

FACHHOCHSCHULE NORDWESTSCHWEIZ  
DISKRETE STOCHASTIK, HS19

## **Bloom Filter**

BONUSAUFGABE

*Stefan Gruber*  
*Roger Kreienbühl*  
*Florian Thiévent*

4. Dezember 2019

# Inhaltsverzeichnis

<b>1</b>	<b>Idee des BloomFilters</b>	<b>2</b>
1.1	Vorteile . . . . .	2
1.2	Nachteile . . . . .	2
<b>2</b>	<b>Beispiel aus der Praxis</b>	<b>3</b>
2.1	Google Chrome . . . . .	3
<b>3</b>	<b>Testergebnisse der Implementierung</b>	<b>4</b>
3.1	Verfahren . . . . .	4
3.2	Resultate . . . . .	4

# Kapitel 1

## Idee des BloomFilters

### 1.1 Vorteile

Der Vorteil beim Einsatz eines Bloom Filters besteht darin, dass schnell und effizient geprüft werden kann ob ein bestimmter Wert vorhanden ist. Man kann also zum Beispiel eine Menge  $n$  an Wörtern im Filter speichern und schnell prüfen ob bei einer Eingabe das eingegebene Wort dem Filter entspricht. Nebst einer schnellen Abfrage ist auch der geringe Speicherplatz welcher ein Bloom Filter belegt einer der Vorteile.

### 1.2 Nachteile

Der grösste Nachteil beim Einsatz eines Bloom Filters sind die sogenannten false positive Matches. Es kann also nie ganz eindeutig bestimmt werden ob ein Wort im Filter enthalten ist.

# Kapitel 2

## Beispiel aus der Praxis

### 2.1 Google Chrome

Der weitverbreitete Browser Google Chrome benutzt Bloom Filter in seiner Malicious URL Implementierung. Dabei werden URL's die von Usern eingegeben werden durch die Browser Engine geprüft und bei einem positiven Match der User mittels einer Meldung darauf aufmerksam gemacht.

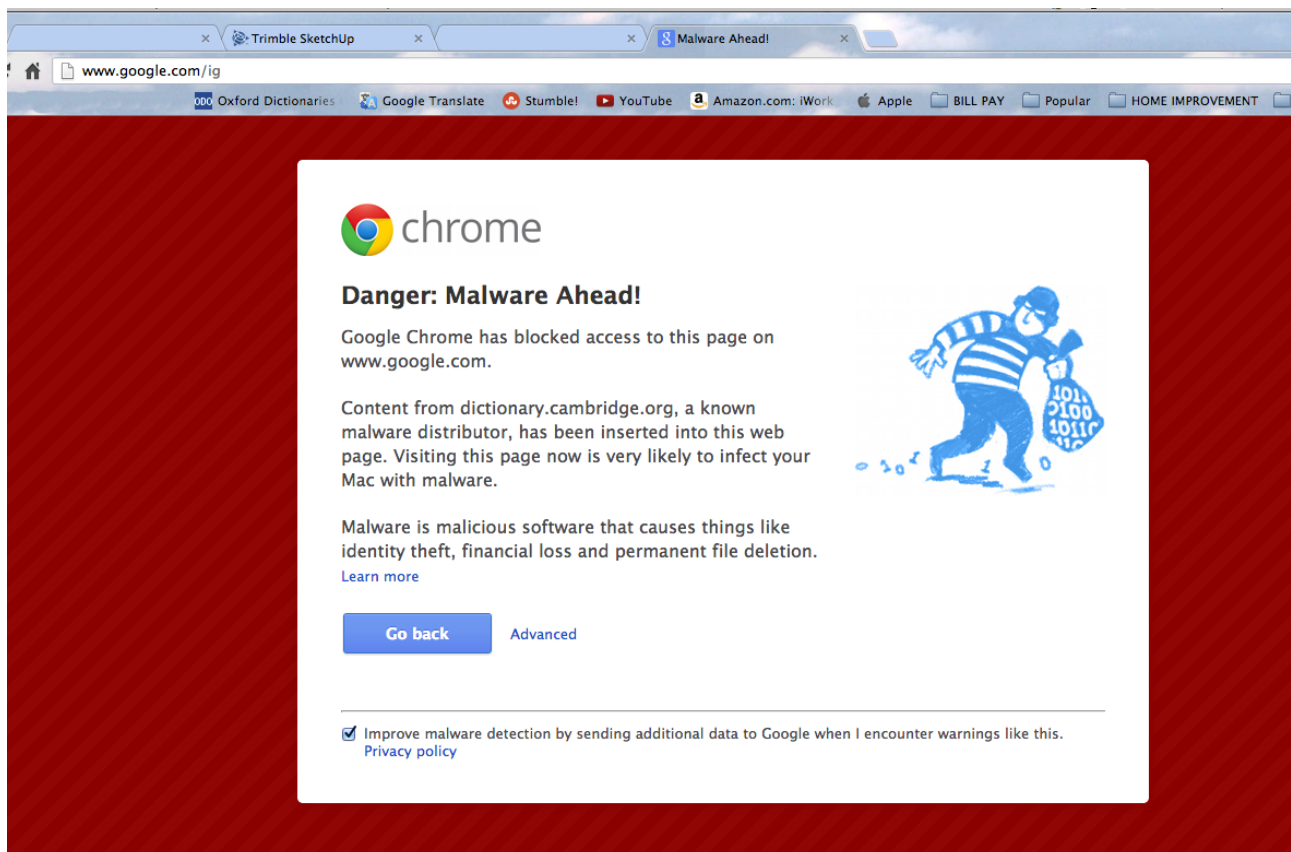


Abbildung 2.1: Google Chrome Malware Hinweis

Die Verwendung eines Bloom Filters bietet sich hier an, da der verwendete Speicherplatz sehr klein gehalten werden kann und die Kommunikation zur Malicious URL API von Google damit einen enorm kleinen Footprint hat. Das bedeutet, dass die Abgleiche auch bei einer langsamen Internetverbindung performant durchgeführt werden können und die User Experience nicht merklich beeinträchtigen.

# Kapitel 3

## Testergebnisse der Implementierung

### 3.1 Verfahren

Um den BloomFilter zu testen wurde eine zweite Wortliste angelegt. Dabei wurde darauf geachtet, dass sich die beiden Wortlisten, also jene zum Training und jene zur Prüfung sich unterscheiden. Mit einer vorgegebenen Wahrscheinlichkeit kann so der Filter relativ leicht geprüft werden.

### 3.2 Resultate

Anbei die Testresultate mit dem oben genannten Verfahren:

```
+-----+
|               Testresults BloomFilter               |
|               implemented by                         |
|      Stefan Gruber, Roger Kreienbühl, Florian Thiévent      |
+-----+
| Right positive: 2887                                     |
| False positive: 293                                     |
| In list detected: 3180                                   |
| Size of TestwordsList: 5824                             |
| Probability: 0.01                                         |
| Percent OK (right positive): 0.9078616352201258 %        |
| Percent NOK (false positive): 0.09213836477987421 %      |
+-----+
```

Abbildung 3.1: Testresultate der Implementierung