Context-Aware Malware Detection Using Topic Modeling

A Thesis Presented By: Wayne Stegner

Committee Members:
Dr. Rashmi Jha (Chair)
Dr. Carla Purdy
Dr. David Kapp
Dr. Temesguen Kebede

September 27, 2021

Acknowledgements

This work was funded by AFRL under DAGSI-SOCHE Award No. RY8-UC-20-1.

Motivation

- Need for cyber security is growing
- SolarWinds cyber attack
  - Malware attack compromised an estimated 100 companies and several U.S. federal agencies
  - Could cost hundreds of millions of dollars to U.S. government alone
- Colonial Pipeline cyber attack
  - Major U.S. oil pipeline was compromised
  - Pipeline operation temporarily halted
- Severe consequences: Need better understanding of cyber threats

Motivation for Context

- A given action is not malicious by nature
- Malice is relative to the desired outcome
- Sort vs. search programs
  - Searching: Changing the data is unexpected
  - Sorting: Reordering is expected (but not changing actual values)
- Autonomous drone camera shutoff
  - Turning off a drone's camera is not always malicious
  - But it *can* be malicious
  - Depends on the context

Malware Analysis Overview

- Static analysis
  - Examining the file without running it
  - Opcodes, system calls, control flow features, etc.
  - Difficult if code is obfuscated
- Dynamic analysis
  - Examining the file by running it
  - Opcodes, system calls, control flow, system changes, network activity, etc.
  - Requires sandbox environment to contain malware

k-Nearest Neighbors (k-NN)

- Simple machine learning classification algorithm
- Only parameter — k, the number of neighbors
- Present labeled training dataset
- Inference the class of new points
  - Majority vote among the k nearest neighbors to the new point
  - Typically uses Euclidean distance

Latent Dirichlet Allocation (LDA)

- Generative statistical topic modeling algorithm [1]
- Learn latent topics from a corpus of documents
  - Each topic is a probability distribution over the vocabulary
- Assign weighted mixture of topics to each document
- Bag-of-words (BoW) preprocessing
  - Document: "cat dog mouse cat cat dog"
  - BoW: $\{``cat'' : 3, ``dog'' : 2, ``mouse'' : 1\}$

LDA Model Evaluation

- Intrinsic evaluation
  - Directly measure quality of topics
  - Perplexity or topic cohesion
  - Unsupervised methods (do not need labeled data)
- Extrinsic evaluation
  - Evaluate topic quality through secondary task
  - Classification accuracy
  - Can be more intuitive to interpret

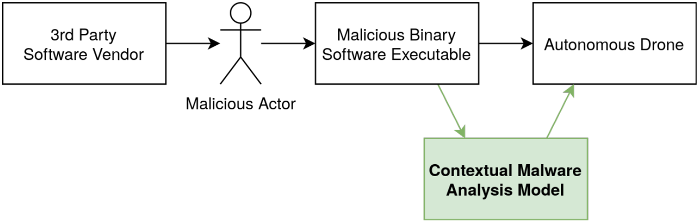
LDA in Malware Classification

- API calls [2]
  - Modeled topics from API calls with LDA
  - Classified topic distributions using various classifiers
  - Maximum accuracy of 98.61%
- Static/dynamic opcodes [3]
  - Modeled topics from both static and dynamic opcode sequences using LDA
  - Showed difference between search and sort programs using LDA topic distributions
  - Analysis was done manually
- Static opcodes [4]
  - Extends [3] to include malware classification
  - Classified Microsoft Malware Classification Challenge (BIG 2015) [5] using k-NN
  - Accuracy of 97.2%

Context in Software Analysis

- Context-based access control [6, 7]
  - Limit allowed actions based on location data
  - Targets smartphone security
- Interaction-based context [8]
  - Require user gestures to allow sensitive actions
  - Targets smartphone security
- Graph-based context [9]
  - Examine entry point of program representation graphs
  - Include whether the user is aware or unaware of the action
  - Targets smartphone security

Threat Model



Context Definition

- Context should encapsulate:
  1. What is the physical context of the system?
  2. How does actual behavior compare to expected behavior?
  3. Why is the software making certain decisions?

Cross Validation

- LDA models have high randomness
  - Accuracy variations of several percent on the same data
  - Difficult to compare parameters
- Solve with k-fold cross validation
  - Split dataset into k folds
  - Cycle through folds as testing partitions
  - k different models trained
  - Take average performance

Model Overview

- Purpose of the model
  - Extrinsic evaluation of LDA features
  - Explore model parameters (number of topics and k)
- Input data
  - Each file is a sequence of static opcodes
- Evaluated using 5-fold cross validation

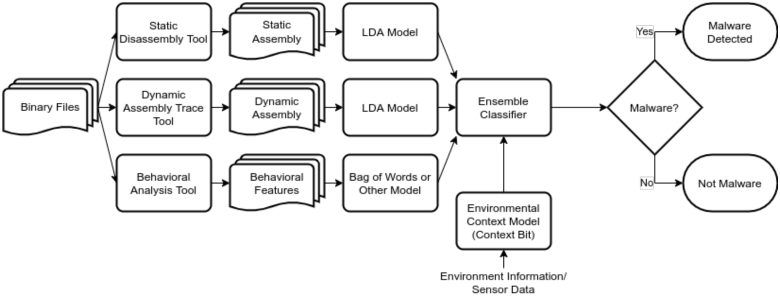
Model Process

1. Transform all documents into BoW documents.
2. Fit LDA model on the training partition.
3. Transform all BoW documents into topic distributions.
4. Fit k-NN classifier on the topic distributions of the training partition.
5. Evaluate k-NN classifier on the topic distributions of the test partition.

Model Overview

- Utilize static, dynamic, and behavioral features
- Extract useful features with LDA/BoW models
- Define context based on physical context
  - Environment data collected from sensors
  - Simplified to a single bit (good vs bad context)

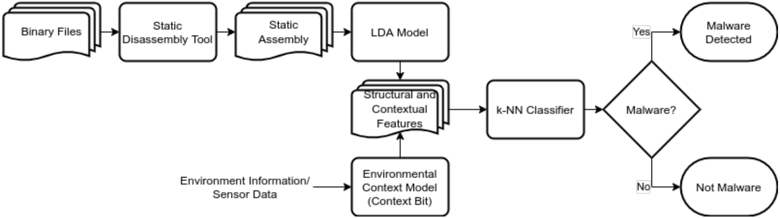
Ensemble Model Diagram



Simplifications

- There is a lot going on
  - Three different feature extraction methods required
- Dynamic features are difficult to collect
  - Difficult to make malware run in a sandbox
  - Took a *long* time to collect
- Simplified model to use only static features

Simplified Model Diagram



Context Integration

- Randomly generate context bit for each file
- Append context bit to LDA topic vector
  - With 15 topics, feature vector is 16-dimensional
- Label whether or not the physical context matches the action (file class)
  - Benign file with good context is operating within proper context
  - Malicious file with bad context is operating within proper context
  - Other cases, file is violating its context

Dataset — Das Malwerk

- Dataset requirements
  - Small dataset for initial testing
  - Live files for dynamic analysis
    - This requirement was later dropped
- Two class dataset
  - Malicious files — 576 samples from Das Malwerk [10]
  - Benign files — 646 samples from default Windows 7 installation

Model Overview

- Similar to simplified context bit model
  - Utilize static disassembly features with LDA
  - Only difference is context definition
- Define context based on expected behavior
  - What type of software do we expect?
  - In practice — vendor description of the software
  - For testing — class label in the dataset

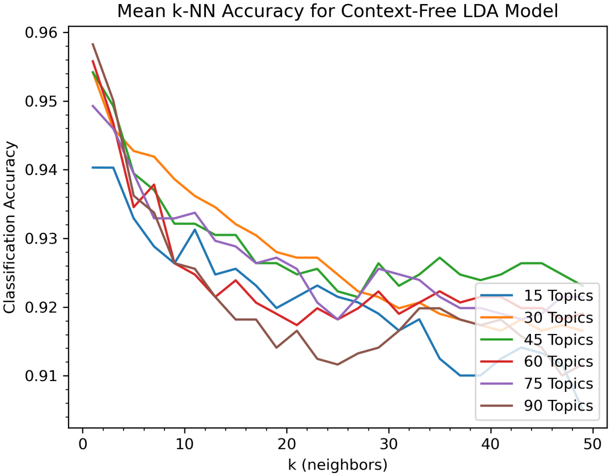
Dataset — BIG 2015

- Limitations of Das Malwerk dataset
  - Only two classes: malicious and benign
  - Classes are too general
- New dataset — BIG 2015 [5]
  - Nine classes separated by specific functionality
  - Already disassembled
- We are not treating these files as inherently malicious
  - Yes, these are all technically malware
  - Treated as just nine different types of software

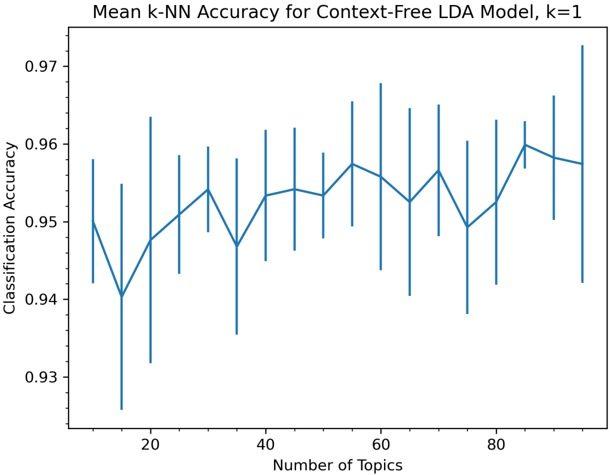
Context Integration

- Goal — simulate receiving software that is not the type we expect
- Change 50% of the class labels
  - Class label represents expected software type
- Append new class label to LDA feature vector
  - One-hot encoding — $3 \rightarrow \{0, 0, 1, 0, 0, 0, 0, 0, 0\}$
  - With 15 LDA topics, feature vector is 24-dimensional
- Label whether or not context is violated
  - File with changed label is violating its context
  - File with original label is operating within proper context

Classification Accuracy — Das Malwerk

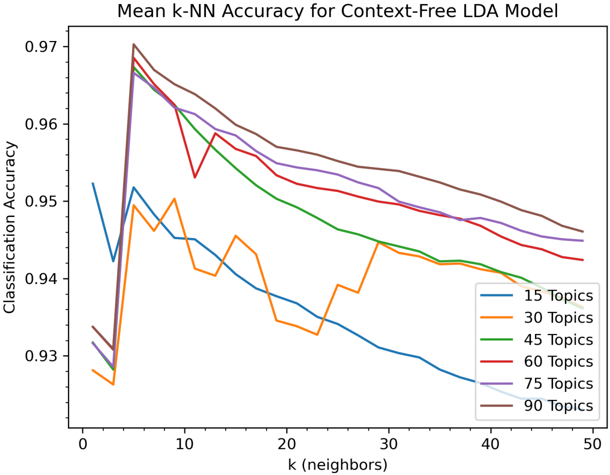


Classification Accuracy — Das Malwerk

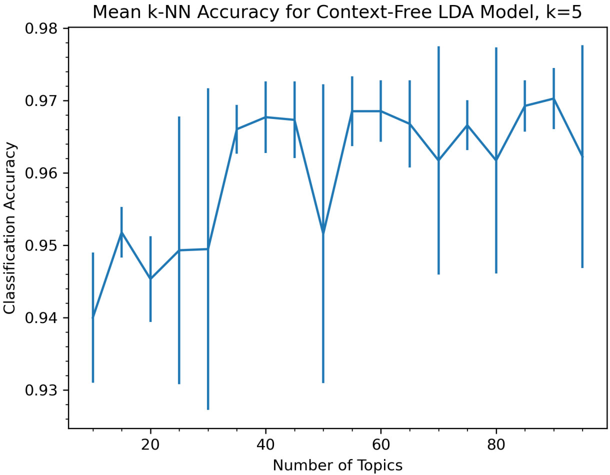


Best accuracy: 85 LDA topics, $k = 1$ — 95.99%

Classification Accuracy — BIG 2015



Classification Accuracy — BIG 2015

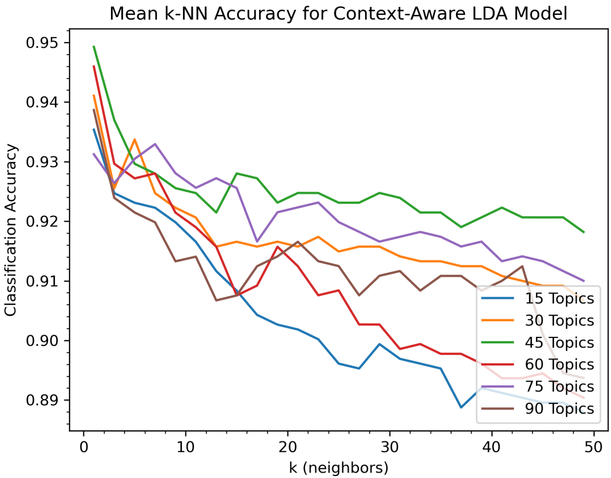


Best accuracy: 90 LDA topics, $k = 5$ — 97.03%

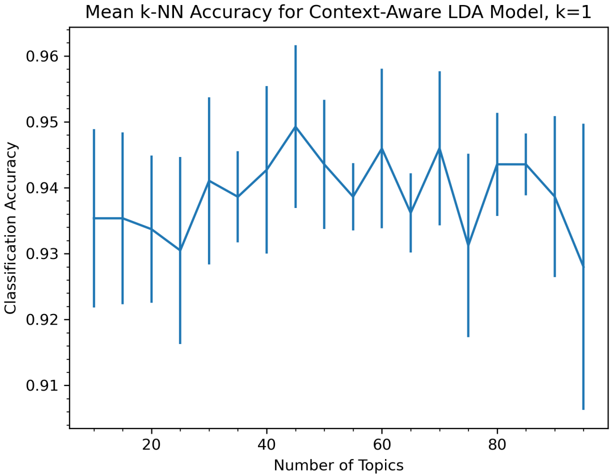
Discussion

- Good performance on both datasets
  - LDA features carry useful information for classification
- Large error bars
  - High variance between models with the same parameters
  - Cannot be confident our best parameters are actually the best

Classification Accuracy



Classification Accuracy

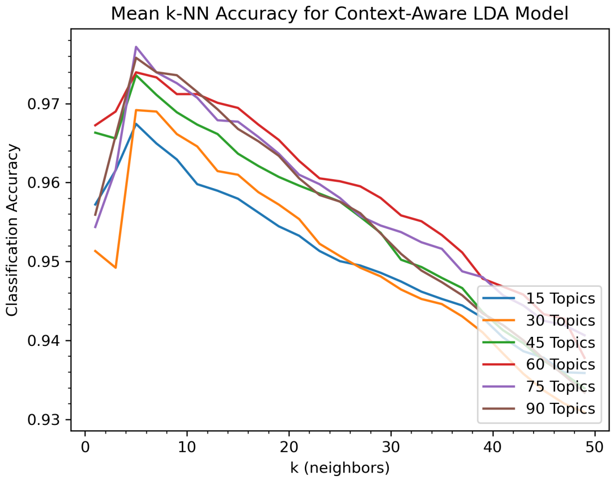


Best accuracy: 45 LDA topics, $k = 1$ — 94.92%

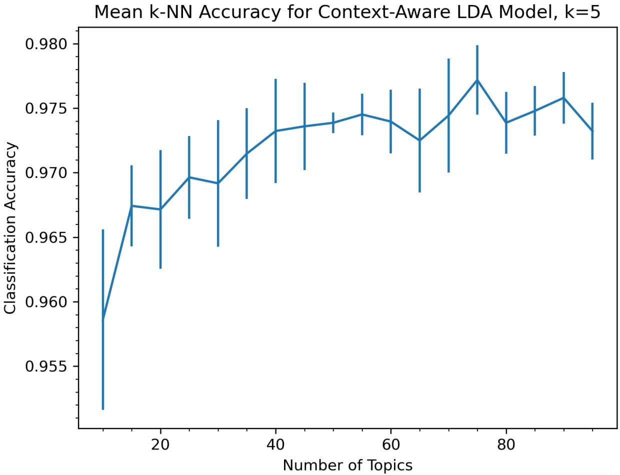
Discussion

- Good performance for the given task
  - Lower than context-free system
  - Large error bars
- Task is oversimplified
  - Context is not a single binary value
  - Require more complex model
- Physical context alone does not form complete context

Classification Accuracy



Classification Accuracy



Best accuracy: 75 LDA topics, $k = 5$ — 97.72%

Discussion

- Good performance on the given task
  - Higher than context-free system
  - Large error bars
- Expected behavior may not fit into discrete classes
  - Some classes may have similar behaviors
  - Some programs could be a mixture of classes

Conclusions

- Explored the definition of context in malware detection
- Presented two proof-of-concept models to address various parts of context
- Context is challenging to define
  - Framing context for software analysis
  - Our questions do not translate directly to computational model
- Proof-of-concept models performed well at their task, but do not make a complete picture of context

Future Work

- Add dynamic analysis
- Biological inspiration
  - Higher-level cognition
- Physical context model improvements
- Combining context models
- More rigorous parameter tuning

List of Publications

W. Stegner, D. Kapp, T. Kebede, and R. Jha, "Context-Aware Malware Detection Using Topic Modeling", *Submitted but not published.*

W. Stegner, T. Westland, D. Kapp, T. Kebede, and R. Jha, "MiBeX: Malware-Inserted Benign Datasets for Explainable Machine Learning", in *Interpretable Artificial Intelligence: A Perspective of Granular Computing*, Feb. 2021, pp. 269–291.

M. Santacroce, W. Stegner, D. Koranek, R. Jha, "A Foray Into Extracting Malicious Features from Executable Code with Neural Network Salience", in *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, USA: IEEE, Jul. 2019, pp. 185–191.

References I

See thesis for full reference list.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *The Journal of Machine Learning Research* 3 (Mar. 2003), pp. 993–1022. ISSN: 1532-4435.

G. Ganesh Sundarkumar et al. "Malware Detection via API Calls, Topic Models and Machine Learning". en. In: *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. Gothenburg, Sweden: IEEE, Aug. 2015, pp. 1212–1217. ISBN: 978-1-4673-8183-3. DOI: 10.1109/CoASE.2015.7294263.

References II

- Jeremiah Greer. "Unsupervised Interpretable Feature Extraction for Binary Executables Using LIBCAISE". en. Masters Thesis. University of Cincinnati, 2019. URL: https://etd.ohiolink.edu/apexprod/rws_olink/r/1501/10?p10_etd_subid=180585&clear=10 (visited on 12/18/2020).
- Ouboti Djaneye-Boundjou et al. "Static Analysis through Topic Modeling and Its Application to Malware Programs Classification". In: *2019 IEEE National Aerospace and Electronics Conference (NAECON)*. July 2019, pp. 226–231. DOI: 10.1109/NAECON46414.2019.9057876.
- Royi Ronen et al. "Microsoft Malware Classification Challenge". In: *arXiv:1802.10135 [cs]* (Feb. 2018). arXiv: 1802.10135 [cs].

References III

Eduardo B. Fernandez, Maria M. Larrondo-Petrie, and Alvaro E. Escobar. "Contexts and Context-Based Access Control". In: *2007 Third International Conference on Wireless and Mobile Communications (ICWMC'07)*. Mar. 2007, pp. 73–73. DOI: 10.1109/ICWMC.2007.30.

Bilal Shebaro, Oyindamola Oluwatimi, and Elisa Bertino. "Context-Based Access Control Systems for Mobile Devices". In: *IEEE Transactions on Dependable and Secure Computing* 12.2 (Mar. 2015), pp. 150–163. ISSN: 1941-0018. DOI: 10.1109/TDSC.2014.2320731.

References IV

Babins Shrestha et al. "Tap-Wave-Rub: Lightweight Human Interaction Approach to Curb Emerging Smartphone Malware". In: *IEEE Transactions on Information Forensics and Security* 10.11 (Nov. 2015), pp. 2270–2283. ISSN: 1556-6021. DOI: 10.1109/TIFS.2015.2436364.

A. Narayanan et al. "Context-Aware, Adaptive, and Scalable Android Malware Detection through Online Learning". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 1.3 (June 2017), pp. 157–175. ISSN: 2471-285X. DOI: 10.1109/TETCI.2017.2699220.

Robert Svensson. *Das Malwerk*. URL: https://www.dasmalwerk.eu/ (visited on 12/21/2020).