UDC 004.891.2
DOI https://doi.org/10.32838/2663-5941/2020.5/19

Suhoniak I.I.
Zhytomyr Polytechnic State University

Yefimenko A.A.
Zhytomyr Polytechnic State University

Marchuk G.V.
Zhytomyr Polytechnic State University

Feschenko D.I.
Zhytomyr Polytechnic State University

DECISION SUPPORT SYSTEM DEVELOPMENT FOR BLOCKING UNWANTED CONTENT BY NEURAL NETWORKS

Context. *The development of information technology and computer technology are main stream of nowadays. Virtual reality become more important than real life. Social measures, electronic mail, video-conferencing, forums, news - all tis technologies are implemented in people activities. That's why spam is part of our information flows too. Spam is what every network user is familiar with. In the classical sense of the word, spam is the mass distribution of certain information, mostly of an advertising nature, without the consent of the people who receive it. This term began to be used actively in the early 90's, when the mass distribution of computers led to the appearance of large advertising companies on the Internet. This problem is important today because of the need to build a complex decision support system for filtering spam, advertising and pornographic content.* ***Objective.*** *The purpose of the paper is analyzing of models and methods for building decision support systems for filtering a variety of spam.* ***Method.*** *Research contain the model of quickly determining and filtering spam content, advertising and adult content combining classical methods with modern ones based on the use of neural networks. The essence of this approach is the joint use of the classical method of classification of text spam based on the Bayesian theorem, as well as modern algorithms based on convolutional neural networks and optical character recognition methods. Existing architectures for classifying a large volume of images based on convolutional neural networks and their modifications were investigated, after which a new integrated solution was developed for spam, advertising and pornographic images based on research data.* ***Results.*** *The purpose of the study is to analyze the models and methods used to build decision support systems for filtering advertising and pornographic content, as well as tools for their implementation and use.* ***Conclusions****. The article examines a new system for filtering spam and advertising in the information environment that provides tools for the rapid recognition and filtering of spam content, advertising and content for adults by combining classical methods with modern ones, based on the use of neural networks.*

Key words: *spam, advertising, neural networks, convolutional neural networks, Android, iOS.*

Introduction. Big companies forced to pay a lot of attention to information security issues, which also includes struggle with spam. The methods include: information verification by human, the use of various gray and black lists, domain blocking, and the use of various statistical methods.

The classic methods of dealing with NSFW (short for Not suitable / safe for work, not dangerous for work), such as checking and confirming every image or video by a person, are no longer relevant. However, new investigations into machine learning and profound neural networks, enable go to new finds in this area. In particular, one of the most effective statistical methods is a method, that based on the use of the Bayes Theorem, which, after training on a large enough sample, can filter out about 95–97% of text spam.

Problem statement. The purpose of the paper is analysing of models and methods for building decision support systems for spam filtering, advertising, pornographic content.

To achieve this goal, following tasks was to be resolved:

− analyze the methods and approaches to the task of recognizing, filtering and blocking spam;

− design a model for filtering spam and pornographic content;

− develop a service for optical text recognition and a service for lexical and morphological text processing;

− develop and implement a decision support system for identifying and blocking spam content, advertising, and pornographic images.

Review of the literature. The particularity of this problem is the few methods and models was been research and published. The main research are [3–11]. Overview of popular machine learning methods and their application to the problem of spam filtering is bounded in [3]. The article is related to apply machine learning methods in practice, and therefore may be of interest mainly to those familiar with the topic. The paper [4] examines the problem of spam filtering and the most common approaches to solving it: on the basis of lists of addresses, signatures, Bayes theorem in comparison with methods of artificial intelligence. An artificial neural network approach is proposed to solve the problem. But this approach requires training and test message sampling to train the classifier, marking important message features, model parameters correction, evaluating classifier accuracy etc. How to overcome these barriers is contained in [5]. The impact of depth convolutional network to accuracy in large-scale customization of image recognition is investigated there. The possibility to use residual learning frameworks as one approach is described in [6]. The mail's messages filtering method on the server side is used in [8] for facilitate the learning of the network. Anti-spam filtering using text categorization methods is possible least for mailing lists and newsgroups [9]. The machine learning methods [11] have highest performance and best results in spam classification, over of all the spam filtering methods. But all methods have some messages mistakenly classified as spam. That it why, new research in anti-spam methods is actually now.

Materials and methods. ***Classical spam and adware method.*** One of the most effective methods for filtering text spam is the use of the Bayes' classifier. The basis of this method is the use of the Bayesian theorem [1]. This method was first used to classify and detect spam in 1996. The majority of modern spam filtering tools use this method, or combine it with other methods to improve the results.

In order to bypass the antispam protection systems that use the Bayes' classifier, the hackers began to convey the ad text as a picture or animation. To secure from such images, the method for image signatures recognition has started to be applied. In order to recognize the image data, a method based on the analysis and recognition of image signatures in the message can be applied and the creation of a corresponding database of collected signatures. Accordingly, after receiving a new message, it is only necessary to obtain the signatures of all the images included in it, and then compare them with those available in the database.

To combat this method of protection, the method of dynamic image formation is quite often used, which can be substantially changed at the moment of generation - the size of the image can be enlarged or reduced, the font and color of the text are changed, and so on.

In order to block ads without spending user traffic, it is necessary to intercept messages and block them before sending a message. The method of the message sender recognition is not based on the analysis of the message content, but namely on the information about its sender.

The advantage of this method is the relative simplicity of implementation and a small requirement for resources.

The disadvantages include the fact that all the letters will come with a certain delay because of such a check, and if there is a failure or error, a large number of users who use the services of one server will be blocked and will not be able to send messages.

The method of using grey lists is based on the fact that the behavior of spam software differs from the behavior of ordinary mail services, for example, spam servers do not attempt to re-send a message, as required by SMTP. Accordingly, in this case, the server is put on the list as potential spammer. Mail clients that are targeted to this list can automatically log messages from this sender to the spam section or simply ignore such messages [2].

Advertising and spam recognition using OCR. Optical Character Recognition (OCR) methods can be used to recognize messages with dynamically generated promotional or spam images. These methods allow to detect a text in an image, recognize it, and then use the obtained text for analysis with other spam filtering methods.

Nowadays, intelligent methods are used to recognize characters that in their turn can recognize not only printed letters, but also to some extent handwritten characters, and so on [3-11].

Recognition NSFW methods. The general NSFW algorithm can be described in two steps. In the first step the images with large areas of flesh colour pixels need to be found. The next step is to find elongated parts in these areas and try to group them into possible human limbs or groups of limbs with the help of specialized grouping modules. These modules contain a large amount of information about the structure of the object.

Today Facebook is using the method of modification to prevent the distribution of its own photos with a certain pornographic content. Yahoo company uses the Convolutional Neural Network (CNN), which has about 50 hidden layers and was trained with the ImageNet 100 class dataset. This dataset contained a large number of images, each of which was labeled NSFW or as a safe image.

They used the CaffeeOnSpark framework to train this network with Hadoop and Spark clusters. They were able to train and preserve the model for the detection of pornographic images. The corresponding model can be used by researchers from all over the world to create their own networks for the recognition of NSFW in combination with the framework for the creation and training of deep neural networks – Caffee.

Setting up the task for developing of spam and advertising DSS filtering. According to the development of modern means such as computer vision, improved network training, algorithms of deep learning, computers can automatically perform recognition and filtering of such content with a high probability.

The protection against unwanted content is quite subjective and does not have clear boundaries, because even a person in different circumstances may misinterpret what content is acceptable, obscene or contains a certain context.

Within this article, only 1 type of NSFW content will be considered, namely pornographic images that allow to appropriately rate a specific image or group of images as probably relevant to the pornographic content.

Based on the comparative analysis of spam, advertising and pornographic content filtering techniques, it has been determined that each of these methods has certain advantages and disadvantages. The classification based on Bayes' theorem is extremely effective in relation to text content, but it is completely helpless against any visual content. OCR methods are extremely effective for cases when it is necessary to take texts on images into account.

Modern methods based on convolutional neural networks allow to quickly teach the model to recognize NSFW with a fairly high probability [4–9].

There is a need to develop a universal application that combines the basic classical methods for solving spam and adware filtering problems, but also contains modern tools for covering a greater number of different scenarios for the use of DSSs.

The service being developed should be flexible enough to be able to add new functionality without having to make significant changes to the existing code. The appropriate solution should be developed in the form of a SaaS application that can be easily placed in the cloud and provide this functionality to any user without the need to install the application, and to maintain its sustainable work

Development of a model for spam and pornographic content filtering based on a combination of classical and machine learning tools. Having analyzed the available methods and approaches to the recognition, filtering and blocking of spam and ad content, it can be concluded that none of them covers all aspects of content analysis and, as a result, prevents complex analysis and filtering of unwanted content.

The decision support system for filtering should not only be focused on the presence of certain keywords, or be entirely based on the assumption that one of these keywords is more likely to occur in spam and others in real content. According to these conditions, it is necessary to build a complex of several services, in particular, an image processing service with the OCR module, a link processing service, a module for the recognition of pornographic images, which will also interact with the module for comparing signatures with the database.

In order to distinguish spam from useful content for the text-processing service, it's advisable to use the Bayes' formula.

For training and verification, CSMining has been selected to collect and distribute public data sets, including the LingSpam dataset, which contains real examples of spam and useful emails. The original set can be downloaded at: http://csmining.org/index.php/ling-spam-datasets.html.

In order to make the service more versatile and flexible, it is necessary to use an improved version of the Bayesian Classifier (modification for many-to-many type classification).

To improve the service of text spam recognition, it is advisable to use appropriate libraries for primary processing. One of the stages of such processing is the algorithm for finding the base word for a given derived word (the so-called "stemming"). One of the most popular implementations of this algorithm is Porter's stemmer. This algorithm was developed by Martin Porter in 1980 and worked only in English. After that, appropriate algorithms for other popular languages were implemented. This algorithm does not use the root of the word, but just applies a series of rules that are intended to cut off the endings and suffixes. At the same time, the algorithm is very fast. The negative point is that this algorithm may produce a faulty result, that is, its accuracy is not absolute [7].

The pre-processing service will be used to translate words into normal form, remove extra words, transliteration, etc. The processed text will be put into the input for the Bayes' classifier and at the same time for the neural network in order to improve the quality of the system.

Beginning since 2012 Convolutional Neural Networks have been continually improving the results of the classic ImageNet Imaging Excellence test. This has led to the appearance of certain modified methods, such as residual networks (RN). Each modification differs from others in speed, equipment requirements and accuracy. The ideal option is a network with the minimum number of connections and parameters, but with the maximum accuracy of the network [5].

To train the network the setoff data, which consists of both positive (belonging to NSFW) and negative (non-NSFW) images that are split into 2 separate directories, is used.

As the network training is a very tedious and lengthy process, it was decided to use the Yahoo CaffeeOnSpark open source framework. This company also provides its own model, which is already somehow trained and can be used for further improvement and research.

There are many different options for building the architecture of residual neural networks, but the ResNet-50thin architecture is most suitable for this task. This architecture was first described in the "Deep Residual Learning for Image Recognition" conducted by Microsoft researchers [6]. The ResNet network with 50 layers of neurons given that each layer contains half of the filters, provides sufficient accuracy for the acceptable network operating time. In general, such a network consumes less than 0.5 seconds of CPU time and approximately 23MB of memory.

The resulting model can be further adjusted for a specific task, thereby improving the results of its work. This is achieved through the use of correction algorithms (such process is known to the researchers as "fine-tuning").

As a metric (which will verify the quality of the resource content and its security in general), it was decided to introduce a new coefficient – the security index. The security index is a value that determines content quality in terms of lack of advertising, spam and pornographic content that a certain resource provides. In general, the security index for a resource is a number from 0 to 5, given that the closer it is to 5, the more qualitative is the content of the resource. This index can be used to automatically block dangerous resources, build a certain ranking of sites and pages, and be actively used as a basis for the parental control tool.

This metric is complex, so it requires the introduction of appropriate new values – the safety index of advertising, as well as the security index from NSFW.

An advertising security index is a number from 0 to 1, given that the larger it is, the smaller the ratio of the number of ads to the total content of the content. According to this definition, the safety index of advertising can be expressed by the formula:

$$I_{adsec} = \frac{C_{ham}}{C_{ham} + C_{ads}}, \qquad (1)$$

where C_{ham} is the total amount of useful content (not advertising), and C_{ads} is the total number of ads.

Depending on the detail, formula (1) can be applied equally well for both a specific content group and for a certain resource as a whole.

The security index from pornographic content is also a floating point number from 0 to 1, given that the larger it is, the less the ratio of the number of ads to the total number of NSFW content is.

$$I_{nsfwsec} = \frac{C_{ham}}{C_{total}}, \qquad (2)$$

where C_{ham} is the total amount of useful content (not NSFW), and C_{total} is the total amount of analizing contet for resources.

Depending on the options set by a user, a certain index can be ignored or taken into account during a general estimation.

To increase the flexibility of the system the ability to customize the parameters of system settings need to be added. In particular, to change the heuristics to determine whether this content is spam, customize image, link and text processing services, enable or disable the features that you want or do not need, etc.

The general scheme of interaction between modules and DSS filtering services for spam, advertising and NSFW presented in fig. 1.

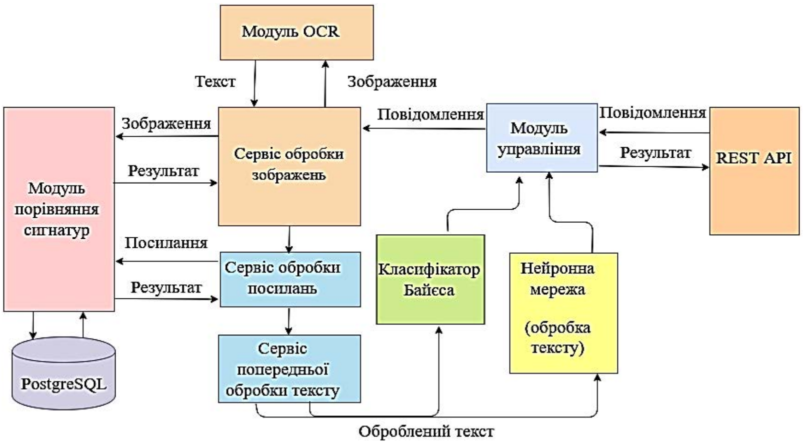


Fig. 1. The general scheme of interaction between modules and DSS filtering services for spam, advertising and NSFW

The system should automatically learn, thereby increasing the efficiency of its work

DSS structures and data warehouse organization. The DSS structure should be modular and provide the opportunity for further improvement and expansion of the functionality.

According to this goal, the most appropriate solution will be the use of a structure consisting of a large number of different services. In this case, each service can use different technologies and data sources. If necessary, the system performance can be adjusted by adding functionality to load balancing and replication of those services that are most often used. Also, this structure allows to place the corresponding services on different working machines with the help of appropriate auxiliaries and software systems.

According to this, it is expedient to use several delimited data sources instead of a single large source. This will allow the use of different databases and database management systems for different services, depending on the tasks and needs.

For communication between services, it is advisable to use one of the existing methods, such as the use of the REST API, SOAP, protobuf, GraphQL, and others. As a result, a set of maximally independent services and ones with the possibility of their re-use is aquired.

At the same time, each service must be responsible for a certain logical part of the application, that is, it is a complete part of business logic that may well solve certain tasks within its own responsibility (Figure 2).

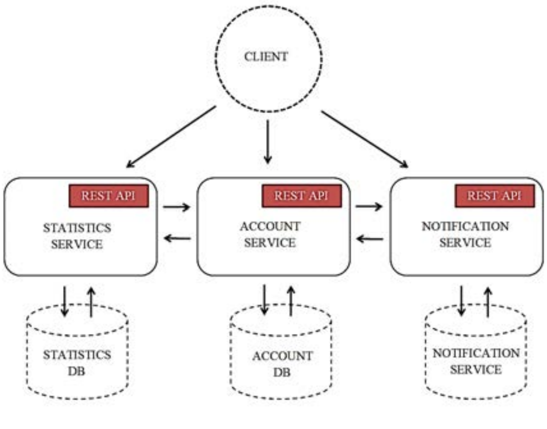


Fig. 2. General scheme of the organization in the form of separate services

According to these requirements, it was decided to create 2 separate data warehouses. The first one is used to store statistics and estimate site security indexes and will be organized in a relational form and contain user data, scanned pages and sites, a list of blocked resources, certain image signatures and the like. The second data warehouse is designed to store service logs that allow to determine the load, the time of recall of various services, errors occurring during the application.

According to this structure, services cannot access the data store that is not directly owned by them. To do this, queries between services need to be used, which makes services more independent and allows to make the structure of the project more flexible, and also allows to use the services repeatedly when necessary.

To provide a single point of access to the application, it is expedient to use the Proxy API Gateway pattern, which will be implemented as a separate service and which will provide access to all internal services and utilities.

Design and implementation of the separate system modules. To implement a service that performs spam recognition based on Bayes' theorem, 3 components need to be created:

− Bayesian classifier that will train and perform the spam recognition function;

− a training system for the classifier that takes data from Lingspam, performs the analysis and preliminary training of data and initializes the classifier;

− a server in Golang, which creates access points to the classifier functions in the form of gRPC and RESTFul API.

The corresponding classifier was created in the form of a package for the programming language Golang, which makes it more versatile and allows to use it for other tasks when needed.

When the service is launched, automatic network training is performed using Lingspam data set, and the overall accuracy of the programme is estimated using the appropriate test message set. To do this, read the data from 2 directories for training, transfer them to the classifier in training mode. To calculate the accuracy of the network a set of real-time emails with 2 directories for training is used. After that, a gRPC server is generated and launched, which allows to communicate with other services and send data in binary format.

To implement the NSFW recognition service the trained model for the Caffee framework is used. The corresponding service is written in the Python programming language (Figure 3).

When launching Docker the Caffee model files and the description of the neural network are copied to the source container. Immediately after loading Python launches a script that initializes the Caffee network using the appropriate model and description. Network initialization usually takes no more than 1 second.

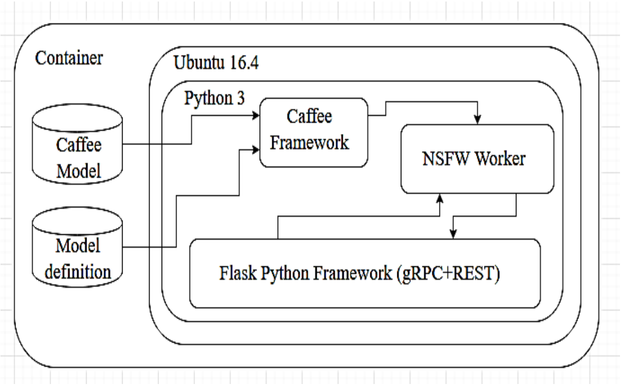


Fig. 3. The general structure of the container for NSFW recognition

Immediately after downloading CNN and all necessary data, the Flask server starts to listen to the corresponding ports (in the case of REST) and raises the gRPC server with the appropriate settings for Service Discovery.

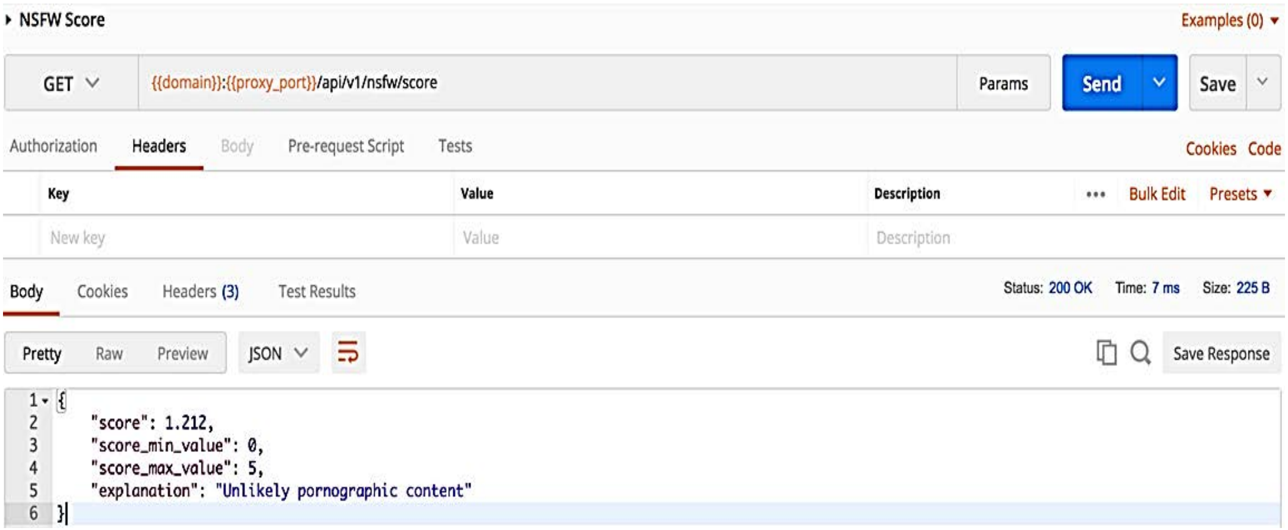
To implement the optical character recognition service, it was also decided to use the Go programming language and Tessaract's service as an engine for recognition. This is done by using the port of the engine from the C ++ programming language to the Golang programming language, which is called gosseract. It comes in the form of appropriate packages that can be easily integrated into own applications.

In order to ensure the integrity of the system and provide the user with a more convenient way to interact with it, it was decided to use the pattern of the Proxy API Gateway. It makes it much easier to interact with services that use different protocols for data transfer between each other, such as REST, GraphQL, APMQ, gRPC, etc. It also allows to implement multiple gateways for different clients without changing the internal structure of services in any way.

The REST architecture as well as the AMPQ and gRPC data transfer protocols were used in the development of this application. Also, it is very important that all customers of this system can be divided into 4 main categories: browser extensions, mobile applications, Web clients and arbitrary developers of their own solutions and APIs.

To implement the Proxy API Gateway a proxy_ api_gateway service has been created that includes clients for all internal gRPC services, is able to serialize and deserialized data for services that use the REST architecture, and also works well with message brokers using the AMPQ protocol. After that the service provides all available access points for this application available to external applications and users.

The basic queries that the Proxy API Gateway can take from customers. Api / v1 / nsfw / score query allowes to determine whether the image belongs to the content class for adults (Figure 4). It returns an evaluation, the minimum and maximum value used by the Google Chrome browser extension, as well as an additional explanation for this value, which is also displayed in the modal window of this extension.



GET {{domain}}:{{proxy_port}}/api/v1/nsfw/score

Status: 200 OK Time: 7 ms Size: 225 B

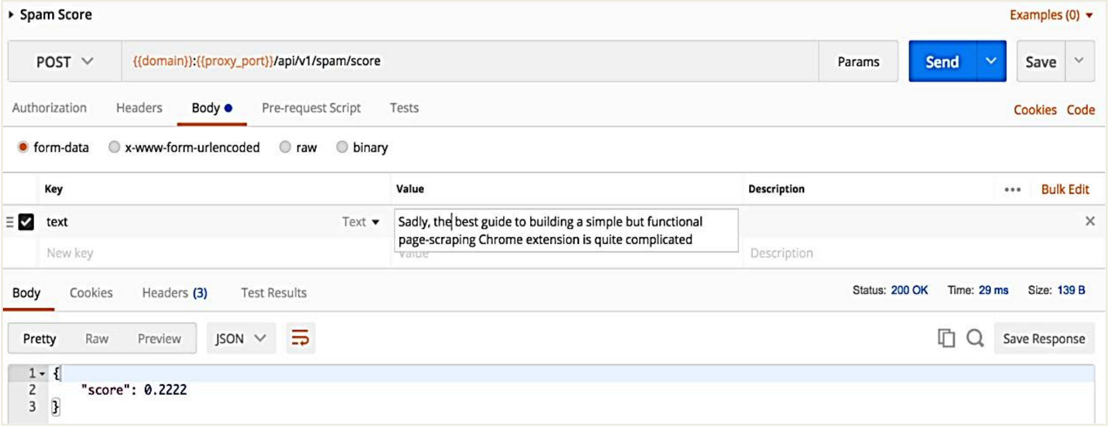
```
{
    "score": 1.212,
    "score_min_value": 0,
    "score_max_value": 5,
    "explanation": "Unlikely pornographic content"
}
```

Fig. 4. Request to Proxy API for NSFW recognizing

Request by pattern api/v1/spam/score should take result, is the text a spam or not (Figure 5).

Api / v1 / batch query combines the previous two queries, and also contains additional fields for the query. An example of the following query parameters in the JSON format:

```
        {"base_url":
        "http://www.tomforth.co.uk/chromeextension/",
"images":["http://www.tomforth.co.uk/chromeextension/e xamplepic.png",
        "http://www.tomforth.co.uk/chromeextension/p ic2.png",
         http://www.tomforth.co.uk/chromeextension/p
        ic3.png" ],
               "articles":  ["Sadly,  the  best  guide  to  building  a  simple  but  functional
page-scraping Chrome extension is quite complicated. So I've learned from it and written
a much simpler Hello World Chrome extension for page scraping.",
               "Want  to  parse  the  content  of  a  website?  More  comfortable  coding  in
javascript and displaying your results in HTML than you are using Scrapy at a Python
command prompt? A google Chrome extension might be perfect for you."
        ]}
```



POST {{domain}}:{{proxy_port}}/api/v1/spam/score

● form-data ○ x-www-form-urlencoded ○ raw ○ binary

Key	Value	Description
☑ text (Text)	Sadly, the best guide to building a simple but functional page-scraping Chrome extension is quite complicated	
New key	Value	Description

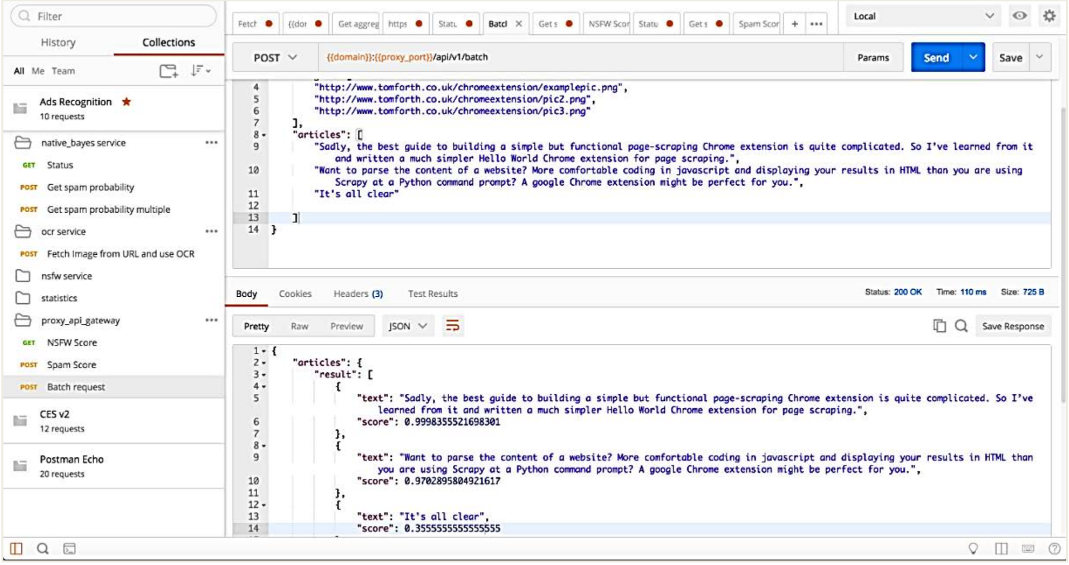
Status: 200 OK Time: 29 ms Size: 139 B

```
{
    "score": 0.2222
}
```

Fig. 5. Query for the Proxy API to recognize spam

In the response the result of the analysis for images on NSFW and spam for the text will be obtained (Figure 6).

Relevant query results will be automatically saved to the database in order to use them to build a general ranking of sites, as well as for caching.



POST {{domain}}:{{proxy_port}}/api/v1/batch

```
        "http://www.tomforth.co.uk/chromeextension/examplepic.png",
        "http://www.tomforth.co.uk/chromeextension/pic2.png",
        "http://www.tomforth.co.uk/chromeextension/pic3.png"
    ],
    "articles": [
        "Sadly, the best guide to building a simple but functional page-scraping Chrome extension is quite complicated. So I've learned from it
            and written a much simpler Hello World Chrome extension for page scraping.",
        "Want to parse the content of a website? More comfortable coding in javascript and displaying your results in HTML than you are using
            Scrapy at a Python command prompt? A google Chrome extension might be perfect for you.",
        "It's all clear"

    ]
}
```

Status: 200 OK Time: 110 ms Size: 725 B

```
{
    "articles": {
        "result": [
            {
                "text": "Sadly, the best guide to building a simple but functional page-scraping Chrome extension is quite complicated. So I've
                    learned from it and written a much simpler Hello World Chrome extension for page scraping.",
                "score": 0.9998355521698301
            },
            {
                "text": "Want to parse the content of a website? More comfortable coding in javascript and displaying your results in HTML than
                    you are using Scrapy at a Python command prompt? A google Chrome extension might be perfect for you.",
                "score": 0.9702895804921617
            },
            {
                "text": "It's all clear",
                "score": 0.3555555555555555
```

Fig. 6. Example of query response from spam, ad, and NSFW content recognition using Postman

In order to compile a particular service, there is a special Makefile file without an extension at the root of this service. Typical structure of this file (example for Proxy API Gateway service):

```
build:
       GOOS=linux GOARCH=amd64 go build        docker build -t proxy_api_gateway .
run:   docker run -e MICRO_REGISTRY=consul proxy_api_gateway
```

This file determines that the standard Golang compiler will be used to compile the Go application. The application itself will be compiled into a binary type for the Linux operating system under the amd64 architecture. This is very important as the developed system works fully under the control of Linux machines, and the data flags provide the correct compilation for this particular platform.

After that Docker is used to build the application. Each Dockerfile describes everything that is needed to run the service, including such data as: the basic image, all the necessary commands for preparing, the application compilation, setting environment variables, settings for various additional utilities, and so on (Figure 7).

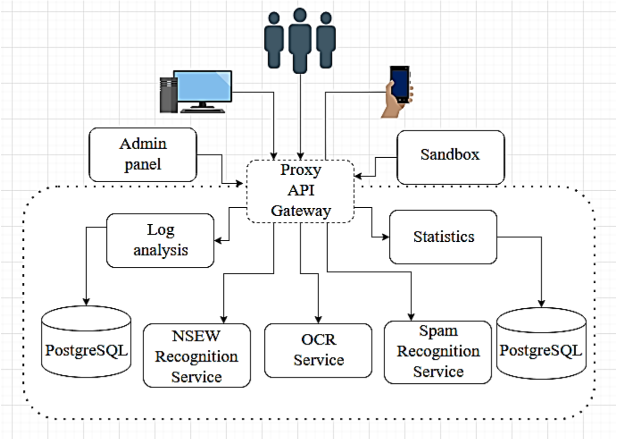


Fig. 7. The services general architecture for the advertising recognition

Implementation of its own extension for Google Chrome consists of several steps. The first step is to create a manifest.json file that contains a description of actions, scripts and a list of required permissions from the user side for the correct operation of the application.

This Manifest indicates that the correct operation of this extension requires access to all active tabs, to both HTTP and HTTPS content, to localStorage to save extension settings and the like. It also determines that for all available tabs, it is necessary to automatically insert scripts of this extension, which will be responsible for blocking of unwanted content.

Extension is written using TypeScript, which is compiled using the Webpack module in JavaScript. An example of the Parser class that finds all images inside the corresponding DOM node:

```
      import  *  as  $  from  'jquery';  import  Image  from  './Image';  class  Parser  {
images: Array<Image>;     constructor() {
              this.images = new Array<Image>();
          }
         /* Finds all images within the given root
      Element.
      *      Filters images to avoid animations (gif) and cut
      *      off all icons, logos and thumbs. Also, includes
      *      only images more than 50px in width and height to
      *      ignore many small images.
      *      Returns an array of images.
      *      @param  $root Root for searching images
      *      @returns array an array of images.
          */     static  parseImages($root:  Document)  :  Array<String>  {          const
imagesSources = [];       const images = $($root).find("img");       for (let i = 0; i
< images.length; i++) {             let $image = $(images[i]);            let src =
$($image).prop('src');
      if (src.includes(".gif")) {continue; }
      if  (src.includes("icon")  ||  src.includes("logo")  ||  src.includes("thumb"))  {
continue; }
      if ($image.width() > 50 && $image.height() > 50) {  imagesSources.push(src);  }
              return imagesSources;
          } } export default Parser;
```

To store image data, articles, and other media data, the appropriate classes of Article, Image, Media are used. After loading the extension, it automatically inserts the content_script.js script into the user's page. This script communicates with the background.js script in order to share the current page data with it. This communication uses postMessage technology, which avoids problems with crossdomain queries, creating a certain channel between the user's page and extensions scripts for Google Chrome.

After this, the initialization of the Parser class object, which in its turn searches for content that can potentially be advertising, spam, or adult content, is initialized. The relevant findings are serialized and sent to the Proxy API Gateway service. Then follow the registration of the corresponding page and site in the system, all necessary preliminary data are stored in the database. In order to reduce the load on the neural network, a query is made to the local cache and it is checked for the certain images in its data.

After that, the optical character recognition service analyses the image ads, after which all recognized texts are sent to the service of advertising and spam classification.

Conclusions. For analysis of pornographic images, the NSFW recognition service is used. After performing the relevant queries, aggregation and serialization of the data in the Proxy API Gateway service takes place, which is given to their client for display.

References:

1. Metsis V. Spam Filtering with Naive Bayes – Which Naive Bayes? *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006)*. Mountain View, CA, USA, 2006.

2. Зайцев О. Технологии рассылки спама и методы защиты от него *КомпьютерПресс*. 2007. № 2. С. 75–78.

3. Tretyakov K. Machine Learning Techniques in Spam Filtering. *University of Tartu. Data Mining Problem-oriented Seminar, Institute of Computer Science.* MTAT.03.177. May 2004. P. 60–79.

4. Ларионова А.В., Хорев П.Б. Метод фильтрации спама на основе искусственной нейронной сети. *Интернетжурнал «НАУКОВЕДЕНИЕ».* 2016. Том 8. № 3 (май–июнь). URL: http://naukovedenie.ru/PDF/04TVN316.pdf

5. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations.* 2015. URL: https://arxiv.org/abs/1409.1556v6

6. Deep Residual Learning for Image Recognition / He Kaimihg et al. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016. P. 770–778. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

7. Porter M. Snowball: A language for stemming algorithms. *Published online.* 2001. URL: http://snowball.tartarus.org/texts/introduction.html

8. Абдуллаев В.Г. Защита от спама в интернет-пространстве. *Радиоэлектроника и информатика*. 2014. № 2. С. 35–38.

9. A memory-based approach to anti-spam filtering for mailing lists / G. Sakkis et al. *Kluwer Academic Publishers. Manufactured in The Netherlands.* 2003. P. 49–73.

10. Seguin K. The Little Go Book. *Published online*. 2014. P. 83. URL: https://openmymind.net/assets/go/go.pdf

11. Mahsa Riahi Asl, Hasan Naderi. Filter Spamming In Computer Networks by Text Mining and Machine Learning Method International *Academic Journal of Science and Engineering.* 2016. Vol. 3. № 8. P. 32–46.

Сугоняк І.І., Єфіменко А.А., Марчук Г.В., Фещенко Д.І. РОЗРОБКА СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ БЛОКУВАННЯ НЕБАЖАНОГО КОНТЕНТУ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

Актуальність. *Розвиток інформаційних технологій і комп'ютерних технологій сьогодні є основним напрямом. Віртуальна реальність стала важливіше реального життя. Соціальні заходи, електронна пошта, відеоконференції, форуми, новини – всі ці технології впроваджені в діяльність людей. Ось чому спам теж є частиною інформаційних потоків. Спам – це те, з чим знайомий кожен користувач мережі. У класичному розумінні цього слова спам – це масове поширення певної інформації, в основному рекламного характеру, без згоди людей, які її отримують. Активно цей термін почали використовувати на початку 90-х років, коли масове поширення комп'ютерів призвело до появи в Інтернеті великих рекламних компаній. Ця проблема важлива сьогодні через необхідність створення комплексної системи підтримки прийняття рішень для фільтрації спаму, реклами і порнографічного контенту.* ***Метою дослідження*** *є аналіз моделей і методів, використовуваних під час побудови систем підтримки прийняття рішень для фільтрації різноманітного спаму.* ***Методи.*** *Дослідження містять* *модель швидкого визначення та фільтрації спам-контенту, реклами і контенту для дорослих, яка поєднуватиме класичні методи із сучасними, які використовують нейронні мережі. Суть даного підходу полягає в спільному використанні класичного методу класифікації текстового спаму на основі теореми Байєса, а також сучасних алгоритмів на основі згорткових нейронних мереж і методів оптичного розпізнавання символів. Були досліджені існуючі архітектури для класифікації великого обсягу зображень на основі згорткових нейронних мереж і їх модифікації, після чого було розроблено нове рішення для визначення спаму, реклами й порнографічних зображень, заснованих на наукових даних.* ***Результати.*** *Мета дослідження полягає в тому, щоб проаналізувати моделі та методи, що використовуються для систем підтримки прийняття рішень для фільтрації спаму різного типу, а також інструменти для їх впровадження і використання.* ***Висновки.*** *У статті розглядається нова система фільтрації спаму і реклами в інформаційному середовищі, яка надає інструменти для швидкого розпізнавання й фільтрації спам-контенту, реклами та контенту для дорослих шляхом поєднання класичних методів із сучасними, заснованими на використанні нейронних мереж.*

Ключові слова: *спам, реклама, нейронні мережі, згорткові нейронні мережі, Android, iOS.*