависношћу (LSTM) су осмишљене да реше проблем дугорочне зависности у једноставним RNN. LSTM мреже имају механизме за одржавање стања (eng. state) које се може одржавати кроз више временских корака. Такође, имају механизме за заборављање информација које нису битне за тренутно стање. Gated Recurrent Units (GRU) је новији тип RNN који је дизајниран да буде једноставнији и ефикаснији од LSTM-а, али са сличним способностима одржавања дугорочне зависности. GRU мреже користе "капије" за контролу тока информација у неуронској мрежи, што им омогућава да селективно забораве или задрже информације.

Треба напоменути да, иако су RNN моћан алат за обраду података, оне такође захтевају одабир одговарајуће архитектуре и подешавање хиперпараметара који су кључни за постизање добрих перформанси. Негативне стране примене RNN-а су њихове склоности да забораве старије информације и њихов захтев за великом количином података за тренирање, што често може да доведе до сложености мреже.

10.3.2. Long Short-Term Memory LSTM

LSTM (Long Short-Term Memory) неуронске мреже, као што је претходно објашњено су специфичан тип рекурентних неуронских мрежа које су ефикасне при обради секвенцијалних података, временских серија и проблема са задатком задржавања информација [94]. Овај модел је направљен за решавање проблема које класичне неуронске мреже не могу да реше, попут памћења информација за дуге периоде времена, проблема нестајања и експлодирања градијената.

LSTM мреже се састоје од низа ћелија, које су повезане међусобно и користе се да би се запамтили подаци из прошлости како би се искористили у тренутној обради. Свака ћелија представља комплексну структуру са више слојева. LSTM ћелије садрже капије, које омогућавају улаз, излаз и заборав информација које више нису битне. Ова својства чине LSTM модел јединственим и омогућавају му да ефикасно управља дуготрајним информацијама. Да бисмо схватили како LSTM мреже функционишу, прво ћемо се детаљније фокусирати на рад једне LSTM ћелије (слика 10.7). Главна компонента LSTM мреже је LSTM ћелија. На улазу, LSTM ћелија прихвата улазне податке x_t , тренутно ћелијско стање $C_{(t-1)}$ и претходно скривено стање $h_{(t-1)}$ и генерише ново ћелијско стање C_t и ново скривено стање h_t .



Слика 10.7 LSTM ћелија

Ток догађаја подразумева следеће пролазе:

1. **Улазна капија и нови кандидат (Input gate)**, ова два корака раде паралелно. Улазна капија користи сигмоидну функцију да одлучи које информације ће бити ажуриране. За ову сврху се користи \tanh активацијска функција да генериша нове вредности које би могле бити додате у ћелијско стање.

Формуле за улазну капију:

$$\text{Сигмоидна функција: } i_t = \sigma(W_i \cdot [h_{(t-1)}, x_t] + b_i) \text{ и} \quad (10.4)$$

$$\text{Хиперболичка тангента: } \tilde{C}_t = \tanh(W_c \cdot [h_{(t-1)}, x_t] + b_c) \quad (10.5)$$

$$\text{Улазна капија: } C_t = i_t \cdot \tilde{C}_t \quad (10.6)$$

2. **Капија за заборављање (Forget gate)**, је први корак у LSTM ћелији. Ова капија користи сигмоидну активацијску функцију за генерисање вредности између 0 и 1. Ове вредности представљају степен до кога ће информације бити задржане или одбачене. Вредност 0 значи да ће информација бити потпуно одбачена, док вредност 1 значи да ће информација бити потпуно задржана.

Формула за капију заборава је:

$$f_t = \sigma(W_f \cdot [h_{(t-1)}, x_t] + b_f) \quad (10.7)$$

3. **Ажурирање ћелијског стања**, се обавља тако што ћелијско стање множи са капијом заборава (одбачене су информације које нису битне) а затим се додају нове информације које су скалиране помоћу улазног врата.

Формула за ажурирање ћелијског стања је:

$$C_t = f_t \cdot C_{(t-1)} + i_t \cdot \tilde{C}_t \quad (10.8)$$

4. **Излазна капија и излазно стање (Output gate)**, где излазна капија користи сигмоидну функцију да одлучи које информације ће бити испоручене на излазу. Излазно стање је затим генерисано тако што се ћелијско стање пропусти кроз \tanh функцију (за нормализацију вредности између -1 и 1) а затим се множи са излазном капијом.

Формуле за излазну капију и излазно стање су:

$$o_t = \sigma(W_o \cdot [h_{(t-1)}, x_t] + b_o) \text{ и} \quad (10.9)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (10.10)$$

где су: σ (сигмоидна функција) и \tanh (хиперболички тангент) активационе функције, C_t ћелијско стање у тренутном временском кораку, W_i, W_f, W_o, W_c су матрице тежина, b_i, b_f, b_o, b_c су вектори одступања (bias), x_t је улаз у претходном временском кораку, h_t скривено стање у тренутном временском кораку, а $h_{(t-1)}$ је претходно стање.

Архитектура LSTM мреже састоји се од низа ових ћелија. Улазни подаци се пропуштају кроз сваку ћелију редом, са ажурирањем ћелијског стања и скривеног стања на сваком кораку. Излаз из једне ћелије постаје улаз у следећу, што омогућава LSTM мрежи

да памти информације кроз време. Ова способност памћења је оно што LSTM мреже чини моћним за обраду секвенцијалних података. Оне могу да уче везе између елемената који су удаљени у времену, што је тешко за традиционалне неуронске мреже.

LSTM метода има широк спектар примене, нарочито у случајевима где је потребан дуготрајни запис информација, укључујући:

- **Предвиђање временских серија.** LSTM мреже се често користе за предвиђање временских серија, као што је предвиђање цена на берзи. Захваљујући својој способности учења дугорочних зависности, LSTM мреже могу ефикасно моделирати сложене обрасце у временским подацима и пружити тачна предвиђања.
- **Обрада природног језика (NLP) [95].** LSTM мреже су изузетно популарне у области обраде природног језика. Оне се користе за различите задатке као што су машинско превођење језика, генерисање текста и анализа сентимента. Захваљујући својој способности разумевања и моделирања секвенцијалних података, LSTM мреже су се показале врло ефикасним у обради језичких информација.
- **Препознавање говора [96].** LSTM мреже су такође ефикасне у препознавању говора. Оне се користе за превод говорног језика у текст, омогућавајући машинама да тумаче и анализирају вербалне изјаве.
- **Генерисање музике.** LSTM мреже могу бити коришћене за генерисање музике. Оне могу научити секвенце нота из тренинг скупа и користити те секвенце за генерисање нових мелодија. Ово отвара могућност креативне примене LSTM мрежа у музичкој индустрији.
- **Анализу слика [97].** Иако LSTM мреже су најчешће повезане са обрадом секвенцијалних података, оне такође могу бити коришћене у анализи слика. Комбинацијом LSTM мрежа са конволуцијским неуронским мрежама (CNN), могуће је постићи ефикасно препознавање и анализу визуелних података.
- **Превод језика [98].** LSTM мреже су се показале врло корисним у задатку превођења језика. Оне могу научити сложене везе између речи и секвенци реченица и користити ту информацију за превод између различитих језика.

Међутим, као и свака технологија, LSTM мреже такође имају и одређене мане које треба узети у обзир. Прво, оне су прилично комплексне и захтевају значајне ресурсе. Обрада великих скупова података LSTM мрежама може бити временски и рачунарски захтевна, што може ограничити њихову применљивост у неким ситуацијама. Друга мана LSTM мрежа је њихова склоност преприлагођавању. Као и остале неуронске мреже, LSTM мреже могу бити подложне преприлагођавању тренинг скупу података, што може резултовати слабијим генерализацијама на новим улазима. Да би се ово решило, могу се применити технике регуларизације као што су Dropout или L1/L2 регуларизација. Трећа мана LSTM мрежа је тешко тумачење. Као дубоке неуронске мреже, LSTM мреже често се описују као "црне кутије". То значи да је тешко разумети како су донете одређене одлуке или предикције. Ово може бити проблематично у неким доменама где је тумачење резултата кључно.

Решење за ове проблеме долази са новом архитектуром неуронских мрежа а то су трансформери [99]. Трансформери представљају архитектуру неуронских мрежа коју чини низ слојева који се називају трансформери, или трансформерски слојеви. Ова архитектура је постигла импресивне резултате у пољима машинског превођења, генерисања текста, препознавање ентитета. Идеја трансформера је замена рекурзивних и конволуционих слојева у неуронским мрежама са механизмом познатим као „пажња“ (енг. *self attention*). Овај механизам омогућава моделу да научи тежину које одређују колико ће се сваки елемент у секвенци фокусирати на друге елементе. Флексибилност у одређивању тежине омогућава моделу да научи и разуме сложене међузависности између елемената у секвенцама.

Трансформери раде на основу улазне секвенце вектора, било да се ради о речима, знаковима или неком другом облику репрезентације текста. Сваки вектор улазне секвенце се затим трансформира кроз низ слојева трансформера. Сваки слој трансформера се састоји од два главна механизма: *self attention* и потпуно повезаних слојева. *Self attention* механизам омогућава моделу да израчуна важност сваког елемента у секвенци у односу на све остале елементе у тој секвенци. Ова важност се назива тежина пажње и користи се за комбиновање информација из свих елемената. Након израчунавања тежине пажње, улазни вектори се комбинују помоћу тих тежина како би се добио контекстуализован вектор за сваки елемент секвенце. Ти контекстуализовани вектори затим пролазе кроз потпуно повезане слојеве који примењују линеарну трансформацију и активациону функцију, као што је ReLU, како би се генерисао коначни излаз модела.

Предност трансформера је у њиховој способности да ефикасно моделирају дугорочне зависности између елемената у секвенци, што је често кључно у обради података. Осим тога, трансформери омогућавају паралелизацију рачунања, јер се тежина пажње може израчунати независно за сваки елемент у секвенци.

10.4. Анализа и предикција временских серија

Предикција временских серија (Time series forecasting) је статистички приступ предвиђања будућих догађаја на основу претходних података. Временска серија представља скуп вредности које се бележе током времена. У процесу предвиђања временских серија, користе се различити модели и технике за учење оквира и трендова у претходним подацима, а затим се користе те информације да би се генерисала предвиђања за будуће догађаје. Модели могу укључивати различите статистичке методе, као што су ARIMA, методе регресије, као и методе машинског учења, као што су неуронске мреже.

Временске серије се обично могу разложити на неколико компоненти, а то су:

- **Тренд.** Ово је генерални правац у којем се креће временска серија на дужи период. Тренд може бити растући (повећава се током времена), опадајући (смањује се током времена) или хоризонталан (релативно константан током времена).
- **Сезоналност.** Сезоналност се односи на редовне флукуације које се јављају у временској серији, а које се понављају у одређеним интервалима, као што су сваки

дан, сваки месец, сваке године итд. Ова сезонска флукуација представља годишњи образац који се понавља сваке године.

- **Циклуси.** Циклуси се односе на флукуације у временској серији које се не јављају у редовним интервалима. Ове флукуације могу бити узроковане различитим факторима, као што су економски услови, и обично их је теже предвидети него трендове или сезоналне флукуације.
- **Неправилности** (бели шум). Неправилности, често назване белим шумом, су случајне, нередовне флукуације које се јављају у временској серији. Ове флукуације обично су непредвидиве и не прате никакав очигледан образац. Оне представљају "шум" у временској серији који није објашњен трендом, сезоналношћу или циклусима.

Још два кључна фактора треба узети у обзир приликом анализе и предвиђања временских серија, а то су аутокорељација и стационарност података. Аутокорељација, такође позната као серијска корелација, је статистички концепт који описује степен сличности између вредности исте временске серије у различитим временима. Другим речима, аутокорељација мери како је једна вредност у временској серији повезана с претходним вредностима.

Аутокорељација може бити позитивна, негативна или нулта. Позитивна аутокорељација постоји када су високе вредности временске серије праћене високим вредностима у будућности, а ниске вредности праћене ниским вредностима. Негативна аутокорељација постоји када су високе вредности праћене ниским вредностима у будућности, и обрнуто. Нулта аутокорељација постоји када нема никакве везе између вредности у различитим временима.

Постоји много разлога зашто је аутокорељација важна у анализи временских серија. На пример, аутокорељација може помоћи у идентификовању трендова или сезоналних образаца у временским серијама. Такође, многи статистички модели временских серија, као што су ARIMA модели, претпостављају да постоји одређени степен аутокорељације у подацима. Да бисмо квантификовали аутокорељацију, можемо користити различите статистичке методе, као што су аутокорељацијска функција (ACF) и парцијална аутокорељацијска функција (PACF). Ове функције нам показују снагу и значај аутокорељације на различитим временским кашњењима.

Стационарност је такође важна претпоставка за многе моделе временских серија. Серија се сматра стационарном ако њена статистичка својства, попут средње вредности, варијансе и аутокорељације, остају константна током времена. То не значи да се свака тачка у временској серији мора подударати, напротив тачке могу бити прилично различите. Међутим, општи обрасци, попут очекиваних средњих вредности и варијансе, се не мењају. Стационарност је важна јер многи статистички модели претпостављају да су временске серије стационарне. Ако временска серија није стационарна, тада модели који претпостављају стационарност могу дати нетачне или завањајуће резултате.

Постоје различите технике за тестирање стационарности временске серије, укључујући Dickey-Fuller тест и KPSS тест. За потребе тестирања стационарности података из дисертације, урађен је Dickey-Fuller тест, који је приказан у табели са резултатима. Ако се утврди да временска серија није стационарна, могуће је применити различите трансформације (попут диференцирања или лог трансформације) да би се учинила стационарном.

Методе које се најчешће користе за анализу временских серија су:

- **Линеарна регресија.** Ово је једноставан приступ који може бити ефикасан када су подаци релативно једноставни и јасно приказују трендове или сезоналне обрасце.
- **Декомпозициони модели (Decomposition models).** Ови модели раде тако што разлажу временску серију на више компоненти, обично тренд, сезоналност и случајност. Једном када су ове компоненте идентификоване, могу се анализирати засебно.
- **ARIMA/SARIMA модели.** Ови модели су ефикасан алат за предвиђање временских серија. ARIMA стоји за Autoregressive Integrated Moving Average, док SARIMA додаје сезонску компоненту. Ови модели користе прошле вредности (ауторегресивни део) и грешке предвиђања из прошлости (покретни просечни део) за предвиђање будућих вредности.
- **State Space Models и Kalman Filter.** Ови модели су прилагодљиви модели који могу моделовати сложене временске серије.
- **Модели машинског учења (Machine Learning Models).** У неким ситуацијама, традиционални статистички модели временских серија нису најефикаснији начин предвиђања. Алгоритми машинског учења, као што су Random Forests или неуронске мреже, користити се за ефикасније предвиђање временских серија.

При анализи временске серије, циљ је често идентификовање и квантификовање ове компоненте. Ово може помоћи у разумевању како се временска серија понаша и самим тим и моделовању прецизнијих предикција о будућим трендовима у серији. У дисертацији су коришћени неки од наведених статистичких модела који су дали различите резултате, али је модел неуронске мреже показао супериорност у резултатима, што је приказано у наредном поглављу.

10.4.1. ARIMA

ARIMA (Ауторегресивни интегрисани покретни просек) модел је поуздани статистички модел који се користи за анализу и предвиђање временских серија [100]. ARIMA модел је уведен од стране Boxa и Jenkinsa, те се понекад назива и Box - Jenkins метод. ARIMA модел комбинује три основна концепта: ауторегресију (AR), интеграцију (I) и покретни просек (MA). Модел је прилагођен временским серијама како би се боље разумели подаци или предвиделе будуће тачке у серији, познате као предвиђање. Акроним модела се добија на основу кључних аспеката самог модела који су:

- **AR:** (Autoregression), Ауторегресија. Модел који користи зависни однос између посматрања и одређеног броја посматрања са кашњењем. Ауторегресија значи да модел користи претходне вредности временске серије за предвиђање будућих вредности. На пример, AR(1) модел би користио једну претходну вредност, AR(2) модел би користио две претходне вредности итд. Математички, AR(p) модел се једноставно може представити формулом:

$$Y_t = c + \varphi_1 Y_{(t-1)} + \varphi_2 Y_{(t-2)} + \dots + \varphi_p Y_{(t-p)} + \varepsilon_t \quad (10.11)$$

где је Y_t тренутна вредност серије, c је константа, φ_i су параметри модела, а ε_t је бели шум;

- **I:** (Integrated), Интегрисан. Ова компонента модела се односи на разликовање података, што значи израчунавање разлике између узастопних опсервација. Циљ је да се учини серија стационарном, јер ARIMA модели захтевају да временска серија буде стационарна:

$$Y_t = (1 - B)^d X_t \quad (10.12)$$

где Y_t представља диференцирану серију, X_t је оригинална серија, а B је оператор диференцирања.

- **MA:** (Moving Average), Покретни просек. Модел који користи зависност између посматрања и преостале грешке из модела покретног просека примењеног на посматрања са кашњењем. На пример, MA(1) модел би користио једну претходну грешку, MA(2) модел би користио две претходне грешке итд. Математички, MA(q) модел се може представити формулом:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{(t-1)} + \theta_2 \varepsilon_{(t-2)} + \dots + \theta_p \varepsilon_{(t-p)} \quad (10.13)$$

где је Y_t тренутна вредност серије, μ је средња вредност серије, θ_i су параметри модела, а ε_t је бели шум.

Стандардна нотација која се користи за ARIMA је p, d, q , где p представља ред ауторегресије (AR), d је ред интеграције (број пута колико је потребно разликовати податке да би постали стационарни), q је ред покретног просека (MA). Параметри p, d и q могу бити само цели бројеви који означавају редослед кашњења, односно број посматрања кашњења укључених у модел, затим степен разлике односно број пута када се необрађена посматрања разликују, и редослед покретног просека, тј. величину оквира покретног просека.

ARIMA при обради користи комбинацију ауторегресије, интеграције и покретног просека. Ова метода се користи за предвиђање будућих вредности на основу прошлих вредности временске серије, посебно када временска серија има јасан тренд и сезонске варијације. Честа примена ARIMA модела је у предвиђању цена акција, производње, потражње и слично.

ARIMA модели имају неке недостатке које треба имати у виду. Један од њих је претпоставка о линеарности, где ARIMA модели претпостављају линеарни однос између променљивих. Ово може бити ограничавајуће за временске серије које имају комплексније обрасце и нелинеарне везе. Други је ручни одабир параметара, иако постоје методе за аутоматско одабирање параметара у ARIMA моделу, често је потребно ручно подешавање параметара како би се постигла оптимална перформанса модела. Ово може захтевати време и стручност у интерпретацији резултата. ARIMA модели се не могу добро носити са нечистим или непотпуним подацима. Такође, могу бити захтевни у одређивању одговарајућег сета хиперпараметара за ARIMA модел. У ситуацијама када временска серија има много нејасноћа, нестабилности или када је потребно предвиђање будућих вредности на основу недостатка прошлих вредности, ARIMA модел може бити неефикасан.

10.4.2. Facebook prophet

Facebook Prophet је open-source алат за предвиђање временских серија развијен од стране Facebook Core Data Science тима. Prophet је осмишљен с циљем да омогући аналитичарима и програмерима да ефикасно извршавају предвиђања високог квалитета временских серија [101]. У својој суштини, Prophet користи декомпозиционе методе које разлажу временску серију на њен тренд, компоненте за сезоналност и празнике како би генерисао предвиђање. Prophet је добар за примену у ситуацијама када је потребно брзо и једноставно предвиђати временске серије са добром тачношћу, као што су предвиђање продаје, тражња, посета итд. Осим тога, Prophet подржава предвиђања са коришћењем календарских ефеката и сезоналних трендова, што га чини добрим за примену у многим ситуацијама.

Математички, Prophet модел може бити изражен следећом формулом:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (10.14)$$

где је $y(t)$ представља вредност временске серије у тренутку t , $g(t)$ компонента тренда, $s(t)$ је компонента за сезоналност, $h(t)$ је ефекат празника, а ε_t је грешка. Тренд компонента $g(t)$ је обично *piecewise linear* или *logistic function*, компонента за сезоналност $s(t)$ је периодична функција, а ефекат празника $h(t)$ је функција која је *pot-null* само када су празници.

Модел временских серија Prophet састоји се од три кључне компоненте:

1. **Trend**, $g(t)$. Тренд представља дугорочну прогресију временске серије, независно од сезоналних флукуација или празника. Prophet подржава два типа трендова - линеарни и логистички. Линеарни тренд је једноставан и погодан за временске серије без дефинисане границе. Логистички тренд је погодан за временске серије са познатим границама (нпр. капацитет).
2. **Seasonality**, $s(t)$. Сезоналност описује промене које се понављају у одређеним интервалима. Prophet аутоматски детектује дневну, недељну и годишњу сезоналност у подацима, али корисник може да дефинише и друге сезонске компоненте.

3. **Holydays, $h(t)$.** Празници, како би се обрадили ефекти празника, Prophet омогућава корисницима да наведу празнике и њихове датуме. Такође, могуће је дефинисати и ширу сезоналност око празника кроз доњи и горњи прозор.

Након што су идентификоване ове компоненте, Prophet користи адитивни модел за комбиновање њихових ефеката, уз додатак термина грешке, t .

Prophet нуди бројне додатне функционалности које омогућавају корисницима да прилагоде модел својим специфичним потребама. Неке од ових функционалности укључују:

- **Промене у тренду.** Prophet омогућава корисницима да експлицитно наведу датуме потенцијалних промена у тренду. То може бити корисно у ситуацијама када су познати догађаји узроковали значајне промене у тренду.
- **Saturation growth model.** Prophet подржава логистички модел раста, који је користан за временске серије које су ограничене капацитетом. За ту потребу користи логистичку функцију која описује раст који постиже zasiћење. Prophet моделира сатурацију раста помоћу три компоненте: тренд раста, годишња или недељна сезоналност, и прилагодљива компонента шума.
- **Регресиони модели.** Prophet омогућава корисницима да додају регресионе променљиве у модел.

Док је адитивни модел који Prophet користи концептуално једноставан, прецизно прилагођавање модела подацима захтева сложене математичке и статистичке технике. На пример, за прилагођавање тренд компоненте, Prophet користи Bayes модел са аутоматском диференцијацијом варијационог инференцирања (ADVI). Ова метода користи софистициране технике оптимизације за одабир параметара модела који најбоље одговарају подацима.

Када су у питању сезонске компоненте, Prophet користи Fourier серију како би моделирао сезонске флукуације, омогућавајући корисницима да одреде број сезонских хармоника које ће се користити у моделу. За празнике, Prophet користи бинарне "dummy" променљиве како би приказао ефекат сваког празника на временску серију. Након обучавања модела, могуће је користити *bootstrap* методу за генерисање интервала одступања за предвиђања. То корисницима пружа не само тачније процене будућих вредности, већ и меру неизвесности око тих процена.

Још једна предност Facebook Prophetа је његова робусност. Он је у стању да се носи са недостајућим вредностима и изванредним вредностима у временским серијама. Ово је посебно корисно јер се стварни подаци често сусрећу са проблемима као што су пропуштена мерења или неочекивани догађаји. Prophet се ефикасно прилагођава тим изазовима и пружа поуздане резултате без потребе за додатном обрадом података. Корисницима је омогућено прилагођавање модела према специфичним потребама, са различитим параметрима и опцијама за подешавање. Prophet омогућава оптимизацију резултата за различите врсте

временских серија, што значи да се може прилагодити специфичним захтевима проблема и оптимизовати перформансе модела.

Међутим, као и сваки алат, Facebook Prophet такође има своје недостатке. Прво, ефикасност алата може опаста када се ради са врло великим скуповима података. Обрада и моделирање великих временских серија могу захтевати додатне рачунарске ресурсе и време за извршавање. Такође, иако Prophet пружа бројне опције прилагођавања, ефикасно коришћење тих опција често захтева ручно подешавање. То може бити захтевно за кориснике без дубљег разумевања временских серија и моделирања.

10.5. Моделовање

10.5.1. Анализа и припрема података

Тестирани период је био у току летњег периода, прикупљени подаци из тог периода нису узети у обзир због тога што су искоришћени за подешавање опреме, исправљање кода и моделовање. Самим тим подаци нису били везани по данима односно имали су прекиде. Подаци који су узети за моделовање су прикупљени током јесењих месеци, када пчеле имају другачија кретања у односу на летњи период. Јесен је период када се пчеле припремају за зимски период. За анализу је коришћено двадесет узастопних дана сниманих у октобру 2021 године. Мерења су вршена у интервалима од 5 минута, док су снимања вршена на сваких 20 секунди. Овакав приступ снимања је омогућио шири спектар манипулације подацима са различитим временским серијама. Читава природа у јесењем периоду има слабију активност па је узети интервал адекватан.

Узимајући у обзир мали број излазака пчела из кошнице, на основу измерених података, и малих промена временских прилика у интервалу од 5 минута у периоду када су вршена мерења, измењен је приступ посматрања временске серије података. На основу тога, временски интервали су консолидовани у 24 сата. За добијање коначне карактеристичне улазне вредности узете су просечне вредности свих повезаних вредности које припадају посматраном сату. С друге стране, излазне вредности се добијају као збир свих излаза из кошнице посебно и улаза у кошницу посебно, у посматраном сату. У табели 10.1. приказана је структура сензора и измерених вредности скупа података са свим коришћеним променљивим и њиховим описима. Читав скуп података коришћених за израду модела и предикцију, као и извор кода, биће јавно доступни на адреси <https://github.com/andrijevicnebojsa/BeeMAP>.

Табела 10.1 Сензори, контролни сензори и опис мерења.

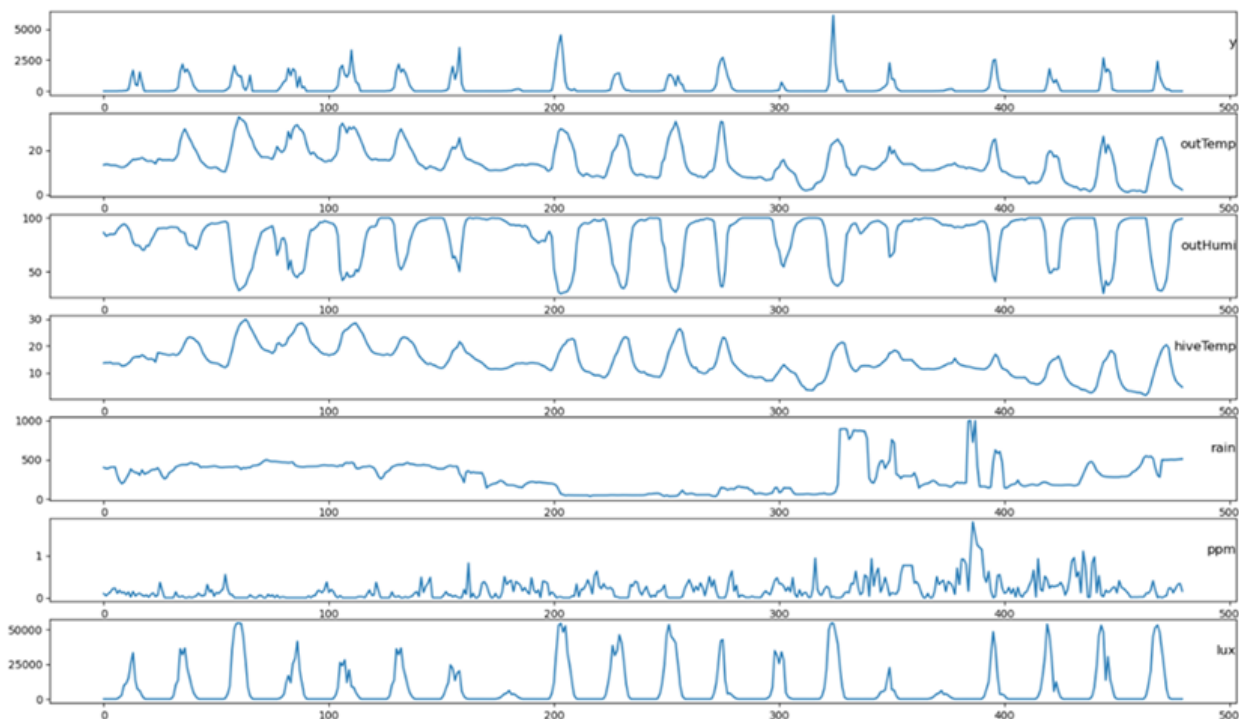
Назив колоне	Опис	Назив колоне	Опис
Date	Датум мерења	BME280_humi	Контрола влажности спољашњег ваздуха
Hour	Време мерења	BME280_pressure	Ваздушни притисак
AM2302_1_Temp	Температура спољашњег ваздуха	BME280_alt	Altitude, Надморска висина

AM2302_1_Humi	<i>Влажност спољашњег ваздуха</i>	SI1145_visible	<i>Интензитет дневне светлости</i>
AM2302_2_Temp HIVE	<i>Температура у кошници</i>	SI1145_IR	<i>Интензитет инфрацрвеног зрачења</i>
AM2302_2_Humi HIVE	<i>Влажност у кошници</i>	SI1145_UV	<i>Контрола УВ индекса</i>
SW420_Vibrate HIVE	<i>Вибрације у кошници</i>	ANEM_voltage	<i>Снага ветра</i>
MHRD_rain	<i>Сензор за кишу</i>	ANEM_windSpeed	<i>Брзина ветра</i>
MQ135_PPM	<i>Сензор квалитета ваздуха</i>	MIC1_freq	<i>Фреквенцијски спектар звука ван кошнице</i>
MICS6814_PPM	<i>Сензор квалитета ваздуха</i>	MIC1_volume	<i>Ниво звука и јачина ван кошнице</i>
MICS5524_PPM HIVE	<i>Сензор квалитета ваздуха у кошници</i>	MIC2_freq HIVE	<i>Фреквенцијски спектар звука у кошници</i>
BH1750_lx	<i>Дневна светлост и интензитет осветљења (lx)</i>	MIC2_volume HIVE	<i>Ниво звука и јачина у кошници</i>
VEML6750_uvindex	<i>УВ индекс</i>	BEE_message OUT	<i>Бројач излазака пчела из кошнице</i>
BME280_temp	<i>Контрола спољашње температуре</i>	BEE_message IN	<i>Бројач улазака пчела у кошницу</i>

Полазну основу чине две главне променљиве, '*BEE_message OUT*' представља број пчела које су изашле из кошнице, док '*BEE_message IN*' представља број пчела које су ушле у кошницу. Ове две променљиве ће се користити као излазне вредности у нашим моделима. На овај начин повезујемо зависне и независне показатеље кретања пчела, односно повезујемо комбинацију независних фактора на утицај кретања пчела са зависним.

Бројање пчелињих улазака и излазака из кошнице, као и мерење стања непосредног окружења, унутар и ван кошнице, од кључног је значаја за иницијализацију стања и тригероване аларма. Такође допуњава резултате других параметара који указују на учесталост кретања пчела добијених из мерења са инсталираних сензора. Чињеница приказана кроз истраживање је повезивање фактора, јер без мерења свих наведених фактора, а посебно временских услова, број пчела које улазе и излазе не би био од већег значаја и био би само статистички детаљ. Супротно томе сви остали подаци и детаљи временских услова не би могли сами да дају значајне резултате. Сви ови подаци, прикупљени из средине у којој пчеле живе и кроз коју се крећу, могу се користити за развој модела који врло прецизно може предвидети фреквенцију тј. кретање пчела.

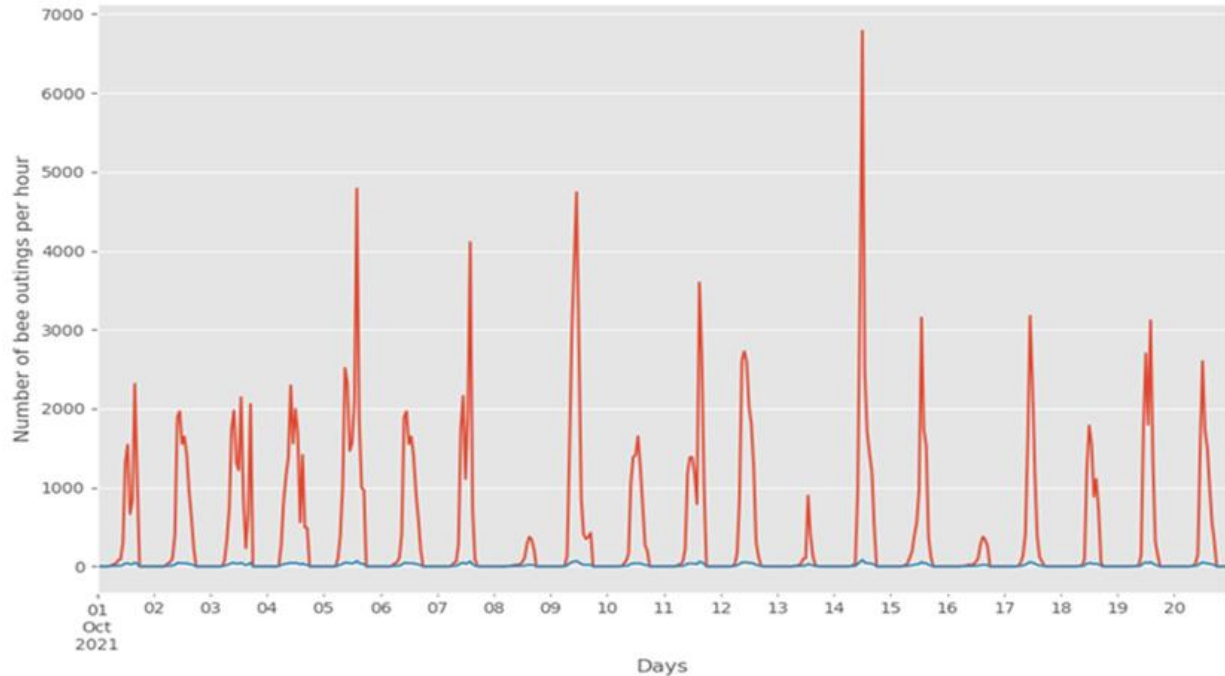
На слици 10.8 дат је пример преклапања неких коришћених променљивих, са излазном променљивом бројем улазака и излазака пчела у кошницу, где облици временских серија одговарају улазима и излазима (више о томе у наредном поглављу). На основу приказа облика временских серија, закључује се да су поједини параметри као што су интензитет дневне светлости или релативна влажност ваздуха ван кошнице, веома значајни на наш модел. Разлог томе је резултат кретања пчела у зениту дана када су параметри повољни за кретање пчела, или супротно томе када је спољашна влажност висока па пчеле не напуштају кошницу.



Слика 10.8 Облик седам временских серија које одговарају излазу и улазима.

У фази израде и дизајна модела, подаци су показали да постоје периоди у току 24 часа када пчеле нису имале активности и када су активне, и то су везани периоди без прекида. Искористили смо тај податак и креирали смо нову бинарну променљиву представљену као део дана, односно дан, временски период у току дана од 05:00 до 18:00 часова, или ноћ која представља преостале сате када пчеле нису активне. Уколико се деси да се јави било каква активност у току периода неактивних сати, мењамо је са вредношћу 0. То се једино може догодити у граничним сатима када пчеле завршавају активност у току дана или рану ујутру када започињу дневну активност, где се појављују миноран број излазака.

Подаци о броју излазака из кошнице и улазака у кошницу, трансформисани су у циљу да се стабилише варијанса серије. Колоне су трансформисане коришћењем трансформације квадратног корена, тзв. трансформација снаге јер се код анализе временских серија врло често сматра да оваква врста трансформација стабилизује варијансу серије. Обзиром да су неке вредности једнаке 0 у том случају логаритамска трансформација се прескаче. На слици 10.9 црвеном бојом је приказана оригинална временска серија излазака пчела из кошнице, а плавом бојом временска серија након примене трансформације квадратног корена. Видимо да је драстично смањен број излазака за поједине дане. Ово се дешава јер су вредности појединих параметара, на пример временски услови, за тај дан били лошији због тога што је падала киша или дувао ветар.



Слика 10.9 Приказ правих и трансформисаних података о броју излазака пчела по сату за 20 дана.

10.5.2. Креирање модела за предикцију у Python-у

Предиктивни модели су једна од кључних апликација ВИ, који се користе за предвиђање будућих догађаја на основу података из прошлости. За креирање модела коришћен је VSCode у Python окружењу. Водећи се специфичним скупом података, креирани су модели за LSTM, ARIMA и Facebook Prophet. У наредном сегменту следи детаљнији опис корака при моделовању.

I. Кораци за примену LSTM модела.

1. **Учитавање података.** Класа DataLoader се користи за учитавање података из .csv фајла. Након што се подаци прочитају, индексирају се по колони 'ds'.

```
DataLoader = DataLoader(data_path, file_name, model_output, sqrt_transform)
bee_df = DataLoader.load_data()
bee_df.set_index('ds', inplace=True, drop=True)
```

2. **Припрема података.** У овом делу, сви подаци се конвертују у *float* тип података, затим се нормализују користећи MinMaxScaler (сви подаци су скалирани да буду између 0 и 1). Након тога, подаци се реконфигуришу да би се представили као проблем надгледаног учења помоћу `series_to_supervised` функције. Затим, колоне које нису потребне за предвиђање се одбацују.

```
values = bee_df.values
# ensure all data is float
```

```

values = values.astype('float32')

# normalize features
scaler = MinMaxScaler(feature_range=(0, 1)).fit(values)
scaled = scaler.transform(values)

# frame as supervised learning
reframed = series_to_supervised(scaled, 1, 1)

# drop columns we don't want to predict [BME280_temp, BME280_humi,
BME280_pressure, BME280_alt, SI1145_visible, SI1145_IR, SI1145_UV, ANEM_voltage,
ANEM_windSpeed]
reframed.drop(reframed.columns[[11, 12, 13, 14, 15, 16, 17, 18, 19]], axis=1,
inplace=True)

```

2. Дефинисање и обучавање модела. У овом сегменту, конструише се LSTM модел користећи Keras библиотеку. Модел се састоји од једног LSTM слоја са 50 неурона и једног густог (Dense) слоја са једним неуроном. Као функција губитка (loss function) користи се средња апсолутна грешка (mean absolute error - mae), а оптимизатор је "adam". Након дефинисања модела, модел се обучава користећи тренинг податке.

```

# design network
model = Sequential()
model.add(LSTM(50, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')

# fit network
history = model.fit(train_X, train_y, epochs=50, batch_size=20,
validation_data=(test_X, test_y),
verbose=2, shuffle=False)

```

3. Евалуација модела. У овом делу се приказује графикон са историјом губитака током тренирања за тренинг и валидациони скуп података.

```

pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()

```

4. Предвиђање. Након што је модел обучен, користи се за предвиђање на тест скупу података. Излаз модела се прво конвертује назад у оригиналне вредности (јер су биле нормализоване), а затим се упоређује са стварним вредностима користећи Root Mean Squared Error (RMSE) као метрику.

```

# make a prediction
yhat = model.predict(test_X)
test_X = test_X.reshape((test_X.shape[0], test_X.shape[2]))

# invert scaling for forecast
inv_yhat = concatenate((yhat, test_X[:, 1:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:, 0]

# invert scaling for actual
test_y = test_y.reshape((len(test_y), 1))
inv_y = concatenate((test_y, test_X[:, 1:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:, 0]

# calculate RMSE
if sqrt_transform:
    rmse = np.sqrt(mean_squared_error(np.power(inv_y, 2), np.power(inv_yhat, 2)))
else:
    rmse = np.sqrt(mean_squared_error(inv_y, inv_yhat))

print('Test RMSE: %.3f' % rmse)

```

II. Кораци за примену *ARIMA* модела.

- 1. Учитавање података.** Дефинисан је метод `load_day` који учитава податке, припрема и групише их по сатима. Затим се учитавају подаци за сваки дан у месецу и сви се спајају у један `DataFrame`.

```

data8 = load_day('BW201017.txt')
...
data22 = load_day('BW201031.txt')
frames = [data8, data9, ..., data22]
result = pd.concat(frames, sort=False)

```

- 2. Претпроцесирање података.** Датум и сат колоне се користе за креирање `DateTime` индекса. Након тога, неке колоне се бришу, а остале се визуализују.

```

result.index = result.Date_Time
result = result.drop(['Date_Time', 'Date', 'hour'], axis=1)
result.plot(kind='line', legend='reverse', title='Visualizing Sensor Array Time-Series')

```

- 3. Тестирање стационарности.** У процесу тестирања стационарности података, прво је потребна подела података на сет за обучавање и сет за тест. Затим се примењује Augmented Dickey-Fuller (ADF) тест на свакој колони сета за обучавање да би се утврдило да ли су подаци стационарни. Уколико вредност ADF теста покаже да подаци нису стационарни, следећи корак је примена разлике над подацима како би постали стационарни.

```
ds_train = result[:int(0.8*(len(result)))]
ds_test = result[int(0.8*(len(result))):]
for i in ds_train.columns:
    print("Column: ", i)
    adf_test(ds_train[i])
ds_differenced = ds_train.diff().dropna()
```

- 4. Моделовање ARIMA.** Након припреме података, користи се ARIMA модел за предвиђање. Прво се дефинише модел са редом (5, 1, 0), где је 5 ред ауторегресије, 1 је степен диференцирања, а 0 је ред покретне просечне вредности.

```
for t in range(len(test)):
    model = ARIMA(history, order=(5, 1, 0))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
```

- 5. Евалуација модела.** На крају се врши евалуација модела коришћењем Root Mean Squared Error (RMSE) метрике. RMSE се израчунава као квадратни корен из средње квадратне грешке између стварних и предвиђених вредности.

```
mse = np.sqrt(mean_squared_error(test, predictions))
print('Test RMSE: %.3f' % rmse)
```

III. Кораци за примену Facebook Prophet модела.

- 1. Учитавање и припрема података.** Пре него што почне обука модела, неопходно је правилно учитати и припремити податке. Овде се подаци учитавају кроз DataLoader класу која се налази у **utils.LoadData**. Подаци се даље чисте и припремају за модел.

```
data_path = 'data/'
file_name = 'bee_October_2021.csv'
model_output = 'BEECNT_message IN'
```



```

sqrt_transform = True

DataLoader = DataLoader(data_path, file_name, model_output, sqrt_transform)
df = DataLoader.load_data()

```

- 2. Постављање Prophet модела.** Подешава се Prophet модел са дневним сезонским факторима и додају се регресори. Ови регресори су додатни фактори на основу којих се модел учи да предвиђа.

```

model = Prophet(yearly_seasonality=False, weekly_seasonality=False,
daily_seasonality=True)
model.add_regressor('outTemp')
model.add_regressor('outHumi')
model.add_regressor('hiveTemp')
model.add_regressor('hiveHumi')
model.add_regressor('rain')
model.add_regressor('ppm')
model.add_regressor('lx')
model.add_regressor('uvindex')
model.add_regressor('Day_night')

```

- 3. Обука модела.** Након што су подаци уčitани и модел постављен, следећи корак је обука модела на припремљеним подацима.

```
model.fit(df)
```

- 4. Предвиђање.** Модел је обучен и може да прави предикције. Модел предвиђа на основу података без циљне колоне 'y'. Ако је предвиђање мање од нуле, поставља се на нулу, пошто негативне вредности не би имале смисла у овом контексту.

```

forecast = model.predict(df.drop(columns="y"))
forecast.loc[forecast.yhat < 0, "yhat"] = 0

```

- 5. Евалуација модела.** Врши се евалуација модела користећи стварне и предвиђене вредности. Израчунава се квадратни корен средње квадратне грешке (RMSE) између стварних и предвиђених вредности.

```

metric_df =
forecast.set_index('ds')[['yhat']].join(df.set_index('ds').y).reset_index()
metric_df.dropna(inplace=True)
if sqrt_transform:
    np.sqrt(mean_squared_error(np.power(metric_df.y, 2), np.power(metric_df.yhat,
2)))
else:
    np.sqrt(mean_squared_error(metric_df.y, metric_df.yhat))

```

6. Подела података и поновна обука. У овом кораку подаци се деле на тренинг и тест сетове користећи методу временских серија. Модел се затим обучава на тренинг сету и евалуира на тест сету. Овај поступак се понавља 5 пута, а за сваку итерацију се рачуна RMSE. На крају се израчунава просечни RMSE за свих 5 итерација.

```
tscv = TimeSeriesSplit(n_splits=5)
day_seas_df = data_gen(df, tscv)
rmsees = []

for i in range(5):
    ...
    rmsees.append(rmse)

print(f"Average RMSE:{np.mean(rmsees)}")
```

При изради модела и приликом тестирања, ради добијања што бољих резултата јако је битно направити правилан избор и подешавање параметара, као што су избор следећих параметара:

1. **Број неурона.** Број неурона у архитектури неуронске мреже се може одредити експериментално или користећи методе као што је унакрсна валидација (cross-validation). Обично се користи поступак постепеног повећавања броја неурона у слојевима неуронске мреже како би се моделу омогућило да научи комплексније шаблоне у подацима. Међутим, превелик број неурона може довести до преобучавања (overfitting) модела.
2. **Епохе.** Епоха представља један пролазак кроз цео скуп података током тренирања неуронске мреже. Током сваке епохе, модел изложи улазне податке неуронској мрежи, израчуна грешку између стварних и предвиђених излаза и прилагоди тежине (параметре) неуронске мреже користећи поступак backpropagation. Обично се модел тренира кроз више епоха како би се постигла боља конвергенција и побољшала предиктивна моћ модела.
3. **Batch size.** Batch size одређује колико примера из скупа података ће бити коришћено за ажурирање параметара модела током једне епохе. Већи batch size може убрзати процес тренирања, али истовремено захтева више меморије. Мањи batch size може успорити тренирање, али може побољшати стабилност модела.
4. **Look back.** Look back (историја улаза) је број претходних временских корака који се користе као улаз за предвиђање следећег временског корака. На пример, ако је look back постављен на 5, модел ће користити податке из претходних 5 временских корака како би предвидео следећи. Одабир одговарајућег look back-а зависи од природе временских серија и жељене способности модела да ухвати сложене шаблоне у подацима.

5. **Скалирање.** Скалирање се односи на поступак претварања вредности улазних података на одређени опсег како би се осигурала стабилност и боље перформансе модела. У контексту временских серија, често се користи стандардно скалирање (standardization) или нормализација (normalization) како би се средња вредност података поставила на нулу, а стандардна девијација на јединицу. То помаже уклањању разлика у мерењу између различитих променљивих и олакшава тренирање неуронске мреже. Скалирање може побољшати конвергенцију модела и спречити доминацију одређених променљивих над другима.
6. **Фитовање података.** Фитовање података односи се на поступак тренирања неуронске мреже са прикладним скупом података. Током фитовања, модел се прилагођава улазним подацима и покушава научити шаблоне и зависности у подацима. Фитовање укључује итеративно ажурирање параметара модела како би се смањила грешка између стварних и предвиђених излаза. Циљ је минимизовати грешку и постићи оптимално прилагођавање подацима.
7. **Слојеви и јединице** у LSTM моделу. LSTM има слојеве који су састављени од јединица или ћелија. Свака јединица у LSTM моделу има унутрашњу структуру која омогућава памћење информација из претходних временских корака. Ове јединице имају механизме за контролу и регулацију протока информација, као што су капије за заборављање (forget gate), капије за улаз (input gate) и капије за излаз (output gate). Ово им омогућава да науче и задрже важне информације о дугорочним зависностима у временским серијама.

Кроз одабир и подешавање свих компоненти, можемо постићи да модел боље научи и предвиђа сложене образце, трендове и зависности од временских серија. Важно је експериментисати са овим параметрима, како би се модел прилагодио специфичним подацима и постигли најбољи резултати предикције.

10.6. Анализа и предикција временских серија података

Подаци о временским серијама су скуп запажања добијених поновљеним мерењима током времена, као што је случај са нашим прикупљеним подацима. У вештачкој интелигенцији, оне се користе за предвиђање будућих вредности на основу претходних вредности и других релевантних фактора. Регресионо предиктивно моделирање је техника која се користи за проналажење зависности између улазних променљивих и одговарајућих излазних променљивих. Код временских серија, за разлику од регресионог предиктивног моделирања, сложеност је утолико већа јер временске серије додају сложеност зависности низа међу улазним променљивим. Основна предност RNN-а у односу на ANN је у томе што RNN може моделовати колекцију записа (тј. колекцију времена) тако да се може претпоставити да сваки образац зависи од претходних. С друге стране, поређења са ETS-ом (грешка, тренд, сезоналност) и ARIMA-ом показује да (полу) аутоматски RNN модели нису баш идеални, али су ипак конкурентне алтернативе у многим ситуацијама.

ETS или Error-Trend-Seasonality у вештачкој интелигенцији представља метод за анализу временских серија. Ова техника се користи за детекцију и моделирање понашања временских серија, укључујући трендове, сезонске ефекте и грешке.

Како бисмо добили најуспешнију методу предикције улазака и излазака пчела из кошнице, користећи временске серије за предвиђање, тестиране су све поменуте методе. Прва метода која је тестирана је ARIMA, која је генерализација једноставнијег ауторегресивног покретног просека и додаје појам интеграције. Након ARIMA-е тестирана су још два напредна приступа Recurrent Neural Networks (LSTM) и Facebook Prophet.

ARIMA ради само са стационарним временским серијама. Стационарне временске серије су оне чија својства не зависе од времена у коме се серија посматра. Један од начина да се објективније утврди да ли је потребно разликовање је коришћење теста јединичног корена. Ово су статистички тестови хипотезе стационарности који су дизајнирани да утврде да ли је потребно разликовање. У ту сврху је коришћен Dickey-Fuller тест (Табела 10.2). Резултати теста за излазне параметре BEE_OUT и BEE_IN су представљени у наставку.

Табела 10.2 Резултати Dickey-Fuller теста.

BEE_OUT		BEE_IN	
<i>Резултати Dickey-Fuller Testa:</i>		<i>Резултати Dickey-Fuller Testa:</i>	
<i>Статистика теста</i>	-8.410305e+00	<i>Статистика теста</i>	-9.169726e+00
<i>p-value</i>	2.112639e-13	<i>p-value</i>	2.406527e-15
<i>#Lags Used</i>	1.400000e+01	<i>#Lags Used</i>	3.000000e+00
<i>Number of Observations Used</i>	4.650000e+02	<i>Number of Observations Used</i>	4.760000e+02
<i>Critical Value (1%)</i>	-3.444491e+00	<i>Critical Value (1%)</i>	-3.444163e+00
<i>Critical Value (5%)</i>	-2.867776e+00	<i>Critical Value (5%)</i>	-2.867631e+00
<i>Critical Value (10%)</i>	-2.570091e+00	<i>Critical Value (10%)</i>	-2.570014e+00

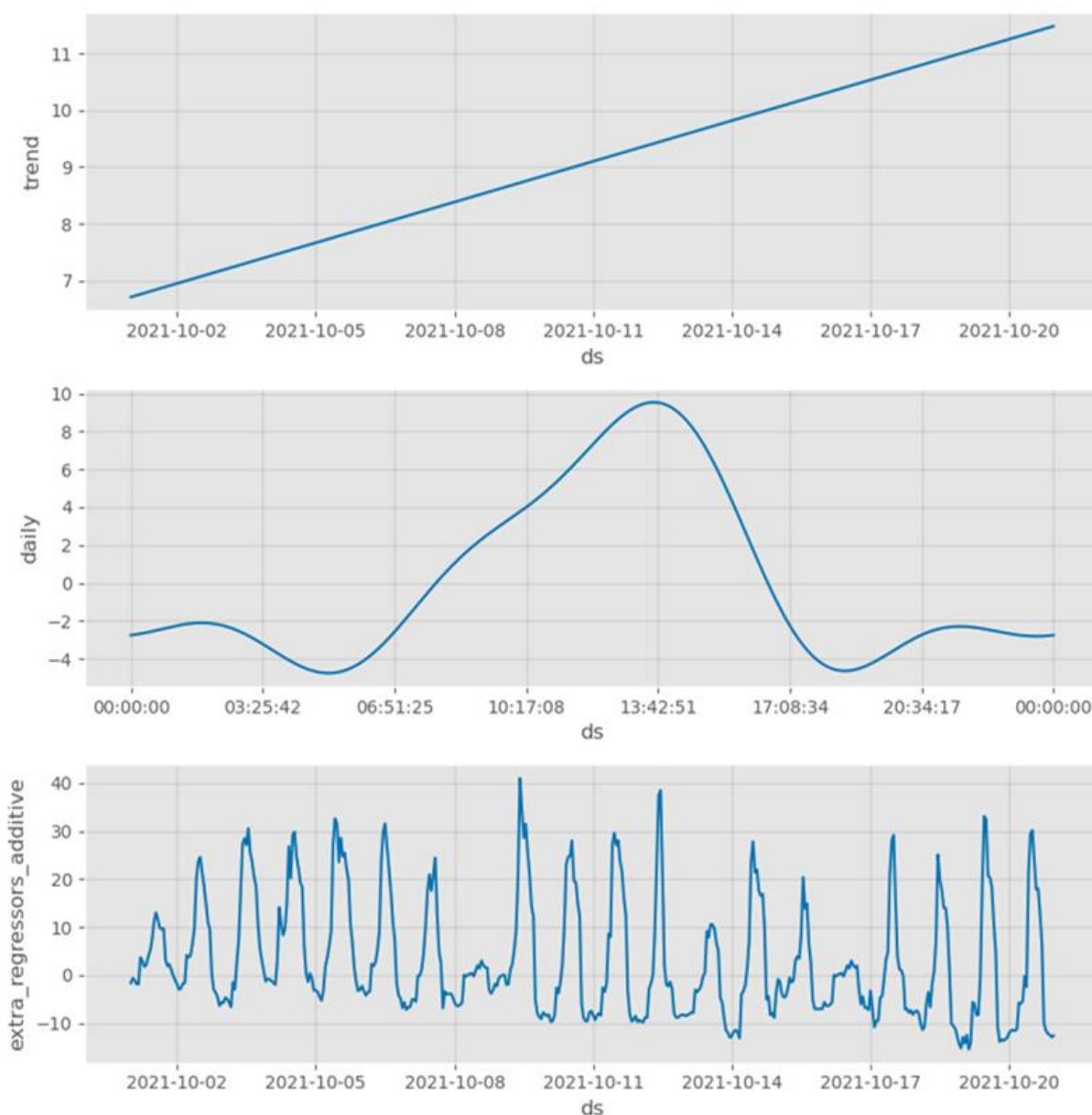
Као што видимо, посматране временске серије су биле стационарне, тако да бисмо могли одбацити нулту хипотезу о јединичном корену на свим значајним заједничким нивоима.

Док ARIMA користи ауторегресивно предвиђање које одговара линији линеарне регресије са вредностима кашњења и условима грешке, **Facebook Prophet** је процедура за предвиђање података временских серија на основу адитивног модела где се нелинеарни трендови уклапају са годишњим, недељним и дневним сезоналностима, плус ефекти празника. Најбоље функционише са временским серијама које имају јаке сезонске ефекте и неколико сезона историјских података. Речено је већ да је Prophet отпоран на податке који недостају, као и на промене у тренду, и обично се добро носи са необичним вредностима и одступањима. Ово се заснива на генерализованим адитивним моделима (GAMs), који обезбеђују општи оквир за проширење стандардног линеарног модела дозвољавајући нелинеарне функције за сваку од променљивих док одржава адитивност. Баш као и линеарни модели, GAM-ови се могу применити и са квантитативним и квалитативним одговорима.

У овом моделу коришћене су три главне компоненте: тренд, сезоналност и празници. Комбиновани су у следећој једначини.

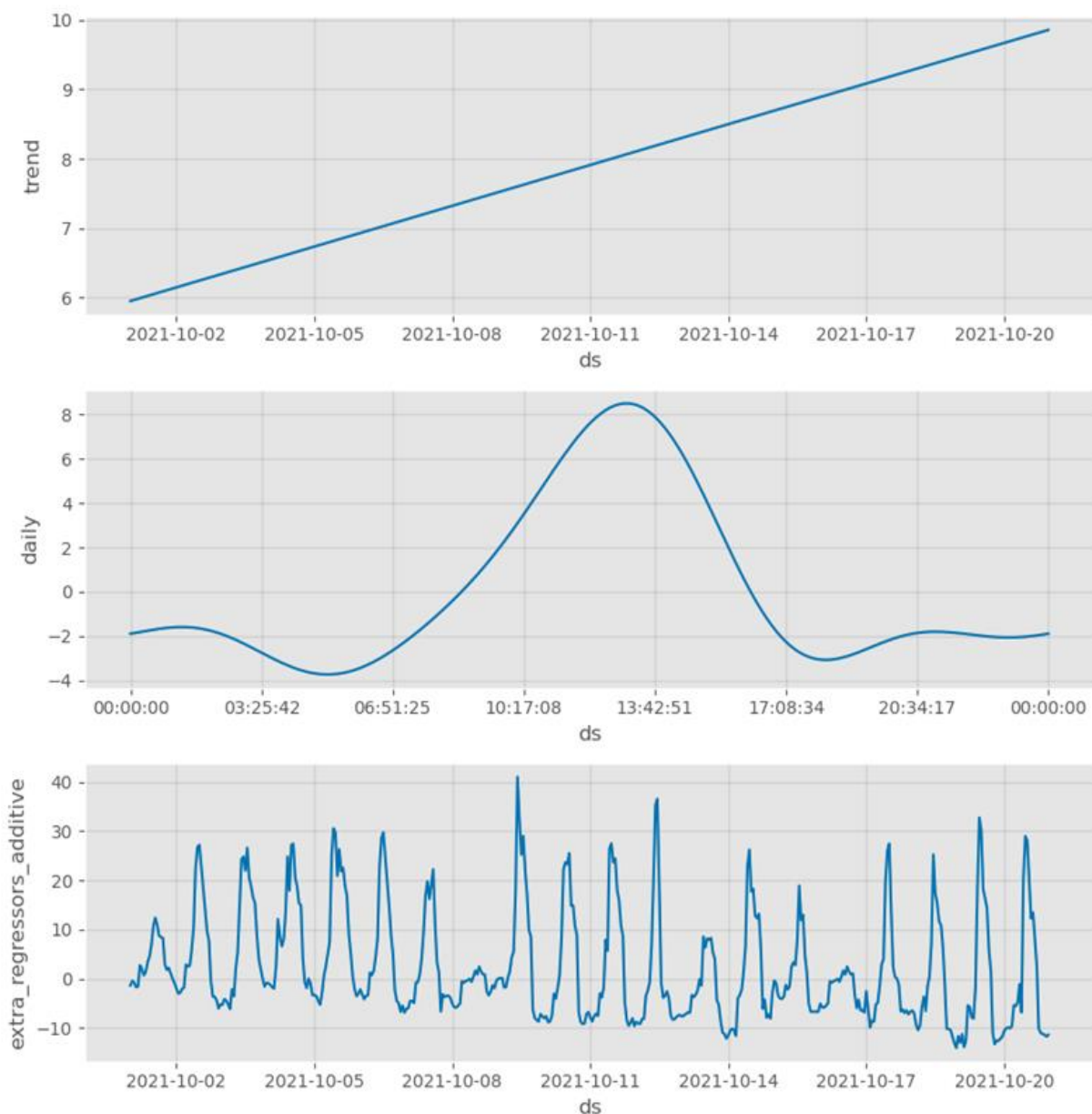
$$y(t) = g(t) + s(t) + h(t) + t, \quad (10.15)$$

где је $g(t)$ функција тренда која моделује неперидичне промене у вредности временске серије, $s(t)$ представља перидичне промене (нпр. седмичну и годишњу сезоналност), а $h(t)$ представља ефекте празника који се дешавају по потенцијално неправилним распоредима током једног или више дана. Термин грешке t представља све идиосинкратичне промене које модел не прихвата. Детектоване компоненте за целу ВЕЕ_OUT временску серију, тренд, дневно понашање, као и утицај додатих регресора, приказани су на слици 10.10. Слични графици су добијени за целу временску серију ВЕЕ_IN (Слика 10.11).



Слика 10.10 Резултати фазе тренинга Facebook Prophet алгоритма за променљиву BEE_OUT.

Facebook Prophet је веома популаран и често коришћен за предвиђање временских серија јер је отпоран на одступања, недостајуће податке и драматичне промене у временским серијама, док је ARIMA склона белом шуму и нестационарним сигнаlima. Постојање одступања и података који недостају у оваквим случајевима је готово сигурно, имајући у виду да се опрема понекад може покварити.



Слика 10.11 Резултати фазе тренинга Facebook Prophet алгоритма за BEE_IN променљиву.

Овде истражујемо проблем флексибилног предвиђања Y на основу неколико предиктора, X_1, \dots, X_p . Могуће улазне променљиве пажљиво су одабране из табеле 10.1. Више информација о изабраним карактеристикама дато је у одељку са резултатима.

11. РЕЗУЛТАТИ МОДЕЛОВАЊА

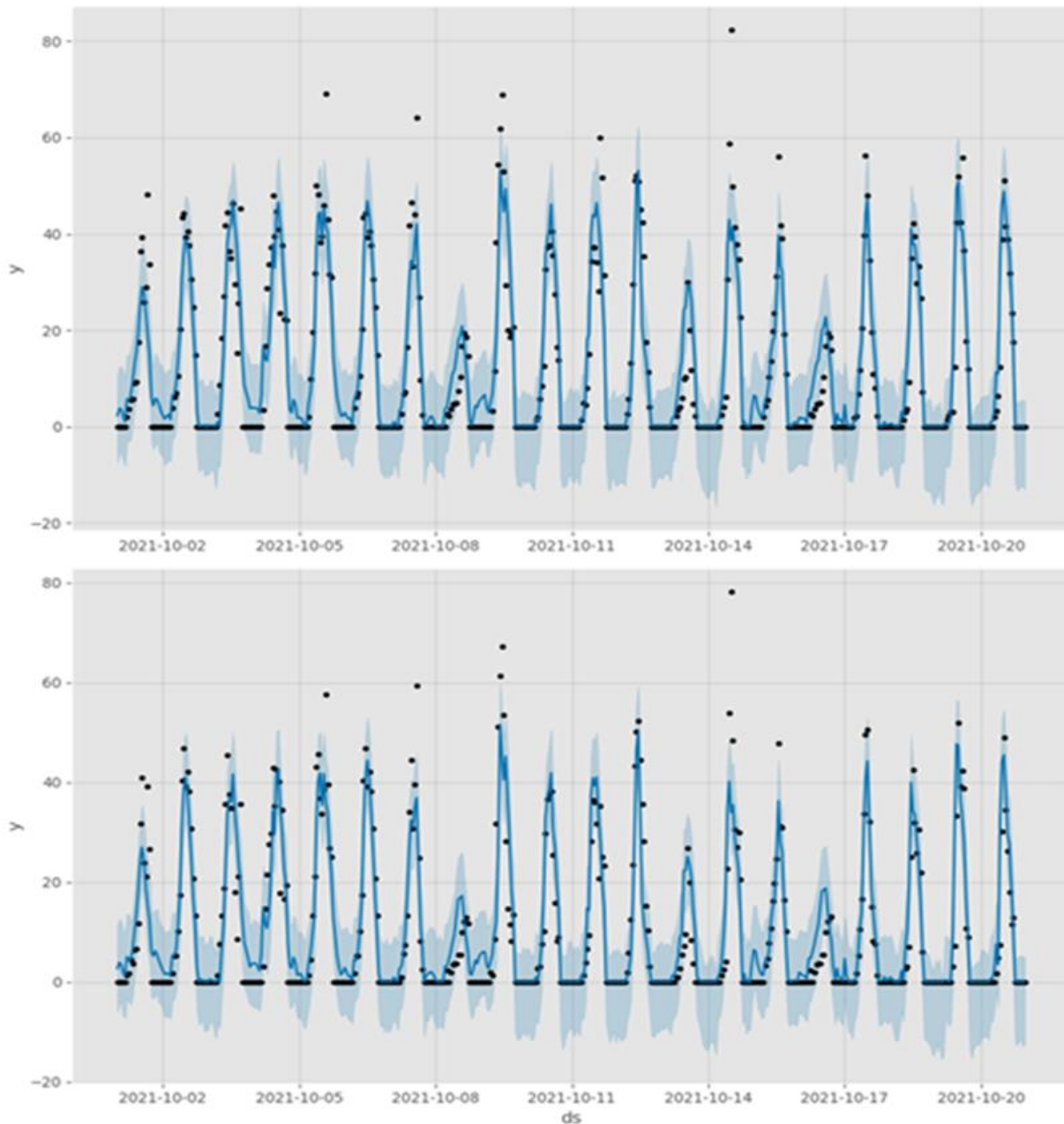
11.1. Експериментална поставка и евалуација

За тестирање робусности модела, у експерименталном делу тестирања је коришћен унакрсни валидатор временских серија. Коришћена је TimeSeriesSplit класа из библиотеке scikit-learn која пружа веома једноставан интерфејс за поделу узорака података временских серија који се посматрају у фиксним временским интервалима у скупове за обуку и тестирање. У свакој подели, показало се да су индекси теста већи него раније; стога је мешање у унакрсном валидатору било неприкладно. Другим речима, овај објекат унакрсне провере је варијација K преклапања, где у k -тој подели враћа првих k преклапања као скуп за тренинг, а преклапање $(k + 1)$ као тест скуп.

- I. **ARIMA**- У нашим експериментима тестиране су различите вредности параметара p , d и q а за даље тестирање је одабран ARIMA модел са најмањом RMSE грешком. За променљиву p тестиране су вредности параметара 0, 1, 2, 4, 6, 8 и 10, док су вредности d и q тестиране за распон вредности од 0 до 3. Комбинација параметара (p, d, q) која је показала најбоље перформансе ARIMA модела за излазне вредности променљиве BEE_OUT и BEE_IN је $(p, d, q) = (10, 0, 2)$, док је за BEE_IN изабрана комбинација $(0, 0, 2)$;
- II. **Facebook Prophet**- Тестиране су различите комбинације улазних променљивих из табеле 1, али су најбољи резултати добијени коришћењем следећих променљивих: AM2302_1_Temp, AM2302_1_Humi, AM2302_2_Temp HIVE, AM2302_2_Humi HIVE, MHRD_rain, MQ135_PPM, BH1750_1x, VEMML6750_uvindex, Day_night;

Параметри који су индицирали највећи утицај на кретања пчела коришћени за израду модела предвиђања били су: температура ваздуха унутар и изван кошнице, релативна влажност ваздуха унутар и изван кошнице, присуство кише, квалитет ваздуха, опсег и интензитет дневне светлости, UV зрачење и смена ноћи и дана, тј. нове променљиве активни и неактивни део дана од 24 часа.

Предвиђање за целу BEE_OUT временску серију приказано је на слици 11.1. Овај резултат је дат само да би се стекао увид у то да Facebook Prophet може успешно да учи из посматраних временских серија. У резултатима, комплетно предвиђање временске серије и представљене метрике су засноване на претходно невидљивим подацима (у питању су тест подаци).

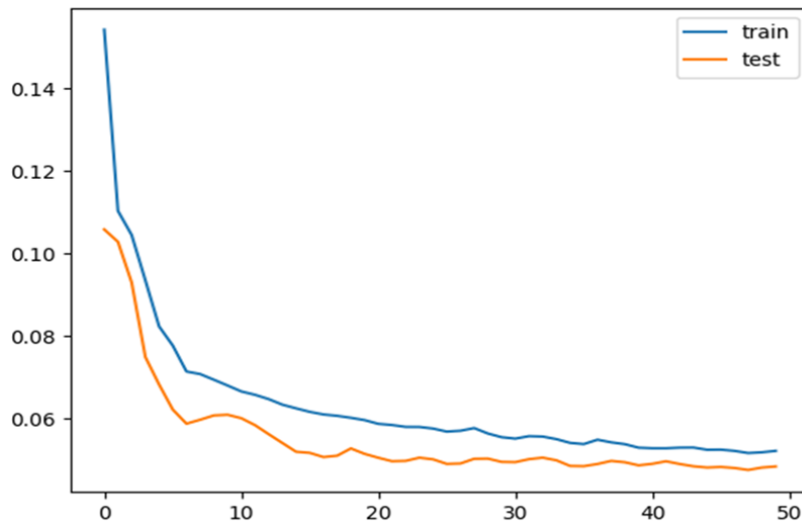


Слика 11.1 Предвиђање за целу временску серију (20 дана) BEE_OUT и BEE_IN .

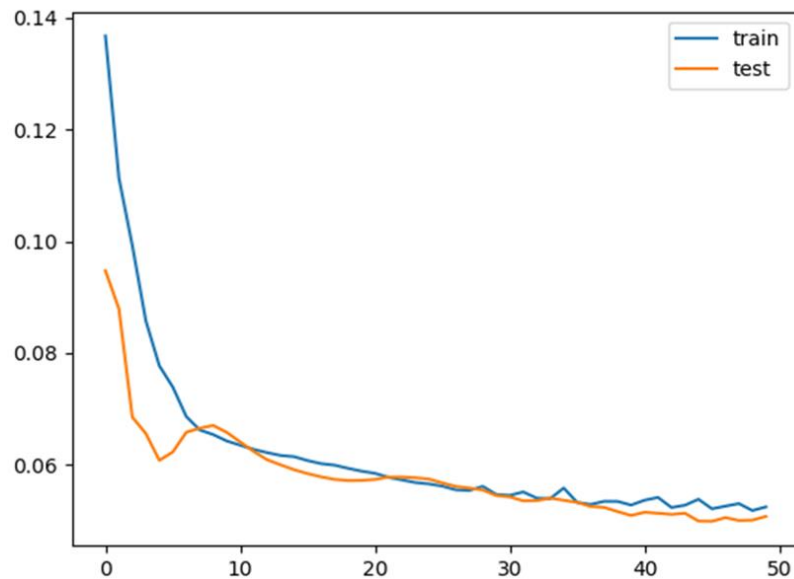
- III. **Рекурентне неуронске мреже**- овде је први корак припрема скупа података ВЕЕ за LSTM. Ово укључује смештање скупа података у оквиру проблема надгледаног учења и нормализацију улазних променљивих. Овде су коришћене исте променљиве које користи и Facebook Prophet алгоритам. Проблем надгледаног учења је уоквирен као предвиђање изласка или уласка пчела у тренутном сату (t) с обзиром на мерење излаза или улаза пчела, и временских услова у претходном временском кораку. Након овог корака трансформације, десет улазних променљивих (улазна серија) и једна излазна променљива (излазак пчела или улаз у тренутном сату) су:

$$var1(t-1), var1(t-1), \dots, var10(t-1), var1(t) \quad (11.1)$$

Дефинисали смо LSTM са 50 неурона у првом скривеном слоју, и 1 неуроном у излазном слоју за предвиђање активности пчела. Улазни облик је био један временски корак са 10 карактеристика. Средња апсолутна грешка (The Mean Absolute Error MAE) је коришћена као функција за израчунавање губитка и ефикасна Adam верзија стохастичког градијентног спуштања. Модел је био подешен за 50 тренинг епоха са величином серије од 20 (batch size). Коначно, пратили смо губитке и за тренинг и за тест током фазе тренинга. На крају тренинга, учртани су губици и на тренингу и на тесту. Добијене криве губитака током фазе обуке и валидације за BEE_OUT и BEE_IN излазе, приказане су на сликама 11.2 и 11.3.

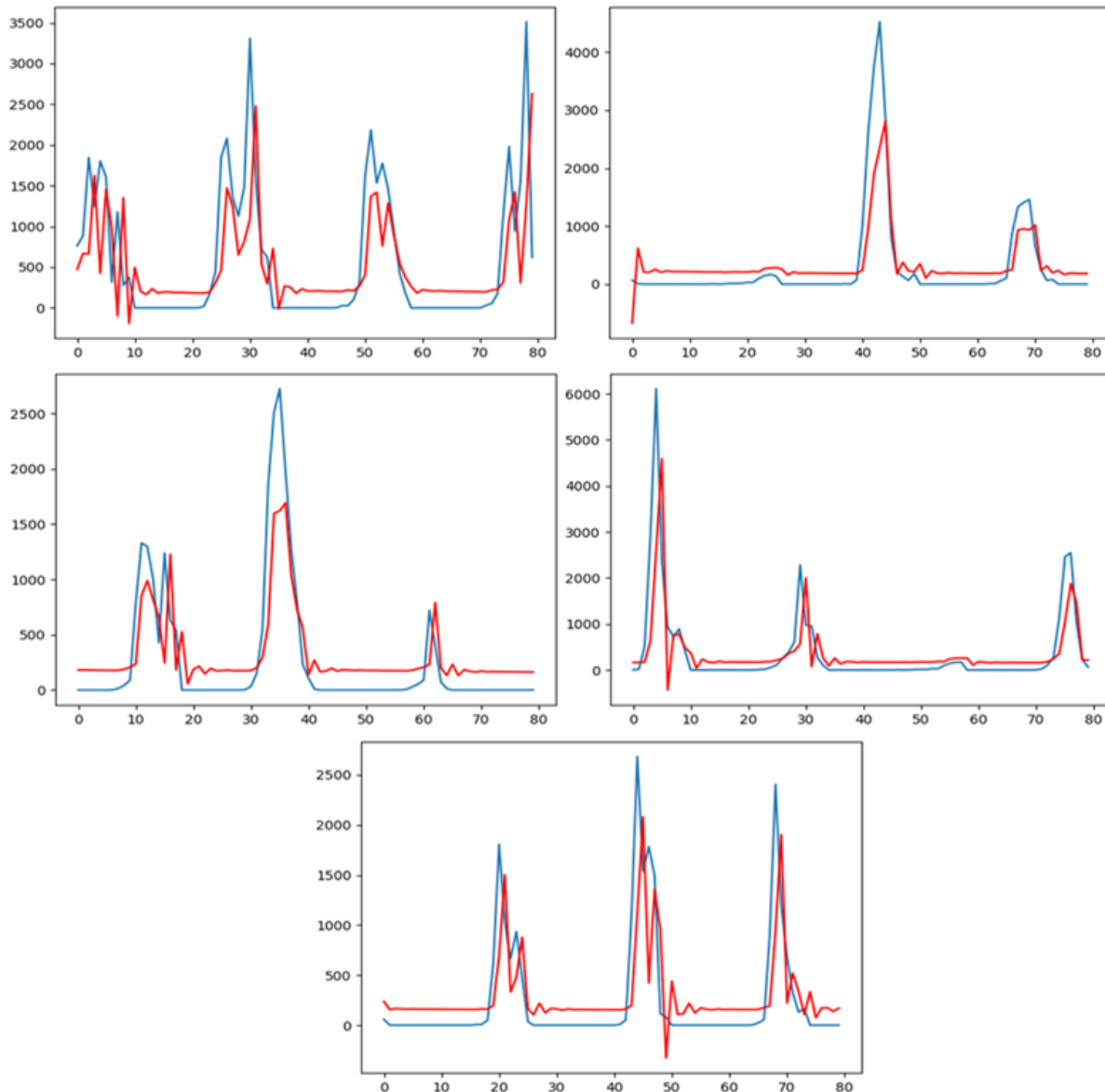


Слика 11.2 Вее OUT фаза тренинга.



Слика 11.3 Вее IN фаза тренинга.

Одвојена предвиђања временских серија на скупу тестова за свако преклапање су приказана на слици 11.4. Машинско учење (ML) примењено на податке временских серија, у овом случају на рекурентне неуронске мреже, је ефикасан и ефективан начин за анализу података, примену алгорита предвиђања и извођење тачне предикције.



Слика 11.4 Плава линија - Оригинални сегмент временске серије BEE_OUT; Црвена линија- предвиђање за свако преклапање (петострука подела временске серије).

Сви збирни резултати су приказани у табели 11.1. Најбољи резултати су постигнути коришћењем рекурентних неуронских мрежа, где је просечна грешка RMSE на тест сетовима била 426.49 за BEE_OUT временску серију; за временску серију BEE_IN RMSE је имао вредност од 378.464.

Табела 11.1 Резултати коришћењем рекурентне неуронске мреже.

MODEL	CV Тест RMSE OUT	CV Тест RMSE IN
ARIMA	894.92	511.77
Facebook Prophet	589.97	475.25
LSTM	426.49	378.464

Приказ сумираних параметара који су дали најбоље резултате код сваке методе посебно, дати су у табели 11.2 као поређење.

Табела 11.2 Поређење сумираних оптималних параметара за све испитиване методе.

Алгоритамски параметри	BEE OUT/IN
ARIMA	За BEE OUT (p, d, q): (10, 0, 2) За BEE IN (p, d, q): (0, 0, 2)
Facebook Prophet	Годишња сезоналност: false Недељна сезоналност: false Дневна сезоналност: true
Recurrent Neural Networks	RNN тип ћелије: LSTM LSTM број: 50 Функција губитка: mean absolute error Величина серије: 20 Оптимизатор: Adam Стопа учења: 1e-3 Број Епоха: 50

11.2. Резултати ИоТ Система

Као први резултат дисертације истиче се изградња ИоТ Система који је био есенцијалан за све добијене резултате приказане у дисертацији, као и будућа истраживања (више о томе у закључку). Инсталирани систем је омогућио прикупљање података који су били основ за даљу анализу података, корелацију параметара и моделовање предикције. Кроз преглед научне литературе и референтних радова, није пронађен систем који би могао да испуни специфичне захтеве овог истраживања. Као одговор на ту потребу, развијен је и имплементиран систем заснован на Atmel и Espressif чиповима, са посебним фокусом на слој перцепције. За слој перцепције примењени су комерцијални сензори, са карактеристикама високе прецизности. Сензори прикупљају податке са стратешких позиција у кошници, обезбеђујући квалитетне и прецизне податке представљене у раду. Сензори у временским оквирима снимају податке и исти се путем ESP32 микроконтролера уписују локално правећи јединствени фајл. Главна јединица у дефинисаним временским итерацијама преко SIM модула шаље податке серверу за упис у базу. Вишеструки је допринос израде приказаног ИоТ Система, који представља основу за даља истраживања и проширења, детаљније описана у наредном поглављу (Закључак). Увид у здравље кошница у реалном времену, могућност благовремене акције и реакције су огроман потенцијал

имплементације ИоТ технологије. Откривање и уклањање потенцијалних проблема чији је највећи иницијатор човек, исти је дужан и да их реши.

Један од значајних доприноса овог система је могућност непрекидног праћења виталних параметара пчелињег друштва у реалном времену. Подаци који се прикупљају омогућавају пчеларима да имају увид у виталне параметре и активности пчела, као и да благовремено реагују на евентуалне проблеме или промене. Ово је од великог значаја за очување здравља пчела и оптимизацију пчеларских пракси. Систем омогућава аутоматско бројање и праћење улазака и излазака пчела из кошнице. Коришћењем рефлектујућих фото-отпорника на улазу, детектује се правац кретања пчела, омогућавајући прецизно бројање и праћење њихових активности. Ови подаци се заједно са осталим мерењима бележе у временским серијама, обезбеђујући корисне информације за даљу анализу и интерпретацију. Ова информација је драгоцену за праћење популације пчела, идентификацију могућих проблема и боље разумевање њиховог понашања.

Овај ИоТ систем представља значајан допринос у домену пчеларства. Омогућава пчеларима да користе напредне технологије за праћење и управљање пчелињим друштвима, унапређујући њихову продуктивност и здравље. Такође, отвара могућности за даља истраживања и развој нових метода за праћење и заштиту пчела. Овај ИоТ систем за пчеларство представља иновативно решење које пружа висок ниво аутоматизације и прецизности у праћењу пчелињих друштава. Његов допринос се огледа у могућности благовременог откривања проблема, унапређењу пчеларских пракси и очувању здравља пчела, чиме се пружа могућност одрживог развоја пчеларства.

Други допринос ове дисертације је сервиски део апликације и база података, који су детаљније описани у поглављу 9. Технологије на којима су базирани серверски део апликације и база података су PHP, Laravel и MySQL. Код је дизајниран тако да прихвати податке који се шаљу са ИоТ Система, упише их у базу података која даље снабдева фронтални део апликације. Web сервис приказан у дисертацији је кључан у оваквим приступима, његова практичност и доступност омогућава контролна и апликативна подешавања без обзира локацију. То иде паралелно са правовременом реакцијом која се очекује од примене оваквог система.

Трећи део доприноса је фронтални део апликације коју чини респонзивни интерфејс задужен за приказ података са базе и аналитичким алармом. Апликацији се може приступити са свих уређаја који имају приступ интернету без обзира на оперативни систем или хардверску конфигурацију. Други и трећи допринос заједно чине Cloud систем, део неопходан за повезивање кошнице са корисником који гаји и негује пчеле. Апликација је организована тако да се лакше манипулише кошницама и сензорима који припадају кошници. Предност је могућност детаљног подешавања кошнице и сензора, преглед и статистика кретања параметара. Статистика је важна због праћења тренда кретања параметара, одступања или отказивања сензора. Могућност подешавања граничних вредности параметара подиже вредност система, јер се укључивањем аналитичког праћења параметара омогућава активација аларма који се приказује на интерфејсу са нотификацијом.

Обзиром да су кошнице удаљене у природи, апликација са алармима игра важну улогу у брзом и адекватном реаговању.

11.3. Предиктивни API

Допринос система није само у прикупљању података, већ и у њиховој анализи и интерпретацији. Кроз имплементацију модула вештачке интелигенције на серверској страни, систем омогућава идентификацију образаца и понашања пчела. Аналитички алати прате податке у стварном времену, препознајући промене у параметрима и активирајући аларме у случају потенцијалних проблема. Ово пружа пчеларима могућност да брзо реагују и предузму одговарајуће мере за заштиту пчела.

Четврти део резултата је примена вештачке интелигенције и модула за предикцију који се базира на LSTM меделу неуронске мреже. Након тренирања и тестирања више модела најбоље резултате у предикцији кретања пчела на основу дефинисаних параметара даје модел описан у претходном поглављу. Циљ предикције је имплементација предиктивног API-ја у покретању аларма, чија се манифестација приказује на фронталном делу апликације.

Базиран на најбољем моделу креиран је API за предикцију интегрисан у апликацију који узима податке из базе, прати и анализира кретање параметара. Уколико дође до предикције одступања кретања пчела на основу неког од параметара, шаље се нотификација на интерфејс апликације (детаљан опис дат у поглављу 9). део кода за слање и упис обавештења дат је у наставку.

Дефинисање путање и методе API endpointa.

```
@app.route("/bee-network-training", methods=["GET"])
def network_training():
    # ...
    return "Model training phase is finished"
```

Овај део кода дефинише путању **/bee-network-training** и методу **GET** за API endpoint. Приликом приступа овом путу преко HTTP GET захтева, извршиће се функција **network_training()**. Тренинг и тест модел су претходно појашњени код дефинисања и примене модела.

Предвиђање на тест скупу и слање обавештења.

```
yhat = model.predict(test_X)
# ...
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:, 0]

# Slanje obaveštenja na osnovu predikcije
if np.any(inv_yhat > threshold):
```

```
send_notification()
```

Овде се врши предвиђање на тест скупу података помоћу функција **predict()**. Након инверзног скалирања, добијене предвиђене вредности се чувају у **inv_yhat**. Затим се проверава да ли било која од тих предвиђених вредности прелази одређени праг **threshold**. Ако је неко од предвиђања изнад прага, позива се функција **send_notification()** која може бити имплементирана за слање обавештења на основу предвиђања.

Чување обавештења о предвиђањима.

```
model_json_name = 'bee_lstm.json'
if not os.path.isfile(model_json_name):
    json_config = model.to_json()
    with open(model_json_name, 'w') as json_file:
        json_file.write(json_config)
```

Овде се чува структура модела у JSON формату у фајлу **bee_lstm.json**. Ако фајл не постоји, конфигурација модела се записује у фајл.

11.4. Предности примена неуронских мрежа у поређењу са другим статистичким моделима

Уколико резултате упоредимо са стандардним статистичким методама и моделима, видимо супериорност примене неуронских мрежа у предикцији. Могућност прилагођавања модела неуронских мрежа различитим типовима података, учење модела и тренинг кроз вишеслојну мрежу неурона, примена великог броја параметара за добијање прецизнијег излаза и резултата предикције. За демонстрацију разлике коришћен је метод линеарне регресије.

За предвиђање зависне варијабле примењена је OLS (Ordinary Least Squares) метода, односно метода најмањих квадрата. OLS је класична метода линеарне регресије која се користи за проналажење најбољег линеарног модела који минимизира збир квадрата разлика између стварних вредности зависне променљиве и предвиђених вредности добијених линеарном комбинацијом независних променљивих. Ова метода се често користи за предвиђање континуираних нумеричких вредности. Резултати су приказани у табелама 11.3 и 11.4.

Табела 11.3 Приказ резултата модела за BEE_messageIN.

Модел 2: OLS, користећи обсервације 1:01-12:05 (T = 269)				
Зависна променљива: BEE_messageIN				
Променљива	Коефицијент	Стандардна грешка	t-ratio	p-вредност
<i>const</i>	-753.449	657.452	-1.146	0.2529
<i>AM2302_1_Temp</i>	50.7499	35.9722	1.411	0.1595
<i>AM2302_1_Humi</i>	-15.4409	3.80868	-4.054	6.67e-05
<i>AM2302_2_TempHIVE</i>	-91.4308	24.3771	-3.751	0.0002
<i>AM2302_2_HumiHIVE</i>	0.421827	4.48443	0.09406	0.9251

<i>MHRD_rain</i>	0.157159	0.215692	0.7286	0.4669
<i>MQ135_PPM</i>	-239.362	194.535	-1.230	0.2197
<i>BH1750_lx</i>	0.0555760	0.0199731	2.783	0.0058
<i>VEML6750_uvindex</i>	-186.900	78.6369	-2.377	0.0182
<i>SI1145_visible</i>	8.71177	2.05020	4.249	3.00e-05
<i>SI1145_IR</i>	-0.978750	0.243537	-4.019	7.69e-05

У следећој табели видимо информације о резултатима прилагођавања модела подацима, што сугерише о значајности статистичких модела (табела 11.4). Искључујући константу, највећа р-вредност је забележена за променљиву *AM2302_2_HumiHIVE*.

Табела 11.4 Значајност статистичких модела за *BEE_messageIN*.

Статистичка мера	Вредност
<i>Mean dependent var</i>	626.9851
<i>S.D. dependent var</i>	882.5902
<i>Sum squared resid</i>	80516694
<i>S.E. of regression</i>	559.7272
<i>R-squared</i>	0.614315
<i>Adjusted R-squared</i>	0.597807
<i>F(11, 257)</i>	37.21333
<i>P-value(F)</i>	5.02e-47
<i>Log-likelihood</i>	-2077.640
<i>Akaike criterion</i>	4179.281
<i>Schwarz criterion</i>	4222.417
<i>Hannan-Quinn</i>	4196.605
<i>rho</i>	0.391670
<i>Durbin-Watson</i>	1.215943

Приказ резултата за релацију параметара и изласка пчела из кошнице је дат у следећој табели.

Табела 11.5 Приказ резултата модела за *BEE_messageOUT*.

Модел 1: OLS, користећи обсервације 1:01-12:05 (T = 269)				
Зависна променљива: <i>BEECNT_messageOUT</i>				
Променљива	Коефицијент	Стандардна грешка	t-ratio	р-вредност
<i>const</i>	-60.4557	790.090	-0.07652	0.9391
<i>AM2302_1_Temp</i>	52.6183	43.2294	1.217	0.2246
<i>AM2302_1_Humi</i>	-19.7141	4.57706	-4.307	2.36e-05
<i>AM2302_2_TempHIVE</i>	-77.8305	29.2950	-2.657	0.0084
<i>AM2302_2_HumiHIVE</i>	5.91006	5.38914	1.097	0.2738
<i>MHRD_rain</i>	0.0856999	0.259207	0.3306	0.7412
<i>MQ135_PPM</i>	-253.163	233.781	-1.083	0.2799
<i>BH1750_lx</i>	0.0541695	0.0240026	2.257	0.0249
<i>VEML6750_uvindex</i>	-139.166	94.5016	-1.473	0.1421
<i>SI1145_visible</i>	4.65956	2.46382	1.891	0.0597

SI1145_IR	-0.577104	0.292670	-1.972	0.0497
------------------	-----------	----------	--------	--------

Искључујући константу, највећа р-вредност је забележена за променљиву MHRD_rain, што говори о најмање значајној променљивој за предикцију. Значајност статистичких модела је дата у следећој табели 11.6.

Табела 11.6 Значајност статистичких модела BEECNT_messageOUT.

Статистичка мера	Вредност
Mean dependent var	805.1784
S.D. dependent var	1023.312
Sum squared resid	1.16e+08
S.E. of regression	672.6498
R-squared	0.585657
Adjusted R-squared	0.567922
F(11, 257)	33.02350
P-value(F)	4.06e-43
Log-likelihood	-2127.076
Akaike criterion	4278.152
Schwarz criterion	4321.289
Hannan-Quinn	4295.476
rho	0.407914
Durbin-Watson	1.183403

Модел 1 (OLS) за улазак пчела (BEE_messageIN), даје резултате да је R-квадрат вредност од 0.614315 што указује да модел има преклапање за око 61.43% са улазним подацима. Р-вредност за F-тест је врло мала (5.02e-47), што сугерише да је модел статистички значајан.

Модел 2 (OLS) за улазак пчела (BEECNT_messageOUT), даје резултате да је R-квадрат вредност од 0.585657 што указује да модел има преклапање за око 58.57% са улазним подацима. Р-вредност за F-тест је врло мала (4.06e-43), што сугерише да је модел статистички значајан.

У поређењу са неуронским мрежама ови статистички модели показују инфериорност што можемо видети у табели 11.7.

Табела 11.7 Резултати свих модела.

Модел	CV Test RMSE OUT	CV Test RMSE IN
ARIMA	894.92	511.77
Facebook Prophet	589.97	475.25
LSTM	426.49	378.464
Модел : OLS	672.6498	559.7272

12. ЗАКЉУЧАК

Резултат ове дисертације је израда система за мониторинг и предикцију услова у кошницама. Представљени систем је базиран на ИоТ и cloud технологијама, вештачкој интелигенцији и анализи и обради података, чиме су утврђене корелације зависних и независних променљивих, као и најчешћи узрочници кретања пчела. У дисертацији је, кроз поглавља, представљен цео ток развоја наведеног система у фазама.

Мотивација и циљ рада произилазе из реалних проблема у непосредном окружењу. Третирањем биљака на неадекватан начин и гушењем природе, латентно се руши природни поредак. Сви фактори који су криви за то су око нас а и сами смо узрочници већине. То питање остаје и даље отворено али смо све ближи решењу, а ова дисертација је један корак ка томе. Водећи се тиме и науком коју желим да наставим да изучавам, а то је примена ИоТ, cloud технологија, као и вештачке интелигенције у пољопривреди, дефинисане су основне хипотезе, методе и технике.

Изради система претходила је детаљна анализа дотадашњих истраживања и решења, представљена литература из поглавља 3 у дисертацији је прочитана и анализирана (више од 100 радова и пројеката). Како би се утврдио значај сваког параметра који је снимљен, искуства из претходних радова су била веома значајна. Обављени су разговори са пчеларима и стручним лицима о томе шта су проблеми и како их решити, што је било од велике користи, прикупљене су квалитетне инструкције и савети за примену важних елемената приликом имплементације система. Два решења су проистекла из тога, аутоматска вентилација и хранилица. Примена ИоТ и cloud технологија је веома захтевна али због своје хетерогености веома остварљива.

Проблеми и изазови савременог пчеларства описани у поглављу 4, указују на читав аспект примене дигиталних технологија у пољопривреди која је сама по себи аналогна. Проширивањем сазнања из ове области увиђамо да није све баш лако применљиво и да постоје рестрикције и ограничења. Имплементација нечег новог и савременог са добрим намерама не значи нужно позитиван исход. Примену технологија треба пратити и константно усавршавати прилагођавајући је природном амбијенту са што мање последица.

Технологија ИоТ је једна од најкомплекснијих и најпримењенијих технологија данашњице, са трендом сталног раста. У поглављу 5 детаљно је описана ИоТ архитектура, подела и примена, технологије бежичних умрежавања које представљају кључни део имплементације. Тренд напретка ИоТ технологија се од трослојне архитектуре проширио на вишеслојну комплексну архитектуру која се проширила на све области информационих технологија, развијајући нове правце и примене.

Израда сваке компоненте посебно, описане у поглављима дизајна Система, захтевала је темељан приступ и месеце тестирања, предан и интензиван рад. Први модел иако не импресиван по изгледу потврдио је труд добрим резултатима. Инсталација и повезивање сензора у главном уређају претходила је инсталацији сензора и уређаја у кошници. Избор сваког сензора вршен је појединачно, набављани су различити типови сензора, различитих

произвођача, чему је претходило истраживање, анализа корисничких искустава и препоруке. Уређај за детекцију кретања пчела и бројач је био најзахтевнији и најскупљи. За његову израду, тестирање и писање кода потрошено је скоро годину дана. Постављени захтеви минимализације целог система отежали су дизајн читавог система. Као што се може видети на сликама компликовано је конектовање свих сензора, модула, регулатора напона, релеја, напонске секције на једном малом месту. Након израде система тестирана је свака јединица посебно, како би се исправиле грешке у коду и како би систем могао да се оптимизује. Било је потребно да све ради и функционише пре него што се монтира и интегрише са кошницом.

Тест фаза целог система показала је много фактора које је требало унапредити, јер је систем комплексан са више јединица. Јавили су се проблеми са потрошњом струје, SIM модул је био најзахтевнији приликом слања података, тако да су се јавили комплексни проблеми који су на крају успешно превазиђени. Поједини сензори су имали проблема са измереним вредностима у раним јутарњим сатима када је влажност ваздуха била преко 90% што је захтевало замену сензора за отпорније моделе, и преправку кода за контролу вредности. Систем је тако тестиран део по део, а затим цео склоп.

Након инсталације цео систем је тестиран на празној кошници, после утврђивања да је систем функционалан прешло се на следећу фазу. Убацивање матице и пчела су одрађени без проблема и пчеле и електроника су интегрисани у један екосистем. Нису уочени проблем у дневним активностима код пчела. Када је кошница постала активна није било могуће радити додатне интервенције. Са првом тестираном кошницом снимљено је било око 90 дана, а у раду се разматра најбољи узорак од 20 везаних дана ране јесени. Активности у осталим данима нису биле толико занимљиве за истраживање јер је почео касни јесењи период, кише, слаба активност пчела.

Анализа обављена над прикупљеним подацима указала је на корелацију фактора са кретањем пчела, где се и по графицима приказаним у претходном сегменту виде кључни узрочници кретања. Пошто је тип података добијен мерењима представљен у временским серијама, изабрани су модели за предикцију који су најпогоднији оваквој обради података. Моделовање предиктивног алгоритма тестирано је помоћу три модела а модел са применом рекурентних неуронских мрежа дао је најбоље резултате. Поређење експерименталних података са моделом показало је да наш модел добро представља посматране процесе. На то указују резултати приказани на сликама претходних секција. Према добијеним резултатима, најбољи модел би могао да постигне поуздано предвиђање активности пчела, са грешком од само 8,9 пропуштених пчела на сат за изласке пчела из и 7,8 пропуштених пчела на сат за улазак пчела у кошницу. Направљен је API за предикцију за интеграцију са апликацијом.

Софтверски део система описан је детаљно кроз поглавље 9, врло је захтеван и односи се на више сегмената. Први се односи на писање кода за сензорске модуле и микроконтролере на Edge слоју, други је серверска страна на cloud-у а трећи клијентски део система. Имплементација нових технологија захтевала је модеран приступ дизајну апликације и серверских услуга, а то је web респонзивна апликација. Систем одлично

функционише у функционалним захтевима и покрива сет додатних функција, као што су аларми, графици, информације о тренду кретања параметара, информације о сензорима на основу којих се добијају кључне информације. Вештачка интелигенција је неизоставан део сваке апликације од које се захтева изведена, управљачка или контролна информација. Цео систем је функционалан али као и све досад описано има јако пуно простора за унапређивање.

Круна свега досад наведеног је рад на SCI листи који је наведен у референтном делу у часопису са импакт фактором 2.69 и категорисан је у M22. Референце обухватају и радове везане за дисертацију а објављене у домаћим часописима и на конференцијама [102]-[105].

Постоје и нека ограничења за приказани приступ. Фаза тестирања је спроведена на две кошнице, а главни подаци су прикупљени из једне кошнице која није померена током експеримента. Експеримент је спроведен у периоду када није било цветања природе која је главни извор хране, и када је активност пчела била мања од оне током пролећа и лета.

Када је у питању наставак истраживања, отвара се више могућих праваца за даље истраживање. Нови модели система који су настали након првог имају квалитетније моделе сензора, унапређен софтверски део апликације на edge слоју и слоју перцепције, бољу интеграцију у кошници (дизајн саме кошнице са сензорима), као и напреднију саму главну јединицу система.

Наредна истраживања могу да се баве другачијим приступом и анализом података о звучним фреквенцијама и јачини звука. Над таквим подацима је потребна дубока анализа и примена метода вештачке интелигенције која би дала боље резултате предикције користећи аналитику звука.

Квалитет ваздуха и загађења су фокус истраживања наредних година. Детекција гасова и различитих врста агенаса је од круцијалног значаја за пчеларе и сам биодиверзитет. Пчеле су активне током целог дана, а пољопривредници са великим бројем хектара не могу чекати време када пчеле престају са активностима, већ опрашују и третирају културе машинама и дроновима. На тај начин пчеле са собом уносе и супстанце садржане у хемијским третманима. Такође проценат угљендиоксида у кошници постаје проблематичан за пчеле. Овде је јако битно користити квалитетне сензоре и кључне позиције за њихову инсталацију. Такође примена метода вештачке интелигенције над оваквим скупом података, због своје специфичности захтева адекватан модел.

Овде је имплементиран и тестиран систем за вентилацију који тек треба да се унапреди. Систем има паралелне вентилаторе који су величине 80x80, веома су тихи и са малим захтевима за напајање од свега 5 V. Систем се активира на два начина, један је када детектује високу влажност, а други при детекцији одређених гасова и дима. Наредна фаза је оптимизација и самосталност система на основу броја обртаја и детекције поменутих параметара. Вентилатори немају директно усмерење ка средини кошнице, већ имају дефлекторе. Такође за контролу и управљање оваквог аутоматизованог система примена модела вештачке интелигенције је неминовна. Систем је успео да обори влажност у

кошници али остаје још фаза тестирања и потврђивања резултата, стога у дисертацији овај део није знатније разматран.

Будућа истраживања могу да се фокусирају на храну и воду коју пчеле немају у јесењем и зимском периоду. Систем за аутоматске хранилице и појилице направљен је за експерименталну кошницу, смештен је испод поклопца на горњем делу кошнице и комплетно је функционалан. Међутим постоје проблеми када су у питању сирупи и храна, а то су паковања из којих се точи сируп који служи за исхрану пчела.

За наредна истраживања, систем приказан у дисертацији ће бити надограђен одговарајућим сензорима за мерење тежине кошнице, сензорима кисеоника/угљен-диоксида, аутоматским храњењем пчела, вентилацијом и системима за затварање капија, као и повезивањем са другим апликацијама и решењима. Већина ових решења је у фази тестирања, тако да ће резултати бити публиковани кроз будуће радове.

Поред хардверских надоградњи ИоТ систем ће бити надограђен и у софтверском апликативном делу. Биће тестиране нове методе вештачке интелигенције за моделовање предикције користећи друге факторе и комбиноване корелације зависних променљивих како би се унапредио систем. Сваки сет података има потенцијалне могућности интеграције у систем за предикцију.

Наредна истраживања могу да се оријентишу ка примени термалних сензора, ради изучавања унутрашњих активности пчелињег друштва и њихову корелацију са осталим параметрима.

Будући радови укључиће и опсежна истраживања која обухватају анализу утицаја микроталаса и присуства електронских компоненти на кретање и живот пчела. Такође је неопходно укључити време као посебан фактор у доношењу закључака, јер из дужег временског периода долази до искуствених закључака, јер свака промена, мерење или анализа захтева одређено време да би резултати били квалитативни.

Очекује се, можда и повећање броја грешака по сату када се мерења врше у пролећним и летњим месецима, што ће узроковати и да се додатно побољшају карактеристике инжењеринга модела.

Примена напредних модела информационих технологија омогућава откривање наглих одступања и поремећаја од нормалног живота пчела и предвиђање потенцијалних узнемирујућих промена. Показали смо да су применом напредних алгоритама могућа свакодневна предвиђања високе прецизности. На овај начин, применом апликације за праћење у реалном времену и pushup обавештења о потенцијалним променама, пчелар има увид у стање у кошницама у реалном времену и може адекватно да реагује како би спречио нежељене исходе.

У дисертацији је показано да развој ИоТ-а помаже у имплементирању савремених технологија у пољопривреду. Сами подаци говоре пуно али изведени подаци анализирани применом вештачке интелигенције отварају могућности превентивне реакције. Будућност апикултуре је управљање кошницама што ће подићи пчеларство на нови ниво. Емпиријско

праћење напредовања оваквих система је кључно уколико желимо да резултати и исходи за пчеларе буду позитивни.

Константне климатске промене мењају природу и указују нам да су дешавања у природи процес који траје, и да их ми морамо пратити како бисмо научили да спречимо девијантне исходе који нарушавају екосистем биљака и животиња, нарочито пчела. Пчеле нас својим понашањем уче о позитивном колективном деловању на околину, па морамо и ми узвратити и сачувати их од нас.

Све компоненте развијеног система су показале своју функционалност и практичну применљивост, што је поткрепљено експерименталним подацима. Систем је тако конципиран да се може проширивати на разне начине, од којих су неки идентификовани и представљени у овом закључку, те су тиме створене могућности за даљи рад и истраживања.

Мед треба да буде последица здравог пчелињег друштва, јер од активности пчела зависе многе природне појаве и стања, а мед је само његов најслађи део.

ЛИТЕРАТУРА

- [1] Van Der Zee, R.; Pisa, L.; Andonov, S.; Brodschneider, R.; Charrière, J.-D.; Chlebo, R.; Coffey, M.; Dahle, B.; Gajda, A.; Gray, A.; et al. Managed Honey Bee Colony Losses in Canada, China, Europe, Israel and Turkey, for the Winters of 2008–9 and 2009–10. *Journal of Apicultural Research* **2012**, *51*, 100–114, doi:10.3896/ibra.1.51.1.12.
- [2] Bacandritsos, N.; Granato, A.; Budge, G.E.; Papanastasiou, I.; Roinioti, E.; Caldon, M.; Falcaro, C.; Gallina, A.; Mutinelli, F. Sudden Deaths and Colony Population Decline in Greek Honey Bee Colonies. *Journal of Invertebrate Pathology* **2010**, *105*, 335–340, doi:10.1016/j.jip.2010.08.004.
- [3] Ngo, T.H.; Rustia, D.J.A.; Yang, E.-C.; Lin, T.-T. Honey Bee Colony Population Daily Loss Rate Forecasting and an Early Warning Method Using Temporal Convolutional Networks. *Sensors* **2021**, *21*, 3900, doi:10.3390/s21113900.
- [4] Hristov, P.Z.; Shumkova, R.; Palova, N.; Neov, B. Factors Associated with Honey Bee Colony Losses: A Mini-Review. *Veterinary Sciences* **2020**, *7*, 166, doi:10.3390/vetsci7040166.
- [5] Watson, K.H.; Stallins, J.A. Honey Bees and Colony Collapse Disorder: A Pluralistic Reframing. *Geography Compass* **2016**, *10*, 222–236, doi:10.1111/gec3.12266.
- [6] Braga, A.L.; Gomes, D.G.; Rogers, R.; Hassler, E.; Freitas, B.M.; Cazier, J.A. A Method for Mining Combined Data from In-Hive Sensors, Weather and Apiary Inspections to Forecast the Health Status of Honey Bee Colonies. *Computers and Electronics in Agriculture* **2020**, *169*, 105161, doi:10.1016/j.compag.2019.105161.
- [7] Clarke, D.M.; Robert, D. Predictive Modelling of Honey Bee Foraging Activity Using Local Weather Conditions. *Apidologie* **2018**, *49*, 386–396, doi:10.1007/s13592-018-0565-3.
- [8] Zabasta, A.; Zhiravetska, A.; Kunicina, N.; Kondratjevs, K. *Technical Implementation of IoT Concept for Bee Colony Monitoring*; **2019**, doi: 10.1109/MECO.2019.8760180,
- [9] Muhammad, Z.; Saxena, N.; Qureshi, I.M.; Ahn, C.W. Hybrid Artificial Bee Colony Algorithm for an Energy Efficient Internet of Things Based on Wireless Sensor Network. *IETE Technical Review* **2017**, *34*, 39–51, doi:10.1080/02564602.2017.1391136.
- [10] Komasilovs, V.; Zacepins, A.; Kviesis, A.; Fiedler, S.; Kirchner, S. Modular Sensory Hardware and Data Processing Solution for Implementation of the Precision Beekeeping. *Agronomy Research* **2019**, *17*, 509–517, doi:10.15159/ar.19.038.
- [11] Zacepins, A.; Brusbardis, V.; Meitalovs, J.; Stalidzans, E. Challenges in the Development of Precision Beekeeping. *Biosystems Engineering* **2015**, *130*, 60–71, doi:10.1016/j.biosystemseng.2014.12.001.

- [12] Zacepins, A.; Kviesis, A.; Pecka, A.; Osadcuks, V. Development of Internet of Things concept for Precision bee-keeping. In Proceedings of the 2017 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 28–31 May **2017**; pp. 28–31.
- [13] Henry, E.; Adamchuk, V.I.; Stanhope, T.P.; Buddle, C.M.; Rindlaub, N. Precision Apiculture: Development of a Wireless Sensor Network for Honeybee Hives. *Computers and Electronics in Agriculture* **2019**, *156*, 138–144, doi:10.1016/j.compag.2018.11.001.
- [14] Catania, P.; Vallone, M. Design of an innovative system for precision beekeeping. IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), 24–26 October **2019**; pp. 323–327, doi: 10.1109/MetroAgriFor.2019.8909256.
- [15] Fiedler, S.; Zacepins, A.; Kviesis, A.; Komasilovs, V.; Wakjira, K.; Nawawi, M.; Hensel, O.; Purnomo, D. *Implementation of the Precision Beekeeping System for Bee Colony Monitoring in Indonesia and Ethiopia*; December **2020**; doi:10.1109/ICCC49264.2020.9257278.
- [16] Bumanis, N. *Data Fusion Challenges in Precision Beekeeping: A Review*; 2020; doi: 10.22616/rrd.26.2020.037
- [17] Zacepins, A.; Kviesis, A.; Komasilovs, V.; Muhammad, F. Monitoring System for Remote Bee Colony State Detection. *Baltic Journal of Modern Computing* **2020**, *8*, 461–470. doi:10.22364/bjmc.2020.8.3.05.
- [18] Pesovic, U.; Markovic, D.; Đurašević, S.; Randić, S. Remote Monitoring of Beehive Activity. *Acta Agriculturae Serbica* **2019**, *24*, 157–165, doi:10.5937/aaser1948157p.
- [19] Dogan, S.; Akbal, E.; Ozmen Koca, G.; Balta, A. Design of a remote Controlled Beehive for Improving Efficiency of Beekeeping Activities. In Proceedings of the 8th International Advanced Technologies Symposium (IATS'17), Elazig, Turkey, 19–22 October 2017; pp. 1084–1090.
- [20] Balta, A.; Dogan, S.; Ozmen Koca, G.; Akbal, E. Software Modeling of Remote Controlled Beehive Design. In Proceedings of the International Conference on Advances and Innovations in Engineering (ICAIE), Elazig, Turkey, 10–12 May 2017; pp. 1133–1137.
- [21] Gil-Lebrero, S.; Quiles-Latorre, F.J.; Ortiz-López, M.; Sánchez-Ruiz, V.; Gámiz-López, V.; Luna-Rodríguez, J.J. Honey Bee Colonies Remote Monitoring System. *Sensors* **2016**, *17*, 55, doi:10.3390/s17010055.
- [22] Rybin, V.G.; Rodionova, E.A.; Karimov, A.I.; Kopets, E.E.; Chernetskiy, E.S. Remote Data Acquisition System for Apiary Monitoring. In Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Russia, 26–29 January **2021**; pp. 1059–1062, doi: 10.1109/ElConRus51938.2021.9396576.

- [23] Sachin, K.; Gagana, M.R.; Rubab, H.; Jalaja, G.S.; Jayanand, J. Monitoring of Honey Bee Hiving System Using Sensor Networks. *International Journal of Engineering Research and Technology* **2020**, *V9*, doi:10.17577/ijertv9is060296.
- [24] Da Silva, D.F.; De Lima Rodrigues, Í.; Braga, A.R.; Nobre, J.S.; Freitas, B.M.; Gomes, D.G. *An Autonomic, Adaptive and High-Precision Statistical Model to Determine Bee Colonies Well-Being Scenarios*; June 2020, doi: 10.5753/wcama.2020.11017
- [25] Abou-Shaara, H.F.; Owayss, A.A.; Ibrahim, Y.Y.; Basuny, N. A Review of Impacts of Temperature and Relative Humidity on Various Activities of Honey Bees. *Insectes Sociaux* **2017**, *64*, 455–463, doi:10.1007/s00040-017-0573-8.
- [26] Kvišis, A.; Komasilovs, V.; Komasilova, O.; Zacepins, A. Application of Fuzzy Logic for Honey Bee Colony State Detection Based on Temperature Data. *Biosystems Engineering* **2020**, *193*, 90–100, doi:10.1016/j.biosystemseng.2020.02.010.
- [27] Meikle, W.G.; Weiss, M.; Maes, P.; Fitz, W.; Snyder, L.A.; Sheehan, T.; Mott, B.M.; Anderson, K.E. Internal Hive Temperature as a Means of Monitoring Honey Bee Colony Health in a Migratory Beekeeping Operation before and during Winter. *Apidologie* **2017**, *48*, 666–680, doi:10.1007/s13592-017-0512-8.
- [28] Zacepins, A. Application of bee hive temperature measurements for recognition of bee colony state. In Proceedings of the 5th International Scientific Conference on Applied Information and Communication Technologies, Jelgava, Latvia, 26–27 April 2012; pp. 216–221.
- [29] Szczurek, A.; Maciejewska, M.; Bąk, B.; Wilde, J.; Siuda, M. Semiconductor Gas Sensor as a Detector of Varroa Destructor Infestation of Honey Bee Colonies – Statistical Evaluation. *Computers and Electronics in Agriculture* **2019**, *162*, 405–411, doi:10.1016/j.compag.2019.04.033.
- [30] Bromenshenk, J.J.; Henderson, C.J.; Seccomb, R.A.; Welch, P.M.; Debnam, S.; Firth, D. Bees as Biosensors: Chemosensory Ability, Honey Bee Monitoring Systems, and Emergent Sensor Technologies Derived from the Pollinator Syndrome. *Biosensors* **2015**, *5*, 678–711, doi:10.3390/bios5040678.
- [31] Cecchi, S.; Terenzi, A.; Orcioni, S.; Riolo, P.; Ruschioni, S.; Isidoro, N. A Preliminary Study of Sounds Emitted by Honey Bees in a Beehive. In *Audio Engineering Society Convention 144*; Audio Engineering Society: Milan, Italy, **2018**.
- [32] Mezquida, D.A.; Martínez, J.L. Short Communication. Platform for Bee-Hives Monitoring Based on Sound Analysis. A Perpetual Warehouse for Swarm Apos;s Daily Activity. *Spanish Journal of Agricultural Research* **2009**, *7*, 824, doi:10.5424/sjar/2009074-1109.
- [33] Du, N.H.; Dong, N.P.; Luu, V.D.; Van Hoang, N.; Thai, P.Q.; Ngoc, P.K.; Long, N.T.; Hong, P.N. Toward Audio Beehive Monitoring Based on IoT-AI Techniques: A Survey and Perspective. *Vietnam Journal of Agricultural Sciences* **2020**, *3*, 530–540, doi:10.31817/vjas.2020.3.1.06.

- [34] Murphy, F.E.; Srbinovski, B.; Magno, M.; Popovici, E.M.; Whelan, P.M. An automatic, wireless audio recording node for analysis of beehives. In Proceedings of the 2015 26th Irish Signals and Systems Conference (ISSC), Carlow, Ireland, 24–25 June 2015; pp. 1–6.
- [35] Robles-Guerrero, A.; Saucedo-Anaya, T.; González-Ramírez, E.; Galván-Tejada, C.E. Frequency Analysis of Honey Bee Buzz for Automatic Recognition of Health Status: A Preliminary Study. *Research in Computing Science* **2017**, *142*, 89–98, doi:10.13053/rcs-142-1-9.
- [36] Hennessy, G.; Harris, C.; Eaton, C.; Wright, P.; Jackson, E.; Goulson, D.; Ratnieks, F.F. Gone with the wind: Effects of wind on honey bee visit rate and foraging behaviour. *Anim. Behav.* **2020**, *161*, 23–31. doi: 10.1016/j.anbehav.2019.12.018.
- [37] Cecchi, S.; Spinsante, S.; Terenzi, A.; Orcioni, S. A Smart Sensor-Based Measurement System for Advanced Bee Hive Monitoring. *Sensors* **2020**, *20*, 2726, doi:10.3390/s20092726.
- [38] Marchal, P.; Buatois, A.; Kraus, S.; Klein, S.; Gómez-Moracho, T.; Lihoreau, M. Automated Monitoring of Bee Behaviour Using Connected Hives: Towards a Computational Apidology. *Apidologie* **2020**, *51*, 356–368, doi:10.1007/s13592-019-00714-8.
- [39] Ngo, T.H.; Wu, K.-C.; Yang, E.-C.; Lin, T.-T. A Real-Time Imaging System for Multiple Honey Bee Tracking and Activity Monitoring. *Computers and Electronics in Agriculture* **2019**, *163*, 104841, doi:10.1016/j.compag.2019.05.050.
- [40] Ochoa, I.Z.; Gutierrez, S.; Rodríguez, F. Internet of Things: Low Cost Monitoring BeeHive System Using Wireless Sensor Network; *IEEE International Conference on Engineering Veracruz (ICEV)*, 2019, pp. 1-7, doi: 10.1109/ICEV.2019.8920622.
- [41] Anand, N.; Raj, V.B.; Ullas, M.S.; Srivastava, A. Swarm Detection and Beehive Monitoring System using Auditory and Microclimatic Analysis. In Proceedings of the 2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C), Bangalore, India, 3–5 October 2018; pp. 1–4, doi: 10.1109/CIMCA.2018.8739710.
- [42] Rybin, V.G.; Butusov, D.N.; Karimov, T.I.; Belkin, D.A.; Kozak, M.N. Embedded data acquisition system for beehive monitoring. In Proceedings of the 2017 IEEE II International Conference on Control in Technical Systems (CTS), Saint Petersburg, Russia, 25–27 October 2017; pp. 387–390, doi: 10.1109/CTSYS.2017.8109576.
- [43] Vidrascu, M.G.; Svasta, P.M. Embedded software for IOT bee hive monitoring node. In Proceedings of the 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME), Constanța, Romania, 26–29 October 2017; pp. 183–188, doi: 10.1109/SIITME.2017.8259887.
- [44] Vidrascu, M.G.; Svasta, P.M.; Vladescu, M. High reliability wireless sensor node for bee hive monitoring. In Proceedings of the 2016 IEEE 22nd International Symposium for Design and Technology in Electronic Packaging (SIITME), Oradea, Romania, 20–23 October 2016; pp. 134–138.

- [45] Chen, W.-S.; Wang, C.-H.; Jiang, J.-A.; Yang, E.-C. Development of a Monitoring System for Honeybee Activities. In Proceedings of the 2015 Minth International Conference on Sensing Technology, Auckland, New Zealand, 8–10 December 2015; pp. 745–750.
- [46] Cousin, P.; Cauia, E.; Siceanu, A.; de Cledat, J. The Development of an Efficient System to Monitor the Honeybee Colonies Depopulations. In Proceedings of the 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 17–21 June 2019, doi:10.1109/GIOTS.2019.8766435 Corpus ID: 198146274.
- [47] Hong, W.; Xu, B.; Chi, X.; Cui, X.; Yinfa, Y.; Li, T. Long-Term and Extensive Monitoring for Bee Colonies Based on Internet of Things. *IEEE Internet of Things Journal* **2020**, *7*, 7148–7155, doi:10.1109/jiot.2020.2981681.
- [48] Kontogiannis, S. An Internet of Things-Based Low-Power Integrated Beekeeping Safety and Conditions Monitoring System. *Inventions* **2019**, *4*, 52, doi:10.3390/inventions4030052.
- [49] Debauche, O.; Moulat, M.E.; Mahmoudi, S.; Boukraa, S.; Manneback, P.; Lebeau, F. Web Monitoring of Bee Health for Researchers and Beekeepers Based on the Internet of Things. *Procedia Computer Science* **2018**, *130*, 991–998, doi:10.1016/j.procs.2018.04.103.
- [50] Dineva, K. Computer system using internet of things for monitoring of bee hives. *Int. Multidiscip. Sci. GeoConference SGEM* **2017**, 27 - 29 November, 2017, Vol. 17, Issue 63, 169-176 pp; doi:10.5593/sgem2017H/63/S25.022.
- [51] Lyu, X.; Zhang, S.; Wang, Q. Design of Intelligent Beehive System based on Internet of Things Technology. In Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019), Chongqing, China, 30–31 May 2019, <https://doi.org/10.2991/iccia-19.2019.18>.
- [52] Zetterman, B. Beekeepers Usage of IoT : Data Collection, Sharing and Visualization in the Domain of Beekeeping. *Master Thesis* **2018**.
- [53] Edwards-Murphy, F.; Magno, M.; Whelan, P.M.; O’Halloran, J.; Popovici, E. B+WSN: Smart Beehive with Preliminary Decision Tree Analysis for Agriculture and Honey Bee Health Monitoring. *Computers and Electronics in Agriculture* **2016**, *124*, 211–219, doi:10.1016/j.compag.2016.04.008.
- [54] Kulyukin, V.; Mukherjee, S.; Amlathe, P. Toward Audio Beehive Monitoring: Deep Learning vs. Standard Machine Learning in Classifying Beehive Audio Samples. *Applied Sciences* **2018**, *8*, 1573, doi:10.3390/app8091573.
- [55] Kviesis, A.; Zacepins, A. Application of neural networks for honey bee colony state identification. In Proceedings of the 2016 17th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 29 May–1 June **2016**; pp. 413–417, doi: 10.1109/CarpathianCC.2016.7501133.
- [56] Jiang, J.-A.; Wang, C.-H.; Chen, C.-H.; Liao, M.-S.; Su, Y.-L.; Chen, W.; Huang, C.-P.; Yang, E.-C.; Chuang, C.-L. A WSN-Based Automatic Monitoring System for the Foraging

- Behavior of Honey Bees and Environmental Factors of Beehives. *Computers and Electronics in Agriculture* **2016**, *123*, 304–318, doi:10.1016/j.compag.2016.03.003.
- [57] Murphy, F.; Magno, M.; Whelan, P.M.; Vici, E.P. *B+WSN: Smart Beehive for Agriculture, Environmental, and Honey Bee Health Monitoring — Preliminary Results and Analysis*; **2015**, pp. 1-6, doi: 10.1109/SAS.2015.7133587.
- [58] Giammarini, M.; Concettoni, E.; Zazzarini, C.C.; N, O.; Albanesi, M.; Cristalli, C. *BeeHive Lab Project - Sensorized Hive for Bee Colonies Life Study*; 2015; pp. 121–126;.
- [59] Cecchi, S.; Terenzi, A.; Orcioni, S.; Spinsante, S.; Primiani, V.M.; Moglie, F.; Ruschioni, S.; Mattei, C.; Riolo, P.; Isidoro, N. Multi-Sensor Platform for Real Time Measurements of Honey Bee Hive Parameters. *IOP Conference Series* **2019**, *275*, 012016, doi:10.1088/1755-1315/275/1/012016.
- [60] Chen, C.W.; Yang, E.-C.; Jiang, J.-A.; Lin, T.-T. An Imaging System for Monitoring the In-and-out Activity of Honey Bees. *Computers and Electronics in Agriculture* **2012**, *89*, 100–109, doi:10.1016/j.compag.2012.08.006.
- [61] Bermig, S.; Odemer, R.; Gombert, A.; Frommberger, M.; Rosenquist, R.; Pistorius, J. Experimental Validation of an Electronic Counting Device to Determine Flight Activity of Honey Bees (*Apis Mellifera* L.). *DOAJ (DOAJ: Directory of Open Access Journals)* **2020**, doi:10.5073/jfk.2020.05.03.
- [62] Arnia: Remote Beehive Monitoring. Доступно online: <https://www.arnia.co/> (последњи приступ 11.06.2023).
- [63] BEECHECK. Доступно online: <https://beecheck.org/> (последњи приступ 11.06.2023).
- [64] Bee counter. Available online: <https://www.beehivemonitoring.com/en/> (последњи приступ 11.06.2023).
- [65] Honey bee counter. Доступно online: <https://www.instructables.com/Honey-Bee-Counter/> (последњи приступ 11.06.2023).
- [66] Fritzing open-source software. Доступно online: <https://fritzing.org> (последњи приступ 11.06.2023).
- [67] Подаци доступни online: <https://github.com/andrijevicnebojsa/BeeMAP> (последњи приступ 11.06.2023).
- [68] Arduino Mega . Доступно online: <https://store.arduino.cc/products/arduino-mega-2560-rev3> (последњи приступ 11.06.2023).
- [69] ESP32 Adafruit Feather. Доступно online: <https://www.adafruit.com/product/3405> (последњи приступ 11.06.2023).
- [70] PHP документација, доступно online: <https://www.php.net/docs.php> (последњи приступ 11.06.2023).

- [71] *Напредни PHP и MySQL*, др Зоран Ловрековић, самостално издање аутора, Нови Сад 2020.
- [72] *Laravel Application Development Cookbook*, Matula T., Packt Publishing, преузето у PDF формату са <https://www.programming-book.com/download/?file=8475> (последњи приступ 11.06.2023).
- [73] *History of Laravel PHP framework*, доступно online: <https://maxoffsky.com/code-blog/history-of-laravel-php-framework-eloquence-emerging/> (последњи приступ 11.06.2023).
- [74] *Features of Laravel*, JavaTPoint, доступно online: <https://www.javatpoint.com/features-of-laravel> (последњи приступ 11.06.2023).
- [75] MVC Framework – Introduction, Tutorialspoint, доступно online: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm (последњи приступ 11.06.2023).
- [76] *How Laravel implements MVC and how to use it effectively*, Pusher Blog, доступно online: <https://blog.pusher.com/laravel-mvc-use/> (последњи приступ 11.06.2023).
- [77] Javascript, Tutorial Point, 2015, стр 12, доступно online: <https://matfuvit.github.io/UVIT/predavanja/literatura/TutorialPoint%20JavaScript.pdf> (последњи приступ 11.06.2023).
- [78] Node.js документација, доступно online: <https://nodejs.org/en/about/> (последњи приступ 11.06.2023).
- [79] Pro Vue.js 2, Freeman Adam, стр 49 – 50, 2018.
- [80] WAMP, доступно online: <https://www.wampserver.com/en/> (последњи приступ 11.06.2023).
- [81] *Visual C++ Redistributable for Visual Studio 2012*, доступно online: <https://www.microsoft.com/en-us/download/details.aspx?id=30679> (последњи приступ 11.06.2023).
- [82] *Composer*, доступно online: <https://getcomposer.org/download/> (последњи приступ 11.06.2023).
- [83] *Node.js*, доступно online: <https://nodejs.org/en/download/> (последњи приступ 11.06.2023).
- [84] Откључана демо кошница , доступно online: <https://www.iobees.rs/singlehive/1> (последњи приступ 11.06.2023).
- [85] Catania, P.; Vallone, M. Application of A Precision Apiculture System to Monitor Honey Daily Production. *Sensors* **2020**, *20*(7):2012, doi:10.3390/s20072012.

- [86] Zabasta, A.; Kunicina, N.; Kondratjevs, K.; Ribickis, L. *IoT Approach Application for Development of Autonomous Beekeeping System*. 2019 International Conference in Engineering Applications (ICEA), **2019**, pp. 1-6, doi: 10.1109/CEAP.2019.8883460.
- [87] Meikle, W.G.; Holst, N. Application of Continuous Monitoring of Honeybee Colonies. *Apidologie* **2015**, *46*, 10–22, doi:10.1007/s13592-014-0298-x.
- [88] Kridi, D.S.; De Carvalho, C.R.R.; Gomes, D.G. Application of Wireless Sensor Networks for Beehive Monitoring and In-Hive Thermal Patterns Detection. *Computers and Electronics in Agriculture* **2016**, *127*, 221–235, doi:10.1016/j.compag.2016.05.013.
- [89] Craig, L.M.; Parry, R.M.; Tashakkori, R.; Watts, I. BeePhon: A Web-Application for Beehive Audio Exploration. In Proceedings of the 2019 SoutheastCon, Huntsville, AL, USA, 11–14 April 2019. doi:10.1109/SoutheastCon42311.2019.9020475.
- [90] Andrijević, N.; Urošević, V.V.; Arsic, B.; Herceg, D.; Savić, B. IoT Monitoring and Prediction Modeling of Honeybee Activity with Alarm. *Electronics* **2022**, *Vol. 11(5)*, 783, doi:10.3390/electronics11050783.
- [91] Supandi; Arkan, F.; Gusa, R.F.; Jumnahdi, M.; Kurniawan, R. Design of System for Setting the Temperature and Monitoring Bees in and out the Hive. *IOP Conference Series: Earth and Environmental Science* **2020**, *599*, 012050, doi:10.1088/1755-1315/599/1/012050.
- [92] Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-Art in Artificial Neural Network Applications: A Survey. *Heliyon* **2018**, *4*, e00938, doi:10.1016/j.heliyon.2018.e00938.
- [93] Hewamalage, H.; Bergmeir, C.; Bandara, K. Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. *International Journal of Forecasting* **2021**, *37*, 388–427, doi:10.1016/j.ijforecast.2020.06.008.
- [94] Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780, doi:10.1162/neco.1997.9.8.1735.
- [95] Otter, D.W.; Medina, J.S.; Kalita, J. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems* **2021**, *32*, 604–624, doi:10.1109/tnnls.2020.2979670.
- [96] Nassif, A.B.; Shahin, I.; Attili, I.B.; Nassif, A.B.; Shaalan, K. Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access* **2019**, *7*, 19143–19165, doi:10.1109/access.2019.2896880.
- [97] Sharma, H.; Agrahari, M.; Singh, S.K.; Firoj, M.; Mishra, R.K. Image captioning: A comprehensive survey. In Proceedings of the 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 28–29 February **2020**; pp. 325–328.

- [98] Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Dean, J. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv* **2016**, arXiv:1609.08144.
- [99] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. *Attention Is All You Need*; Cornell University, **2017**; Vol. 30, pp. 5998–6008, <https://arxiv.org/pdf/1706.03762v5>.
- [100] Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*, 5th ed.; John Wiley & Sons: Hoboken, NJ, USA, **2015**; ISBN 9781118674925.
- [101] Taylor, S.V.; Letham, B. Forecasting at Scale. *The American Statistician* **2018**, *72*, 37–45, doi:10.1080/00031305.2017.1380080.
- [102] Andrijević, N.; Urošević, V. Koncept smart košnica sa sistemom za zaštitu pčela od toksičnih poremećaja u vazduhu. *Konferencija YU-INFO*, **2023**, рад је прихваћен и презентован, чека се зборник радова.
- [103] Andrijević, N.; Urošević, V. Razvoj softvera za kontrolu ekosistema pčelinjeg društva sa alarmom. *Konferencija YU-INFO*, **2022**, ISBN 978-86-85525-27-8, str.173–175.
- [104] Andrijević, N.; Urošević, V. Razvoj Sistema Za Kontinuirano Praćenje Stanja Pčelinjeg Društva. *Konferencija YU-INFO* **2021**, ISBN 978-86-85525-25-4, str.123–126.
- [105] Andrijević, N.; Urošević, V. Koncept Adaptivne Smart IoT Tehnologije Za Primenu u Pametnoj Poljoprivredi. *Konferencija Politehnika* **2021**, ISBN-978-86-7498-087-3, str.749.

ПРИЛОЗИ

Прилог 7.1 Техничка спецификација микроконтролера Arduino Mega 2560.

Спецификација	Опис
Микроконтролер	ATmega2560
Архитектура	AVR
Брзина такта	16 MHz
Број пинова	54 дигитална улазно/излазна пина, 15 аналогних улаза
Број PWM пина	15
Меморија	256 KB флеш меморија, 8 KB SRAM, 4 KB EEPROM
Напајање	7 – 12 V преко USB порта или спољног извора напајања
Интерфејси	USB, UART, SPI, I2C, ICSP
Дигитални пинови	54, напон: 5 V, струја на излазу: 20 mA
Аналогни пинови	16, 10 бита, референтни напон: 5 V, распон: 0 - 5 V
PWM пинови	15, 8 бита, ширина пулса: 0 - 255
Комуникациони интерфејси	UART: 4 порта, SPI: 1 порт, I2C: 1 порт
Меморија	Flash: 256 KB, SRAM: 8 KB, EEPROM: 4 KB

Прилог 7.2 Техничка спецификација за Mega Sensor Shield V2.0.

Спецификација	Опис
Компатибилност	Компатибилан са Arduino Mega 2560 и Arduino Mega ADK
Пинови	- 54 дигиталних улазно/излазних пина (укључујући 14 PWM пина) - 16 аналогних улазних пина
Интерфејси	- 4 серијска порта (UART) - 1 SPI интерфејс - 1 I2C интерфејс
Напајање	- Напајање преко Arduino Mega 2560 плоче - Напајање напона: 7 - 12 V
Конектори	- Конектор за серво моторе (Power + Signal) - Конектори за степенасте водове
Интерфејси сензора	- 3,3 V и 5 V прикључни пинови су осетљиви - Додатни пинови за GND и 5 V напајање
Друге функционалности	- Подршка за додатне модуле и компоненте, укључујући моторе, сензоре, дисплеје и друге додатке

Прилог 7.3 Техничка спецификација ESP32 микроконтролера.

Спецификација	Опис
Микропроцесор	LX6 микропроцесор са једним или двојезгарним моделом, радна фреквенција до 240 MHz
Меморија	520 KB SRAM, 4 MB flash
Wi-Fi	Интегрисана антена 802.11 b/g/n Wi-Fi модул са брзинама до 150 Mbps
Bluetooth	Подршка за Bluetooth v4.2 и Bluetooth Low Energy (BLE) спецификације

GPIO	34 програмабилних GPIO пина
Аналогни улаз/излаз	12 x ADC улазних канала, 2 x I2S Audio, 2 x DAC
Комуникација	3 x SPI, 2 x I2C, 2 x I2S, 3 x UART
Етернет	Интегрисани Ethernet MAC за физичку LAN комуникацију
SD картице	1 host контролер за SD/SDIO/MMC, 1 slave контролер за SDIO/SPI
PWM	Мотор PWM и до 16 канала LED PWM
Сигурност	Безбедно покретање и флеш шифровање, хардверски криптографски акцелератори (AES, Hash, RSA, ECC, RNG)
Напајање	Напон 3.3 V, подршка за различите режиме напајања
Друге карактеристике	32 kHz crystal oscillator, подршка за мобилне и ИОТ апликације, мала потрошња енергије

Прилог 7.4 Техничка спецификација за RTC 3231 модул реалног времена (Real-Time Clock).

Спецификација	Опис
Чипсет	DS3231
Врста	Модул реалног времена (RTC)
Повезивање	Комуникација преко I2C интерфејса
Тачност	±2 ppm (±1 минут у години)
Датум и време	- Формат времена: Часови, минути, секунде (24-сатни формат) - Формат датума: Дан, месец, година
Календарски аларми	Подршка за аларме на основу датума и времена
Батерија за бекап	Интегрисана батерија (CR2032) за бекап података при прекиду напајања
Радни напон	3.3 V - 5 V
Температурни сензор	Интегрисан температурни сензор са резолуцијом 0.25 °C

Прилог 7.5 Техничка спецификација за SIM800L GPRS GSM модул.

Спецификација	Опис
Мрежна подршка	GSM/GPRS мрежа (2G)
Чипсет	SIM800L
Фреквенције	- GSM: 850/900/1800/1900 MHz - GPRS: Класа 12
Подржане технологије	- GSM: GPRS, EDGE - GPRS: Класа 12
GSM команде	Подршка за GSM команде AT
Сим-карта	1 SIM-картица (микро SIM-картица)
Антена	Унутрашња или спољна антена (U.FL конектор)
Напајање	Радни напон: 3.4 V - 4.4 V Потрошња струје: 2 A,
Комуникација	Серијски интерфејс (UART) са брзином преноса података од 1200 bps до 115200 bps
Аудио	Подржава микрофон и звучник интерфејс за гласовну комуникацију
Дигитални пинови	- GPIO пинови за контролу и комуникацију са спољним уређајима

- ADC пинови

Интернет протоколи TCP/IP, HTTP, FTP, SMTP, POP3*Прилог 7.6 Техничка спецификација за DHT-22 (AM2302) модул.*

Спецификација	Опис
Тип сензора	DHT-22 (AM2302)
Мерени параметри	Температура у целзијусима (°C) (Температура: ±0.5 °C) Влажност у ваздуху (%) (Влажност: ±2%)
Напајање	3.3 V - 5.5 V (DC)
Комуникација	Једноставан дигитални комуникациони протокол
Резолуција	Температура: 0.1 °C Влажност: 0.1%
Мерни опсег	Влажност: 0 - 100% RH Температура: -40 °C – 80 °C
Сензор	Polymer humidity capacitor
Брзина одзива	2 секунде
Додатне функционалности	Подршка за калибрацију

Прилог 7.7 Техничка спецификација за сензор вибрације SW-420.

Спецификација	Опис
Радни напон	Између 3.3 V и 5 V
Радна струја	15 mA
Изназ	Дигитални излаз
LED индикатори	Има уграђене LED диоде које показују статус излаза и снагу
Интегрисана капацитивност	Заснован на LM393 интегрисаном колу
Филтер шума	Има филтер за смањење шума и превенцију лажних сигнала

Прилог 7.8 Техничка спецификација за сензор MHRD rain.

Спецификација	Опис
Радни напон	5 V
Изназни формат	Дигитални прекидачки излаз (0 и 1) и аналогни напонски излаз АО
Подешавање осетљивост	Користи LM393 компаратор са широким опсегом напона Осетљивост сензора се подешава помоћу потенциометра
Изназни сигнал компаратора	Чист таласни облик доброг квалитета, погодан за проводљивост, преко 15 mA
Отпорност материјала	Висококвалитетан RF-04 материјал са двостраном површином Сензор је отпоран на оксидацију и проводљивост, са дугим веком трајања

Прилог 7.9 Техничка спецификација за сензор MQ135.

Спецификација	Опис
Тип сензора	MQ135
Радни напон	5 V
Изназ	Аналогни излаз
Квалитет ваздуха	Мери концентрацију PM2.5 (честице у ваздуху величине 2.5 микрометра) и PM10 (честице величине 10 микрометра)
Детекција гасова	амонијак (NH_3), сумпор-диоксид (SO_2), бензен (C_6H_6), NO_x , алкохол, дим и други штетни гасови
Распон мерења	амонијак (NH_3): 10-1000 ppm сумпор-диоксид (SO_2): 0.1-1000 ppm бензен (C_6H_6): 0.1-1000 ppm
Резолуција	0.1 ppm
Време одзива	Мање од 10 секунди
Радна температура	Од -10 °C до 50 °C
Радна влажност	95% RH максимално

Прилог 7.10 Техничка спецификација за сензор BH1750.

Спецификација	Опис
Тип сензора	BH1750
Радни напон	3.3 V – 5 V
Комуникација	I2C (TWI)
Распон мерења	Од 1 lx до 65535 lx
Време одзива	120 ms (High Resolution Mode) 16 ms (Low Resolution Mode)
Осетљивост	Спектрална осетљивост приближно људском виду (Human Eye Response Curve)
Радна температура	Од -40 °C до +85 °C
Потрошња струје	0,12 mA
Додатне функционалности	Аутоматско мерење амбијенталног осветљења и конверзија у дигитални излаз

Прилог 7.11 Техничка спецификација за сензор VEML6075.

Спецификација	Опис
Тип сензора	VEML6075
Радни напон	3.3 V – 5 V
Комуникација	I2C (TWI)
Тип мерења	UV-A, UV-B, UV indeks
Распон мерења	UV-A: $\lambda 0.5$ у распону од 315 nm до 340 nm UV-B: $\lambda 0.5$ у распону од 350 nm до 375 nm
Време одзива	Мање од 1 секунде
Радна температура	Од -40 °C до +85 °C

Радна влажност	Од 0% до 100% RH
Додатне функционалности	Аутоматско мерење UV индекса, UV indeks: 0 - 11+

Прилог 7.12 Техничка спецификација за сензор SI1145.

Спецификација	Опис
Тип сензора	SI1145
Радни напон	3.3 V – 5 V
Комуникација	I2C (TWI)
Тип мерења	UV зрачење, видљива светлост, IR инфрацрвено зрачење
Распон мерења	UV: спектар таласне дужине 200 nm - 370 nm Visible: спектар таласне дужине 400 nm - 800 nm Infrared: спектар таласне дужине 550 nm - 1000 nm
Време одзива	Мање од 25 ms
Радна температура	Од -40 °C до +85 °C
Додатне функционалности	Аутоматско мерење UV индекса

Прилог 7.13 Техничка спецификација за сензор BME280.

Спецификација	Опис
Тип сензора	BME280
Радни напон	3.3 V или 5 V
Комуникација	I2C (TWI) или SPI
Тип мерења	Температура, влажност ваздуха, притисак, алтитуда
Резолуција	Температура: 0.01 °C Влажност ваздуха: 0.008% RH Притисак: 0.18 Pa
Распон мерења	Температура: од -40 °C до +85 °C (тачност: ±1.0 °C) Влажност ваздуха: од 0% до 100% RH (тачност: ±3% RH) Притисак: од 300 до 1100 hPa (тачност: ±1.0 hPa)
Време одзива	Мање од 1 секунде
Радна температура	од -40 °C до +85 °C
Радна влажност	Од 0% до 100% RH
Додатне функционалности	Могућност компензације температурних и варијација ваздушног притиска

Прилог 7.14 Техничка спецификација за сензор Анемометер.

Спецификација	Опис
Тип сензора	Анемометер (сензор брзине ветра)
Радни напон	DC 9-24 V
Издаз	Пулсни издаз (број импулса у одређеном временском интервалу)
Распон мерења	0 - 70 m/s
Резолуција	0.1m/s
Почетна брзина ветра	0.2 m/s
Издазна снага	0-20 mA
Радна температура	Од -40 °C до 80 °C
Радна влажност	Од 35% до 85%.

Додатне функционалности	Могућност калибрације и подешавања осетљивости, отпорности на временске услове
--------------------------------	--

Прилог 7.15 Техничка спецификација за сензор МАХ4466.

Спецификација	Опис
Тип сензора	МАХ4466
Радни напон	2.4 V - 5.5 V
Тип излаза	Аналогни
Појачање	Од 25x до 125x (подесиво)
Фреквенцијски одзив	Од 20 Hz до 20 kHz
Шум	0.003% THD+N
SNR (Signal-to-Noise Ratio)	Више од 100 dB при појачању 125x
Радна температура	Од -40 °C до +85 °C
Висок AVOL	125 dB (RL = 100 kΩ)
Ниска струја мировања	од 24 μA
Пропусни опсег појачања	600 kHz
Додатне функционалности	Подесиво појачање, ниски шум, широк фреквенцијски одзив, ниска потрошња

Прилог 7.16 Техничка спецификација за сензор МАХ9814.

Спецификација	Опис
Тип сензора	МАХ9814
Радни напон	2.7 V – 5 V
Тип излаза	Аналогни
Појачање	Од 20x до 60x (подесиво)
Фреквенцијски одзив	Од 20 Hz до 20 kHz
SNR (Signal-to-Noise Ratio)	≥58 dB
Радна температура	Од -40 °C до +85 °C
Осетљивост	-42 ±3 dB
Аутоматска контрола нивоа излазног сигнала	40 dB, 50 dB или 60 dB
Додатне функционалности	Подесиво појачање, низак ниво шума, широк фреквентни одзив, ниска потрошња

БИОГРАФИЈА АУТОРА

Лични подаци:

Небојша Андријевић рођен је 11. октобра 1980. године у Призрену, где је завршио основну и средњу техничку школу. Ожењен је и отац четворо деце.

Образовање:

Факултет техничких наука у Чачку, интегрисане мастер студије, модул Професор технике и информатике завршио је 2014. године. Докторске студије у Чачку уписао је 2019. године, на студијском програму Информационе технологије. Током докторских студија објавио је више радова, од којих је један у истакнутом међународном часопису, три рада на међународним скуповима, 4 рада на националним конференцијама, као и две публикације са учешћа на пројектима из области високог образовања у оквиру програма Европске уније и других међународних организација.

Педагошко искуство:

Своје педагошко искуство започиње 2016. године на Високој техничкој школи струковних студија у Новом Саду, а тренутно је запослен на Академији техничких струковних студија у Београду. Ангажован је на више стручних предмета из области информациоих технологија.

Активности ангажовање у стручним телима и органима:

Био је члан и рецезент Комисије за акредитацију и проверу квалитета високошколских установа (КАПК садашњи НАТ), члан Националног савета за високо образовање (НСВО), председник Студентске конференције академија струковних студија Србије (СКАССС), члан радне групе за израду Закона о студентском организовању, као и члан више радних група из области високог образовања.

Учешће на пројектима:

Учествовао је на више **TEMPUS** пројеката, међу којима су **TEMPUS SIGMUS SM 511332-2010**, **TEMPUS ECBAC 517200- TEMPUS-1-2011-1-BE**, **TEMPUS GOMES – JP 158926 – 2009**.

Семинари и конференције:

Организатор је и коорганизатор више до 15 стручних конференција из области високог образовања.

Образац 1

ИЗЈАВА АУТОРА О ОРИГИНАЛНОСТИ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Изјављујем да докторска дисертација под насловом:

„СИСТЕМ ЗА МОНИТОРИНГ И ПРЕДИКЦИЈУ АКТИВНОСТИ МЕДОНОСНИХ
ПЧЕЛА БАЗИРАН НА ИОТ ТЕХНОЛОГИЈИ И ПРИМЕНИ ВЕШТАЧКЕ
ИНТЕЛИГЕНЦИЈЕ “

представља *оригинално ауторско дело* настало као резултат *сопственог истраживачког рада*.

Овом Изјавом такође потврђујем:

- да сам *једини аутор* наведене докторске дисертације,
- да у наведеној докторској дисертацији *нисам извршио/ла повреду* ауторског нити другог права интелектуалне својине других лица,

У Чачку, 12.07.2023. године,



потпис аутора

Образац 2

**ИЗЈАВА АУТОРА О ИСТОВЕТНОСТИ ШТАМПАНЕ И ЕЛЕКТРОНСКЕ ВЕРЗИЈЕ
ДОКТОРСКЕ ДИСЕРТАЦИЈЕ**

Изјављујем да су штампана и електронска верзија докторске дисертације под насловом:
„ СИСТЕМ ЗА МОНИТОРИНГ И ПРЕДИКЦИЈУ АКТИВНОСТИ МЕДОНОСНИХ
ПЧЕЛА БАЗИРАН НА ИОТ ТЕХНОЛОГИЈИ И ПРИМЕНИ ВЕШТАЧКЕ
ИНТЕЛИГЕНЦИЈЕ “

истоветне.

У Чачку, 12.07.2023. године,



потпис аутора

Образац 3

ИЗЈАВА АУТОРА О ИСКОРИШЋАВАЊУ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Ја, Небојша М. Андријевић,

дозвољавам

не дозвољавам

Универзитетској библиотеци у Крагујевцу да начини два трајна умножена примерка у електронској форми докторске дисертације под насловом:

„СИСТЕМ ЗА МОНИТОРИНГ И ПРЕДИКЦИЈУ АКТИВНОСТИ МЕДОНОСНИХ
ПЧЕЛА БАЗИРАН НА ИОТ ТЕХНОЛОГИЈИ И ПРИМЕНИ ВЕШТАЧКЕ
ИНТЕЛИГЕНЦИЈЕ“

и то у целини, као и да по један примерак тако умножене докторске дисертације учини трајно доступним јавности путем дигиталног репозиторијума Универзитета у Крагујевцу и централног репозиторијума надлежног министарства, тако да припадници јавности могу начинити трајне умножене примерке у електронској форми наведене докторске дисертације путем *преузимања*.

Овом Изјавом такође

дозвољавам

не дозвољавам¹

¹ Уколико аутор изабере да не дозволи припадницима јавности да тако доступну докторску дисертацију користе под условима утврђеним једном од *Creative Commons* лиценци, то не искључује право припадника јавности да наведену докторску дисертацију користе у складу са одредбама Закона о ауторском и сродним правима.

припадницима јавности да тако доступну докторску дисертацију користе под условима утврђеним једном од следећих *Creative Commons* лиценци:

- 1) Ауторство
- 2) Ауторство - делим под истим условима**
- 3) Ауторство - без прерада
- 4) Ауторство - некомерцијално
- 5) Ауторство - некомерцијално - делим под истим условима
- 6) Ауторство - некомерцијално - без прерада²

У Чачку, 12.07.2023. године,



потпис аутора

² Молимо ауторе који су изабрали да дозволе припадницима јавности да тако доступну докторску дисертацију користе под условима утврђеним једном од *Creative Commons* лиценци да заокруже једну од понуђених лиценци. Детаљан садржај наведених лиценци доступан је на: <http://creativecommons.org.rs/>