Turun yliopisto
University of Turku

Exploring the Use of Deprecated PHP Releases in the Wild Internet: Still a LAMP Issue?

Jukka Ruohonen

Sami Hyrynsalmi

Ville Leppänen

WIMS 2016, Nîmes

Research Question

- Is the use of deprecated PHP releases associated with LAMP?
  $\Longrightarrow$ The so-called "personal home page" (a.k.a. **PHP**) is still clearly the most popular server-side web programing language
  $\Longrightarrow$ Here, the term **deprecation** refers to major PHP release branches that are no longer supported by the PHP project
  $\Longrightarrow$ The abbreviation **LAMP** (Linux, Apache, MySQL, PHP) is taken as an idiom for a still typical open source web stack

Motivation

Turun yliopisto
University of Turku

1. **Analytics**

   $\Longrightarrow$ A well-established niche for web **fingerprinting**

   $\Longrightarrow$ Software and **release engineering**

2. **Security**

   $\Longrightarrow$ PHP applications are widely exploited, but also the language itself has been exposed to many **vulnerabilities**

   $\Longrightarrow$ Automatic scanners frequently probe for deprecated releases (cf. WordPress); national cyber security concerns

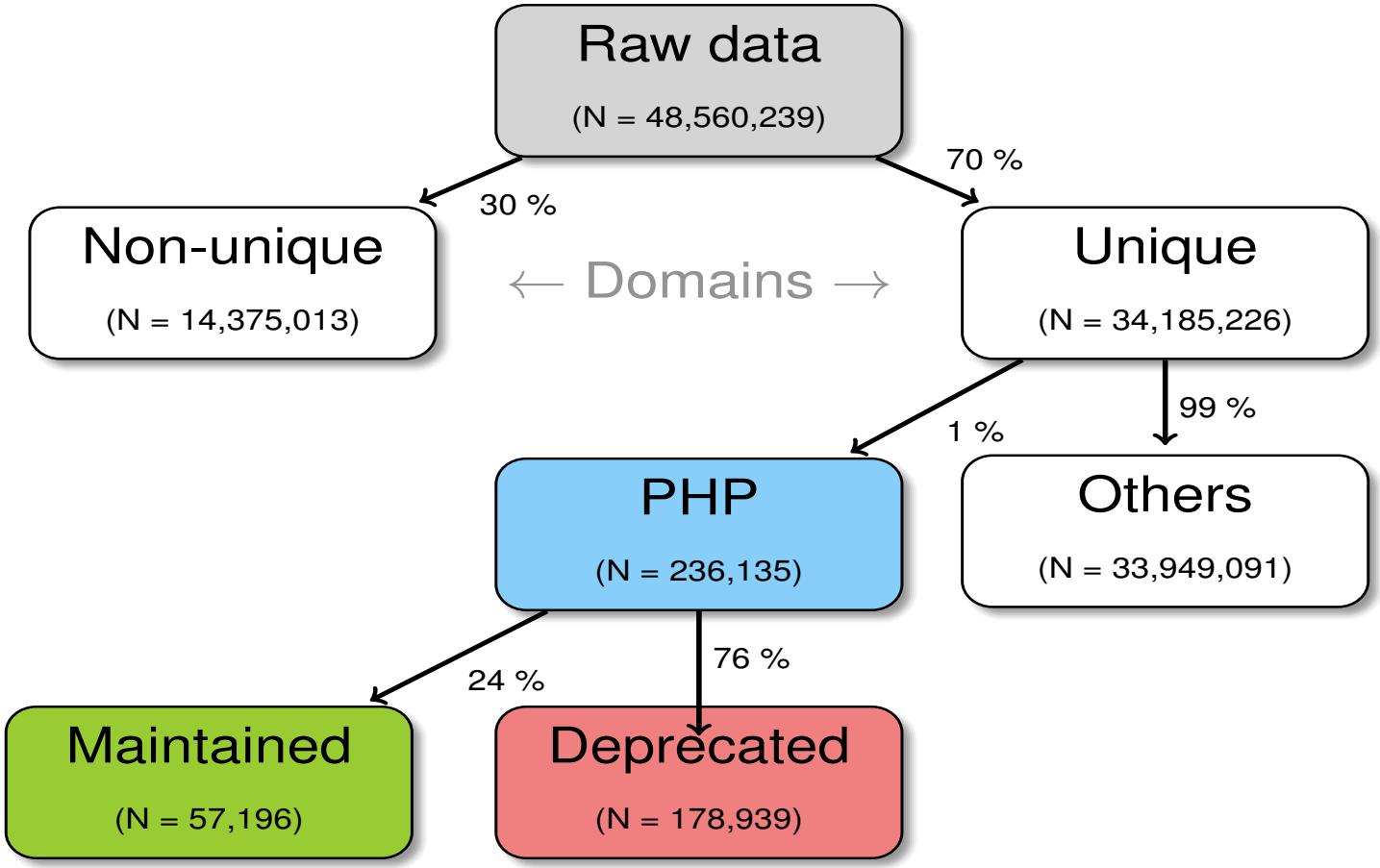
3. **Data Mining**

   $\Longrightarrow$ What can be done with existing **HTTP header data**?

Materials

- The **HTTP Archive** web crawling project
  - $\Longrightarrow$ Covers most of the popular web sites
  - $\Longrightarrow$ Sponsored by Google, Mozilla, O'Reilly, etc.
  - $\Longrightarrow$ Previously, mainly used for performance evaluations
- Generally **big data** (high-volume, potentially high-variety)
  - $\Longrightarrow$ Though, a **bi-monthly** crawling schedule (low-velocity)
  - $\Longrightarrow$ Only a single **snapshot** utilized for this research
- Third-party data sources imply **research constraints**
  - $\Longrightarrow$ Notably, only HTTP headers are used for fingerprinting

Sample



Variables

- **Dependent**: PHP deprecation (0/1)
- **Independent**: five categories derived from HTTP headers;
  1. HTTP servers (e.g., Apache, IIS, Nginx, etc.)
  2. Operating systems (e.g., Linux, Windows, Ubuntu, etc.)
  3. Apache-specific modules (e.g., OpenSSL)
  4. One PHP-specific variable (Suhosin)
  5. Control variables (e.g., top-30 TLDs)
- **Operationalization** with regular expression matching

  $\Longrightarrow$ When possible, includes also rough release information about the independent variables

  $\Longrightarrow$ Thus, false positives (negatives) are likely

Methods (1/2)

- **Estimation** with the standard logistic regression:

$$p_i = \Pr\{\text{PHP is deprecated} = 1 \mid \mathbf{x}_i\},$$
$$= F(\mathbf{x}_i'\boldsymbol{\beta}) = \frac{\exp(\mathbf{x}_i'\boldsymbol{\beta})}{1 + \exp(\mathbf{x}_i'\boldsymbol{\beta})},$$

- In total, **57 parameters**, including a constant
- **Parameter evaluation** with the so-called **marginal effects**:

$$\widehat{\beta}_j = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i'\widehat{\boldsymbol{\beta}})\widehat{\beta}_j \quad \text{(continuous)}$$

$$\widehat{\beta}_j = \frac{1}{n}\sum_{i=1}^{n}\left[F(\mathbf{x}_i'\widehat{\boldsymbol{\beta}}) - F\left(\widehat{\beta}_0 + \sum_{s=1, s\neq j}^{k}\widehat{\beta}_s x_{is}\right)\right] \quad \text{(discrete)}$$

Methods (2/2)

- **Model reduction** sought with a combined forward and backward stepwise selection algorithm
  $\Longrightarrow$ Drops uninformative variables (based on AIC)
  $\Longrightarrow$ Implemented in most statistical software packages
- Reduced model compared also with a **likelihood ratio test**
- **Model evaluation** with a five-fold cross-validation and the three conventional metrics (accuracy, precision, and recall)
- **Resampling** (during model training) used for accounting the **unbalanced sample** (i.e., about 76 % deprecated)

Results (1/4)

Table: Deprecated PHP Releases (as of 10 Dec 2015)

Branch	Deprecation date	Days since EOL
5.4	3 September 2015	89
5.3	14 August 2014	474
5.2	6 January 2011	1790
5.1	24 August 2006	3386
5.0	5 September 2005	3739
4.4	7 August 2008	2672
4.3	31 March 2005	3897
4.2	6 September 2002	4834
4.1	12 March 2002	5012
4.0	23 June 2001	5274
3.0	20 October 2000	5520

Results (2/4)

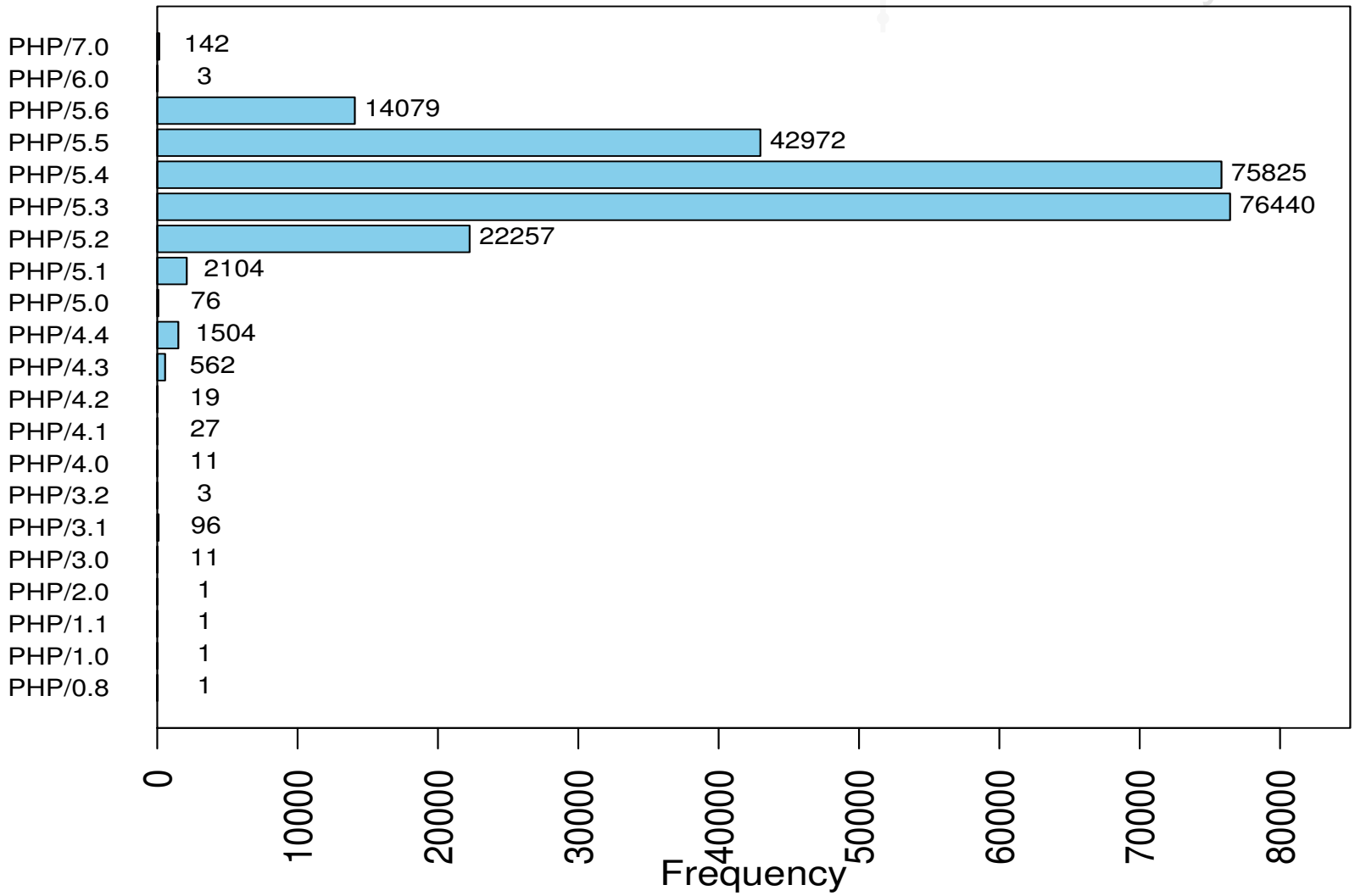


Figure: PHP releases

Results (3/4)

Table: Performance of $\mathcal{M}^i_{\text{Full}}$

$i =$	1	2	3	4	5
Unbalanced					
Accuracy	0.780	0.775	0.778	0.778	0.780
Precision	0.970	0.968	0.969	0.971	0.970
Recall	0.789	0.785	0.787	0.786	0.788
Balanced					
Accuracy	0.635	0.636	0.635	0.638	0.633
Precision	0.640	0.643	0.640	0.645	0.636
Recall	0.842	0.839	0.841	0.840	0.840

Results (4/4)

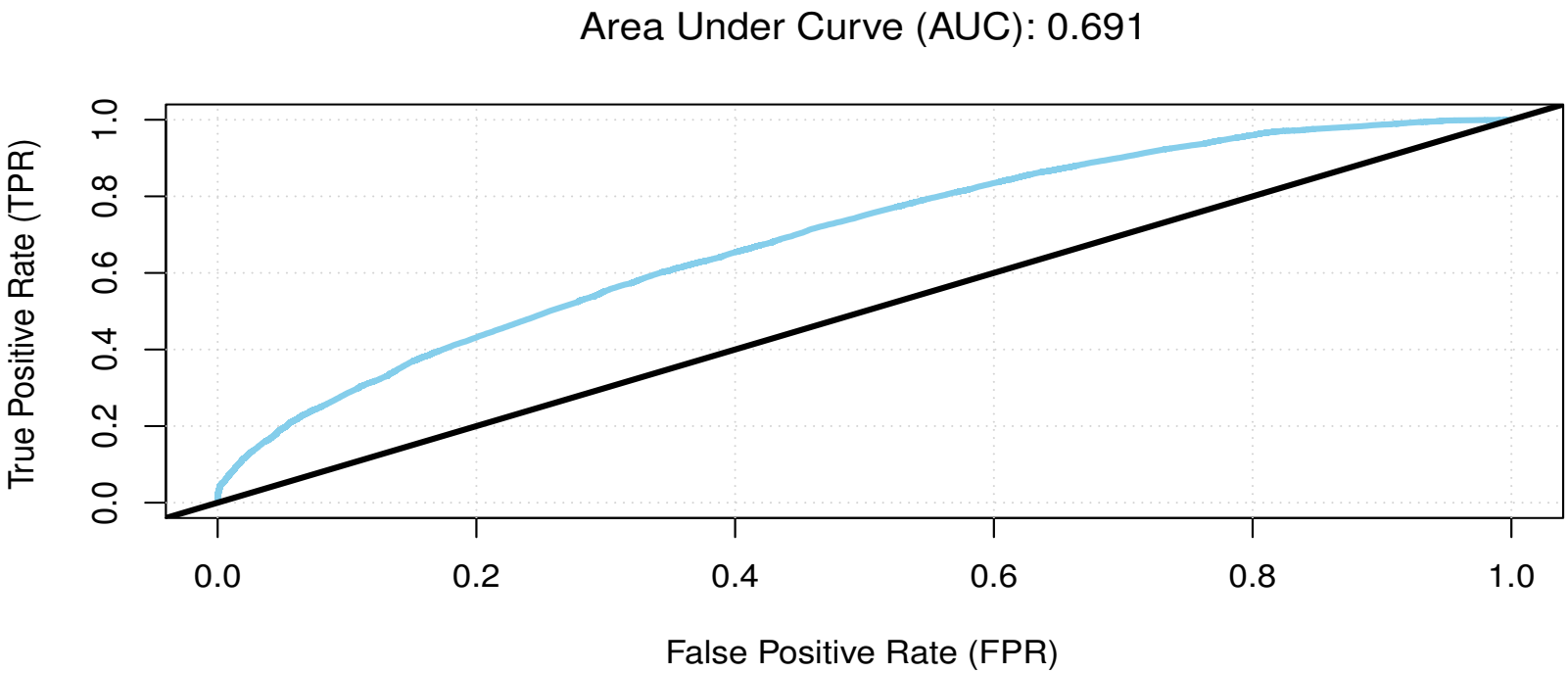


Figure: Performance of Unbalanced $\mathcal{M}^1_{\text{Full}}$

Conclusion

Turun yliopisto
University of Turku

- Thus, classification **performance is modest** at best
  - $\Longrightarrow$ Irrespective of the balancing
  - $\Longrightarrow$ Unlikely due to logistic regression *per se*
  - $\Longrightarrow$ Though, all of the variables pass the model reduction
- The marginal effects indicate **interpretable** results
  - $\Longrightarrow$ In particular, the **LAMP hypothesis holds**; deprecated PHP releases are strongly associated with Linux and Apache
  - $\Longrightarrow$ No notable "geographic" variation (given popular TLDs)
  - $\Longrightarrow$ Some variation across Linux distributions
- In general, HTTP **header data has limited use**

Future Work

1. **Fingerprinting** improvements

   $\Longrightarrow$ HTTP headers are adequate for web server and application (PHP) layers; operating system probing is much more difficult (though, cf. network scanners)

2. **Combination of data** sources

   $\Longrightarrow$ Popularity ranks, PageRank, DNS, etc.

3. **Longitudinal** analysis

   $\Longrightarrow$ The evolution of PHP is arguably more interesting from a release engineering perspective

4. Security and **malware analysis**

   $\Longrightarrow$ The use of deprecated PHP releases is presumably more relevant as an independent rather than dependent variable

Thank you

Questions?