WaveAccess
We Make IT Easy

Machine learning

based module to detect users' fraudulent activity

About

WaveAccess is a results focused software development company that provides high quality software outsourcing services to hundreds of emerging and established companies globally. We use our technical expertise to increase business efficiencies, optimize slow or unreliable systems, recover projects that have gone off track and bring ambitious ideas to life.

22
years of delivering successful outcomes for customers

800+
talented and passionate professionals

8
R&D centers and regional offices

17+
industry verticals from banking to healthcare

500+
successful projects delivered and counting

96%
customer satisfaction index

Awards and Recognitions

Microsoft Partner — Microsoft — 2017 Partner of the year Business Analytics Award

Microsoft Partner — Microsoft — 2018 Partner of the year Artificial Intelligence Award

Microsoft Partner — Microsoft — 2019 Partner of the year Media & Communications Award

Microsoft Partner Network — WORLDWIDE DYNAMICS PARTNER OF THE YEAR — 2009, 2010, 2011 & 2016

GDPR

Information Security — ISO/IEC 27001

SCI-TECH AWARDS (ACADEMY OF MOTION PICTURE ARTS AND SCIENCES)

Project overview

The subsystem that is based on machine learning and fuzzy logic algorithms developed for the big telecom company in Russia. The company needs to protect its web portal with more than million users from data fraud.

The problem

Our client needed the protection module that can tell different fraudulent actions from millions of normal actions of the portal's users.

The fraudulent activity is hard to predict or to prevent. Be it access from some fake IP, or attempt to steal a credit card number, as well as several dozen other unpredictable events - cybercrimes are inventive and unpredictable.

To recognize suspicious actions as threats, our customer needed a module based on algorithms of fuzzy logic and machine learning. We were required not only to develop, but also to train it.

Why WaveAccess?

We specialize in developing and implementing machine learning and fuzzy logic based systems. Our clients use these solutions to implement:

- Trend and event forecasting.
- Object recognition and classification.
- Personal recommendations.

Being a full circle developer allows us to immediately offer the whole working team, and to provide our client with comfortable project monitoring tools.

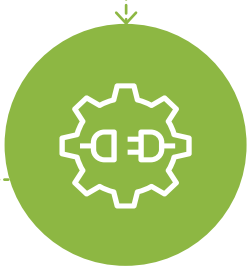
Work stages

1 **Analysis of possible attacks** and formalization of 16 types of threats to the system.

2 **Implementation of fuzzy logic:** selection of the input output engine, caching of elements and creation of API.

3 **Implementation of machine learning** to customize the analytics module with training data.

4 **External API** for integration with the developed module.

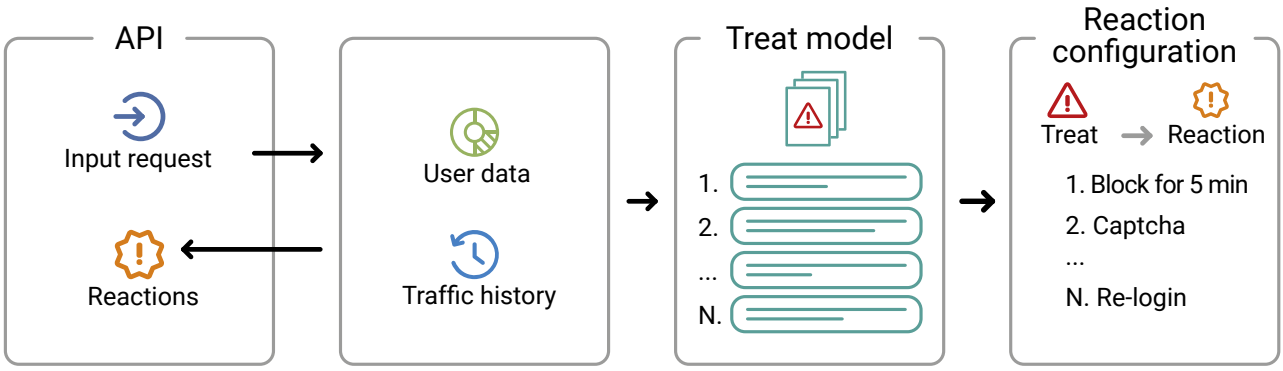


5 **Development of the administration module** with the user interface.

6 **Development of a «test utility»** for performance, load and integration testing.

7 **Testing:** autotests, installation testing, interface tests.

First, we analyzed most of possible threats to the portal, which allowed us to define and formalize 16 main threats types. For each of these threats we developed an algorithm which determines how a threat will be defined and dealt with. For some threats, a user's feedback is requested, which determines the reaction to a threat.



At the next stage we started developing the protection module. We started from the analytics submodule, an API, and the testing utility.

The analytics module is the core of the system. It will process data with algorithms of fuzzy logic. Algorithms were implemented in two different realizations: small classes containing calculations and statistics gathering structures with complex architecture and use of fuzzy logic.

Implementing the fuzzy logic turned out to be a very complex task. We chose an template based input/output engine, then customized it: cached some of its elements, then created an API to interact with templates.

We also implemented instant cash regeneration if the fuzzy logics calculation values change (boundary values).

The machine learning allows for tuning the analytics module using the training data. This helps to detect fraud even if a template has deviations. This approach covers more similar situations, however, it requires resources to maintain the tuning and testing.

Machine learning is divided to two stages:

1. **Detection of anomalies.** Helps to detect and analyze unknown patterns of suspicious activity

2. **Classification of patterns** (both known, and detected in the Detection of anomalies stage)

While implementing the external API for integration with the Anomalies Monitor, we followed REST architecture guidelines, API is kept small. The data coming from the API is subjected to basic validation.

Both API services and the whole system are built by the guidelines of stateless architecture, which allows us to easily scale the developed product if needed. The cluster consists of four servers of applications with redundancy and load balancing.

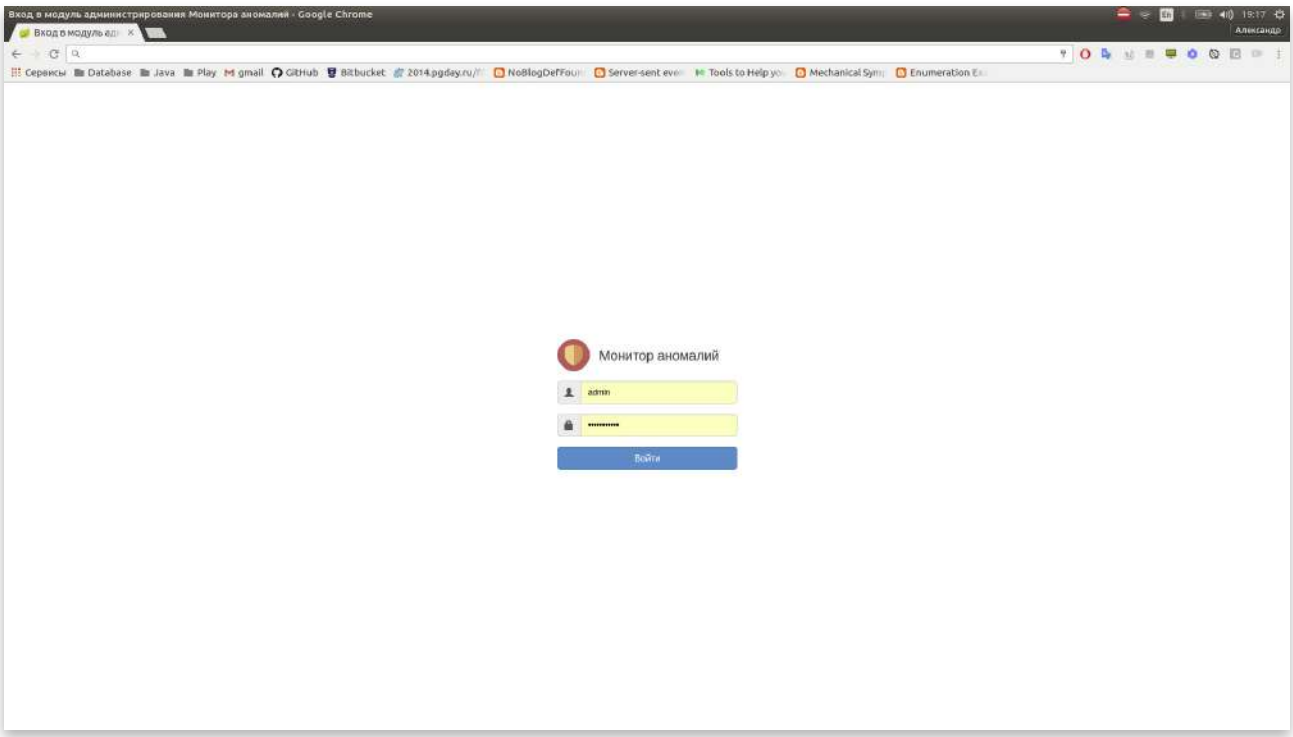
Design and development of an administrator's submodule

To keep setup and monitoring processes simple, we designed and developed the submodule with UI. Despite the fact that our Client didn't expect getting this improvement, it turned out to be very useful. This module allows the client to monitor event logs and security logs and setup the Anomalies Monitor's subsystems.

QA

To ensure a top quality project, we implemented different types of testing.

For **Integration testing** we developed the testing utility (named scripts imitator). This approach saved time and resources that are normally spent on manual testing. It also allowed us to add the testing into Continuous Integration.



Integration tests may be developed for all of the module's scripts. Input data for every test was parameterized and saved into a file. When a test was launched, the system generated the data from the file. After the required amount of data was generated, the real results were checked and compared with expected results (which were also saved into the file).

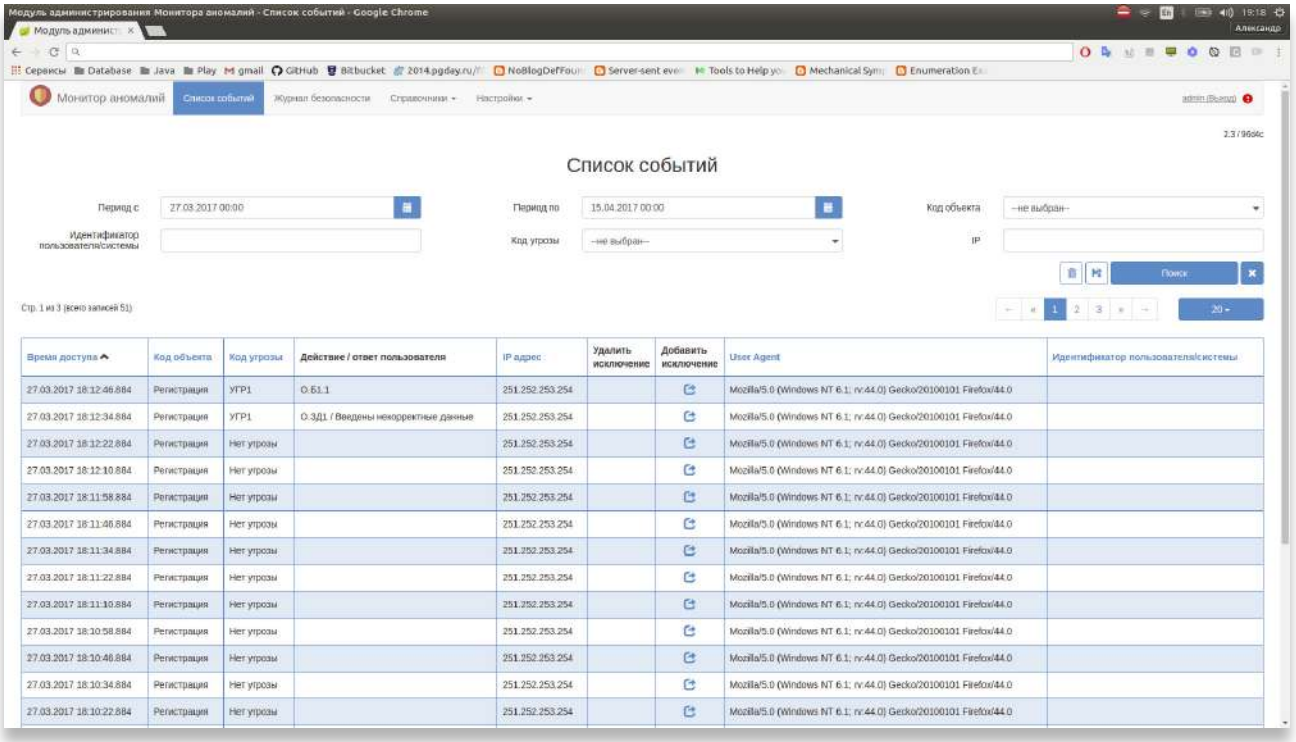
Anomaly Monitor subsystem is designed to handle a huge amount of input data, so we meticulously **tested the load and performance**. We created auto tests to determine the amount of correctly processed requests and to see the time spent on request processing.

To load the system and to test different aspects of it, we used the same testing utility. It can show if the algorithm functions correctly, it also generates the data flow to test the performance.

To create data overload, this utility "attacks" the service with REST-requests in the multithreading mode. This behavior can be configured by the XLS-file that contains settings for every threat detection algorithm.

The utility uses the thread pool. The interval running of thread pools is also provided.

The part of the utility which is responsible for Anomaly Monitor's load, uses several thread pools. Interactions are executed according to the order; this keeps the memory and processor free of ongoing inquiries, while the pool which is responsible for sending inquiries doesn't stay idle. We optimized weak points detected while testing (mostly database issues). We optimized inquiries, cleaned up the max amount of join tables and slow inquiries. We created covering indices and simplified data types to reduce the table's size.



The test also showed that, while the Anomaly Monitor runs constantly, it accumulates statistic data too fast, so the base grows, and the inquiries get slower. To avoid these problems, we optimized the data keeping model, adding the deletion of useless statistical data. Automated monthly partitioning was used for big tables.

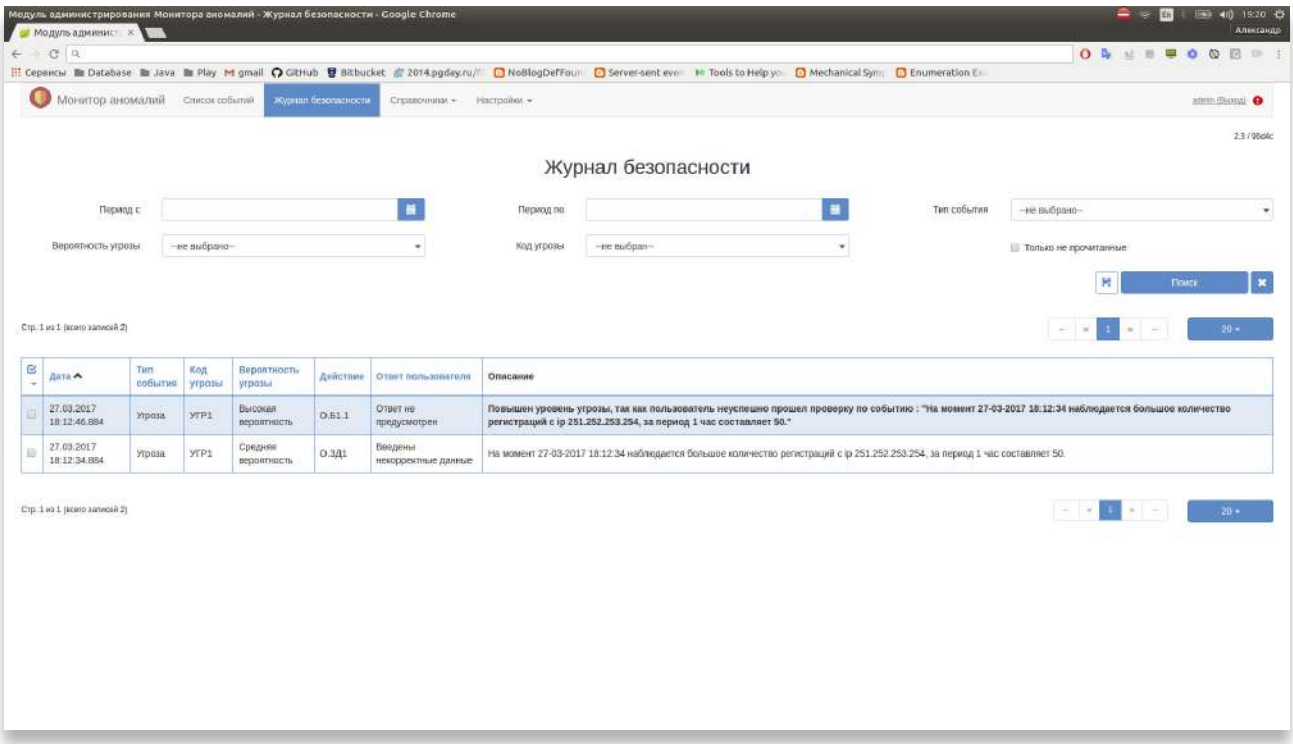
As a result, we achieved **performance and load that were even better than our customer expected**, so there was potential to grow the flow of processed data and to grow the Monitor's functionality.

To assure the quality of the system's logic we used **functional testing** with the SoapUI tool. Also, the automated tests were developed, which were sending REST-requests with specific parameters.

To test the administration module with a UI we used **manual functional testing** according to the patterns developed in the analysis stage of testing procedures.

UI automated testing. We used Selenium to create the main testing scripts, the scripts run every time a new project is assembled.

Installation testing. We tested the installation of the system on physically different processors with and without load balancing utility. Results proved that the system behaves just as expected, and there's no need to remaster it.



Documentation

We prepared the project documentation according to GOST (Russian standards for technical documentation).

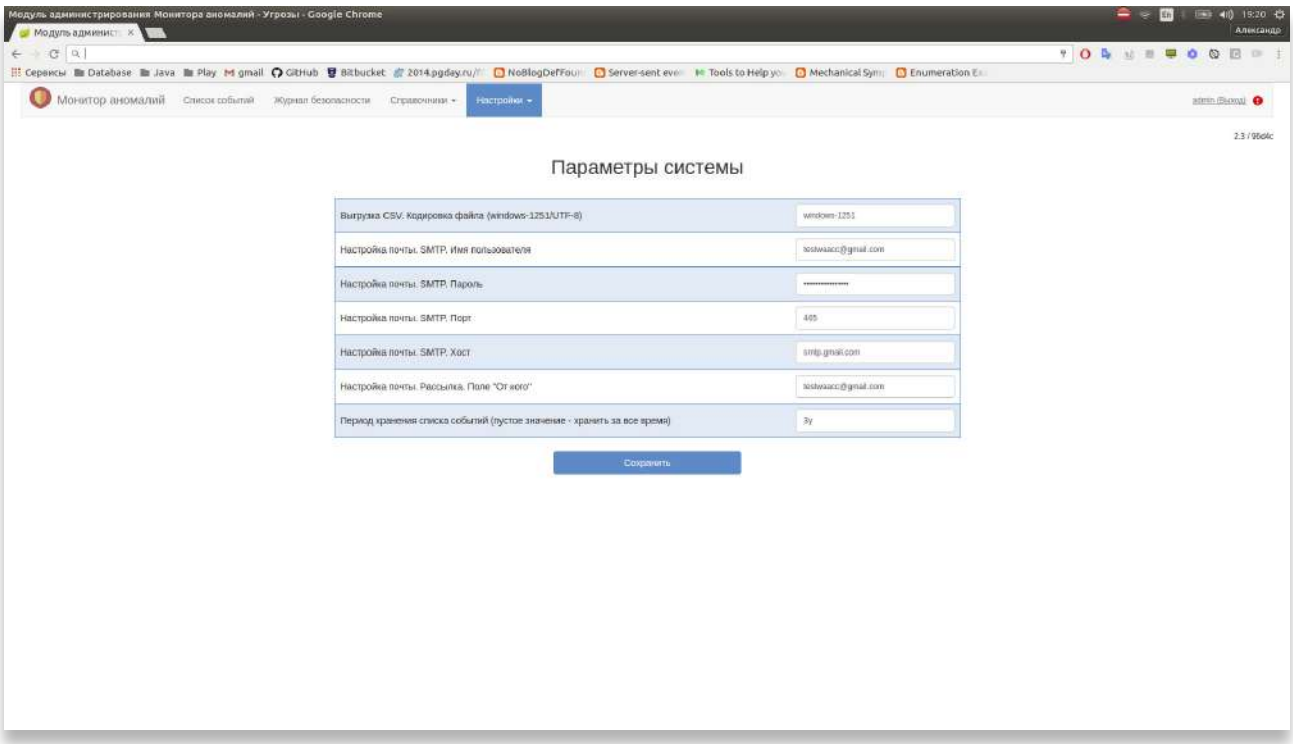
Technologies

1. **Backend:** Java 8, Spring 4, Spring Boot 1.4, JDBC Template, jFuzzyLogic 1.3, FlywayDB, ThymeLeaf, OpenCSV
2. **Client part:** Bootstrap, JQuery
3. **Database:** PostgreSQL 9.4+
4. **Testing:** JUnit, Selenium + PhantomJS driver, JXLS Reader, SoapUI.

Result

We developed the Anomaly Monitor subsystem that provides the following functions:

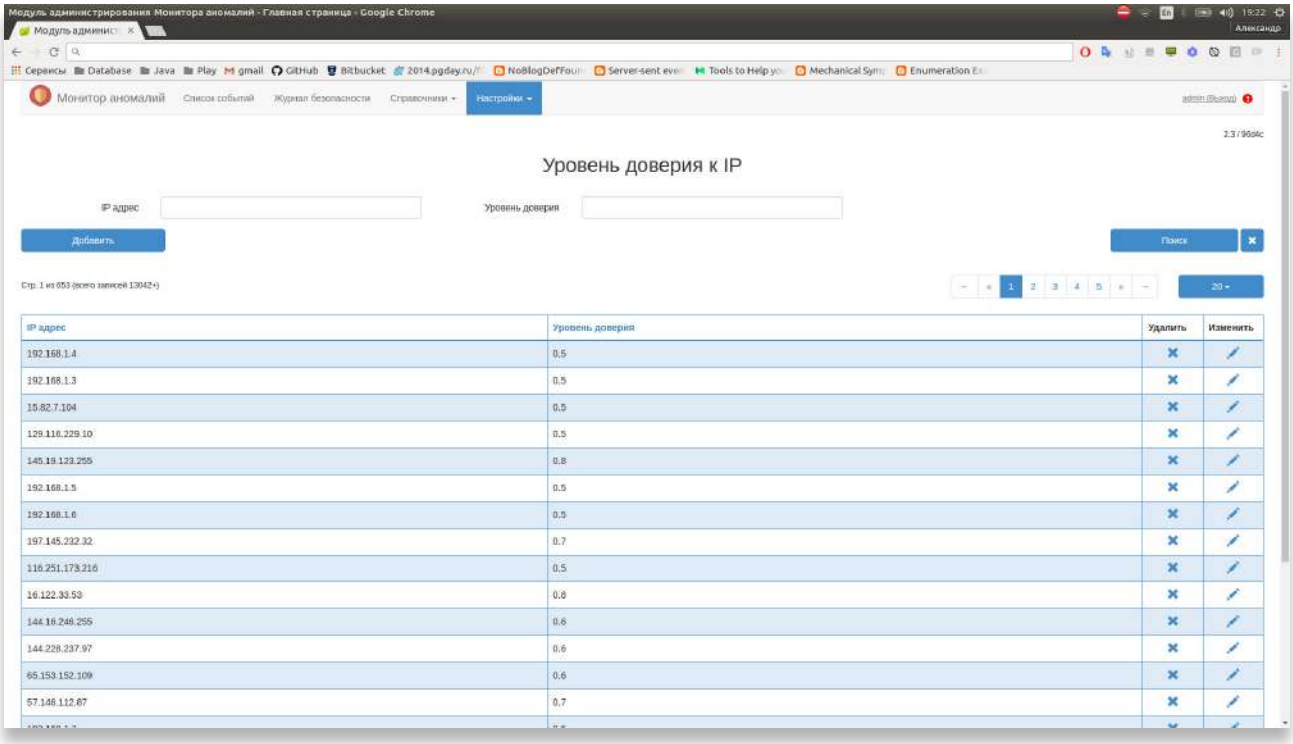
- Data monitoring and gathering;
- Keeping of gathered data, intermediate results, and service data;
- Sorting of threads' attributes and analyzing threads probability;
- Determining the actions in response of an incident.



The testing utility was also developed and given to our customer. It will simulate the load on the subsystem and help to implement integral testing of the main module using templates.

The main subsystem has passed all our internal tests. Our client has already started to use it in a trial mode, and plan to implement it in practice in the near future.

Meanwhile we are working on the second stage of the project: self-learning algorithms which are able to analyze threads' probability and choose a reaction according to this probability.



Customer's testimonial

«WaveAccess accomplished the project on schedule, and with exceptional quality, despite changing our requirements several times after the project had started. We appreciate their ability to deliver a high performance product, and to exceed our every expectation.

WaveAccess was always there to help. Developers, analysts and managers answered all of our questions, when we started to test the deployed project. The managers informed us about changes in the tasks' states right away, so we were always aware of any changes throughout the entire process.»

WaveAccess
We Make IT Easy

If you would like to evaluate the capabilities of machine learning for your project, we will be happy to share our experience:

Tel.: +1 866 311 24 67

E-mail: hello@wave-access.com

Read more at

www.wave-access.com