Kivonat

A szokásostól eltérő adatpontok, ún. anomáliák automatikus detektálása számos szakterületen kiemelt jelentőségű, hiszen a ritkán előforduló, a többiektől viselkedésükben különböző megfigyelések hátterében gyakran veszélyes jelenségek állnak, amelyek számos esetben (például pénzügyi szektor csalásait vagy egy számítógépes infrastruktúra betörési kísérleteinél) nagy anyagi vagy erkölcsi károkat okoznak.

Ezeken a szakterületeken az adatpontok időbeli kapcsolata hangsúlyos, emiatt valós idejű kiértékelés válik szükségessé. Ezekben az esetekben tehát adatfolyamokon értelmezett, online anomália detektálási módszerek használata gyakori.

Különböző **detektáló algoritmusok** különböző feltételezésekkel élnek az anomáliák jellemzőire vonatkozóan, emiatt más és más módszerekkel próbálják megtalálni ezeket a furcsa eseményeket. Jelen szakdolgozat célja az irodalomban fellelhető, jellemzően használt algoritmusok vizsgálata.

A szakdolgozatban emiatt részletesen ismertetek több anomáliadetektáló algoritmust, melyek közül hármat Pythonban implementáltam és az **Apache Storm** adatfolyam feldolgozó környezetbe integráltam. Ez a három leimplementált algoritmus a távolság alapú **Exact-Storm** és a klaszterezési elemzést használó **Korm** és **DenStream**.

A szakdolgozatban bemutatom az általam felépített kísérleti környezetet, mely adatfolyam feldolgozási keretrendszerként az Apache Stormot használja, az algoritmusok eredményeinek valós idejű ábrázolását egy kliens oldali vizualizációs könyvtár (**Bokeh)** és a kettő közötti köztes kapcsolatot pedig egy Python alapú webszerver (**Web.py)** valósítja meg.

Az algoritmusok eredményeinek vizualizációjára párhuzamos koordináta és szórásdiagramot használtam.

Összeállítottam egy, az algoritmusok alapvető összehasonlításra alkalmas reprezentatív szintetikus adatkészlet halmazt és ezeken vizsgáltam az implementált algoritmusok detektálási sikerességét. Az algoritmusok használhatóságát egy, a pénzügyi szektorból származó éles adathalmazon vizsgáltam meg.

Abstract

The automatic detection of abnormal data points, called anomalies, is a priority in many specialities. The main cause of these rarely occurring abnormal observations are mostly dangerous events, which cause big financial or moral damage (like financial scams or attempted computer infrastructure break ins).

In these specialities the time relation of the data points is emphasised, and real-time evaluation becomes necessary. So the method of online anomaly detection over data streams is commonly used.

Different **detection algorithms** have different assumptions regarding the nature of the anomalies, therefore these algorithms have different methods to identify these strange events. The goal of this thesis is to examine the typically used algorithms found in the scientific literature.

So in this thesis I will describe several anomaly detection algorithms in detail. From the several algorithms I implemented three using Python and integrated them into the **Apache Storm** stream processing environment. The three implemented algorithms are the distance based **Exact-Storm** and the cluster analysis using **Korm** and **DenStream**.

In this thesis, I will present the experimental environment I have built, which uses the Apache Storm for stream processing, uses a client side visualization library (**Bokeh**) to represent the results of the algorithms in real-time and uses a Python based webserver library (**Web.py**) as the intermediate connection of these two.

To visualize the results of the algorithms I have used scatter plot and parallel coordinate diagram.

I have produced several representative synthetic data sets and used them to compare the success of detection of the implemented algorithms. I have examined the usability of the algorithms on a live financial data set.