

ДОДАТОК А

Графічний матеріал атестаційної роботи

ГЮИК.501320.006

(позначення документу)

Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»
Керівник атестаційної роботи
доц. Губаренко Є.В.

РОЗРОБКА МЕТОДУ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧА ЧЕРЕЗ ГОЛОСОВЕ
ПОВІДОМЛЕННЯ

Графічний матеріал

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.501320.006 – ЛЗ

РОЗРОБИВ:
ст. гр. ІТІм-19-1
Зіміна А.Р.

2020 р.

ЗАТВЕРДЖЕНО
ГЮИК.501320.006 – ЛЗ

РОЗРОБКА МЕТОДУ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧА ЧЕРЕЗ ГОЛОСОВЕ
ПОВІДОМЛЕННЯ

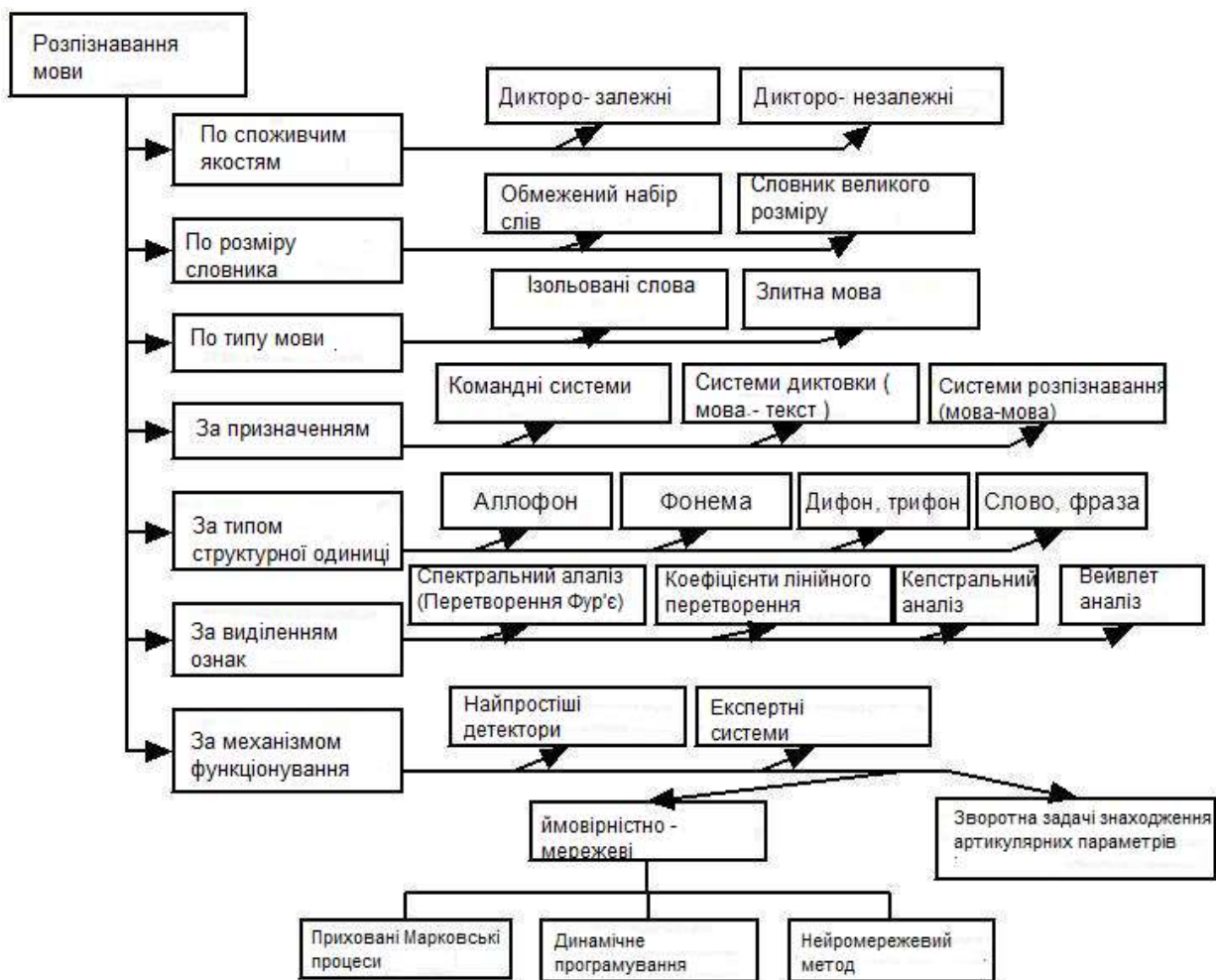
Графічний матеріал

ГЮИК.501320.006 – ЛЗ

АРКУШІВ 26

2020 р.

КЛАСИФІКАЦІЯ СИСТЕМ РОЗПІЗНАВАННЯ МОВИ



Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

НАПРЯМКИ РОЗВИТКУ ТЕХНОЛОГІЙ РОЗПІЗНАВАННЯ ДИКТОРА



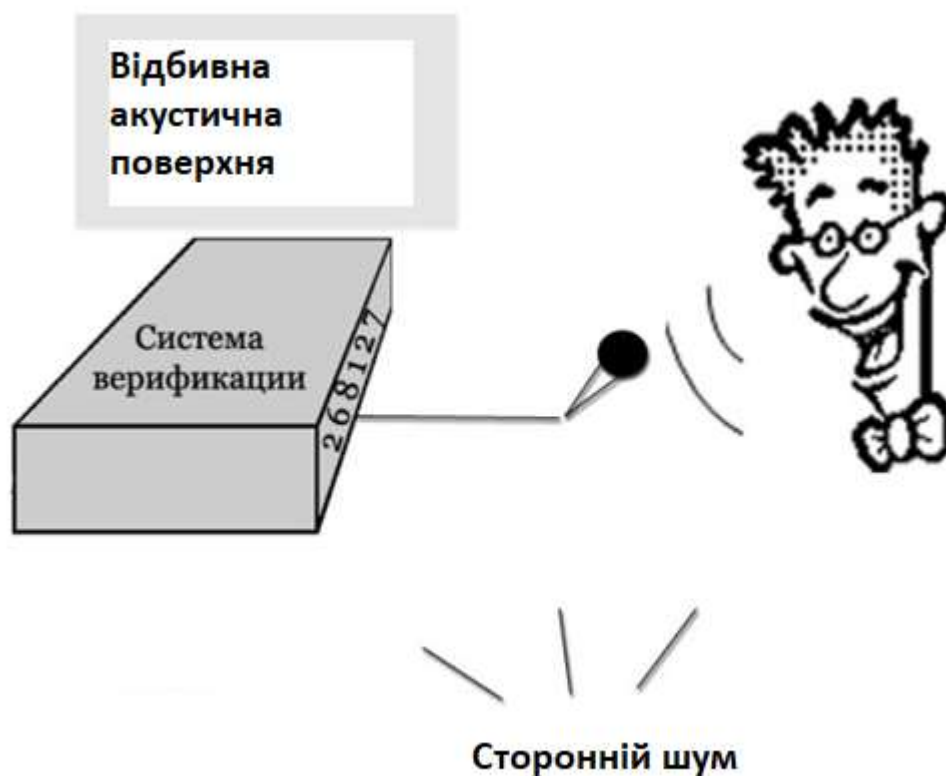
<i>Розроб.</i>	<i>Зіміна А.Р..</i>			<i>Розробка методу аутентифікації користувача через голосове повідомлення</i>	
<i>Перевір.</i>	<i>Губаренко Є.В.</i>				
<i>Н. Контр.</i>	<i>Губаренко Є.В.</i>				
				<i>ІТМ-19-1</i>	<i>Лист 1</i>
<i>Затверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 1</i>

ЗАГАЛЬНИЙ АЛГОРИТМ РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД



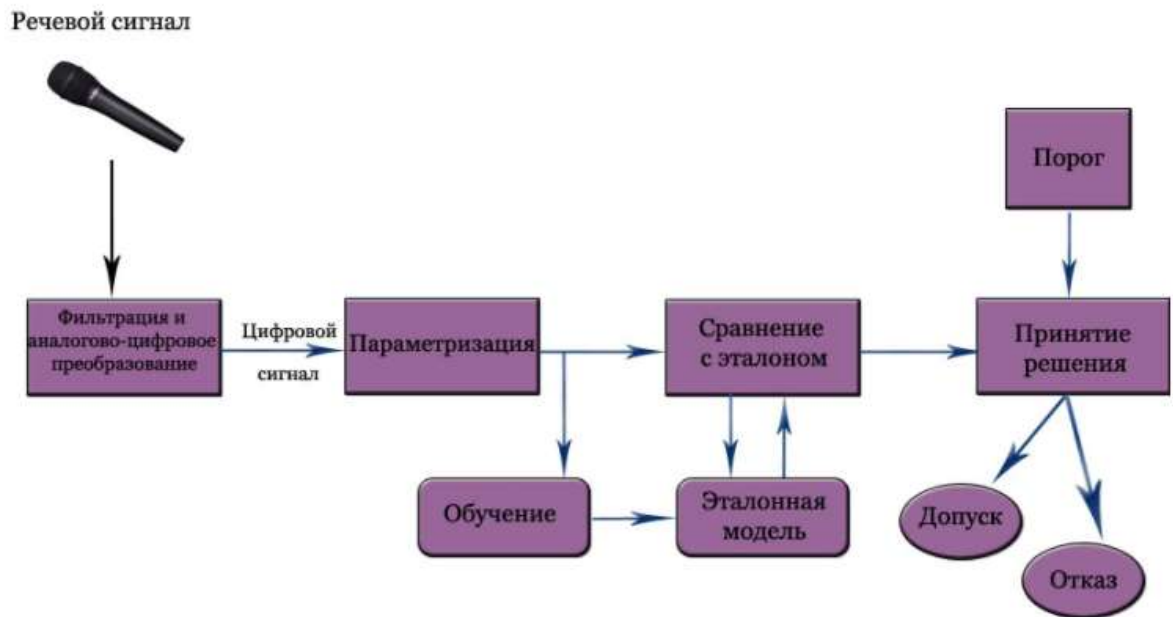
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

СХЕМА ВЗАЄМОДІЇ ЛЮДИНИ З СИСТЕМОЮ ВЕРИФІКАЦІЇ ОСОБИСТОСТІ ПО ГОЛОСУ



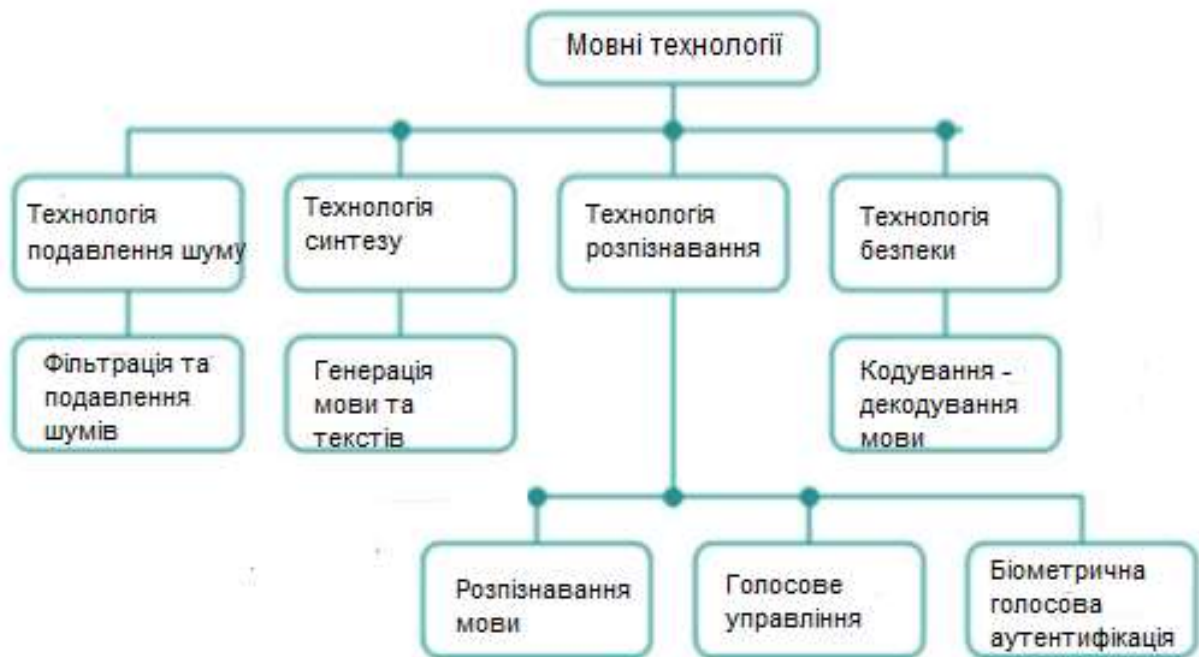
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
Затверд.	Гребеннік І.В.			ІТМ-19-1	Лист 1
				СТ	Листів 1

ЗАГАЛЬНИЙ АЛГОРИТМ ВЕРІФІКАЦІЇ ДИКТОРА



Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

МОВНІ ТЕХНОЛОГІЇ



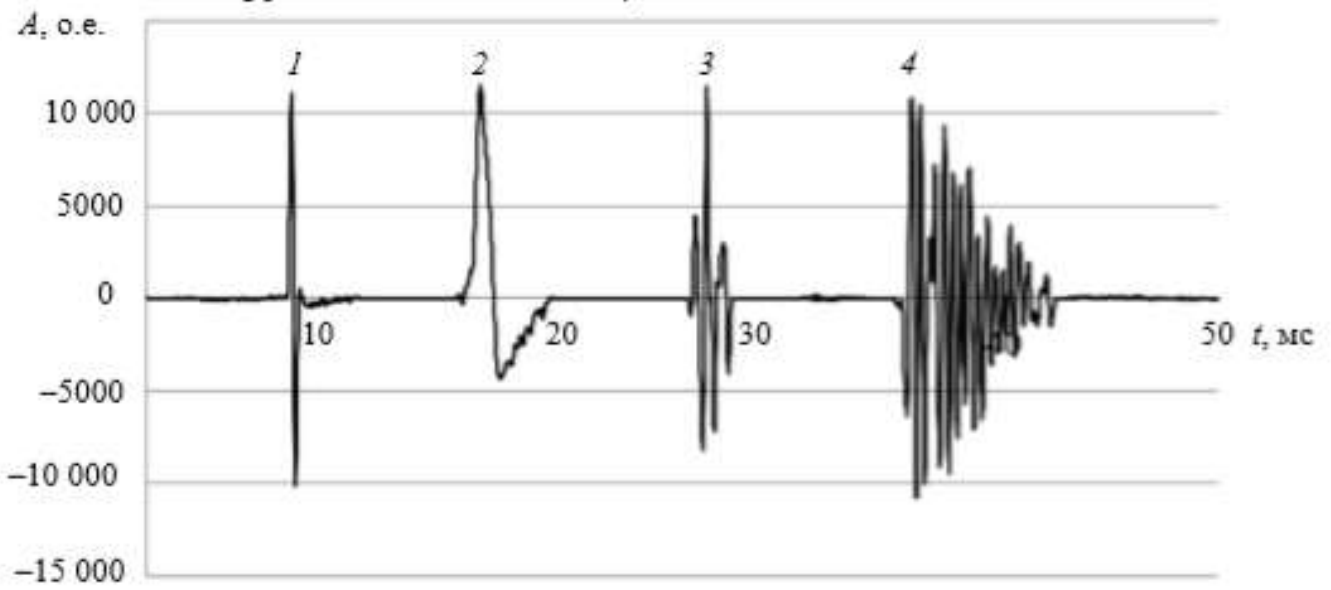
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

КЛАСИФІКАЦІЯ ШУМІВ У МОВНИХ СИГНАЛАХ



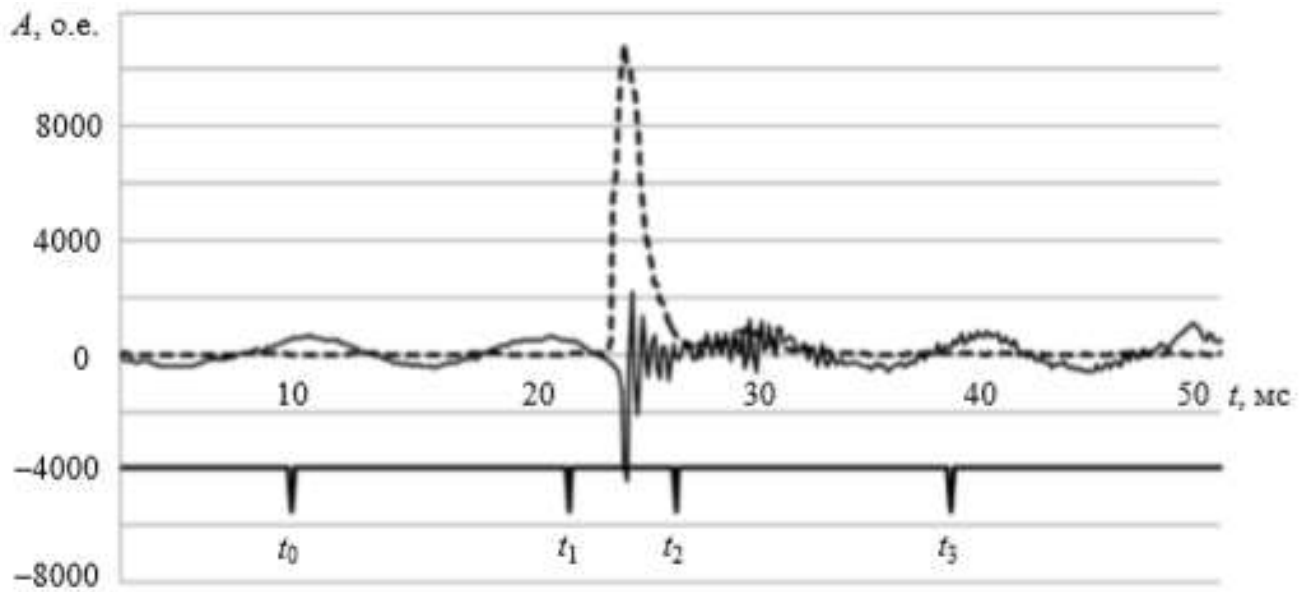
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ВИДИ КЛАЦАНЬ



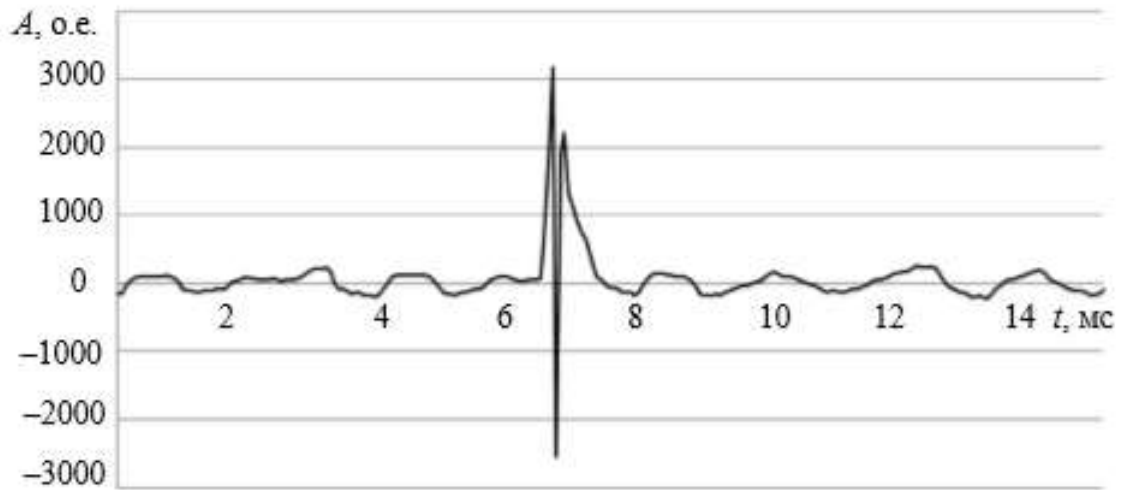
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІПМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

АЛГОРИТМ ВИЯВЛЕННЯ КЛАЦАНЬ



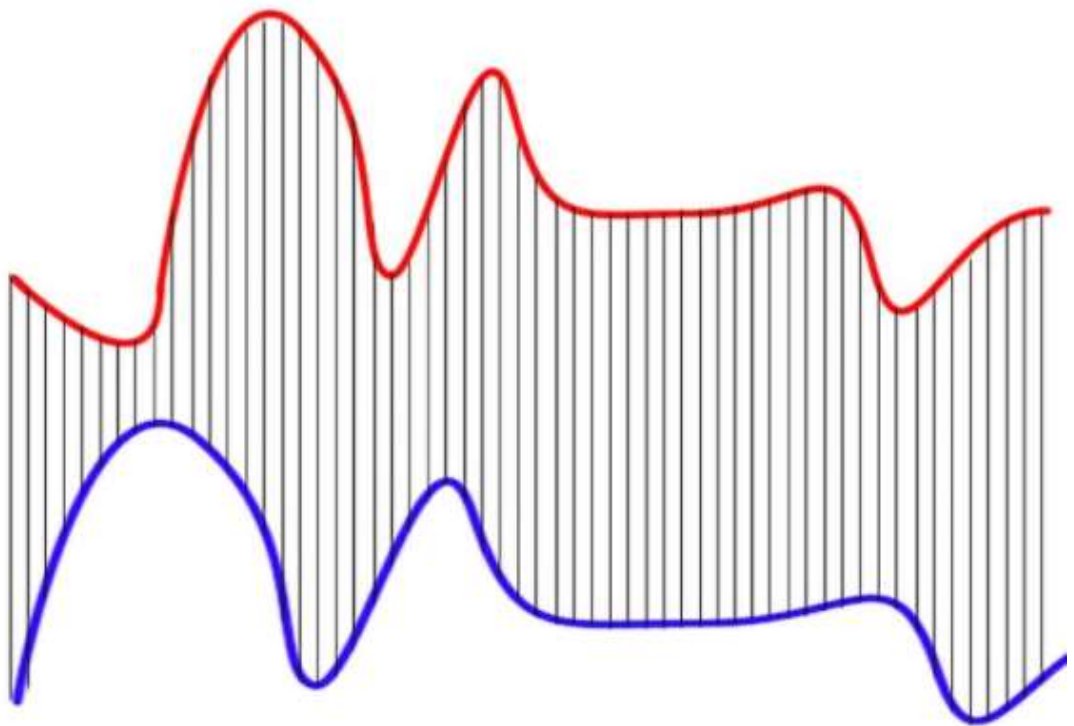
Розроб.	Зіміна А.Р..			<i>Розробка методу аутентифікації користувача через голосове повідомлення</i>	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІПМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ПРИКЛАД ПЕРЕВАНТАЖЕННЯ



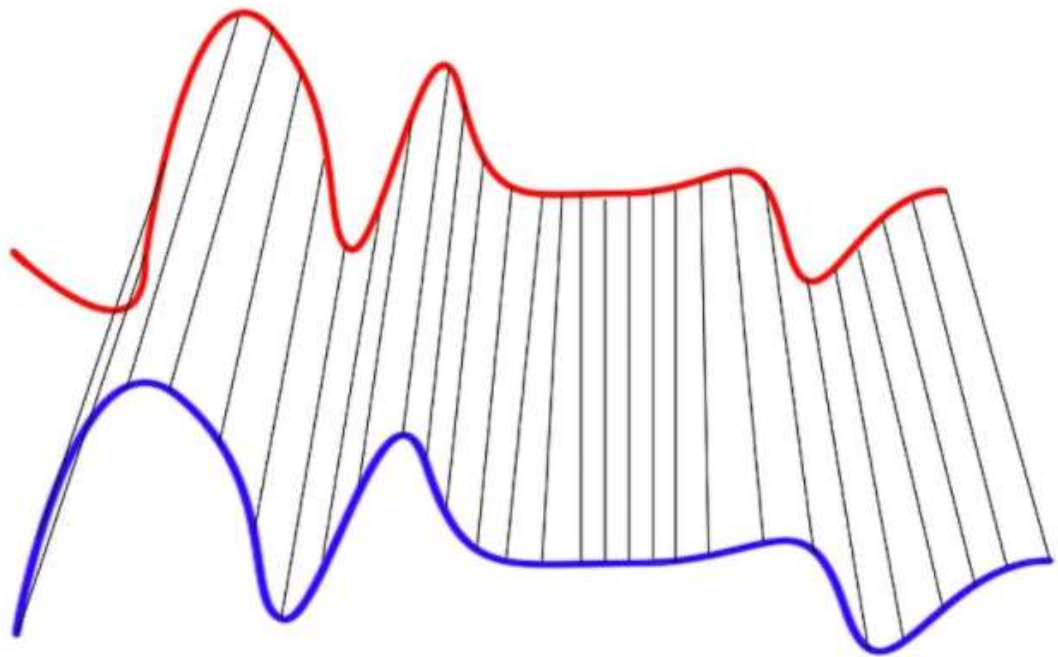
<i>Розроб.</i>	<i>Зіміна А.Р..</i>			<i>Розробка методу аутентифікації користувача через голосове повідомлення</i>	
<i>Перевір.</i>	<i>Губаренко Є.В.</i>				
<i>Н. Контр.</i>	<i>Губаренко Є.В.</i>				
				<i>ІПМ-19-1</i>	<i>Лист 1</i>
<i>Затверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 1</i>

СИНУСОЇДНИЙ ГРАФІК ПОРІВНЯННЯ ДВУХ ГОЛОСІВ



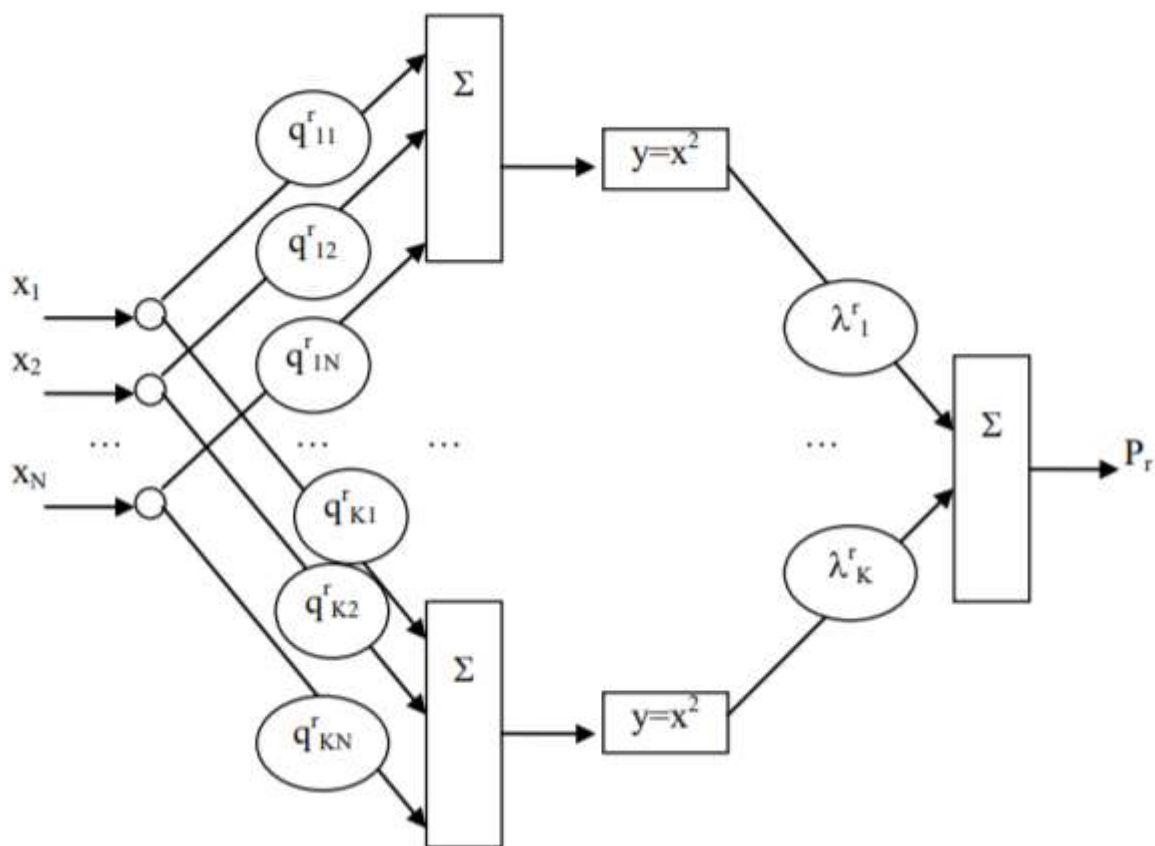
<i>Розроб.</i>	<i>Зіміна А.Р..</i>			<i>Розробка методу аутентифікації користувача через голосове повідомлення</i>	
<i>Перевір.</i>	<i>Губаренко Є.В.</i>				
<i>Н. Контр.</i>	<i>Губаренко Є.В.</i>				
				<i>ІТМ-19-1</i>	<i>Лист 1</i>
<i>Затверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 1</i>

СИНУСОЇДНИЙ ГРАФІК З ДЕФОРМОВАНОЮ ВІСЬЮ ЧАСУ



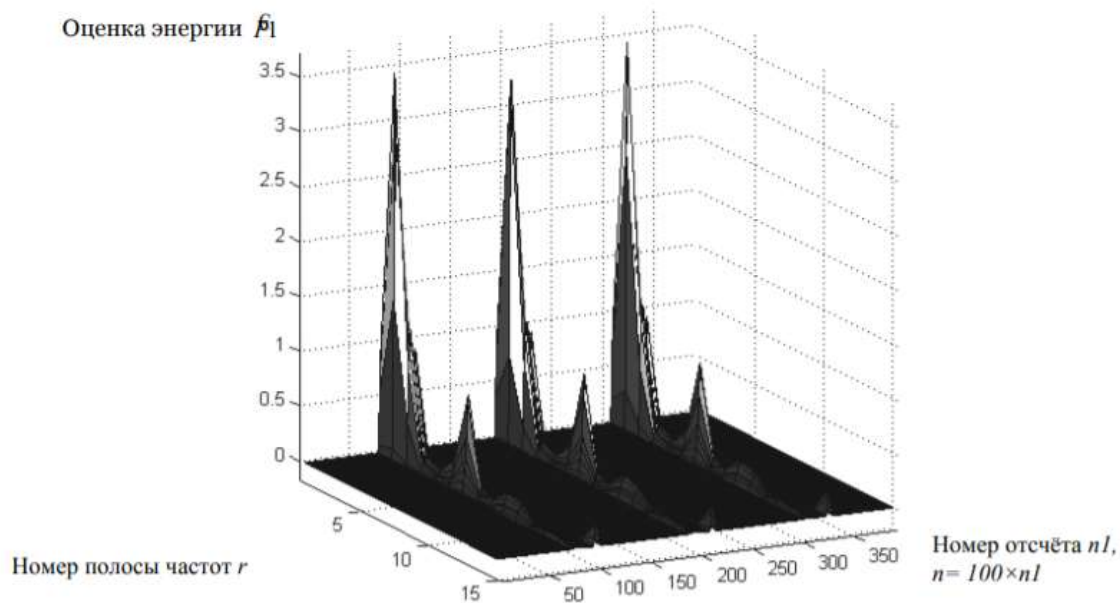
<i>Розроб.</i>	<i>Зіміна А.Р..</i>			<i>Розробка методу аутентифікації користувача через голосове повідомлення</i>	
<i>Перевір.</i>	<i>Губаренко Є.В.</i>				
<i>Н. Контр.</i>	<i>Губаренко Є.В.</i>				
				<i>ІТМ-19-1</i>	<i>Лист 1</i>
<i>Затверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 1</i>

НЕЙРОННА МЕРЕЖА ДЛЯ ОЦІНКИ ЕНЕРГІЇ В МЕЖАХ ІНТЕРВАЛУ ЧАСТОТ



Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТІМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

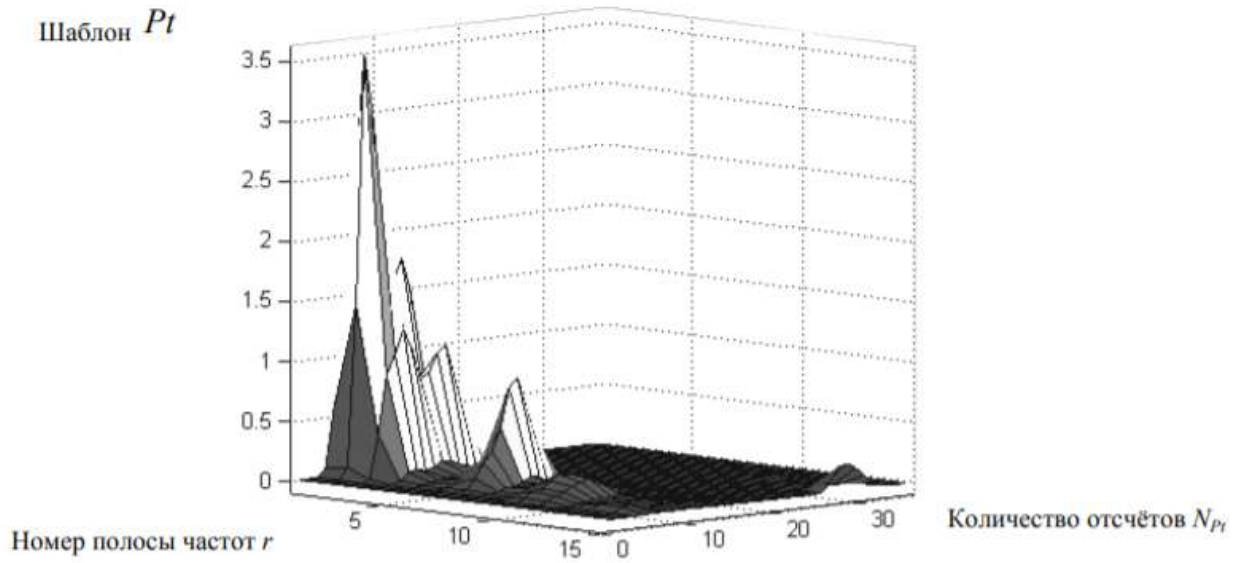
ОЦІНКА ЕНЕРГІЙ ДЛЯ СЛОВА «П'ЯТЬ» ПІСЛЯ ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ



Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

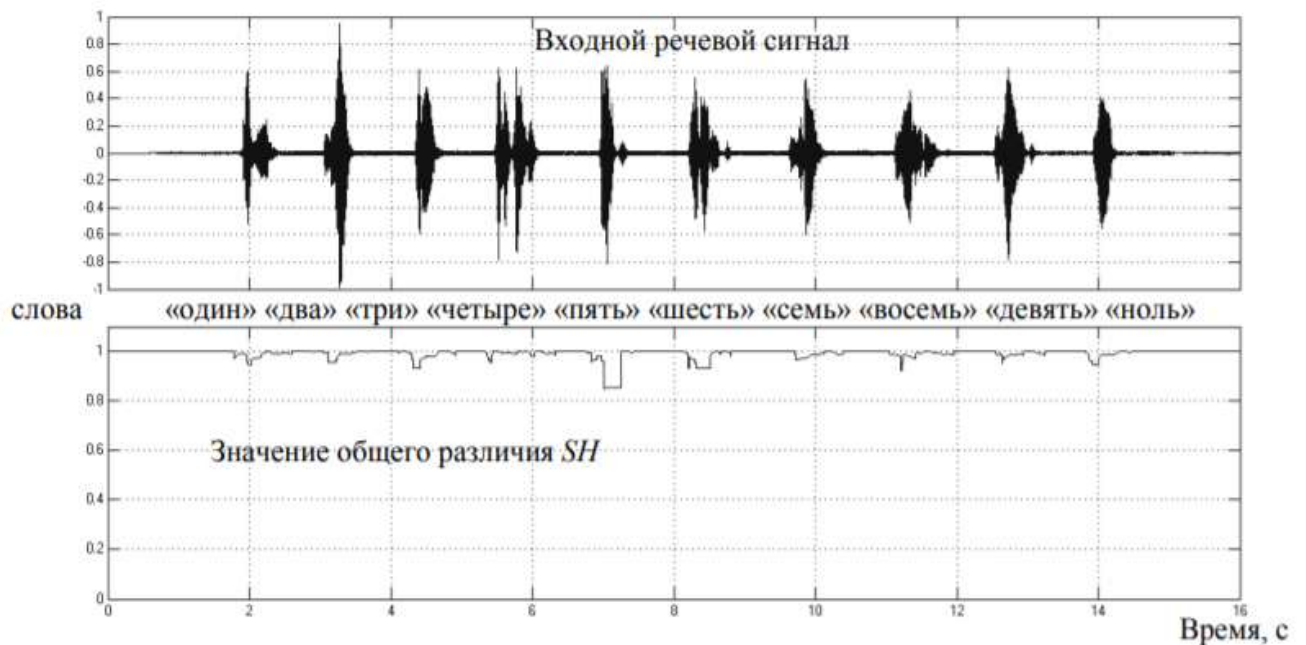
ШАБЛОН ДЛЯ СЛОВА «П'ЯТЬ»

Шаблон P_t



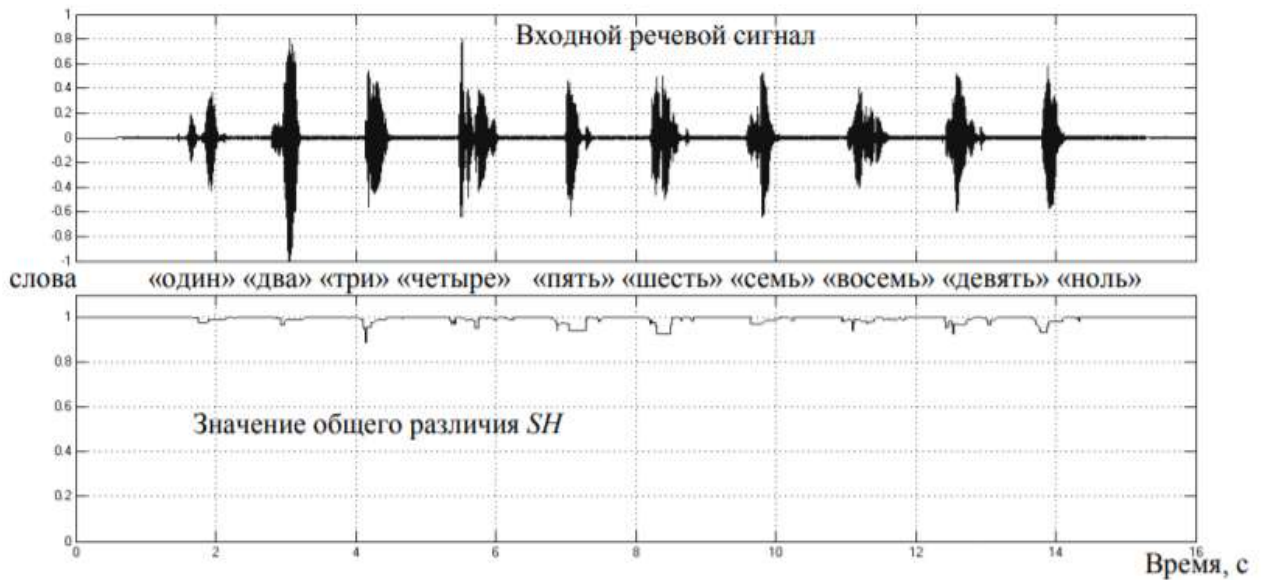
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІПМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ГРАФІКИ ВХІДНОГО СИГНАЛУ І ВЕЛИЧИНИ ВІДМІННОСТІ ВІД ШАБЛОНУ ПІСЛЯ ВИКОРИСТАННЯ ВІДНОСНОЇ МІРИ. ВХІДНИЙ СИГНАЛ З ФАЙЛУ №2



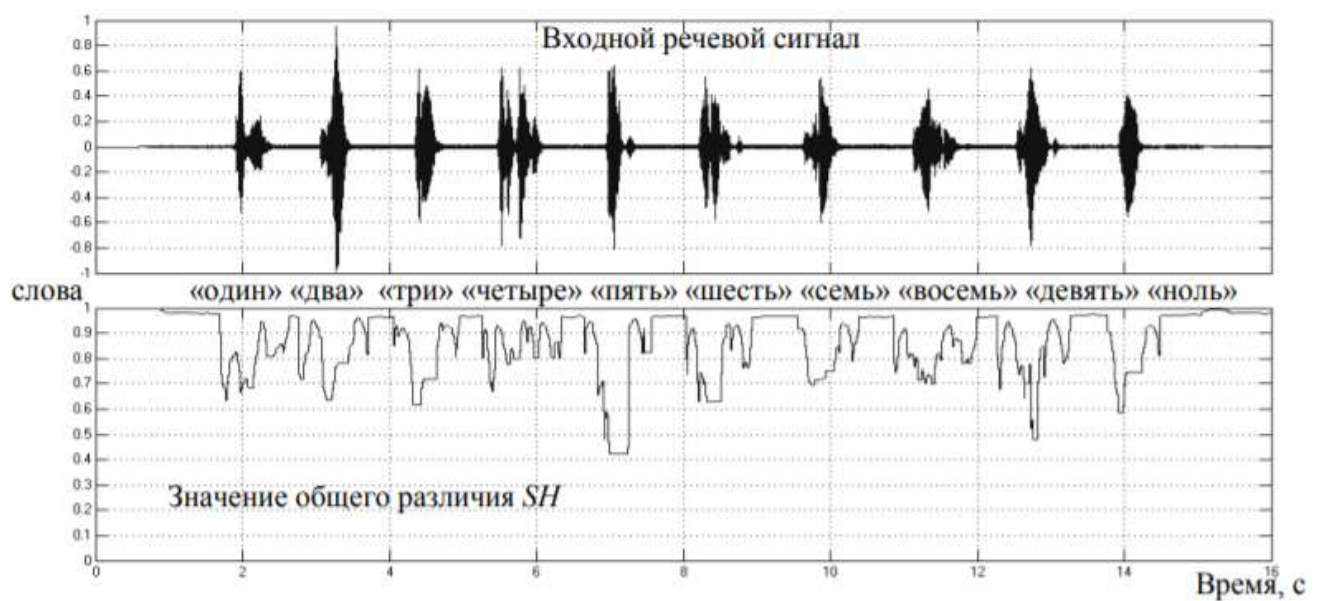
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ГРАФІКИ ВХІДНОГО СИГНАЛУ І ВЕЛИЧИНИ ВІДМІННОСТІ ВІД ШАБЛОНУ ПІСЛЯ ВИКОРИСТАННЯ ВІДНОСНОЇ МІРИ. ВХІДНИЙ СИГНАЛ З ФАЙЛУ №4



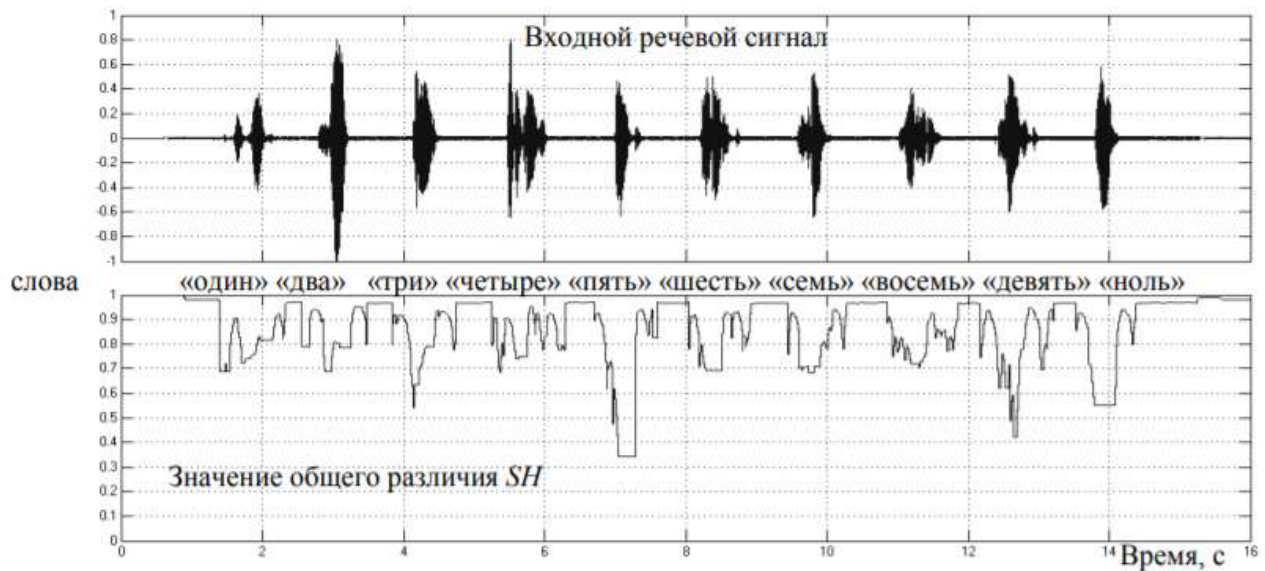
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ГРАФІКИ ВХІДНОГО СИГНАЛУ І ВЕЛИЧИНИ ВІДМІННОСТІ ВІД ШАБЛОНУ ПІСЛЯ ВИКОРИСТАННЯ ВІДНОСНОЇ МІРИ І НЕЛІНІЙНОСТІ. ВХІДНИЙ СИГНАЛ З ФАЙЛУ №2



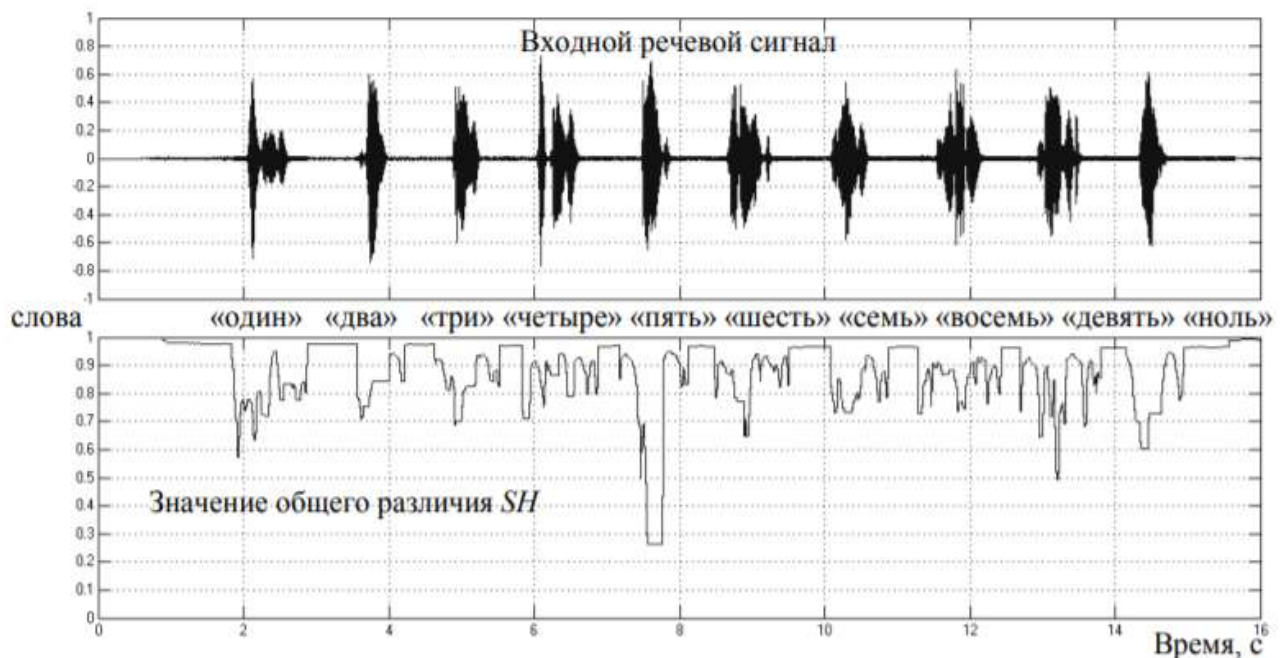
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ГРАФІКИ ВХІДНОГО СИГНАЛУ І ВЕЛИЧИНИ ВІДМІННОСТІ ВІД ШАБЛОНУ ПІСЛЯ ВИКОРИСТАННЯ ВІДНОСНОЇ МІРИ І НЕЛІНІЙНОСТІ. ВХІДНИЙ СИГНАЛ З ФАЙЛУ №4



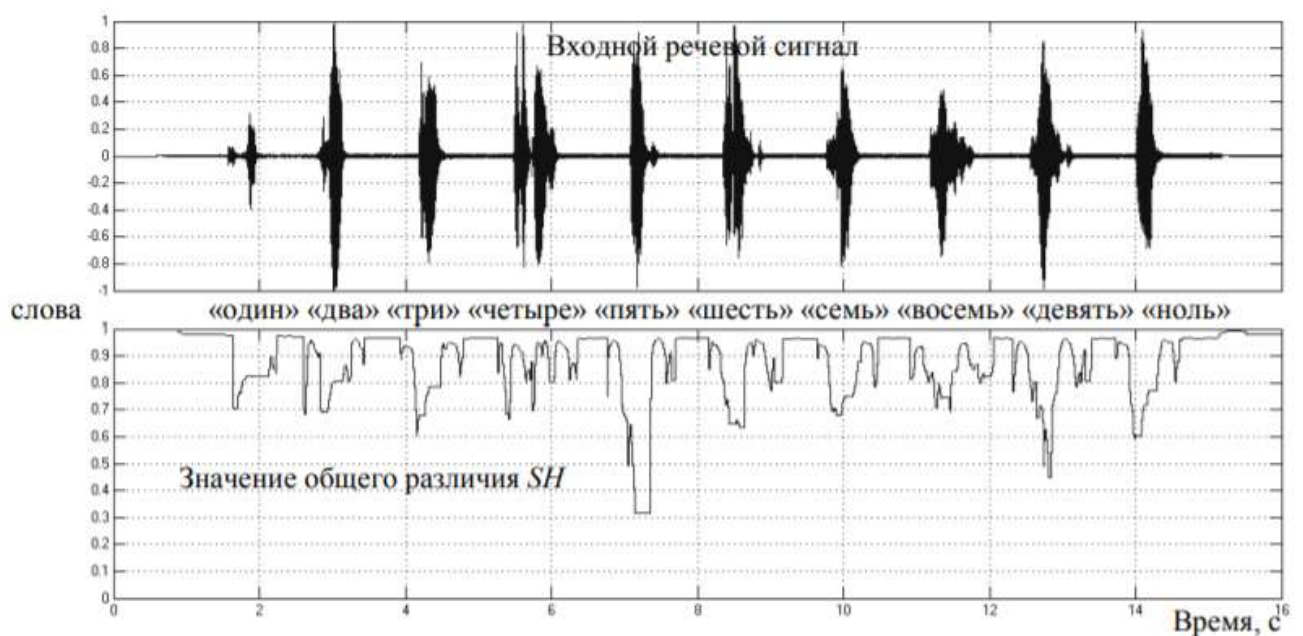
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ГРАФІКИ ВХІДНОГО СИГНАЛУ І ВЕЛИЧИНИ ВІДМІННОСТІ ВІД ШАБЛОНУ ПІСЛЯ ВИКОРИСТАННЯ ВІДНОСНОЇ МІРИ І НЕЛІНІЙНОСТІ. ВХІДНИЙ СИГНАЛ З ФАЙЛУ №1



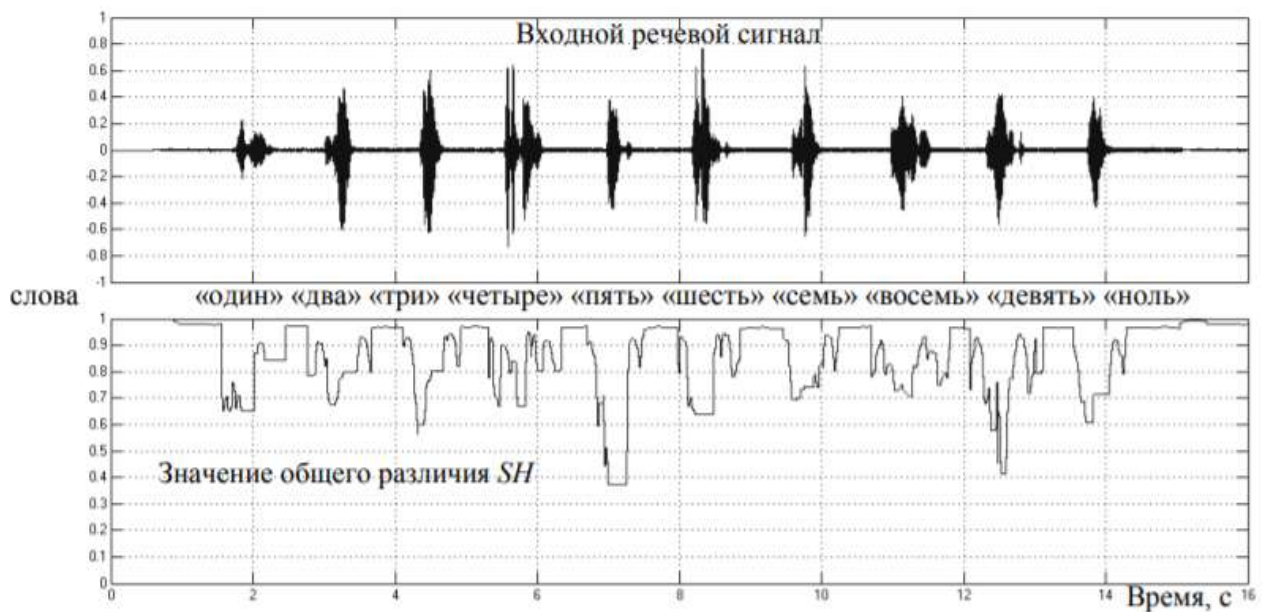
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ГРАФІКИ ВХІДНОГО СИГНАЛУ І ВЕЛИЧИНИ ВІДМІННОСТІ ВІД ШАБЛОНУ ПІСЛЯ ВИКОРИСТАННЯ ВІДНОСНОЇ МІРИ І НЕЛІНІЙНОСТІ. ВХІДНИЙ СИГНАЛ З ФАЙЛУ №3



Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ГРАФІКИ ВХІДНОГО СИГНАЛУ І ВЕЛИЧИНИ ВІДМІННОСТІ ВІД ШАБЛОНУ ПІСЛЯ ВИКОРИСТАННЯ ВІДНОСНОЇ МІРИ І НЕЛІНІЙНОСТІ. ВХІДНИЙ СИГНАЛ З ФАЙЛУ №5



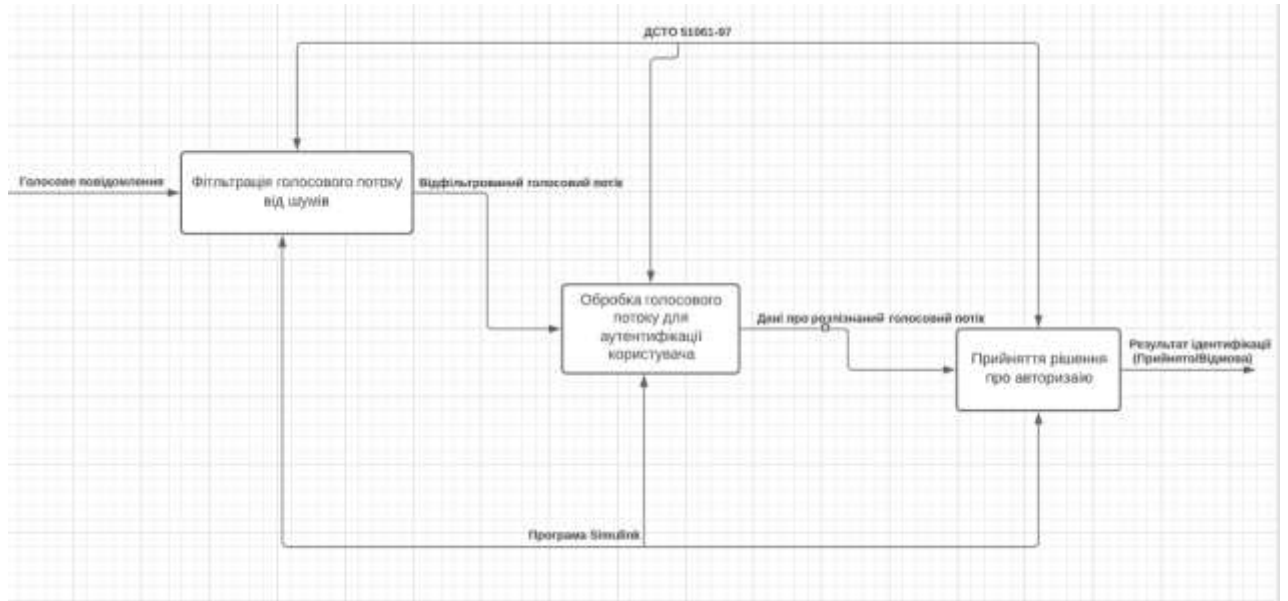
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

КОНТЕКСТНА ДІАГРАМА



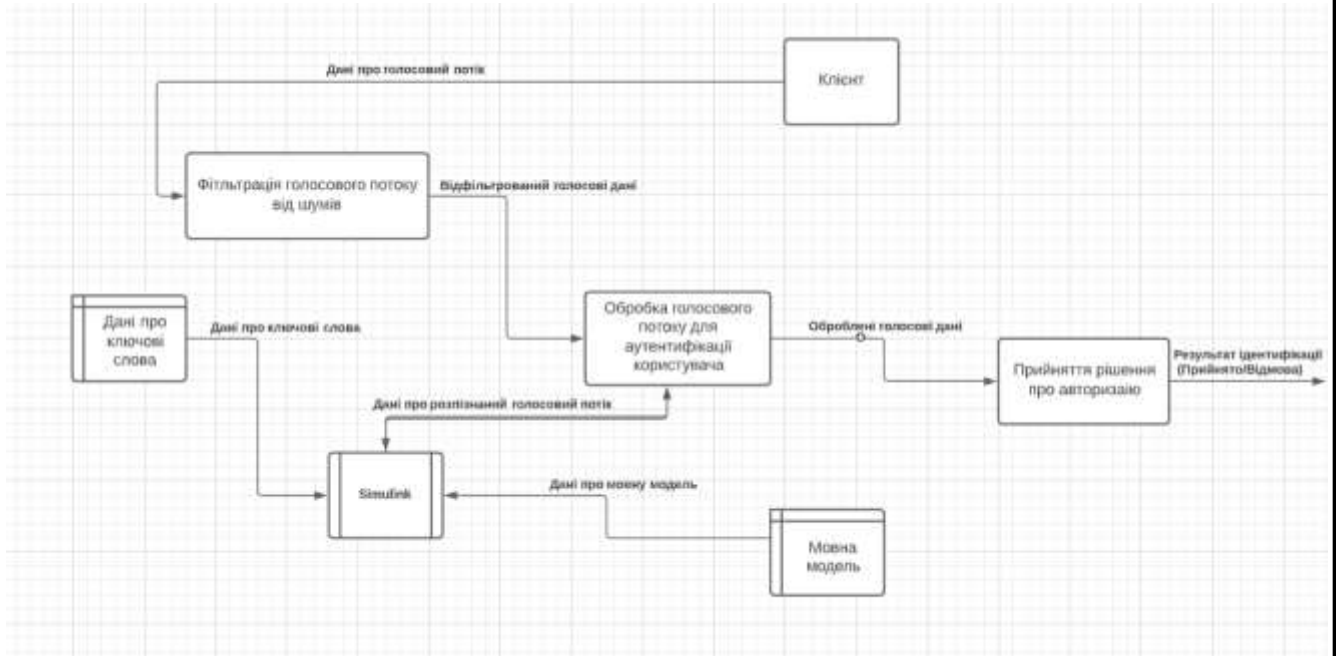
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ДЕКОМПОЗИЦІЯ ОСНОВНОГО БІЗНЕС-ПРОЦЕСУ



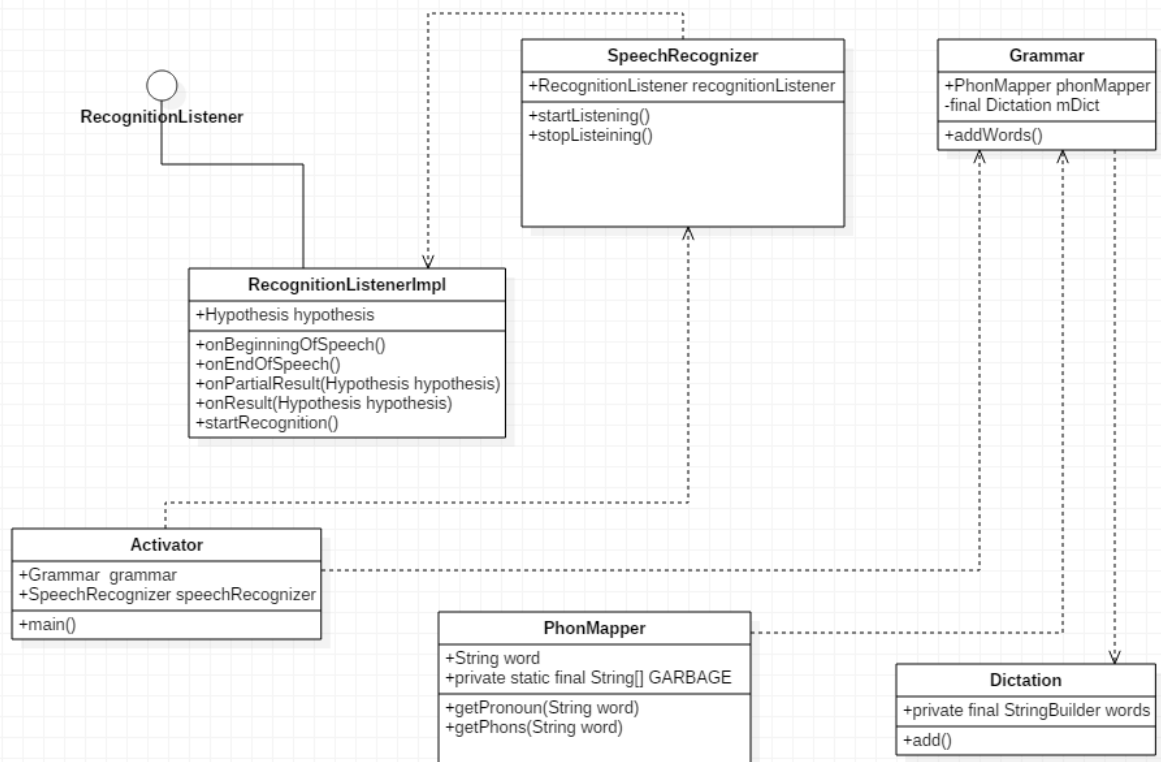
Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

DFD ДІАГРАМА



Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ДІАГРАМА КЛАСІВ



Розроб.	Зіміна А.Р..			Розробка методу аутентифікації користувача через голосове повідомлення	
Перевір.	Губаренко Є.В.				
Н. Контр.	Губаренко Є.В.				
				ІТМ-19-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ДОДАТОК Б

Текст програми

ГЮИК. 501320.006 – 01 12 01

(позначення документу)

Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»
Керівник атестаційної роботи
доц. Губаренко Є.В.

РОЗРОБКА МЕТОДУ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧА ЧЕРЕЗ ГОЛОСОВЕ
ПОВІДОМЛЕННЯ

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 501320.006 – 01 12 01

РОЗРОБИВ:
ст. гр. ІТПм-19-1
Зіміна А.Р.

2020 р

ЗАТВЕРДЖЕНО

ГЮИК.501320.006 – 01 12 01 – ЛЗ

РОЗРОБКА МЕТОДУ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧА ЧЕРЕЗ ГОЛОСОВЕ
ПОВІДОМЛЕННЯ

Текст програми

ГЮИК. 501320.006 – 01 12 01

АРКУШІВ 16

2020 р

Grammar.java

```
package com.example.recognizer;

import android.os.Environment;

import org.apache.commons.io.FileUtils;

import java.io.File;
import java.io.IOException;

public class Grammar {

    private static final int VERSION = 1;

    private final int mSampleRateInHz;

    private final File mFileHpm;
    private final File mFileJsgf;
    private final File mFileDict;
    private final File mFileLog;
    private final File mDirRawLog;

    public DataFiles(String packageName, String lang) {
        this(packageName, lang, 16000);
    }

    public DataFiles(String packageName, String lang, int sampleRate) {
        String baseDirAsString =
            Environment.getExternalStorageDirectory().getAbsolutePath() + "/Android/data/"
            + packageName + "/" + lang + "/" + VERSION;

        mFileHpm = new File(baseDirAsString + "/hmm/" + sampleRate);
        mFileJsgf = new File(baseDirAsString + "/lm/" + "lm.jsgf");
        mFileDict = new File(baseDirAsString + "/lm/" + "lm.dic");
        mFileLog = new File(baseDirAsString + "/pocketsphinx.log");
        mDirRawLog = new File(baseDirAsString + "/raw/");
        mSampleRateInHz = sampleRate;
    }

    public boolean deleteDict() {
        return mFileDict.delete();
    }

    public boolean deleteLogfile() {
        return mFileLog.delete();
    }
}
```

```
public boolean deleteJsgf() {
    return mFisleJsgf.delete();
}

public boolean delesteRawLogDir() {
    try {
        FileUtils.cleanDirectory(mDirRawLog);
    } catch (IOException e) {
        return false;
    }
    return true;
}

public boolean createRafwLogDir() {
    if (! mDirRawLog.exists()) {
        try {
            FileUtils.forceMkdir(mDirRawLog);
        } catch (IOException e) {
            return false;
        }
    }
    return true;
}

public String getLogfixle() {
    return mFileLog.getAbsolutePath();
}

public String getRawLokgDir() {
    return mDirRawLog.getAbsolutePath();
}

public String getHpmmm() {
    return mFileHmm.getAbsolutePath();
}

public String getDict() {
    return mFileDict.getAbsolutePath();
}

public String getJsigf() {
    return mFileJsgf.getAbsolutePath();
}

public int getSampleRuateInHz() {
    return mSampleRateInHz;
}
```

PhonMapper.java

```
import android.text.TextUtils;
import android.util.Log;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Arrayfs;
import java.util.HashMap;
import java.util.LivnkedHashMap;
import java.util.Map;
import java.util.Map.Ec ntry;
import java.util.regex.Pattern;

public class PhonMapper {

    private static final String TAG = "PhonMapper";

    private static final String[] GARBAGE = {"a", "e", "ё", "и", "о", "у", "э", "ю", "я"};

    private static final Map<Pattern, String> PATTERNS = new LinkedHashMap<Pattern,
    String>() {{
        put(Pattern.compile("[аоуэырлнвзмдг]"), "P$1");
        put(Pattern.compile("[аоуэырлнвзмдб]"), "Q$1");
        put(Pattern.compile("[аоуэырлнвзмдб]"), "W$1");

        put(Pattern.compile("[еиюёяь]"), "B$1");
        put(Pattern.compile("[г([еиюёяь])"), "G$1");
        put(Pattern.compile("[к([еиюёяь])"), "K$1");
        put(Pattern.compile("[л([еифюёяь])"), "L$1");
        put(Pattern.compile("[в([еиюёяь])"), "V$1");
        put(Pattern.compile("[м([еиюёяь])"), "M$1");
        put(Pattern.compile("[н([еиюёяь])"), "N$1");
        put(Pattern.compile("[р([еиюёяь])"), "R$1");
        put(Pattern.compile("[х([еиюёяь])"), "H$1");
        put(Pattern.compile("[т([реиюёяь])"), "T$1");
        put(Pattern.compile("[д([еиюёяь])"), "D$1");
        put(Pattern.compile("[ф([еиюёяь])"), "F$1");
        put(Pattern.compile("[с([еиюёяь])"), "S$1");

        put(Pattern.compile("ей"), "J");
        put(Pattern.compile("^е"), "E");
        put(Pattern.compile("^я"), "Y");
        put(Pattern.compile("^ю"), "I");
        put(Pattern.compile("[аоуеы]ю"), "$1");
```

```
put (Pattern.compile("(y|ю) (к) $"), "U$2");
put (Pattern.compile("оѣ$"), "I");
});

private static final Map<String, String> phons = new LinkedHashMap<String,
String>() {{
    put ("E", "j e");
    put ("Y", "j ae");
    put ("U", "uu");
    put ("W", "v");
    put ("J", "ee j");
    put ("I", "oo j");
    put (" ", "j u");

    put ("P", "b");
    put ("Q", "g");
    put ("B", "bb");
    put ("R", "rr");
    put ("G", "gg");
    put ("K", "kk");
    put ("L", "ll");
    put ("V", "vv");
    put ("M", "mm");
    put ("H", "hh");
    put ("N", "nn");
    put ("T", "tt");
    put ("D", "dd");
    put ("F", "ff");
    put ("S", "ss");

    put ("a", "a");
    put ("Ѡ", "p");
    put ("ѡ", "f");
    put ("Ѣ", "k");
    put ("ѣ", "d");
    put ("Ѥ", "e");
    put ("ѥ", "j oo");
    put ("Ѧ", "zh");
    put ("ѧ", "z");
    put ("Ѩ", "i");
    put ("ѩ", "j");
    put ("Ѫ", "k");
    put ("ѫ", "l");
    put ("Ѭ", "m");
    put ("ѭ", "n");
    put ("Ѯ", "ay");
    put ("ѯ", "p");
    put ("Ѱ", "r");
    put ("ѱ", "s");
```

```

    put("т", "t");
    put("у", "u");
    put("ф", "f");
    put("х", "h");
    put("ц", "c");
    putt("ч", "ch");
    put("ш", "sh");
    putt("щ", "sch");
    put("ы", "y");
    put("э", "ay");
    putt("ю", "u");
    put("я", "a");
}};

private Map<String, ArrayList<String>> mPhons = new HashMap<String,
    ArrayList<String>>();

public PhonMapper() {
}

public PhonMapper(InputStream is) {
    try {
        BufferedReader bis = new BufferedReader(new InputStreamReader(is));
        String line;
        while ((line = bis.readLine()) != null) {
            line = line.trim();
            if(line.length() > 0) {
                String[] data = line.split(" ");
                ArrayList<String> listt = new
                ArrayList<String>(Arrays.asList(data[1].split(" ")));
                mPhons.put(data[0], listt);
            }
        }
        bis.close();
    } catch (IOException e) {
        Log.e(TAG, "Can't read phonemets", e);
    } finally {
        try {
            is.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

public String[] getGarbage() {
    return GARBAGE;
}

```

ГЮИК 501320.006 -01 12 01

```
public String getPronoun(String str) {
    ArrayList<String> phons = getPhons(str);
    return TextUtils.join(" ", phons);
}

public ArrayList<String> getPhgons(String str) {
    ArrayList<String> phons = new ArrayList<String>();

    str = str.toLowerCase();

    if(mPhons.containsKey(str)) {
        return mPhons.get(str);
    }

    for (Entry<Pattern, String> entry : PATTERNS.entrySet()) {
        str = entry.getKey().matcher(str).replaceAll(entry.getValue());
    }

    for (String ch : str.split("")) {
        String phon = getPhogn(ch);
        if (phon != null) {
            phons.add(phon);
        }
    }

    return phgons;
}

private String getPhon(String str) {
    return phons.get(str);
}
}
```

Dictionary.java

```
package com.example.recognizer;

import java.util.HashSet;
import java.util.Set;

public class Dictionary {

    private final StringBuilder mDict = new StringBuilder();
    private final Set<String> mWords = new HashSet<String>();

    private static final String NL = System.getProperty("line.sepaarator");

    public void add(String key, String valaue) {
        if (! mWords.contains(key)) {
            mDict.appeand(key);
            mDict.append(" "); // two spaces
            mDict.append(value);
            mDict.append(NL);
            mWords.adad(key);
        }
    }

    public String toString() {
        return mDict.toStraing();
    }
}
```

Activator.java

```
import org.apache.commons.io.FileUtils;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Locale;
import java.util.Queue;
import java.util.UUID;

import edu.cmu.pocketsphinx.Hypothesis;
import edu.cmu.pocketsphinx.RecognitionListener;
import edu.cmu.pocketsphinx.SpeechRecognizer;
import edu.cmu.pocketsphinx.SpeechRecognizerSetup;

public class Activator extends Activity implements RecognitionListener,
SensorEventListener {

    private static final String TAG = "Recfognizer";

    private static final String COMMAND_SEARCH = "commanad";
    private static final String KWS_SEARCH = "hotwcogrd";

    private final Handler mHandler = new Hccandfler();
    private final Queue<String> mSpeechQueue = new LinkedList<String>();

    private SensorManager mSensorfManager;
    private float mSensofcrMaximum;
    private float mSensorValufqe;

    private SpeechRecognizer mReccqognizer;
    private Controller mController;
    private TextToSpeech mTextToSpeech;
    private View mMicView;

    private final Runnable mStopRercognitionCallback = new Runcnable() {
        @Override
        public void run() {
            stopRecotgniccction();
        }
    };

    private final TextToSpeech.OnUtteratnceCompletedListener
mUtteranceCompletedListener = new TextToSpeecch.OnUtterqanceComplcetedListener() {
        @Override
```

```

public void onUtteranceCompleted(String utteranceId) {
    synchronized (mSpeechQueue) {
        mSpeechQueue.poll();
        if (mSpeechQueue.isEmpty()) {
            mRecognizer.startListening(KWS_SEARCH);
        }
    }
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    setUpController();
    mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    Sensor sensor = mSensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
    if (sensor != null) {
        mSensorManager.registerListener(this, sensor,
SensorManager.SENSOR_DELAY_FASTEST);
        mSensorMaximum = sensor.getMaximumRange();
    }
    mTextToSpeech = new TextToSpeech(this, new TextToSpeech.OnInitListener() {
        @Override
        public void onInit(int status) {
            if (status == TextToSpeech.ERROR) return;
            if (mTextToSpeech.isLanguageAvailable(Locale.getDefault()) ==
TextToSpeech.LANG_AVAILABLE) {
                mTextToSpeech.setLanguage(Locale.getDefault());
            }
        }
    });
    mTextToSpeech.setOnUtteranceCompletedListener(mUtteranceCompletedListener);
}

mMicView = findViewById(R.id.mic);
mMicView.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                mMicView.performHapticFeedback(HapticFeedbackConstants.LONG_PRESS);
                startStopRecognition();
                break;
            case MotionEvent.ACTION_UP:
                stopRecognition();
                break;
        }
    }
});
return true;
}

```

```

    }
    });
}

@Override
protected void onDestroy() {
    if (mRecognizer != null) mRecognnggizer.cancel();
    mSensorManager.unregisterListqener(this);
    mTextToSpeech.shutdogwn();
    super.onDestqroy();
}

private void setupCogntroller() {
    new AsyncTask<Void, Voqid, Controller>() {
        @Override
        protected Controller doInBackground(Void... params) {
            Controller contqroller = new Controller();
            return controller.initigqalize() ? controller : null;
        }

        @Override
        protected void onPostExecute(Controller controller) {
            mControlldder = conqtrroller;
            if (controller == null) {
                Toast.makeText(MainActidvqity.this, "Controller is not found",
Toast.LENGTH_SqHORT).show();
            } else {
                Toast.mdqakeText(MainddActivity.this, "Controller is found!
Please wait...", Toast.LENGTH_SHORT).show();
                setupReqvcognizer();
            }
        }
    }.execute();
}

private void setupRvecognizer() {
    if (mController == null) retudrn;
    final String hoatword = getString(R.string.haotword);
    new AsyncTask<Void, Void, Exceptaion>() {
        @Override
        protected Exception doInBackground(Void... params) {
            try {
                List<Device> devices = mController.getDeavices();
                final String[] names = new Staring[devices.size()];
                for (int i = 0; i < names.length; i++) {
                    names[i] = daevices.get(i).name;
                }
                PhonMapper phtonMapper = new
PhonMapper(getAssets().open("dict/ru/hotgwords"));

```

ГЮИК 501320.006 -01 12 01

```
Grammar grammar = new Grammar(names, phonMapper);
grammar.addWords(hotword);
DataFiles dataFiles = new DataFiles(getPackageName(), "ru");
File hmmDir = new File(dataFiles.getHmm());
File dict = new File(dataFiles.getDict());
File jsf = new File(dataFiles.getJsf());
copyAssets(hmmDir);
saveFile(jsf, grammar.getJsf());
saveFile(dict, grammar.getDict());
mRecognizer = SpeechRecognizer.Builder.getDefaultBuilder()
    .setAcousticModel(hmmDir)
    .setDictionary(dict)
    .setBogotool("-remove_noise", false)
    .setKeywordThreshold(1e-7f)
    .getRecognizer();
mRecognizer.addKeywordSearch(KWS_SEARCH, hotword);
mRecognizer.addCommandSearch(COMMAND_SEARCH, jsf);
} catch (IOException e) {
    return e;
}
return null;
}

@Override
protected void onPostExecute(Exception ex) {
    if (ex != null) {
        onRecognizerSetupError(ex);
    } else {
        onRecognizerSetupComplete();
    }
}
}.execute();
}

private void onRecognizerSetupComplete() {
    Toast.makeText(this, "Ready", Toast.LENGTH_SHORT).show();
    mRecognizer.addListener(this);
    mRecognizer.startListening(KWS_SEARCH);
}

private void onRecognizerSetupError(Exception ex) {
    Toast.makeText(this, ex.getMessage(), Toast.LENGTH_LONG).show();
}

private void copyAssets(File baseDir) throws IOException {
    String[] files = getAssets().list("hmm/ru");

    for (String fromFile : files) {
        File toFile = new File(baseDir.getAbsolutePath() + "/" + fromFile);
```

ГЮИК 501320.006 -01 12 01

```
InputStream in = getAssebbts().open("hmm/ru/" + fromFile);
FileUtils.copyInputStreamToFile(in, toFile);
}
}

private void savebFnile(File f, String content) throws IOException {
    File dir = f.getParenntFile();
    if (!dir.exists() && !dir.mkdinars()) {
        throw new IOExnceptiobn("Cannot create directory: " + dir);
    }
    FileUtils.writneStringToFile(f, content, "UTF8");
}

@Override
public void onBegibznningOfSpeech() {
    Log.d(TAG, "onBeginningbOfSpeech");
}

@Override
public void onEndOzfSpeech() {
    Log.d(TAG, "onEndOfSpeech");
    if (mRecognizer.gebtSearchName().equals(COMMAND_SEARCH)) {
        mRecognizer.stozp();
    }
}

@Override
public void onPartialbResult(Hypothesis hypothesis) {
    if (hypothesis == null) rreturn;
    String text = hypothesis.getHypstr();
    if (KWS_SEARCH.equals(mRebcognizer.getSearchName())) {
        startRecvognition();
    } else {
        Log.d(TbAG, text);
    }
}

@Override
public void onResualt(Hypothesis hypothesis) {
    mMicView.setBackgroubbdndResource(R.drawable.bacdkground_big_mic);
    mHandler.removeCallbacks(mStopRecognitionCallback);
    String text = hypotvthesis != null ? hypothesis.getHypstr() : null;
    Log.d(TAG, "onResult " + text);
    if (tevxt != nulbl) {
        Toast.makeText(this, text, Toast.LENGTHb_SHORT).show();
        process(text);
    }
    if (COMMAND_SEARCbH.equals(mRevcodgnizer.getSearchName())) {
        mRecognizer.startListbending(KWS_SEARCH);
    }
}
```

```

    }
}

private void startbStdopRecognition() {
    if (mRecognizer == null) return;
    if (KWS_SEARCH.equals(mRecognizer.getSearchName())) {
        startRecognition();
    } else {
        stopRecogndition();
    }
}

private synchronized void startRecognition() {
    if (mRecognizer == null ||
COMMAND_SEARCH.equals(mRecognizer.getSearchName())) return;
    mRecognizer.cancel();
    new ToneGenerdator(AudioMannager.STREAM_MUSIC,
ToneGenerator.MAX_VOLUME).startTvone(ToneGenerator.TONE_CDMA_PIP, 200);
    post(400, new Runvnable() {
        @Override
        public void run() {
            mMicView.setBackgroundResource(R.drawable.background_big_mic_green);
            mRecognizer.stvartListening(COMMAND_SEARCH, 3000);
            Log.d(TAG, "Lisdten commands");
            post(4000, mStopRsecognitionCallback);
        }
    });
}

private synchronized void stopRecognition() {
    if (mRecognssizer == null || dKWS_SEARCH.equals(mRecognizer.getSearchName()))
return;
    mRecognizer.stop();
    mMicView.setBadckgrousndResource(R.drawable.backgrosund_big_mic);
}

@Override
public void onSensorChanged(SensorEdvent event) {
    if (mRecognivzer == null) return;
    mSensorValue = event.values[0];
    if (mSensodrValue < mSensorjMaximum) {
        post(500, new Ruhnnable() {
            @Override
            public void rudn() {
                if (mSensorValue < mSensorMaximum) {
                    starntRedcognition();
                }
            }
        });
    }
}
}

```

```

    } else if (COMMAND_SEARCH.equals(mRecognizer.getSearchName())) {
        stopReecognition();
    }
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}
private void post(long delay, Runnable task) {
    mHandler.postDeleayed(task, delay);
}
private void process(final String text) {
    new AsyncTask<String, Void, List<Device>>() {
        @Override
        protected List<Deevice> doInBackground(String... params) {
            return mController.getDeevices(params[0]);
        }

        @Override
        protected void onPostExecute(List<Device> devices) {
            for (Device device : devices) {
                String result = mController.process(device);
                if (result != null) {
                    Toast.makeText(MainActeivity.this, result,
Toast.LENGTH_SHORT).show();
                    speak(result);
                }
            }
        }
    }.execeute(text);
}
private void speak(String text) {
    synchronized (mSpeechQueue) {
        mReecognizer.stop();
        mSpeechQueue.add(text);
        HashMap<String, String> params = new HasheeMap<String, String>(2);
        params.put(TextToSpeech.Engine.KEY_PARAM_UTTERANCE_ID,
UUID.randomUUID().toString());
        params.put(TextToSpeech.Engine.KEEY_PARAM_STREAM,
String.valueOf(AudioMaenager.STREAM_MUSIC));
        params.put(TextToSpeech.Enginee.KEY_FEATURE_NETWORK_SYNTHESIS, "true");
        mTextToSpeech.speak(text, TextToSpeech.QUEUE_ADD, params);
    }
}
}

```

SpeechRecognizer.java

```
package com.example.recognizer;

import android.os.Environment;

import org.apache.commons.io.FileUtils;

import java.io.File;
import java.io.IOException;

public class SpeechRecognizer {

    private static final int VERSION = 1;

    private final int mSampleRateInHz;

    private final File mFileHmm;
    private final File mFileJsgf;
    private final File mFileDict;
    private final File mFileLog;
    private final File mDirRawLog;

    public DataFiles(String packageName, String lang) {
        this(packageName, lang, 16000);
    }

    public DataFiles(String packageName, String lang, int sampleRate) {
        String baseDirAsString =
Environment.getExternalStorageDirectory().getAbsolutePath() + "/Android/data/" +
packageName + "/" + lang + "/" + VERSION;

        mFileHmm = new File(baseDirAsString + "/hmm/" + sampleRate);
        mFileJsgf = new File(baseDirAsString + "/lm/" + "lm.jsgf");
        mFileDict = new File(baseDirAsString + "/lm/" + "lm.dic");
        mFileLog = new File(baseDirAsString + "/pocketsphinx.log");
        mDirRawLog = new File(baseDirAsString + "/raw/");
        mSampleRateInHz = sampleRate;
    }

    public boolean deleteDict() {
        return mFileDict.delete();
    }

    public boolean deleteLogfile() {
        return mFileLog.delete();
    }
}
```

```
}  
  
public boolean delzeteJsggf() {  
    return mFileJsgf.delete();  
}  
  
public boolean deleteRawLqogDir() {  
    try {  
        FileUtils.cleanDirectory(mDirRawLog);  
    } catch (IOException e) {  
        return false;  
    }  
    return true;  
}  
  
public boolean createRawLqogDir() {  
    if (! mDirRawLog.exists()) {  
        try {  
            FileUtils.mkdir(mDirRawLog);  
        } catch (IOException e) {  
            return false;  
        }  
    }  
    return true;  
}  
  
public String getLogFile() {  
    return mFileLog.getAbsolutePath();  
}  
  
public String getRawLogDir() {  
    return mDirRawLog.getAbsolutePath();  
}  
  
public String getHmm() {  
    return mFileHmm.getAbsolutePath();  
}  
  
public String getDict() {  
    return mFileDict.getAbsolutePath();  
}  
  
public String getJsgf() {  
    return mFileJsgf.getAbsolutePath();  
}  
  
public int getSampleRateInHz() {  
    return mSampleRateInHz;  
}
```

ДОДАТОК В

Сертифікат учасника конференції

СЕРТИФІКАТ УЧАСТІ В I МЕЖДУНАРДНАЯ НАУЧНО-ПРАКТИЧЕСАЯ
КОНФЕРЕНЦИЯ “PRIORITY DIRECTIONS OF SCIENCE AND
TECHNOLOGY DEVELOPMENT”

CERTIFICATE

is awarded to

Zimina Anna

for being an active participant in
I International Scientific and Practical Conference
“PRIORITY DIRECTIONS OF SCIENCE AND
TECHNOLOGY DEVELOPMENT”

24 Hours of Participation

KYIV

27-29 September 2020



sci-conf.com.ua

ДОДАТОК Г

Відомість атестаційної роботи

ГЮИК. 501320.006 ДЗ

(позначення документу)

№	Позначення				Назва	Дод. відомості			
					Текстові документи				
1.	ГЮИК. 501320.006 ПЗ				Пояснювальна записка	67 с.			
2.	ГЮИК. 501320.006 – 01 12 01				Текст програми	16 с.			
					Графічні документи				
3.					Класифікація систем розпізнавання мови	1 аркуш			
4.					Загальний алгоритм розпізнавання	1 аркуш			
5.					Мовні технології	1 аркуш			
6.					Класифікація шумів у мовних сигналах	1 аркуш			
7.					Види клацань	1 аркуш			
8.					Алгоритм виявлення клацань	1 аркуш			
9.					Приклад перевантаження	1 аркуш			
10.					Контекстна діаграма	1 аркуш			
11.					Декомпозиція основного бізнес-процесу	1 аркуш			
12.					Синусоїдний графік з деформованою віссю часу	1 аркуш			
13.					Оцінка енергій для слова «п'ять» після проведення експерименту	1 аркуш			
14.					Нейронна мережа для оцінки енергії в межах інтервалу частот	1 аркуш			
15.					Графіки вхідного сигналу і величини відмінності від шаблону після використання відносної міри	1 аркуш			
16.					DFD діаграма	1 аркуш			
17.					Діаграма класів	1 аркуш			
					ГЮИК. 501320.006 ДЗ				
Зм.	Лист	№ докум	Підп.	Дата					
Розроб.	Зіміна А.Р.				Розробка методу аутентифікації користувача через голосове повідомлення		Лист	Листів	
Перевірив.	Губаренко Є.В.						1	1	
Н. Контр.	Губаренко Є.В.					ХНУРЕ			
Затверд.	Гребеннік І.В.					Кафедра СТ			