# ISOLATED SPEECH COMMANDS RECOGNITION BY USING HYBRID APPROACH

Tomas Rasymas
Kaunas faculty, Vilnius University, Muitinės str. 8, Kaunas, Lithuania
tomas.rasymas@gmail.com https://www.vu.lt

Vytautas Rudžionis
Kaunas faculty, Vilnius University, Muitinės str. 8, Kaunas, Lithuania
vytautas.rudzionis@knf.vu.lt .https://www.vu.lt

**Abstract This paper presents our results obtained by experimenting with different classifiers and different types of features to combine different speech recognizers in order to increase isolated speech commands recognition accuracy. Two different speech corpora were used to evaluate classifiers and features performance: "Asmens kodai" – person code which consists of 11 digits, "Skaiciai" – digits from 0 to 9. Compared suggested speech recognizers combination method with best single recognizer accuracy increased based on corpora: "Asmens kodai" – 0.352 % and "Skaiciai" – 1.057 %. Compared with combination that was made using voting method accuracy increased based on corpora: "Asmens kodai" – 18.7 % and "Skaiciai" – 2.9 %. Classifiers that were evaluated in different speech recognizers combination task: Naïve Bayes, Random forest, Nearest neighbors, CART and Support vector classifiers. Highest accuracy increase obtained by using Random forest classifier.**

**Keywords:** Hybrid speech recognition; recognizers combination; intelligent systems; classifiers evaluation.

## 1. Introduction

There are a lot of different methods in the world introduced to perform speech recognition: different speech signal features, different classification methods, different training methods and so on. Different speech recognition methods have different pros and cons, what is more they use only part of useful speech signal information and another part of information is left unused. It is not possible to except methods that are best suited for speech recognition task. It depends on a lot of factors: environment that system will work, size of training data, voice properties of end users and so on. Reach high speech recognition accuracy by using standard methods is very hard or even impossible task. For that reason, hybrid approaches become more and more popular, because they allow to combine different standard methods for speech recognition and increase overall recognition system accuracy. Hybrid approach is one of the ways to achieve higher recognition accuracy of speech processing system. By the term hybrid approach, we understand the incorporation of several different recognition algorithms or methods. The basic idea behind the hybrid approach is that different recognition methods are able to extract and to process different kinds of information present in the acoustic signal and these types of information aren't completely correlated. It means that if they are used together this could lead to the overall increase of recognition accuracy and robustness.

What is more hybrid approach is one of the ways to create accurate speech recognition systems with minimum resources. Better solution would be to use hybrid speech recognition approach with adapted foreign language recognizers. Potentially even better solution could be use of several foreign language recognizers and adapting them for our needs with the hope that different recognizers will provide capabilities in different situations. In other words, we need try to use adapted foreign language recognizers in hybrid recognition approach. This approach is very important for all under resourced languages, because the development of any speech recognition system requires enormous resources: both material and human. It is difficult to gather such resources in countries were relatively not widely spoken languages are used as a primary tool for communication.

The idea of creating hybrid speech recognizer and adapting other languages acoustic models is not new [Rasymas and Rudžionis (2015a), Rasymas and Rudžionis (2015b), Rudžionis et al (2013), Lojka and Juhar (2014)]. These kinds of researches are especially important for all under resourced languages. There were successful attempts to estimate acoustic models for new target language using speech data from varied source languages, but only limited data from the target language [Schultz et al (2001), Wang et al (2003)]. Features combination is another area of research which allows increase recognition accuracy [Zolnay et al (2005)]. For continue speech recognizers combination researchers are experimenting with ROVER and confusion network methods which shows good results in different speech recognition systems combination [Siohan and Rybach

(2015), Meneido and Neto (2000)]. As for isolated commands recognition, there are too few researches on different recognizers combination for isolated command recognition.

Almost all popular speech recognizers combination methods (like Rover, etc.) are designed for continues speech recognition systems. While isolated speech commands recognition problem stays open. In this paper, we are proposing new architecture for isolated speech commands recognition by using hybrid approach. We will use seven different speech recognizers for combination: two Lithuanian language (Google Speech API ("lt_g") and LIEPA acoustic model ("lt")), two adapted English acoustic models (CMU Sphinx ("en") and Wall Street Journal ("en_wsj")), two Russian acoustic models (CMU Sphinx ("ru_cmu") and VoxForge ("ru")) and one Spanish acoustic model (CMU Sphinx ("es")). Foreign speech recognizers adaptation was made by using transcription rewriting rules that were obtained experimentally. For the quicker classification methods realization, we used scikit-learn library. Scikit-learn is an open source machine learning library for the Python programming language [Pedregosa et al (2011)]. PocketSphinx speech recognition toolkit was used to perform recognition with HMM based acoustic models [Huggins-Daines et al (2006)]. What is interesting that Lithuanian language Google Speech API recognizer is based on deep neural network remaining recognizers are based on hidden Markov models.

## 2. Proposed method

Proposed hybrid isolated speech command recognition method consists of three main parts: single different recognizers, features from recognizers and classifier to combine different recognizers. Diagram of such system is displayed in Fig. 1. In proposed system, each single speech recognizer is considered as "black box". Each recognizer gets speech signal and produces its output. It does not matter what speech signal features are used (MFCC, LPC and etc.), what classification method recognizer is based on (HMM or deep learning).

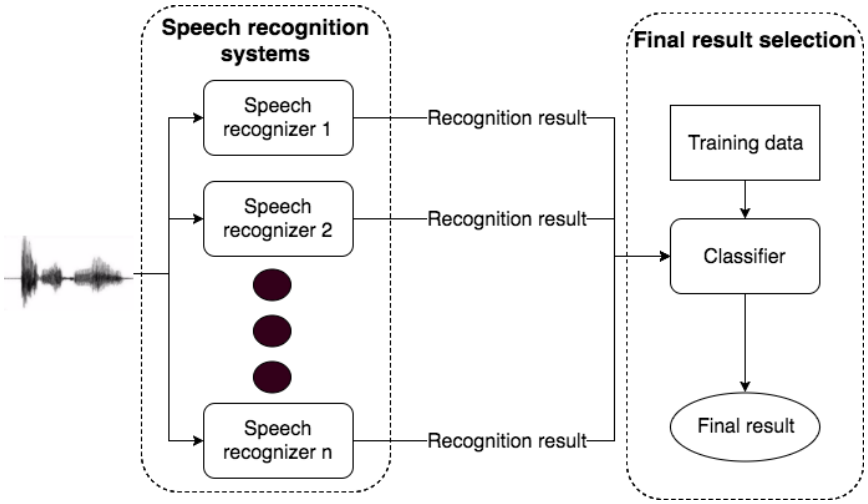


Fig. 1. This is the caption for the figure. If the caption is less than one line then it needs to be manually centered.

Recognition process can be described as follows: speech signal is passed to all different speech recognizers. They perform recognition and returns their recognition results. Typically, recognition result contains of best hypothesis and some alternative hypothesis. Hypothesis – recognized text and some statistical recognition values, like probability. When all recognizers finish recognition process, and has recognition results, those results are pasted to trained classifier to predict which recognizers result to use as final. To use such system two problems must be solved: which features to use for classifier training and predicting and what classifier to use for final prediction making.

As result each speech recognizer produces best hypothesis and some alternatives hypothesis. In proposed system evaluation, we are using best hypothesis and 4 alternative best hypotheses. Hypothesis can be imagined as array of three elements: recognized text, recognition probability and score of how well audio matches model. Only recognition probability and score of how well audio matches model from hypotheses was used to create feature vectors for combination process. So, from one recognizer we get five hypothesis of recognition probability ("prob") and score of how well audio matches model ("score") pairs. From those features we generated few more additional features for every recognizer: average of recognition probability ("prob_best_avg"), average of score of how well audio matches model ("score_best_avg"), average of recognition probability if hypothesis text matches ("prob_best_if_avg"), average of score of how well audio matches model if hypothesis text matches ("score_best_avg"), maximum recognition probability ("prob_max"), maximum score of how well audio matches model ("score_max"). Cause we are using seven recognizers we have big feature vector and not all features are valuable for us. So, Mutual information feature selection method

was used for final feature vector creation. We created two feature vectors with best 10 and 20 features. What is more for each corpus we created different feature vectors. All feature sets used are displayed in Table 1.

Table 1. Feature sets used for evaluation.

| Features | Description |
|---|---|
| prob_lt_g, score_lt_g, score_lt, prob_en, prob_en_wsj, prob_ru, score_ru_1, score_ru_4, prob_es, score_ru_best_max | Feature set name: **asmens_kodai_10**, corpus: "Asmens kodai". |
| prob_lt_g, score_lt_g, prob_lt, score_lt, score_lt_1, prob_en_wsj, prob_ru_cmu, prob_ru, score_ru_1, score_ru_3, score_ru_4, prob_es, score_es_3, score_en_wsj_best_avg, score_ru_best_avg, score_es_best_avg, score_lt_best_if_avg, score_ru_cmu_best_if_avg, score_ru_best_if_avg, score_ru_best_max | Feature set name: **asmens_kodai_20**, corpus: "Asmens kodai". |
| prob_lt_g, score_lt_g, score_lt, score_lt_1, score_en_wsj, score_en_wsj_1, score_lt_best_avg, score_en_wsj_best_avg, score_lt_best_if_avg, score_en_wsj_best_if_avg | Feature set name: **skaiciai_10**, corpus: "Skaiciai". |
| prob_lt_g, score_lt_g, prob_lt, score_lt, score_lt_1, prob_en, score_en, score_en_1, score_en_wsj, score_en_wsj_1, score_ru_cmu, score_ru_cmu_1, score_ru_cmu_2, prob_ru, score_ru_1, prob_es, score_lt_best_avg, score_en_wsj_best_avg, score_lt_best_if_avg, score_en_wsj_best_if_avg | Feature set name: **skaiciai_20**, corpus: "Skaiciai". |

As final decision making classifier we analyzed five different classifiers: Naïve Bayes, Random forest, Nearest neighbors, Support vector and CART. Those classifiers were selected based on other researchers experiments and obtained results [Wu et al (2008), Sharam et al (2013), Jain et al (2016), Bozkurt et al (2015), Tchendjou et al (2016) and Delgado et al (2014)]. Those classifiers showed very high classification accuracy in different areas, so we are evaluating them in speech recognizers combination task. To evaluate classifier efficiency, we selected few main hyperparameters that influence classifier accuracy most and tried to search optimal hyperparameters values that produces best classification accuracy.

## 3. Experiment data preparation

Two different corpus were created in order to evaluate proposed method: "Skaiciai" and "Asmens kodai". Data for each corpus were recorded in silent room. Gathered recordings were saved in wave format (16000 Hz, 16 bytes, one channel). Corpus "Asmens kodai" consists of person personal code. Person code consist of 11 digit. First digit must be 3 (male) or 4 (female). 11 persons (4 women and 7 men) took part in gathering recordings for "Asmens kodai" corpus. Every person repeated its personal code for 20 times, so we gathered 220 recordings. "Asmens kodai" corpus grammar in JSGF format is displayed in Table 2.

Table 2. JSGF grammar of "Asmens kodai" corpus.

```
#JSGF V1.0;

grammar PersonalCode;
public <main> = <first> <number> <number> <number> <number> <number> <number> <number>
<number> <number> <number>;

<first> = (3 | 4);
<number> = (0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9);
```

Corpus "Skaiciai" consist of digits from 0 to 9. Corpus recordings were gathered from different sources, so number of recordings of every participant is different. We gathered 500 recordings of every digit. "Skaiciai" corpus grammar in JSGF format is displayed in Table 3.

Table 3. JSGF grammar of "Skaiciai" corpus.

```
#JSGF V1.0;

grammar numbers;
public <numbers> = nulis | vienas | du | trys | keturi | penki | šeši | septyni | aštuoni | devyni;
```

First of all, single speech recognizers were evaluated with created corpus. Results are displayed in Fig. 2.

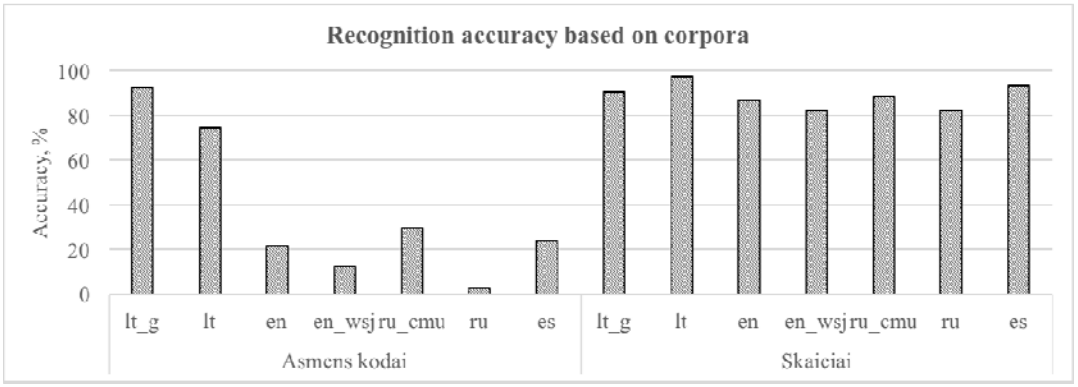


Fig. 2. Recognition results of single recognizers with every corpus.

As we can see corpora "Asmens kodai" is recognized better with native recognizers, adapted recognizers accuracy is lower than 30 %. Best results are obtained by using "lt_g" recognizer which is based on deep learning technology. While "Skaiciai" corpus is recognized good with all recognizers. All recognizers show accuracy higher than 80 %. Best results are obtained by using "lt" recognizer.

## 4. Proposed method experimental evaluation

As mentioned before five classification methods will be evaluated in different speech recognizers combination task: Naïve Bayes, Random forest, Nearest neighbors, CART and Support vector classifiers. These classifiers have a lot hyperparameter that may be tuned for better classification performance. Hyperparameter tuning is hard and time consuming task, so in this case we will tune only few main hyperparameters of each classifier: Naïve Bayes – has no tunable hyperparameters so no tuning for this classifier was performed; Random forest – number of trees in forest with values [20, 50, 80, 120, 160, 210] and max tree depth with values [10, 20, 50, 80, 120, 160, 200]; Nearest neighbors – number of neighbors with values [1, 3, 5, 11, 15, 21, 25, 31, 35, 41, 45, 50]; CART – maximum depth of the tree with values [2, 4, 8, 16, 32] and minimum number of samples required to split with values [2, 4, 8, 16, 32]; Support vector – penalty parameter with values [0.1, 1, 10, 20, 30, 50] and kernel coefficient with values [0.001, 0.01, 0.1, 1], "rbf" was used as kernel function. Classifiers evaluation and hyperparameter tuning will be performed as explained in Fig. 3.

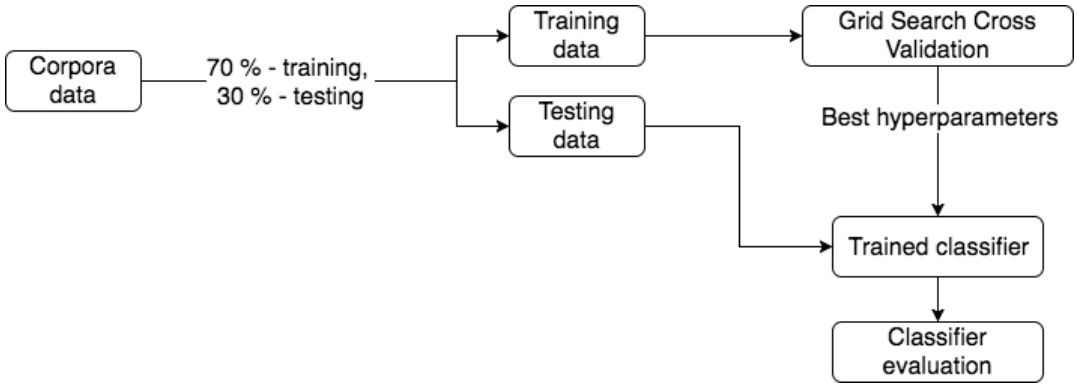


Fig. 3. Classifiers evaluation and hyperparameter tuning process.

First of all, data was split into two sets: 70 % for training and hyperparameter tuning and 30 % for testing. For hyperparameter tuning and classifiers evaluation Grid search cross validation method was used. Cross validation bins number was based on corpora: "Asmens kodai" – 3 bins, "Skaiciai" – 10 bins. Best classifiers with best hyperparameters were evaluated using testing data. After evaluating all classifiers with all

hyperparameters using created corpora we achieved results displayed in Table 4. Only highest accuracy achieved classifiers are displayed.

Table 4. Highest accuracy achieved based on classifiers.

| Classifiers and hyperparameters | Corpora | Feature vector | Accuracy (with testing data) |
|---|---|---|---|
| **Naïve Bayes** | "Skaiciai" | skaiciai_10 | 89.7 |
| **Naïve Bayes** | | skaiciai_20 | 89.6 |
| **Naïve Bayes** | "Asmens kodai" | asmens_kodai_10 | 93.9 |
| **Naïve Bayes** | | asmens_kodai _20 | 92.4 |
| **Random forest**: number of trees in forest - 80, max tree depth – 10 | "Skaiciai" | skaiciai_10 | 98.46 |
| **Random forest**: number of trees in forest - 160, max tree depth – 20 | | skaiciai_20 | 98.73 |
| **Random forest**: number of trees in forest - 50, max tree depth – 10 | "Asmens kodai" | asmens_kodai_10 | 93.93 |
| **Random forest**: number of trees in forest - 80, max tree depth – 20 | | asmens_kodai _20 | 93.93 |
| **Nearest neighbors**: number of neighbors – 11 | "Skaiciai" | skaiciai_10 | 92.78 |
| **Nearest neighbors**: number of neighbors – 5 | | skaiciai_20 | 93.05 |
| **Nearest neighbors**: number of neighbors – 5 | "Asmens kodai" | asmens_kodai_10 | 90.91 |
| **Nearest neighbors**: number of neighbors – 3 | | asmens_kodai _20 | 89.4 |
| **CART**: maximum depth of the tree - 2, minimum number of samples required to split – 2 | "Skaiciai" | skaiciai_10 | 98.33 |
| **CART**: maximum depth of the tree - 2, minimum number of samples required to split – 2 | | skaiciai_20 | 98.33 |
| **CART**: maximum depth of the tree - 4, minimum number of samples required to split – 2 | "Asmens kodai" | asmens_kodai_10 | 93.4 |
| **CART**: maximum depth of the tree - 4, minimum number of samples required to split - 2 | | asmens_kodai _20 | 93.4 |
| **Support vector**: penalty parameter – 0.1, kernel coefficient – 0.01 | "Skaiciai" | skaiciai_10 | 92.31 |
| **Support vector**: penalty parameter – 0.1, kernel coefficient – 0.01 | | skaiciai_20 | 92.31 |
| **Support vector**: penalty parameter – 0.1, kernel coefficient – 0.01 | "Asmens kodai" | asmens_kodai_10 | 92.42 |
| **Support vector**: penalty parameter – 0.1, kernel coefficient – 0.01 | | asmens_kodai _20 | 92.42 |

As we can see highest accuracy was achieved be using Random forest classifier. We noticed that tree architecture based classifiers performs better in such data classification. After the experiments, we compared achieved results with single best recognizers accuracy and if voting method would be used for combination task. Results are displayed in Fig. 4.

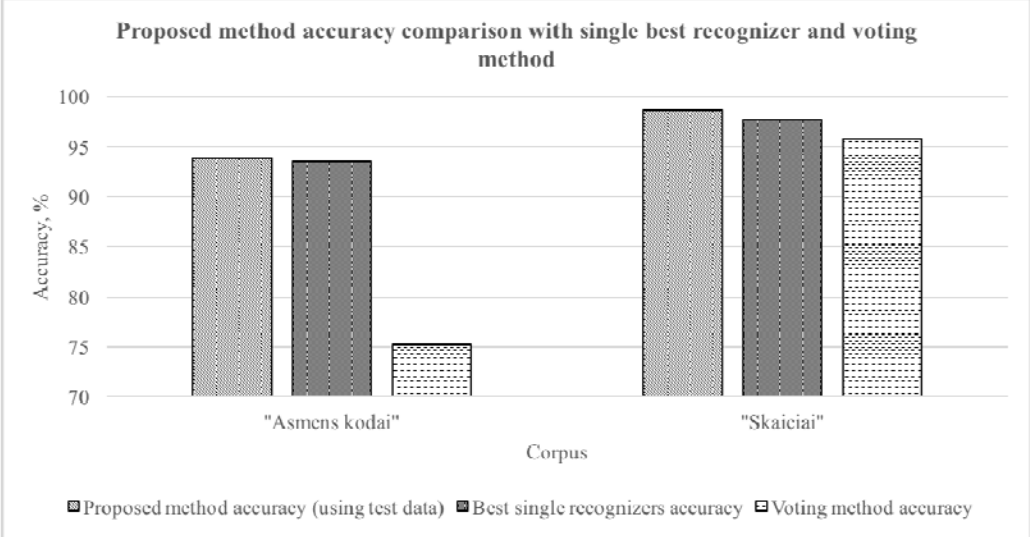


Fig. 4. Proposed method comparison with alternative methods.

After comparing proposed method with other methods, we can see that in every way we achieve accuracy increase. Accuracy increase depends on corpora that is being recognized. Detailed accuracy increase is displayed in Table 5.

Table 5. Achieved accuracy increase compared with other methods.

| **Corpora** | **Accuracy increased compared with best single recognizers accuracy** | **Accuracy increased compared with voting method accuracy** |
|---|---|---|
| "Asmens kodai" | 0.352 | 18.7 |
| "Skaiciai" | 1.057 | 2.9 |

Advantages of suggested method:

Method allows to combine different speech recognizers which are based on different speech features, different training methods and classification methods;

Need less transcribed recordings for training;

Allows to use already trained acoustic models, even foreign adapted acoustic models;

Allows to combine open source and even closed source speech recognizers;

Allows to create reliable speech recognition system with minimal resources.

Main limitation of such method is that at least one speech recognizer must produce correct recognition result.

## 5. Conclusions

In this paper, we proposed state of the art method for isolated speech commands recognition using hybrid approach. Proposed method allows to increase speech recognition accuracy compared with other methods. In our experiments, we achieved recognition accuracy increase compared with best single recognizer: "Asmens kodai" – 0.352 %, "Skaiciai" – 1.057 % and compared with voting method: "Asmens kodai" – 18.7 %, "Skaiciai" – 2.9 %. What is more we evaluated different classification methods for speech recognizers combination task and results shows that Random forest classifier produces highest classification accuracy: "Asmens_kodai" – 93.93 %, "Skaiciai" – 98.73 %. Second best results were obtained by using CART classifier. Tree architecture based classifiers shows best results in different recognizers combination task. What is more obtained results proves that it is possible to adapt foreign language speech recognizers for Lithuanian voice commands recognition by using transcription rewriting rules. Recognition accuracy of adapted foreign recognizers depends on corpora.

## References


[1] Bozkurt, S.; Elibol, G.; Gunal, S.; Yayan, U. (2015). A comparative study on machine learning algorithms for indoor positioning, 2015 International Symposium on Innovations in Intelligent SysTems and Applications.

[2] Delgado, M. F.; Cernadas, E.; Barro, S.; Amorim, D. (2014). Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?, Journal of Machine Learning Research, 15, pp. 3133–3181.

[3] Huggins-Daines, D.; Kumar, M.; Chan, A.; Block, A. W.; Ravishankar, M.; Rudnicky, A. I. (2006). Pocketsphinx: a free, real-time continuous speech recognition system for hand-held devices, IEEE ICASSP 2006 Proceedings, 1, pp. 185–188.

[4] Jain, V.; Dubey, A.; Gupta, A.; Sharma, S. (2016). Comparative analysis of machine learning algorithms in OCR, 3rd International Conference on Computing for Sustainable Global Development, pp. 1089–1092.

[5] Lojka, M.; Juhar, J. (2014). Hypothesis combination for Slovak dictation speech recognition, ELMAR (ELMAR), 2014 56th International Symposium, IEEE, pp. 43–46.

[6] Meneido, H.; Neto, J. (2000). Combination of acoustic models in continuous speech recognition hybrid systems, Proceedings of the International Conference in Spoken Language Processing, 9, pp. 1000–1029.

[7] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. (2011). Scikitlearn: Machine Learning in Python, The Journal of Machine Learning Research, 12, pp. 2825–2830.

[8] Rasymas, T.; Rudžionis, V. (2015). Evaluation of Methods to Combine Different Speech Recognizers, Proceedings of the Federated Conference on Computer Science and Information Systems, DOI: 10.15439/2015F62, 5, pp. 1043–1047.

[9] Rasymas, T.; Rudžionis, V. (2015). Lithuanian Digits Recognition by using Hybrid Approach by Combining Lithuanian Google Recognizer and some Foreign Language Recognizers, The 21st International Conference on Information and Software Technologies (ICIST 2015), DOI: 10.1007/978-3-319-24770-0, pp. 449 – 459.

[10] Rudžionis, V.; Ratkevičius, K.; Rudžionis, A.; Raškinis, G.; Maskeliūnas, R. (2013). Recognition of Voice Commands Using Hybrid Approach, ICIST2013, pp. 249–260.

[11] Schultz, T.; Waibel, A. (2001). Language-independent and language-adaptive acoustic modeling for speech recognition, Speech Communication, 35(1), pp. 31–52.

[12] Sharma, S.; Agrawal, J.; Agarwal, S.; Sharma, S. (2013). Machine learning techniques for data mining: A survey, IEEE International Conference on Computational Intelligence and Computing Research.

[13] Siohan, O.; Rybach, D. (2015). Multitask learning and system combination for automatic speech recognition, Automatic Speech Recognition and Understanding (ASRU), pp. 589–595.

[14] Tchendjou, G. T.; Alhakim, R.; Simeu, E.; Lebowsky, F. (2016). Evaluation of machine learning algorithms for image quality assessment, 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design, pp. 193–194.

[15] Wang, Z.; Schultz, T.; Waibel, A. (2003). Comparison of Acoustic Model Adaptation Techniques on Non-Native Speech, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 540–543.

[16] Wu, X.; Kumar, V.; Quinlan, J. R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G. J.; Ng, A.; Liu, B.; Yu, P. S.; Zhou, Z. H.; Steinbach, M.; Hand, D. J.; Steinberg, D. (2008). Top 10 algorithms in data mining, Knowledge and Information Systems, 14(1), pp. 1–37.

[17] Zolnay, A.; Schluter, R.; Ney, H. (2005). Acoustic feature combination for robust speech recognition, Acoustics, Speech, and Signal Processing, pp. 457–460.