An Analysis Framework for Product Prices and Supplies in Darknet Marketplaces

York Yannikos
Fraunhofer Institute for Secure Information Technology SIT
Darmstadt, Germany
york.yannikos@sit.fraunhofer.de

Julian Heeger
Fraunhofer Institute for Secure Information Technology SIT
Darmstadt, Germany
julian.heeger@sit.fraunhofer.de

Maria Brockmeyer
Technische Universität Darmstadt
Darmstadt, Germany
maria.seidens@stud.tu-darmstadt.de

ABSTRACT

Darknet marketplaces are an interesting research area. Most marketplaces are hosted in the Tor network as an anonymous onion service. In this paper we provide a generic approach to build a framework for the collection and analysis of product prices and supplies in such marketplaces. We focus on the technical details how to implement such a framework and how to collect, organize, and analyze the product data. For our framework implementation we provide an evaluation based on collected data from three large marketplaces and present an approach to enrich the analyzed data with additional information from external sources.

KEYWORDS

Darknet marketplaces; analysis framework; web scraping

ACM Reference Format:
York Yannikos, Julian Heeger, and Maria Brockmeyer. 2019. An Analysis Framework for Product Prices and Supplies in Darknet Marketplaces. In *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES 2019) (ARES '19), August 26–29, 2019, Canterbury, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3339252.3341485

1 INTRODUCTION

The Tor network, in the media commonly used as a synonym for the *darknet*, is (in)famous for its onion services (previously called hidden services). This technology allows any participant in the Tor network to host a service in an anonymous way without revealing the location of the host. Tor participants quickly started using onion services to create marketplaces where various mostly illegal goods can be bought and sold anonymously. After the first version of the very popular marketplace *Silk Road* caught media attention and was taken down eventually, the number of darknet marketplaces increased.

Previous scientific research has been done regarding the type, amount, and quality of goods sold on darknet marketplaces. Also bitcoin, the most popular cryptocurrency used on these marketplaces, was subject to comprehensive previous research. However, due to the anonymity Tor provides, it is difficult to passively gather more information about transactions on marketplaces without participating as buyer or seller. For example, it is difficult to learn about what events outside of the darknet may influence product sales on the marketplaces, i. e. driving product prices or supplies. Such additional information could be relevant for researchers and investigators but are usually not available and must be gathered manually.

Contribution

In this paper we propose a framework that helps with the analysis of product prices and supplies on darknet marketplaces. More precisely, the framework combines the automatic collection of product data on marketplaces (e. g. type of goods, supply, price, date), the analysis of the collected price and supply data (e. g. regarding outliers), and an automatic approach to correlate analysis results with information from external sources, i. e. news websites. We describe our framework architecture and its technical components in detail and provide an evaluation of our framework implementation with, in our opinion, promising results.

Outline

This paper is structured as follows: In Section 2 we provide a short introduction to Tor and its onion services. Section 3 gives an overview about Tor marketplaces and their popularity. In Section 4 we summarize current research on Tor marketplaces with similar research questions. In Section 5 we describe the architecture of a framework for product data analysis. Section 6 describes our framework implementation and Section 7 provides the results of our evaluation. Section 8 gives a conclusion and mentions further research possibilities.

2 TOR AND ONION SERVICES

The Onion Router or *Tor* is an overlay network that provides anonymity for its participants. Tor relies on *onion routing*, a smart way of routing packets combined with several layers of encryption [5]. Onion routing ensures that every packet in the Tor network is routed through a predefined path with a minimum number of three nodes such that the source node of a packet is never directly connected to the destination node. This routing path is called Tor circuit. Before sending a packet through a Tor circuit, one separate layer of asymmetric encryption is added for each node, effectively building an onion-like packet wrapped in multiple layers of encryption. This ensures that the nodes the packet passes can each only decrypt the outermost layer to learn where the packet should

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ARES '19, August 26–29, 2019, Canterbury, United Kingdom
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7164-3/19/08…$15.00
https://doi.org/10.1145/3339252.3341485

be sent to next. Additionally, the Tor nodes are not aware of the purpose of the other nodes in the circuit. Therefore, even the last node in a circuit does not know that it is sending the packet to its final destination (exceptions are Tor exit/entry nodes that are directly connected to the destination/source of a packet). Tor changes circuits every 10 minutes for new connections.

By using Tor, participants can either access a destination in the clearnet (outside of Tor) in an anonymous way or they can anonymously access specific services within the Tor network called *onion services*. Onion services (previously called *hidden services*) are Tor nodes that typically host websites (although every other service can also be hosted, e. g. IRC, FTP, SMTP servers, etc.) and can only be reached via Tor. This is explicitly done by using *.onion* addresses with a special top-level domain DNS servers cannot resolve. Tor however knows how to resolve these addresses and starts building the required circuits to contact the requested onion service.

3 TOR MARKETPLACES

Because onion services allow anonymous hosting without disclosing the location or other information about the owner of the service, they quickly started to gain popularity. While people that lived in countries with oppressive regimes found a way to freely communicate without fearing consequences, others started setting up marketplaces to buy and sell all kinds of illegal goods. Since the days of the first version of *Silk Road*, other large marketplaces went up, many of them being online for several years before silently going offline, presumably performing an exit scam, or being taken down by law enforcement agencies.

Some of the largest marketplaces listed on the darknet market index page *DeepDotWeb* [4] in 2018 were *Dream Market*, *Wall Street Market*, and *Tochka Market*. By now, both Dream Market and Wall Street Market are offline: Dream Market announced that they allegedly became victim of a DDoS attack, promising to restart soon with a new onion service [7]. Wall Street Market was taken down on May 3, 2019 as result of a coordinated operation of law enforcement agencies [11]. The same happened a few days later with the index page itself: DeepDotWeb was seized and taken offline on May 6–7, 2019 [8].

From a technical point of view, Tor marketplaces are typically built like a normal shopping websites in the clearnet, e. g. Amazon. Available product categories range from illegal drugs and pharmaceuticals to digital goods like stolen credit cards or PayPal accounts and service offers like website defacements or Facebook account overtakes. Although marketplaces typically share a similar structure regarding product categories, this is often not the case when looking at the underlying HTML structure. Therefore, for each marketplace that we want to collect data from, a separate web scraper needs to be built.

4 RELATED WORK

From 2013 to 2015, Branwen scraped a large amount of data from 89 Tor marketplaces and made this collection available for the public [1]. Since then many researchers took a closer look on the data. In 2017, Broséus *et al.* did a comprehensive analysis of the data regarding global trafficking on darknet marketplace [2]. They analyzed the scraped data regarding type of goods, originating countries, and specialization of the dealers. Also in 2017, Décary-Hétu and Giommoni published their work regarding whether or not police crackdowns disrupt the darknet drug market [3]. The authors found that *Operation Onymous*, a coordinated global operation of law enforcement agencies against darknet marketplaces, did indeed affect the darknet drug market, but only for a short time. Laadegard did research on the influence of media reports about darknet marketplaces, police operations, or criminal convictions on darknet drug markets [12]. He scraped data from large marketplaces like *Agora* and *Evolution* and analyzed news articles especially regarding the case of Ross Ulbricht, also known as "Dread Pirate Roberts", the founder of *Silk Road*. His results showed that there was an increase of trade activity on the darknet marketplaces after the conviction and severe sentencing of Ulbricht. The author concluded that the media coverage significantly contributed to the increased trades. Ladegaard's approach of scraping news articles greatly inspired our own work on the analysis component of our framework which is described in the following section. In 2018, we collected product data and bitcoin wallet addresses from several small darknet shops. By analyzing the bitcoin transactions for the found wallet addresses, we were able to draw conclusions about the type of goods were sold [21]. However, we did not consider larger marketplaces with many different dealers and customers at this time.

5 FRAMEWORK ARCHITECTURE

In the following we describe a generic framework architecture for the collection and analysis of product prices and supply on Tor marketplaces. We divided the framework into three main components: one for data collection, one for data management and storage, and one for analysis tasks. The data collection component heavily relies on web scraping and crawling technology. The data management and storage component provides tools to store, access, and export the data in a convenient way. The analysis component provides methods for statistical analysis of the collected data and for enrichment with information from external sources.

5.1 Collection of Product Data

To efficiently collect relevant data from Tor marketplaces, we utilize web scraping. Web scraping in general is a relatively easy task due to the large number of available tools and libraries that include efficient implementations of most of the needed functionality. For our data collection component we formulate the following list of requirements:

(1) Supporting all types of HTTP requests (GET, POST, PUT, HEAD, ...)
(2) Extraction of relevant elements from HTML DOM
(3) Taking screenshots of arbitrary websites
(4) Multithreading (for efficiency reasons)
(5) SOCKS proxy support (Tor)

Basic web scraping libraries usually provide functionality to send HTTP requests with arbitrary header data and payload. Therefore, meeting requirement 1 is relatively easy. The same holds for requirement 2: After downloading a HTML website, its content can be easily parsed using one of the many existing XML parsers. Query languages like XPath provide a convenient way of selecting relevant elements of the HTML DOM.

Requirement 3 can be met with additional use of web testing frameworks like Selenium that provide a full HTML rendering engine [14]. Therefore, Selenium is able to take screenshots and simulate user interactions (e. g. clicking buttons, typing text, etc.) on scraped websites. It also allows scraping websites that extensively make use of JavaScript libraries to provide dynamic content (however, JavaScript is rarely used on Tor marketplaces due to security concerns).

We also require support for multithreading (requirement 4) because this significantly speeds up any scraping tasks: The list of web pages to be scraped (i. e. different Tor marketplaces, product categories, subcategories) is available in advance and can be processed in parallel instead of sequentially. And even if a web scraping library or framework does not support multithreading, it can usually be implemented with only little overhead.

SOCKS proxy support (requirement 5) is necessary to connect to Tor.

5.2 Data Storage and Management

For storage and management of the collected data we formulate the following requirements:

(1) Easily extensible
(2) Very good performance, stability, and scaling even with large datasets
(3) Able to handle text and binary data (e. g. product pictures)
(4) Data filtering, cleansing, and normalization
(5) Simple and effective data access (e. g. search, filtering)
(6) Data export to standard data formats

Although many different alternatives exist for storing unstructured data like web pages, we felt that an relational database management system like MySQL or PostgreSQL provides the best overall solution that meets the requirements 1–3, especially regarding stability.

Tools that satisfy requirement 4 heavily depend on the actual data that was collected. For instance, if some products on a Tor marketplace are offered with a price in U.S. dollar (USD) and others in Bitcoin (BTC), the prices have to be converted to a single currency to make them comparable. Another example are badly written product texts that need additional filtering to correctly find unit size, product type, and so on. Many of these tools can be implemented using regular expressions.

For data management (requirement 5–6) we use a RESTful API. This allows to hide complicated database queries behind a predefined set of simple HTTP calls providing functionality like search, filtering, or export of the collected data.

5.3 Analysis of Product Data

For the analysis of product data we formulate the following requirements:

(1) Statistical methods for outlier detection
(2) Graphical user interface for visual analysis
(3) Collection of additional information through external sources

For an automatic analysis approach we apply outlier detection on the collected data and use a toolset that enriches found outliers with helpful additional information from external sources. Outliers can be detected in various ways with different tolerance and accuracy. Basic statistical methods like regression analysis can already provide sufficient results that are worth further looking into.

To find helpful information for identified outliers we consider scraping external sources that provide possibly relevant information like news websites. The idea is the following: If we find an outlier in the product data, e. g. a jump in cocaine prices starting from a specific day, we search news websites for articles that were published around this time and are possibly related to this drug, seller, and/or Tor marketplace. This is closely related to the work done by Ladegaard that we mentioned earlier in Section 4, however we focused on searching news articles in a fully automated way and only on demand, i. e. when an outlier was found.

Besides collecting news articles using web scraping, there are also specific news APIs available that provide more convenient access to news data, e. g. with detailed search and filtering functionalities [18].

6 FRAMEWORK IMPLEMENTATION

In the following we describe our framework implementation in Python. For the different components (data collection, storage/management, and analysis) we used the following libraries:

Data Collection Component. We implemented the web scraping using Selenium and Python's requests library with SOCKS proxy support [17]. For multithreading we implemented a message queue using RabbitMQ [13] that handles all URLs of the different marketplaces with their categories and subcategories. For each data collection task we start 10 web scraping threads. Each thread takes one URL from the queue, performs the corresponding scraping task, and posts the scraped content to our RESTful API (see data storage/management component) in order to store it in our database. To take screenshots of specific offers we start a Selenium instance on demand. Using these libraries and tools, we could satisfy our requirements mentioned in Section 5.1.

Data Storage/Management Component. For our data storage we used PostgreSQL with its Python bindings. Since we wanted to work with the collected data in a simple and effective way, we implemented a RESTful API using Python's Flask framework [15]. By defining API endpoints in advance for operations like storing data in the database or searching in the stored data, we removed the need to write complex database queries for later operations, using simple HTTP requests instead. By that we satisfied our requirements mentioned in Section 5.2.

Data Analysis Component. To perform statistical analysis tasks on the collected data we utilized Python's Pandas library [20] and implemented a visualization of the results using Chart.js [19]. For the visualization of basic statistics about the collected data we used Redash [16]. To enrich our analysis results (i. e. detected outliers in price of supply data) with additional information, we utilized the website NewsRiver [18]: NewsRiver crawls more than 500,000 daily news articles from many newspapers and blogs and provides convenient access to its database via its news API. Using the request library and JSON parsing, we could efficiently search through many news articles to find useful additional information. Therefore, we met the requirements we formulated in Section 5.3.

Figure 1 provides an overview about our framework implementation with the corresponding libraries and tools used per component. Figure 2 depicts a screenshot of our framework showing statistics of the collected data using Redash.

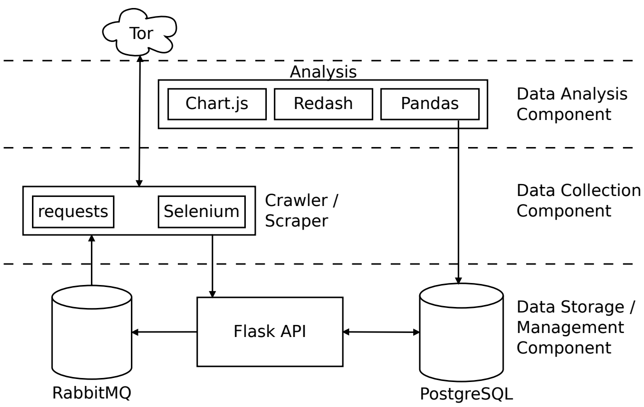

Figure 1: Overview of the framework implementation

7 EVALUATION

To evaluate our framework implementation we took three of the largest Tor marketplaces at that time (according to DeepDotWeb): *Dream Market*, *Wall Street Market*, and *Toshka Market*. We looked into their product categories and found that illegal drugs were by far the most offered products. This was not really surprising to us as this has often been observed before by other researchers [1, 6]. Therefore, we decided to crawl the product data of illegal drugs offered on these three marketplaces. We performed the data collection over a duration of a little more than 7 weeks, starting from September 18, 2018 until November 19, 2018.

Table 1: Number of data sets of illegal drugs offers on three Tor marketplaces (including recurrent offers)

Tor marketplace	Number of offers
Dream Market	2,184,413
Wall Street Market	273,757
Tochka Market	80,591
Total	2,538,761

Table 1 shows the total number of the data sets we collected from all product offers on the three marketplaces. The numbers include duplicates, i. e. we counted each visit of our web scraper on the same product offer separately. As we can see, the most data sets we could collect came from Dream Market (more than 86 % of all collected data sets).

Table 2 provides a closer view on what kind of drugs were offered on all three marketplaces. Almost one third of all data sets we collected came from offers of weed and cocaine. Therefore, we decided to look closer into these two drug categories.

Table 2: Number of data sets in different categories of illegal drugs (including recurrent offers)

Illegal drug type	Number of offers
Weed	550,142
Cocaine	267,677
Opiates	239,647
Ecstasy	215,965
MDMA	210,806
Hashish	190,390
Steroids	185,821
Benzos	174,911
Concentrates	169,679
LSD	137,202
Speed	108,623
Crystal Meth	87,898
Total	2,538,761

Table 3 shows the amount of weed offers per marketplace, again with Dream Market providing almost 80 % of all data sets. In Table 4 the amount of cocaine offers per marketplace is shown. While more than 92 % of the data was collected from Dream Market, we found that Wall Street Market provided four times more data sets compared to Tochka Market. Compared with the weed offers, cocaine seemed to be a lot less popular on all three marketplaces.

Interestingly, the number of unique vendors offering weed and cocaine did not differ much, as shown in Table 5 and 6: The total number of weed vendors we collected data from is only 40 percent higher than the number of cocaine vendors[1].

Table 3: Number of data sets of weed offers on three Tor marketplaces (including recurrent offers)

Tor marketplace	Number of offers
Dream Market	437,563
Wall Street Market	61,253
Tochka Market	51,326
Total	550,142

Table 4: Number of data sets of cocaine offers on three Tor marketplaces (including recurrent offers)

Tor marketplace	Number of offers
Dream Market	247,163
Wall Street Market	16,356
Tochka Market	4,158
Total	267,677

[1]The total number of unique vendors on the three marketplaces in Table 5 and 6 is lower than the sum since several vendors offered drugs on more than one marketplace.

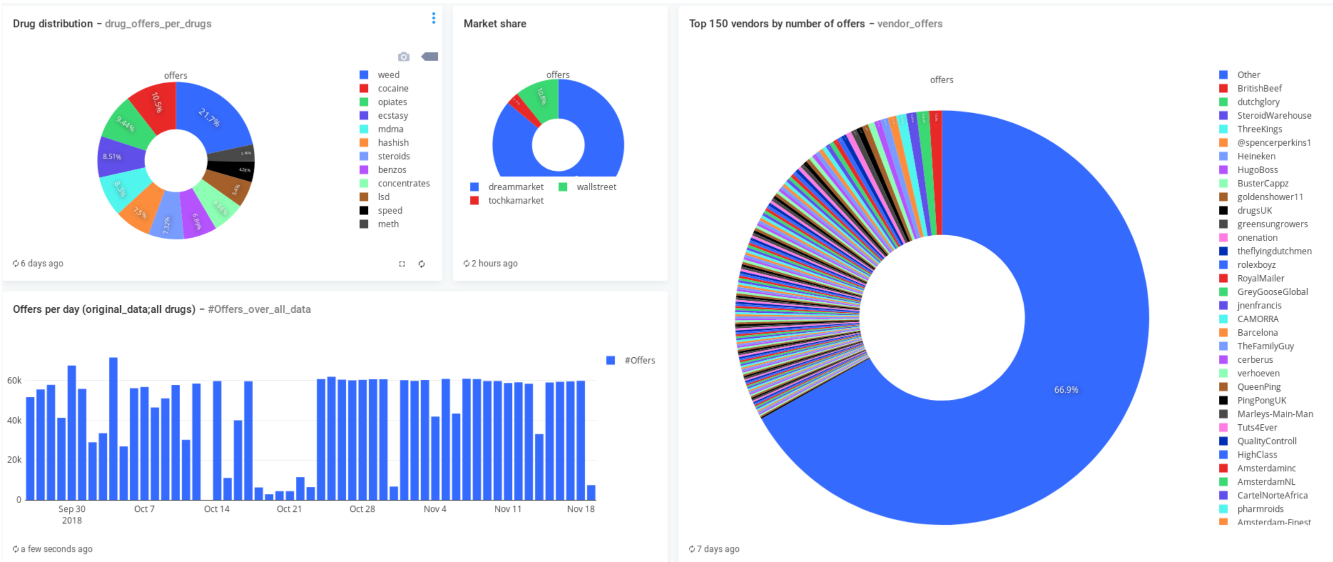

Figure 2: Screenshot of the framework GUI showing statistics of collected data

Table 5: Number of (unique) weed vendors on three Tor marketplaces

Tor marketplace	Number of offers
Dream Market	993
Wall Street Market	454
Tochka Market	149
Total	1,497

Table 6: Number of (unique) cocaine vendors on three Tor marketplaces

Tor marketplace	Number of offers
Dream Market	792
Wall Street Market	240
Tochka Market	107
Total	1,069

To further evaluate the analysis component, we randomly selected offers of weed and cocaine and conducted a more detailed analysis: For each offer we plotted the price changes over time and performed a rolling regression analysis. We then calculated the residuals and the residual standard deviation. For our analysis we defined that all residuals greater than the standard deviation are considered as outliers. For each outlier we performed an automatic search using the news API of NewsRiver to find events that caused media attention and could possibly be related to our observed price change/outlier.

Figures 3 and 4 show the data points with regression line and residuals for two specific offers where we identified outliers. For the weed offer with the price data shown in Figure 3, we identified two outliers, one at October 17, 2018 and one at October 25, 2018. Our automated news API scraper was able to find an article by the Regina Leader Post around this date with the following title: *"First day of legalization quiet for police, busy for cannabis stores"* [10].

Since the vendor offering the weed mentioned that he/she is located somewhere in Canada, we believe it is reasonable to consider this search result as interesting information that could possibly be related to the price change: Canada legalized cannabis on October 17, 2018. This correlates with the price drop we observed, but, of course, we cannot conclude that the observed price change is really caused by the mentioned event.

For the cocaine offer depicted in Figure 4 we identified an outlier in our dataset at September 25, 2018. The automated search via the news API resulted in a news article by the Daily Mail with the following title: *"Cocaine worth $18M is discovered inside shipment of bananas donated to Texas prison system"* [9].

Regarding the vendor description of the offer we analyzed, the vendor's location was in the U.S.. We believe that this news is also reasonable to consider as interesting additional information.

8 CONCLUSION

In this paper we proposed an approach to a generic analysis framework for the collection and analysis of product data from Tor marketplaces. We described the general architecture of the framework and explained the functionality of the three framework components we use to collect, store, manage, and analyze the data. We then implemented the framework using Python tools and libraries for web scraping and analysis, a PostgreSQL database, and a GUI based on Redash and Chart.js to perform and visualize basic statistics and analysis results. After collecting data from three large Tor marketplaces over 7 weeks, we provided information about the different amount of goods offered on these marketplaces, mainly focusing on illegal drugs. For several offers we conducted a more detailed

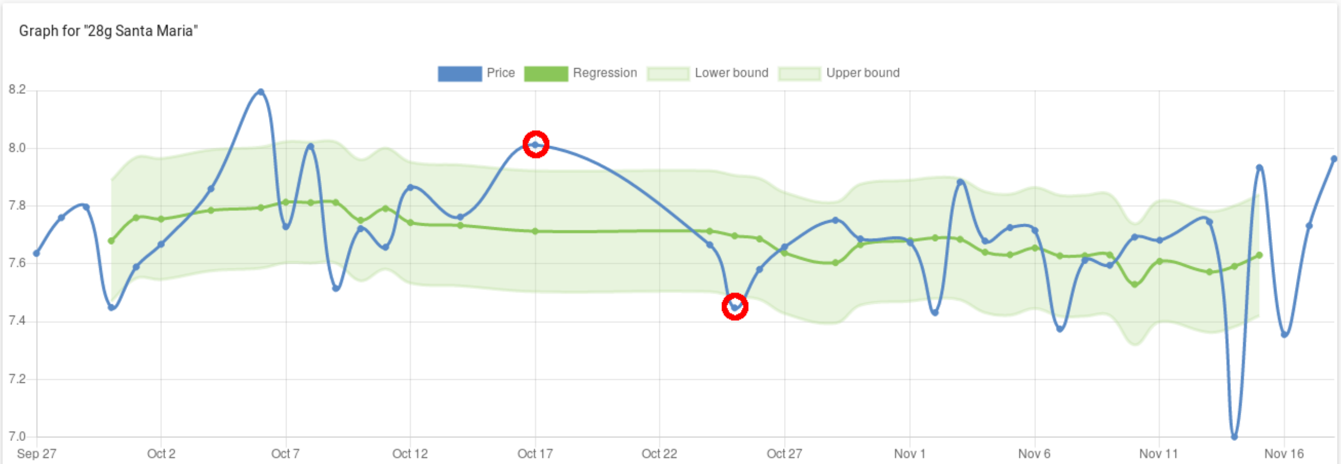

Figure 3: Identified outliers of a weed offer based on rolling window regression

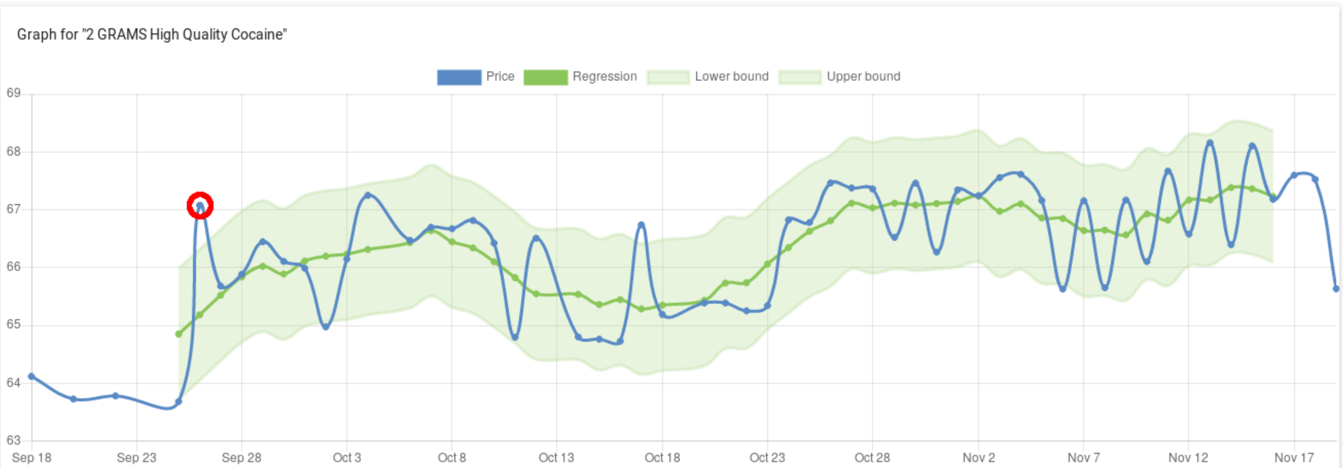

Figure 4: Identified outliers of a cocaine offer based on rolling window regression

analysis based on a rolling-window regression, identified outliers, and performed a search on a large dataset of news articles to find events that were possibly related to these outliers.

Although the duration of our data collection was rather short, we believe that we could provide an interesting view on the products offered on mentioned Tor marketplaces. While two of these marketplaces, Wall Street Market and Dream Market, are now offline (probably forever), we are confident that they will be replaced by many other marketplaces in the future. Therefore, we plan to conduct further research including other product categories and to extend our framework with additional features for automated analysis tasks. We also plan to implement the framework components as microservices using container technology like Docker for easier scaling and deployment.

ACKNOWLEDGMENTS

We would like to thank the members of the PANDA project (supported by the German Federal Ministry of Education and Research (BMBF) – https://panda-projekt.de) for all the discussions and comments that improved this work.

REFERENCES

[1] Gwern Branwen. 2019. Darknet Market Archives (2013–2015). https://www.gwern.net/DNM-archives
[2] Julian Broséus, Damien Rhumorbarbe, Marie Morelato, Ludovic Staehli, and Quentin Rossy. 2017. A geographical analysis of trafficking on a popular darknet market. *Forensic science international* 277 (2017), 88–102.
[3] David Décary-Hétu and Luca Giommoni. 2017. Do police crackdowns disrupt drug cryptomarkets? A longitudinal analysis of the effects of Operation Onymous. *Crime, Law and Social Change* 67, 1 (2017), 55–75.
[4] DeepDotWeb.com Team. 2018. DeepDotWeb. https://www.deepdotweb.com/
[5] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. *Tor: The second-generation onion router.* Technical Report. Naval Research Lab Washington DC.
[6] The Economist. 2016. Shedding light on the dark web. https://www.economist.com/international/2016/07/16/shedding-light-on-the-dark-web

[7] Kai Sedgwick for Bitcoin.com. 2019. The Darknet's Largest Marketplace Is Closing – But a Replacement Is on Its Way. https://news.bitcoin.com/the-darknets-largest-marketplace-is-closing-but-a-replacement-is-on-its-way/

[8] David Hundeyin for CNN. 2019. FBI Dismantles Deep Dot Web in Seismic Dark Market Platform Bust. https://www.ccn.com/fbi-dismantles-deep-dot-web-in-seismic-dark-market-platform-bust

[9] Megan Sheets for Daily Mail. 2018. Cocaine worth $18M is discovered inside shipment of bananas donated to Texas prison system. https://www.dailymail.co.uk/news/article-6198507

[10] Mark Melnychuk for Regina Leader Post. 2018. First day of legalization quiet for police, busy for cannabis stores. https://leaderpost.com/news/local-news/first-day-of-legalization-quiet-for-police-busy-for-cannabis-stores

[11] Brian Krebs. [n. d.]. Feds Bust Up Dark Web Hub Wall Street Market. https://krebsonsecurity.com/2019/05/feds-bust-up-dark-web-hub-wall-street-market/

[12] Isak Ladegaard. 2017. We know where you are, what you are doing and we will catch you: Testing deterrence theory in digital drug markets. *The British Journal of Criminology* 58, 2 (2017), 414–433.

[13] Pivotal. 2019. RabbitMQ. https://www.rabbitmq.com/

[14] The Selenium project. 2019. Selenium. https://www.seleniumhq.org/

[15] The Pallets Projects. 2019. Flask. http://flask.pocoo.org/

[16] Redash. 2019. Redash. https://redash.io/

[17] Kenneth Reitz. 2019. Requests. https://2.python-requests.org/en/master/

[18] NewsRiver Team. 2019. NewsRiver News API. https://newsriver.io/

[19] The Chart.js Team. 2019. Chart.js. https://www.chartjs.org/

[20] The Pandas Team. 2019. Pandas. https://pandas.pydata.org/

[21] York Yannikos, Annika Schäfer, and Martin Steinebach. 2018. Monitoring Product Sales in Darknet Shops. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, 59.