Inner parameters' optimization in the artificial neural network for the traffic data classification in radiofrequency applications

Classification of nonstationary data using the machine learning algorithm "random forest"

Kalashnikov Evgeniy Alexandrovich
Department of Automated Control Systems
National University of Science and Technology "MISiS"
119991, Russian Federation, Moscow, Leninsky Prospect, 4
e-mail ek.misis@gmail.com
kalashnikov.e.2018@mail.ru

Kondybayeva Almagul Baurzhanovna
Department of Automated Control Systems
National Research Technological University "MISiS"
119991, Russian Federation, Moscow, Leninsky Prospekt, 4
e-mail almakonde18@gmail.com

Ositis Anastasia Petrovna
International Telecommunication Academy
119034, Russia, Moscow, ul. Prechistenka, 4, building 2
info@ita.org.ru

***Abstract*—The number of devices connected to the World Wide Web and the requirements of subscribers to the speed of mobile Internet access are increasing every year. Developers of telecom equipment and telecom operators, trying to meet new challenges, are preparing to seriously change the architecture of networks and interaction regulations in radiofrequency applications.**

Availability of required radiofrequencies in radiofrequency applications is one of the main necessary factors for the development of such networks, along with the readiness of the network architecture and infrastructure, business models and subscriber devices [1].

In the development of fifth-generation technologies in radiofrequency applications, there is a certain set of "input" characteristics that serve as a guide for a new standard. For example, compared to the best existing LTE networks, the data transfer speed in 5G networks should be 10-100 times higher, the response time is 5 times less, the network should support the number of devices 100 times more[2-4].

The modern telecommunication market is at a stage when operators have a favorable opportunity to bypass all the convergence difficulties inherent in the networks of the past, and go directly to the next-generation networks based on technology, which received the working name NGN - "New Generation Network". In order to make this breakthrough and join the number of high-tech operators, new solutions are needed to be constructed in the field of creating and providing high-performance services. NGN - technology of building a network - is designed to provide data transmission services and voice services in radiofrequency applications. It removes a number of restrictions and barriers that exist now, and this is its economic productivity[4].

DPI (Deep Packet Inspection) - technology for the classification and filtering of traffic by its content. DPI is able to define application protocols (HTTP, HTTPS, Skype, BitTorrent), the type of data transferred (web pages, audio and video files), it can extract information specific to specific protocols (URL for HTTP, domain name for HTTPS, distribution identifier BitTorrent). At the moment, most Russian providers use DPI as a means of blocking websites from a single register of prohibited information. The use of DPI for this purpose allows providers to block specific links made to the registry, rather than an IP address or a whole domain[1-4].

Having enough data about the traffic data related to the user, it is possible to successfully determine the thematic categories reflecting the interests and preferences of this user or detect risky activities.

In the process of choosing the optimal model for traffic data analisys the so-called “random forest” algorithm was chosen due to the accuracy quality and equation stability: a large number of parameters affecting the implementation of the task are taken into account for solving the classification problem[2].

Keywords—artificial intelligence, machine learning, classification, stationary, signal, traffic clusterization, traffic analysis, recommendations of new services to the telecommunications market, detection of terrorists, GSM, IP, NGN, radiofrequency applications.

978-1-5386-5622-8/18/$31.00 ©2018 IEEE

I. Introduction

In statistics, there are two main views on how the solution of the classification problem should look. The first point of view assumes that the object belongs to one and only one of the classes. For example, when it comes to classifying traffic data, classes can be "social network" and "not social network". There may be some uncertainty about the classification issue (the traffic data may be a bit like social network), but the result is only the final decision - social network or not social network. Uncertainty, if it was, does not leave the limits of the classification algorithm.

The second approach consists in obtaining a vector of conditional probabilities - a vector whose components are equal to the probabilities of the subset belonging to all possible classes. The algorithm does not decide on the classification of the traffic data. It just tells what the probability is that a particular traffic data is “social network” - and what is the probability of what is not. The adoption of decisions based on this information is transferred to the operator.

The second approach is more flexible than the first, and more reasonable. How can the classification algorithm know which priorities the operator is guided by? In some cases, it is necessary to minimize an error on one of the classes.

II. Approach to the solution of the problem

Nominal variables can be encoded in several ways: as an integer, using "1-of-N" encoding or "1-of-N-1" encoding.

To improve the performance of computational subroutines, it is recommended that the following data representation scheme be followed [5]:

- Real variables are stored as is.

- Nominal variables that take two values are coded with one number as "0" or as "1" (i.e., using the "1-of-N-1" scheme).

- Nominal variables that take three or more values are encoded using the "1-of-N" scheme (for example, "red", "yellow" and "green" can be encoded as "1 0 0", "0 1 0" , "0 0 1").

- Nominal variable, regardless of the number of values to be received, can be encoded with an integer numbers(0, 1, 2, ...). However, this method of encoding is recommended to be used only if the values of the variable can be meaningfully ordered and only for nonlinear models (neural networks, decision trees). For example, the values "cold", "room temperature", "heat" can be ordered by increasing temperature and encoded as "0", "1", "2". At the same time, the values "sour", "bitter", "sweet", "salty" are difficult to somehow intelligently streamline [5].

At the time of writing, none of the algorithms, except for the Bayesian classifier, can work with missed values[5-10]. However, one can work around this limitation by adding one more value to the possible values of the variable, denoting the pass. For example, if the variable takes the values "0", "1", "2", then one can encode the missing values like "1 0 0 0", "0 1 0 0", "0 0 1 0", and the missing value - as "0 0 0 1". The analogue for the real variable will be the encoding method: the unallocated value of x is encoded as "x 0", and the omitted value is "0 1".

Another option is to replace the missing value with the mean (or most probable) value for a given variable.

Work with forest solutions consists of the following steps (work with decision trees is carried out in the following sequence):

1. Select the values of the adjustable parameters of the algorithm (see below).

2. Construction of the forest with the help of the corresponding subprogram.

3. Work with the resulting model (data processing, model copying / serialization, etc.).

The algorithm contains three main parameters that need to be tuned: the coefficient r, telling us which part of the training set we will use to construct individual trees; number of trees NTrees; as well as the number of NFeatures - the number of variables used to build individual trees. Below these parameters are considered in more detail.

The coefficient r, which is in the range from 0 to 1, affects the tolerance of the algorithm to noise in the learning set. The weak point of the original Braimean algorithm is that it does not fully compensate the propensity of individual trees for retraining. Recall that an unregularized decision tree accurately remembers the learning set. The procedure used in the original algorithm for selecting N elements with repetitions leads to the fact that approximately 0.632 elements of the total training set fall into the training sample (which corresponds to the value r = 0.632) - which means that approximately 63% of individual trees will be accurately remembered by any arbitrarily chosen element of the learning set and will receive a majority during the voting procedure. Thus, in the presence of noise in the learning set, the Braimean algorithm with a high degree of accuracy will repeat all the errors of the training set, failing to restore the noise-suppressed pattern. This problem has different solutions: it is possible to regularize individual trees [5]. Recommended values r - from 0.66 (low noise level) to 0.05 (very high noise level). The choice is made based on the relationship between the error on the training set and the generalization error (calculated using a test set or out-of-bag evaluation) - if the ratio is significantly less than one, reduce r and rebuild the model.

Other parameters of the algorithm are much easier to configure. The number of NTrees trees is recommended to be made at the level of 50-100, and the number of NFeatures is automatically selected and set at half of the total number of variables [5-10].

In this solving problem it is necessary to collect data for training. The training data set is a set of observations for which the values of the input and output variables are indicated. The first question to be solved is which variables to use and how many (and what) observations to collect.

Neural networks can work with numeric data lying in a certain limited range. This creates problems in cases where the data has a non-standard scale when there are missing values in it, and when the data is non-numeric. Numerical data is scaled to a suitable range for the network, and the missing values can be replaced by the average value (or other statistics) of this variable for all available training examples [10-12].

A more difficult task is to work with non-numerical data. Most often non-numerical data can be represented in the form of nominal variables of the type traffic = {personal, service, ...}. Variables with nominal values can be represented in numerical form. However, neural networks do not give good results when working with nominal variables that can take many different values. Unfortunately, in this case it will be very difficult to train a neural network, and instead it is better to assign a certain rating to each district (based on expert estimates).

Non-numeric data of other types can either be converted to numerical form, or declared insignificant. The values of dates and times, if they are needed, can be converted to numeric, subtracting from them the start date (time).

The question of how many observations are needed to be done for learning a network is often not easy. A number of heuristic rules are known, linking the number of necessary observations to the size of the network (the simplest of them says that the number of observations should be ten times greater than the number of connections in the network). In fact, this number also depends on the (previously unknown) complexity of the mapping that the neural network tends to reproduce. With the increase in the number of variables, the number of required observations grows nonlinearly, so that even with a rather small (eg, fifty) number of variables, a huge number of observations may be required. This difficulty is known as the "curse of dimension" [7].

For most real problems, several hundred or thousands of observations are sufficient. For particularly complex tasks, an even greater number may be required, but very rarely (even a trivial) task may occur, where there would be less than a hundred observations. If the data is less than what is said here, in fact there is not enough information to learn the network, and the best thing one can do is to try to fit some linear model to the data.

In many real problems one has to deal with not quite reliable data. The values of some variables can be distorted by noise or partially absent. So if there is no so much data, one can include cases with missing values (although, of course, it's best to avoid it). In addition, neural networks are generally resistant to noise. However, this stability has a limit. For example, emissions, moreover values that lie very far from the range of normal values of some variable can distort the result of learning. In such cases, it is best to try to detect and remove these emissions (either by removing the relevant observations or by converting the emissions into missed values). If emissions are difficult to identify, then the process of training can be made sustainable by emissions (using the urban-type error function [8]), but such emissions-tolerant training is generally less effective than standard.

The basic recommendation with the approach is to choose variables that are suspected to affect the outcome.

With numerical and nominal variables one can work directly. Variables of other types should be converted to specified types or declared insignificant.

For the analysis it is necessary to have about hundreds or thousands of observations;

The more variables are in the problem , the more observations need to be made.

If necessary, one can work with observations that contain missing values. The presence of emissions in the data can create difficulties. If possible, the process of the removing the emissions should be done. If the data is sufficient, one can remove from consideration the observations with the missing values.

Each neural network accepts numeric values in the input and numeric values at the output.

A. Model Error

If it is a classification problem, then five error measures can be used. The first and most widely known is the classification error (the number or percentage of incorrectly classified cases). The second, no less well known measure of error is the cross-entropy. Using the average cross-entropy (instead of the total cross-entropy) allows us to obtain comparable estimates for different test sets [5-10].

The remaining three measures of error are again the root-mean-square, mean and average relative errors. However, unlike the regression problem, here they characterize the error in calculating the conditional probability vector. The error is understood as the probability vector calculated by the classification algorithm differs from the vector obtained on the basis of the test set (the components of this vector are 0 or 1, depending on which class the object belongs to). The meaning of mean-square and mean errors is clear - this is a mistake in approximating conditional probabilities, averaged over all probabilities. The average relative error is the average error in the approximation of the probability that the image belongs to the correct class (remember that this error is considered only for non-zero elements of the probability set) [5-10].

In Deep Packet Inspection (DPI) facilities in radiofrequency applications, as a rule, the object of classification is a traffic flow of traffic level - it is a set of IP packets that have the same transport layer protocol as well as an unordered pair of endpoints: <(source ip, source port), destination ip, port destination)>.

Since there is no single standard for DPI, there is a large number of implementations from different DPI solution providers that differ in the type of connection and type of operation[1-4]. There are two common types of DPI connection: active and passive.

Before defining these metrics, we introduce definitions[3]:

• Client - the initiator of the TCP connection or the sender of the first UDP-datagram of the stream, depending on the transport layer protocol;

• The server is the receiving end of the TCP connection or the destination of the first UDP datagram of the stream, depending on the transport layer protocol.

• Data Portion is a collection of application-layer payload that was transferred from one side to the other (from the client to the server or vice versa), and was not interrupted by the payload on the other hand [3].

Active DPI

Active DPI - DPI, connected to the network of the provider in the usual way, like any other network device. The provider configures routing so that DPI receives traffic from users to blocked IP addresses or domains, and DPI already decides whether to skip or block traffic. Active DPI can check both outgoing and incoming traffic, however, if the provider applies DPI only to block sites from the registry, it is most often configured to check only outbound traffic.

Passive DPI

Passive DPI - DPI, connected to the provider network in parallel (not in the section) either through a passive optical splitter, or using the mirroring of user traffic. This connection does not slow down the speed of the network provider in the case of insufficient DPI performance, which is why it is used by large providers. DPI with this type of connection can technically only detect an attempt to request a prohibited content, but not to suppress it. To bypass this restriction and block access to a denied site, DPI sends a specially crafted HTTP packet to the user requesting the blocked URL, redirecting to the stub page of the provider, as if the requested resource itself had been sent by the user (the IP address of the sender and the TCP sequence are forged). Due to the fact that the DPI is physically located closer to the user than the requested site, the forged response reaches the user's device faster than the real response from the site.

Sometimes passive DPI is used to substitute the response of a DNS server, not the site itself[1-4].

"Normal" DPI

By "regular" DPI is meant a DPI that filters a certain type of traffic only on the most common ports for this type. For example, "normal" DPI detects and blocks blocked HTTP traffic only on port 80, HTTPS traffic on port 443. This type of DPI will not track prohibited content if one sends a request with a blocked URL to an unlocked IP or non-standard port.

"Full" DPI

Unlike "regular" DPI, this type of DPI classifies traffic regardless of the IP address and port. Thus, blocked sites will not open, even if one uses a proxy server on a completely different port and an unlocked IP address.

B. *Measurements*

- Traffic measurements are performed in order to obtain numerical data on the load on the system, which allows to calculate the required network sizes.
- Under traffic research, we mean any collection of traffic data.
- Billing of telephone calls also corresponds to traffic measurements in which the amount of money spent is used as a unit of measure.
- The volume and type of measurements, as well as the measured parameters (traffic characteristics) must in each case be selected in accordance with the needs in such a way that the minimum technical and administrative costs bring maximum information and benefits.
- In accordance with the nature of the traffic, the measurement for a limited period of time corresponds to the registration of a specific implementation of the traffic transfer process.
- Measurement is a sample of one and several random variables.
- By making repeated measurements, we usually get a different value, and in general we can only state that an unknown parameter (the sample parameter, for example, the average value of the transmitted traffic) lies with a given probability in a certain interval, called a trusted one.

C. *Equations*

We denote the set of attributes as follows:

(1.1) $$F = \{f_i\}, i = 1 \ldots n;$$

For each characteristic, we can select a set of its values, based either on the training set, or using other apriori information about the problem, we denote the finite set of characteristic values as follows:

(1.2) $$\forall f_i \epsilon\, F \exists d_{f_i} \complement R;$$

It is also necessary to introduce the so-called measure of the inhomogeneity of the set with respect to its labels:

(1.3) $$\forall A \subseteq T : 0 \le \hat{p}_k(A) \le 1;$$

Thus, we defined an empirical discrete probability distribution of labels in a subset of observations. A measure of the inhomogeneity of this subset will be the function of the following form, where K(A) is the total number of labels of the subset A:

(1.4) $$\phi : [0,1]^{K(A)} \to \mathbb{R};$$

The measure of heterogeneity is set in such a way that the value of the function as possibility increases with increasing diversity of the set, reaching its maximum when the set consists of the same number of possible labels, and the minimum if the set consists only from labels from one class:

(1.5) $$\phi(\bar{p}) = \sum_{i=1}^{m} p_i (1 - p_i)$$

The algorithm for constructing a binary decision tree works according to the scheme of a greedy algorithm: at each iteration for an input subset of the training set, such a partition of the space is constructed by a hyperplane (orthogonal to one of their coordinate axes) that minimizes the average inhomogeneity measure of the two received subsets. This procedure is performed recursively for each received subset until the stop criteria are met. Let's write it more formally, for the input set A we find the pair <sign, the value of the characteristic>, that the measure of the inhomogeneity will be minimal.

D. Results of the experiment

Experimentally, a list of statistical characteristics of the sessions was developed, which can be regarded as independent and, at the same time, successfully used to classify traffic:

- Number of external ports;
- Number of internal ports;
- The share of outgoing traffic.

We formulate the statistical characteristics of the data stream, starting from these four series of numbers:

1. Average package size from the client side;

2. Standard deviation of the packet size from the client side;

3. The average packet size on the server side;

4. The standard deviation of the packet size from the server side;

5. Average size of a portion of data from the client side;

6. Standard deviation of the size of the data portion from the client side;

7. The average size of the data portion from the server side;

8. The standard deviation of the portion size from the server side;

9. The average number of packets per data portion from the client side;

10. The average number of packets per data portion from the server side;

11. Client efficiency: where it is the amount of applied application loaded and transferred divided by the total number of applied application and transport layer transfers;

12. Efficiency of the server;

13. Byte ratio - how many times the client transmitted more bytes than the server;

14. Payload ratio - how many times the client transmitted more bytes than the server;

15. Packet ratio - how many times the client transmitted more packets than the server;

16. Total number of transmitted bytes from the client side;

17. Total amount of transferred application-level load from the client side;

18. Total number of transported segments of the transport layer from the client side;

19. Total number of transferred portions of data from the client side;

20. Total number of transmitted bytes from the server side;

21. Total amount of transferred application-level load on the server side;

22. The total number of transported segments of the transport layer from the server side;

23. Total number of transferred portions of data from the server side;

24. The size of the first segment of the transport layer from the client side;

25. The size of the second segment of the transport layer from the client side;

26. The size of the first segment of the transport layer from the server side;

27. The size of the second transport-level segment from the server side;

28. The size of the first portion of data from the client side;

29. Size of the second portion of data from the client side;

30. The size of the first portion of data from the server side;

31. The size of the second portion of data from the server side;

32. The type of the transport layer protocol (0 - UDP, 1 - TCP).

As parameters of the algorithm Random Forest, after several experiments, the following values were chosen:

- Number of Trees: 27;
- Criterion: entropy;
- Maximum depth of the tree: 9.

Data Records	**Experiment**		
	Protocol	***Records in the training sample***	***Records in the test sample***
1	Skype	34	340
2	SSL	78	780
3	HTTP	769	7690
4	DNS	465	4650
5	BitTorrent	6	60

Data Records	**Experiment**		
	Protocol	***Records in the training sample***	***Records in the test sample***
All		1352	13520

Fig. 1. The results of the experiment on the selection of optimal parameters for the classification of network traffic.

Sign importance	**Experiment**	
	Sign	***Importance***
1	Number of packages	0,1
2	Average customer data size	0,1
3	The size of the first data packet	0,08
4	Number of bytes transferred	0,07
5	Standard deviation of the packet size	6
6	The size of the second packet from the server side	1352

Fig. 2. The results of measurments in the experiment on the selection of optimal parameters for the classification of network traffic.

Protocol	**Experiment**		
	Accuracy	***Completeness***	***Number***
Skype	0,94	0,99	319
SSL	1	0,99	780
HTTP	0,99	1	7613
DNS	1	1	4650
BitTorrent	0,98	0,89	58

Fig. 3. The results of data records in the experiment on the selection of optimal parameters for the classification of network traffic

III. Conclusions

It is hardly possible to imagine a world of modern network technologies without DPI (deep packet inspection). On it are systems for detecting network attacks, the lion's share of corporate network security policies, shaping and blocking of user traffic by the operator[13-15].

And yet, for all its relevance, DPI has some drawbacks. The main one is that the DPI tools need to see the payload of the parsed packages. And what if the client uses encryption? Or, for example, if there is no DPI here and now, but in the long term it will be necessary to conduct some kind of analysis of the current traffic on the network - then there will be only a possibility of the entire payload for subsequent analysis, which is very inconvenient.

The possibile application of the performed approach of the traffic data classifiation using machine learning algorithm “random forest” in radiofrequency applications can be, for example, the problem of identifying the application protocols used in applications in which the payload is encrypted or for preventive statistics collection (for defining an application-level protocol, it is sufficient to have value storage not more than 256 bytes of information per stream which represents 32 floating-point numbers).

The choice of variables (at least the initial one) is done intuitively. The experience in this subject area will help determine which variables are important. For starters, it makes sense to include all the variables that, in their opinion, can influence the result - in subsequent stages it is possible to reduce this set.

Advantages of the algorithm are:

- High learning speed;
- Non-iterative training - the algorithm is completed for a fixed number of operations;
- Scalability (ability to handle large amounts of data);
- High quality of the received models (comparable with neural networks and ensembles of neural networks [10]);
- Small number of configurable parameters;
- Internal assessment of the ability of the model to generalize;

At the same time, one can note the drawbacks of the algorithm:

- The model constructed occupies a large amount of memory. If there will be built a committee of K trees based on the training set of size N, then the memory requirements will be O (K · N). For example, for K = 100 and N = 1000;
- The trained model works somewhat slower than the other algorithms (if there are 100 trees in the model, we must walk through all to get the result);
- The algorithm is prone for retraining, especially - on noisy tasks. Partly this problem can be overcome by setting the parameter r (see below). A similar, only more pronounced problem is observed in the original Random Forest algorithm [10]. It should be noted that this shortcoming was not noticed by its authors, who assumed that the algorithm is not prone to retraining, and this error is shared by some practitioners and theorists of machine learning;
- Like decision trees, the algorithm is absolutely incapable of extrapolating.

Acknowledgment

This research was supported by International Telecommunication Academy and Department of Automated Control Systems National University of Science and Technology "MISiS". We thank our colleagues from the Department of Automated Control Systems in National University of Science and Technology "MISiS" who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

We thank Doroshkevich Olga, Director of Development in LLC "ID Media Publishers" for assistance with publishing, and Boris Alexandrovich Lastovich from the International Telecommunication Academy for comments that greatly improved the manuscript.

We would also like to show our gratitude to the Butromenko Elena from the International Telecommunication Academy for sharing her pearls of wisdom with us during the course of this research, and we thank 3 “anonymous” reviewers for their so-called insights. Although any errors are our own and should not tarnish the reputations of these esteemed persons.

References

[1] I.G. Baklanov organization / under. Ed. Yu Chernyshova. NGN: principles of construction and organization. Moscow: Eco-Trends, 2008, 400 p.

[2] V.I. Bitner, C.T. Mikhailova. Next Generation Network - NGN. Textbook for high schools. Moscow: Hot line – Telecom, 2011, - 226 p.

[3] V.G. Olifer, N.A. Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Peter, 2002, - 672 p.

[4] Yu.V. Semenov. Designing communication networks of the next generation. St. Petersburg: Science and Technology, 2005, - 240 p.

[5] Dorogov A.Yu. Structural and Topological Invariants of Fast Tunable Transformations. VIII All-Russian Scientific and Technical Conference "Neuroin Format-2006". Scientific session of MEPhI-2006. In 3 parts. Part 1. Moscow: MIFI, 2006, - pp. 39 - 50.

[6] Kohonen T. The "Neural" Phonetic Typewriter. IEEE Computer, vol. IV, no 1, 1988, - pp.11 - 22.

[7] 7. Glezer V.D. Sight and thinking. SPb .: Science, 1993, - 284 p..

[8] Glezer V.D. Sight and thinking. Izd.2-e .- SPb .: Science, 1993. - 284 p.

[9] Dyakonov V. MATLAB. Signal and image processing. Special directory / In Dyakonov., I. Abramenkova - St. Petersburg .: Peter, 2002. -608 p.

[10] Gorban AN A generalized approximation theorem and the computational capabilities of neural networks / A.N. Gorban // Siberian Journal of Computational Mathematics. - Novosibirsk: Russian Academy of Sciences. Sib. Division, 1998. - 1, No. 1.-C. 11-24.

[11] Lankin Yu.P. Training without a teacher based on neural networks with self-adaptation / Yu.P. Lankin, T.F. Baskanova // Relational, continuous-logical neural networks and models. Proceedings of the International Conference

[12] "Continual algebraic logic, calculus and neuromathematics in science, engineering and economics - KLIN - 2002". Volume 3. - Ulyanovsk: UlSTU, 2002. - P. 39- 41.

[13] I.G. Baklanov, NGN: principles of construction and organization / under. Ed. Yu Chernyshova. - Moscow: Eco-Trends, 2008. - 400 p.

[14] V.I. Bitner, C.T. Mikhailova, Next Generation Network - NGN. Textbook for high schools. - M.: Hot line - Telecom, 2011. - 226 with.

[15] V.G. Olifer, N.A. Olifer, Computer networks. Principles, technologies, protocols. - St. Petersburg: Peter, 2002. - 672 p.