ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse237942020.pdf
https://doi.org/10.30534/ijatcse/2020/237942020

Neural Networks and Algorithms for Modern IT Systems

Olga V. Kitova[1], Vladimir A. Kitov[2]
[1] Plekhanov Russian University of Economics, Moscow, Russia
[2] Plekhanov Russian University of Economics, Moscow, Russia

ABSTRACT

Systems with speech recognition voice messages commander used in the field of automation and systems IoT (Internet of things). Systems IoT offer large quantity options interactions, one of which speech recognition voice messages commands. Development in the region speech recognition and speech synthesis speech contributes to when voice messages appear assistants, which can conduct a dialog with the user and perform various voice commands. This option communications increases productivity at work, thanks to the convenience in interaction with intelligent features systems.

In as a tool implementations speech recognition systems voice messages commander neural network. Neural networks they can perform various tasks, including number and speech recognition voices, and have some advantages in comparison with the traditional ones methods. For using neural networks can be implemented adaptive features systems with a constant opportunity training. Thanks to it, neural networks ad networks acquire high popularity because they can learn decide any task.

The largest interest in neural networks leaves the sphere medicine, industry, automotive industry, avionics. Such the systems have huge potential to training and ability decide various Bok tasks side by side with a person. Based on neural networks ad networks are being created intelligent features systems, among which have voice commands assistants.

Intelligent features systems are technical or software version a system that able to solve issues that need to be resolved traditionally counted creative. Intelligent features the systems include includes a database knowledge, intellectual interface and acceptance mechanism solutions. These systems are being studied within the framework of artificial intelligence. There is a certain number of views intellectual systems, including which ones should be selected a hybrid system.

Key words : Phonetics, neural network, algorithm, IoT.

1. INTRODUCTION

For today difficult day reply at what level located recognition quality language, because accuracy referral systems depends on various side factors. Very difficult achieve this goal same accuracy speech recognition, which it demonstrates a person.

Accuracy any system it may depend from various sources parameters:

- size dictionary;
- dependent system speaker or not;
- varieties broadcasting: isolated, with an interruption and continuous.

Usually, very easy distinguish a word among them small number of words, but the frequency the error rate increases, if it increases dictionary inventory [5].

The system, which depends from the speaker, usually easier to implement, however, it can manage only one person. Very difficult get an independent from the speaker the system, so what does she learn understand the specific a human being.

Each one it has its own specifics C spoken words, intonation, speed, clarity, etc. Usually error rate independent systems in 3-5 times higher education. Also available implementation options multi-acoustic systems (when uses the system a small amount people) and adaptive (systems that set up for anyone the speaker) [7].

Distinguish three types of speech:

- isolated speech – available types of single files words and sounds;
- broadcasting with an interruption – these are complete sentences, in which the words artificially selected silence;
- continuous speech – it's fine spoken words offers, without VAT artificial pauses.

Speech recognition isolated speech and broadcasting from with an abort relatively easier in implementation, because the word and its boundaries easier to highlight. Continuous broadcasting is more complicated in implementation, since the words they don't have clear lines borders, sometimes incomprehensible, they can also be present other noises.

Even with a fixed price vocabulary, productivity it will change depending from the sequence words in a sentence and character spoken words. Some restrictions they may be semantic links, grammatical features etc. [6].

Grammar rules, usually, built basically expectations specific words, which can go for the preliminary one. Complexity tasks, delivered before the system better to measure not deep dictionary, and a metric that answers for the surprise systems.

Systems can be divided on the ones that perceive a clear sequence words in a sentence (template) or on systems spontaneous speech. Systems with a clear the sequence they are clearly marked expressed the sequence commands. If this sequence will be violated, then the system will work with errors or omissions it will stop altogether function.

Such systems are lighter in implementation and in some cases situations should advantage, because what works only on clear ones the script.

However more flexible systems, is spontaneous learning systems speech. Such systems such as usually, they have lots of problems in implementation. This and the sounds that they violate our privacy policy. clear speech, incomprehensible, fuzzy words, very quiet or fast speech, disadvantages of a thing, illness, emotional state. Also great what matters is that what is a dictionary there is a reserve in fact unlimited. A person can build one and the same sentence different words.

2. MATERIALS AND METHODS

Neural networks are often learned to calculate probability and are well suited to solving the mapping problem. They have an advantage over hidden Markov models, because, unlike them, they can accept input data with a constant value, and therefore do not contain quantization errors. In contrast to fuel and energy systems with constant density, they do not make questionable assumptions about the parametric form of the density function [1-3].

There are many ways to design and train neural networks for such tasks.

One of the simplest is training a neural network based on frames. This approach is called frame learning. An alternative method is segmental learning, in which the neural network receives an entire segment of language as input, rather than a single frame of speech. This allows the network to better use the ratio that is contained between all segment frames, and also makes it easier to execute segmental information, such as duration. The disadvantage of this approach is that the speech is first followed by the segment of speech, before the neural network can evaluate the segments. A time-delayed neural network (TDNN) is a variant of segment-level learning, but it can only be used for phoneme recognition, not for word recognition [4].

The segmental neural network is trained to classify phonemes from speech segments with different durations. During training, you should correctly classify each segment of each utterance. During testing, the segmental neural network receives segmentation of the n-best hypotheses and finds the compiled score for each offer. Such a system has 9% word errors in the database [8-10].

The next level of learning is word-level learning, in which the neural network receives an entire word as input and is trained to optimize the classification of the word. Learning at the word level is the best option, because it still meets the learning criteria closer to the final test for checking the accuracy of sentence recognition. Unfortunately, the extension is not trivial, because unlike a simple phoneme, a word cannot be adequately modeled by a single state, but requires a sequence of States. Also, the activation of these States cannot simply be summed up over time in time-delayed neural networks, but must first be segmented using a dynamic time-warping procedure that determines which States are applied in which frames.

Thus, word-level learning requires the presence of a temporary dynamic algorithm in the neural network.

Thus, the language and context are fed into independent parts, and each text effectively contributes to a different offset of source units.

Figure 1: Example of a standard and efficient implementation

Figure 1 shows us example of a standard implementation in neural network.

The latter approach is based on factorization. When a neural network is trained as a phoneme classifier, it evaluates *P(q,x)*:

$$P(q,c \mid x) = P(q \mid x) * P(c \mid q,x) \quad (1)$$

Where: q is the phoneme class, x-introduction of the language, c-phonetic context.

That is, context dependent probability is equal to the product of two items:

$$P(x \mid q,c) = \frac{P(q,c \mid x) * P(x)}{P(q,c)}, \quad (2)$$

where *P(x)* can be ignored during speech recognition, since it is a constant in each frame and the previous *P(q,c)* can be evaluated directly from the training set.

This factorization approach can easily be extended to the tryphoneme of modeling. For try phonemes, we estimate $P(q,c_l,c_r,x)$.

$$P(q,c,c_l,c_r,x) = P(q \mid x) * P(c_l \mid q,x) * P(c_r \mid c_l,q,x), \quad (3)$$

where $c_l$is the left phonetic context, c_r – the right phonetic series. Similarly

$$P(q,c_l,c_r) = P(q) * P(c_l \mid q) * P(c_r \mid c_l,q). \quad (4)$$

These six summands can be evaluated by neural networks whose inputs and outputs correspond to each input and output *P(o|i)*therefore, the part of the equation can be converted to probability by Bayes ' rule:

$$P(x \mid q,c_l,c_r) = \frac{P(q,c_l,c_r \mid x) * P(x)}{P(q,c_l,c_r)} = \frac{P(q \mid x) * P(c_r \mid c_l,q,x)}{P(q) * P(c_l \mid q) * P(c_r \mid c_l,q)} * P(x), \quad (5)$$

where *P(x)* can again be ignored during recognition, and the other six terms can be taken from the six neural network outputs. This probability can be used to align the Viterbi [10].

Neural networks can be taught to compute smooth, nonlinear, nonparametric functions from any input space to any output space. The General types of functions are forecasting and classification, as shown in figure 2 [9]. In a prediction

network, inputs are multiple frames of a language, and outputs are predictions of the next frame of a language; using multiple prediction networks, one for each phoneme, their prediction errors can be compared, and the smallest prediction error is considered the best match for this segment of the language (figure 3) [9]. On the contrary, in a classification network, input data consists of several language frames, but the results directly classify the speech segment into one of the specified classes.

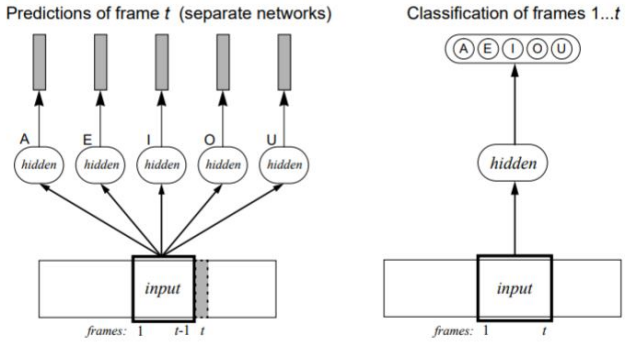

Figure 2: Networks prediction and classification

To work with acoustic models, we use thepredictive neural network (pnnn). It is designed for recognizing a large dictionary of both isolated words and continuous speech. It is based on models of a common phoneme, that is, models that were linked in different contexts [12].

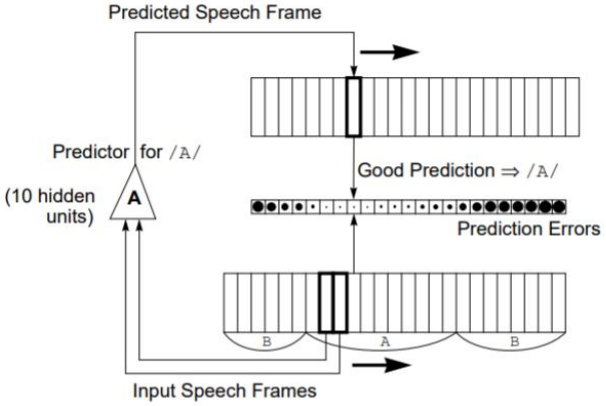

Figure 3:. Principle of operation of the network prediction

LPNN performs phoneme recognition using prediction. The network shown as a triangle accepts *K* adjacent frames of the language (usually K = 2), passes them through a hidden layer, and tries to anticipate the next frame of the language. The projected frame is then compared to the actual frame. If the error is small, the network is considered a good model for this segment of the language. If you could teach the network to make accurate predictions only during segments that respond to /A / phonemes (for example) and poor forecasting in other places, then it wouldhave an effective phoneme Recognizer /A/ due to its contrast with other phoneme models [11].
LPNN satisfies this condition by using its learning algorithm, so that we get a collection of phonemes with one model per phoneme. LPNN is a hybrid of NN-HMM, which means that acoustic modeling is performed using predictive networks, and temporal modeling is performed using HMM.
LPNN is a state-based system, such that each predictive network corresponds to a state in (autoregressive) HMM. As in HMM, phonemes can be modeled in finer detail using sub-phonetic state models. Typically, three States (prediction networks) per phoneme are used, as shown in the following diagrams. Just like in HMM, States (prediction networks) are hierarchically sequenced into words and sentences, observing the limitations of the dictionary and grammar.
There are many ways to develop a classification network for speech recognition. Designs depend on five main dimensions: network architecture, input representation, broadcast models, and training and testing procedures. There are many issues to consider in each of these dimensions. A single-layer and multi-layer perceptron (Figure 4) [10].

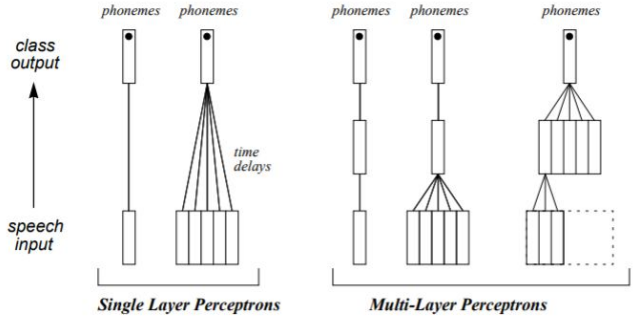

Figure 4:. Single-Layer and multi-layer perceptron

If a multilayer perceptron is asymptotically prepared classifier 1 with N, using the mean square error (SKO) or any similar criterion, then its activation is close to the probability of class P (class / input), the accuracy of which improves with the size of the training set (Figure 5) [15].

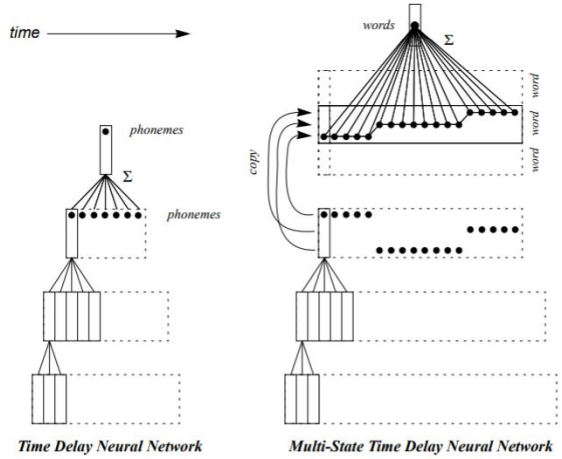

Figure 5:. Networks with time delays

When building a neural network, you should first consider whether it will contain a hidden layer. A network that does not have a hidden layer (a single-layer perceptron) can only form regions of linear resolution, but the classification accuracy can reach 100% if the training set is linearly slow. In contrast, a system that contains a hidden layer (a multi-layer perceptron)

can form nonlinear regions of the solution, but there is always a small chance that the system can get stuck in a local minimum [13].

A multi-layer perceptron is considered better than a single-layer one for speech recognition, since speech is a nonlinear domain, and the problem of local minima in multi-layer networks is insignificant. Multilayer systems have an advantage in accuracy of up to 30%, so the hidden layer is really useful (Figure 6) [15].

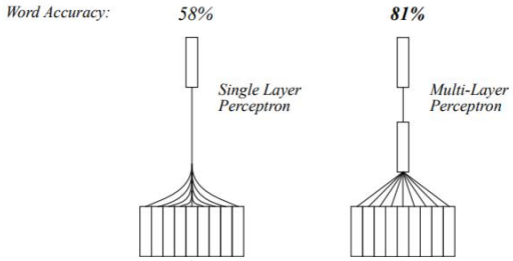

Figure 6:. Accuracy of single-layer and multi-layer perceptron systems

A system with a single hidden layer can solve problems just like those with multiple hidden layers. Also, as the number of hidden layers increases, the time to learn increases [16].

The next question is exactly how to reduce the learning speed to get the optimal value. To do this, take the base learning speed, which is high, and reduce it geometrically by using a coefficient less than 1, after each iteration of training. On the graph, you can see the results obtained as a result of adjusting the learning speed using a coefficient that varies from 0.5 to 1.the Coefficient 1, on the contrary, leads to too high a level of learning, so the system becomes unstable. The best result is the average value, which gives enough time for the system to get out of local minima even before the efficiency of the learning speed decreases to 0 [10].

The coefficient of 0.5 gives a better learning accuracy, but this advantage is soon lost, as the learning speed decreases so rapidly that the system cannot get out of local minima (figure 7) [14].

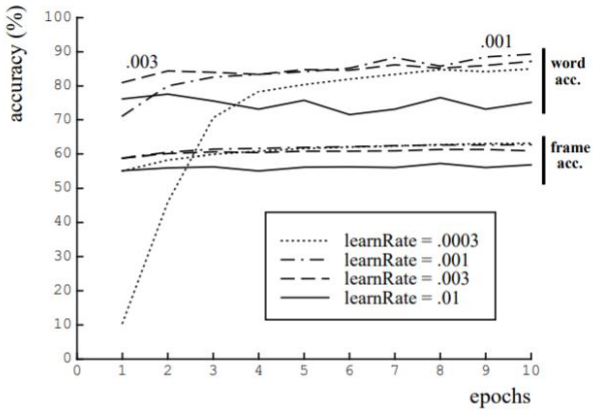

Figure 7:. Graph showing the system's accuracy lode as a function of the learning speed

Although geometric coefficients are useful for detecting learning speed, this approach may still not be optimal, because it is impossible to know for sure that the system has reached peak accuracy on this high-level data. Therefore, you should consider changing the coefficient beyond the dynamic range. That is, starting from the initial learning curve in the first iteration, the results of evaluating the accuracy of the system are taken. Then, in the next iteration, the system learns from only half of the data, and the system's accuracy is evaluated again. Comparing these two results, we can conclude whether the learning rate for the first stage of learning is greater or less than this value. Then we double or decrease the nearest training level and try again. Then we continue to increase or decrease the level of training, so that the accuracy does not deteriorate to the threshold level.

Then we perform a quadratic interpolation on the best data point and its left and right neighbor to find the learning indicators, between the obtained points where, x = learning rate, and y = accuracy (figure 8) [11].

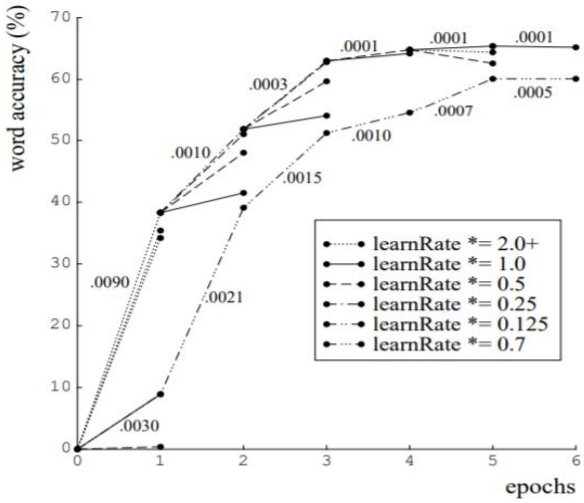

Figure 8: The search for the optimal learning rate

The optimal learning speed is influenced at least by the following factors:

the value of the training set. A larger training Suite provides lower learning rates for each iteration (figure 9) [18]. This is because the optimal learning rate curve decreases slightly after each change in the weight ratio;

normalization of inputs. A higher standard deviation of input data provides for higher learning rates to compensate for the fact that latent neurons are saturating;

activation function;

the number of neurons on the input and hidden layer.

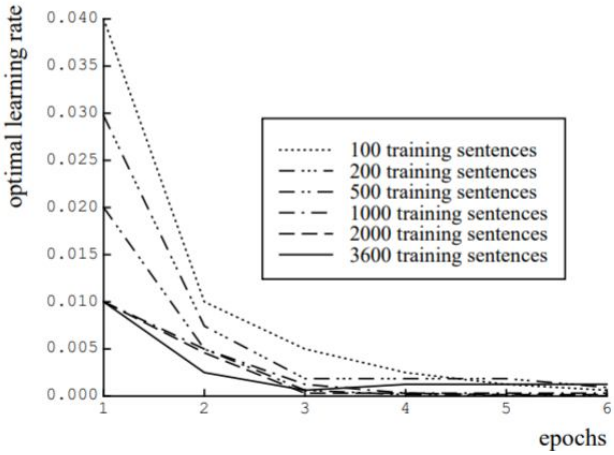

Figure 9:. Dependence of the system's accuracy on the size of the training set

The learning speed is least affected by factors such as the input representation, delay hierarchy, number of speakers, and labels that are used during training. However, it remains unclear whether the frequency of updating weight coefficients or using different error criteria has a big impact.

The RNNLM framework RNNLM(recurrent neural network language models) was used to train the network.

RNM (recurrent neural networks) is a class of artificial neural networks, the connections between the nodes of which form a graph oriented in time. This creates an internal state of the network that allows it to exhibit dynamic behavior over time. Unlike direct propagation neural networks, RNNS can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as recognizing non-segmented continuous handwritten text and speech recognition (figure 10).

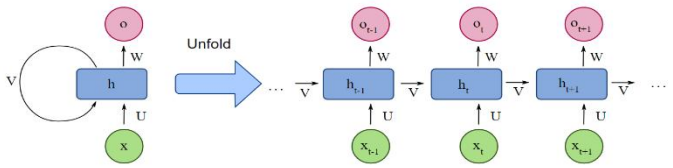

Figure 10:. Architecture of a recurrent neural network model

Recurrent models of neural networks whose input layer uses a 1-of-N representation of the previous word, which is combined with the previous state of the hidden layer *s(t)*, using the sigmoid activation function. The output layer *y(t)* it has the same dimension *as w(t)*. Training takes place using the stochastic gradient algorithm.

3. CONCLUSION

The standard method is called reverse propagation over time (RPC) and is a generalization of reverse propagation for direct propagation networks. Computationally more expensive interactive option is called real-time recurrent learning (RRR), and is a sample of automatic differentiation in sequential accumulation mode with added tangent vectors. In contrast to the HRM, this algorithm is local in time, but not local in space.

In this context, local in space means that the node's weight coefficient vector can only be refined using information stored in the connected nodes and the node itself, so that the complexity of refining a single node is linear with respect to the dimension of the weight coefficient vector. Local in time means that callouts occur continuously and depend only on the most recent clock cycle, and not on several clock cycles within a given time interval, as in the PO. Biological neural networks appear to be local, both in time and space.

Obtaining weight coefficients in a neural network can be modeled as a nonlinear global optimization problem. An objective function for estimating the fitness or error of a certain weight vector can be formed in this way: first, the weights in the network are set according to this weight vector. Next, the network is evaluated by the training sequence. As a rule, to represent the error of the current weight vector, the sum of squares of the differences between forecasts and target values specified in the training sequence is used. Then arbitrary methods of global optimization can be applied to minimize this target function.

This research was performed in the framework of the state task in the field of scientific activity of the Ministry of Science and Higher Education of the Russian Federation, project "Development of the methodology and a software platform for the construction of digital twins, intellectual analysis and forecast of complex economic systems", grant no. FSSW-2020-0008.

REFERENCES

1. Alperovich, I. A. **Development of production of volume coloring** / I. A. Alperovich, G. I. Votyeva, V. K. Kryukov / / Building materials. - 1992. - No. 3-4. - Pp. 2-4.
2. Filatova, E. V. **Neural network recurrent model with input layers**: dis.. tech. Sciences: 05.23.05 / Filatova Ekaterina Vladimirovna. - Novocherkassk, 2004. - 149c.
3. Han, L. X. **Voice recognition of neural network** / L. X. Han, J. Han, F. M. Sun, Y. J. Huo / / Advanced Materials Research. - 2011. - Vols. 160-162. - P. 880-885.
4. Kleerekoper, A. van den Dobbelsteen, E. van den Ham, T. Hordijk, C. Martin / / Urban Climate-2015. - Vol. 14-Pp. 290-300.
5. Kleerekoper, L. **Creating drafts in voices**: Exploring a new climate adaptation measure based on thermal stratification / L.
6. Mojsov, G. N. **Development of multi-purpose additives**: dis... tech. Sciences: 05.23.05 / moysov Georgy Leonidovich. - Stavropol, 2003. - 164 p.
7. Razgovorov, P. B. **Scientific bases of creation of composite materials from technical and natural silicates**: dis. ... d-RA tekhn. Sciences: 05.17.01. / Pavel Borisovich Razgovorov-Ivanovo, 2008, - 357 p.
8. Salakhov, A. M. **Experience of surface treatment of ceramic materials for construction purposes** / Building materials. - 2017. - No. 4. Pp. 42-46.
9. Zubekhin, A. P. A**ngoby on the basis of red-hot fusible clays** / A. p. Zubekhin, N. D. Yatsenko, V. P. ratkova, E.

O. ratkova, K. A. Verevkin / / Building materials. - 2009. - No. 3. Pp. 40-41.

10. Lubysheva, L. I., & Abramov, R. A. (2014). **Innovative model of olympic education of primary schoolchildren based on information and communication technologies**. Teoriya i Praktika Fizicheskoy Kultury, (7), 87–89.
11. Rodnyansky, D. V, Abramov, R. A., Repin, M. L., & Nekrasova, E. A. (2019). **Estimation of innovative clusters efficiency based on information management and basic models of data envelopment analysis**. International Journal of Supply Chain Management, 8(5), 929–936.
12. Rodnyansky, D., Abramov, R., Valeeva, G., Makarov, I., & Levchegov, O. (2019). **Methods to evaluate public administration efficiency**: The case of the volga region. International Journal of Engineering and Advanced Technology, 8(5), 2261–2271.
13. Klyuev S.V., Bratanovskiy S.N., Trukhanov S.V., Manukyan H.A. **Strengthening of concrete structures with composite based on carbon fiber** // Journal of Computational and Theoretical Nanoscience. 2019. V.16. №7. P. 2810 – 2814.
14. **Scenario analysis**. International Journal of Advanced Trends in Computer Science and Engineering, 8(1.4 S1), 1–8. https://doi.org/10.30534/ijatcse/2019/0181.42019
15. Doke, A. R., Garla, N., & Radha, D. (2019). Analysis of human gene-disease association as a social network. International Journal of Advanced Trends in Computer Science and Engineering, 8(4), 1062–1068. https://doi.org/10.30534/ijatcse/2019/12842019
16. Doshi, N. (2019). **Analysis of efficient and privacy-preserving metering protocols for smart grid systems**. International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 2882–2886. https://doi.org/10.30534/ijatcse/2019/32862019