Machine Learning Classification of Internet Worms and Ransomware Attacks and Effect of BGP Feature Properties

---

M. A. Sc. Thesis
Hardeep Kaur Takhar
School of Engineering Science
Simon Fraser University

Roadmap

- Motivation and Introduction
- Dataset Description
- Dimension Reduction and Clustering
- Feature Selection
- Feature Analysis Based on Goodness of Fit Test
- Machine Learning Approaches
- Performance Evaluation
- Conclusion and References

Roadmap

- **Motivation and Introduction**
- Dataset Description
- Dimension Reduction and Clustering
- Feature Selection
- Feature Analysis Based on Goodness of Fit Test
- Machine Learning Approaches
- Performance Evaluation
- Conclusion and References

Motivation

- **Internet:**
  - Digital presence and device connectivity have immensely grown:
    - due to continued Internet expansion
    - to meet increased user demands
  - Evolution of attacks and increase in the complexity of devices connected to the Internet have outpaced current infrastructural and cybersecurity solutions

J. Kurose and K. Ross, "*Computer Networking: A Top-Down Approach*, 8th ed., New Jersey, USA: Pearson, 2021, pp. 1-80.

Motivation

- **Internet:**
  - Global network that facilitates communication, collaboration, access to information
  - Managing secure connections of devices connecting to the Internet is a major challenge
- **Intrusion detection systems (IDSs):**
  - Host-based: installed on host devices to monitor operating system files
  - Network-based: installed to monitor network traffic
    - signature-based: match the attack signature against the database
    - anomaly-based: search for an activity deviating from regular behavior

J. Kurose and K. Ross, "*Computer Networking: A Top-Down Approach*, 8th ed., New Jersey, USA: Pearson, 2021, pp. 1-80.

Background

- **Attacks:**
  - Malware: malicious software
    - spyware, keyloggers, rootkits, malicious adware, bots, trojans, viruses, worms, and ransomware
  - Attack vectors: method of obtaining unauthorized access to a system to perform an illicit activity by delivering malware in the form of attachments
    - SQL injection, exploits, phishing, and passive monitoring
  - Exploits: tools that search for vulnerabilities in a system to launch an attack by installing malware
- **Example:** attack vectors such as phishing emails are used to spread infected attachments

S. Donaldson, S. Siegel, C. K. Williams, and A. Aslam, *Enterprise Cybersecurity*. Apress, 2015.

Machine Learning Techniques

- **Various machine learning approaches are widely used for intrusion detection:**
  - Unsupervised: labels unavailable
  - Supervised: labels available
  - Semi-supervised: partial labels available
- **Training time is important for designing scalable and real-time IDSs:**
  - Short training time enables:
    - scalability
    - computational effectiveness

Summary of Research Contributions

- **Main contributions:**
  - Feature analysis:
    - goodness of fit Kolmogorov – Smirnov (K-S) test
  - Classification of anomalies using machine learning:
    - unsupervised
    - supervised

Research Publications

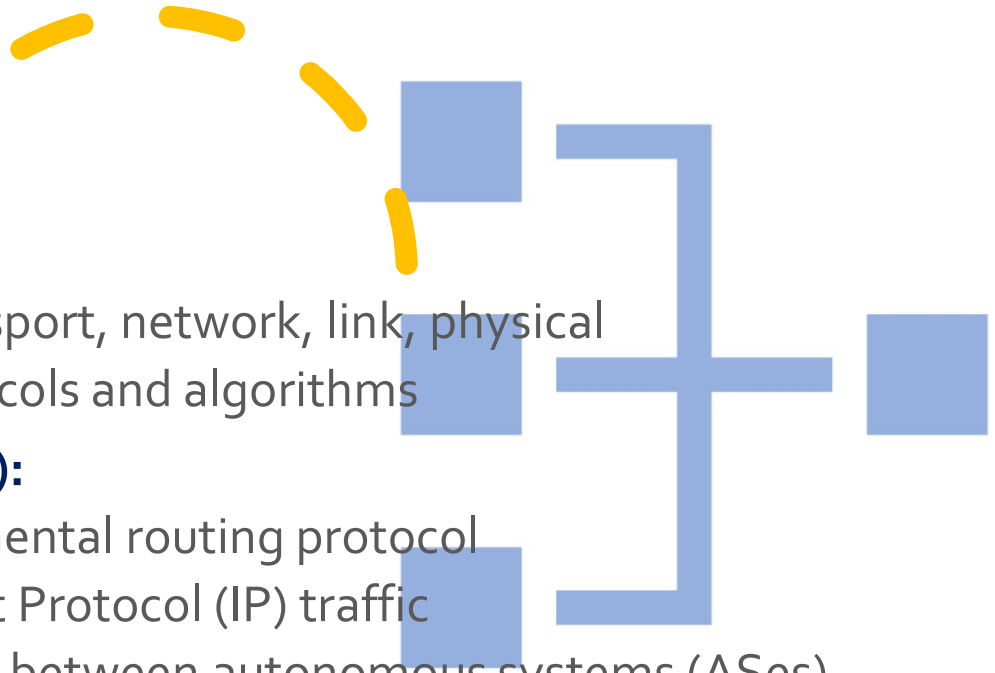
- H. K. Takhar and Lj. Trajković, "Internet worms and ransomware datasets: feature analysis," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics,* Maui, Hawaii, USA, submitted.
- H. K. Takhar, A. L. Gonzalez Rios, and Lj. Trajković, "Comparison of virtual network embedding algorithms for data center networks," in *Proc. IEEE International Symposium on Circuits and Systems*, Austin, Texas, USA, May 2022, pp. 1660—1664 (virtual).

Background

- **Anomalies:**
  - Deviations (non-adhering patterns) from expected behavior
  - Outliers, discordant observations, exceptions
  - Point, contextual, collective anomalies
  - Severe economic consequences for individuals and corporations due to cyber and network attacks

V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, July 2009.

Background



- **Internet data network model:**
  - Five layers: application, transport, network, link, physical
  - Network layer: routing protocols and algorithms
- **Border Gateway Protocol (BGP):**
  - De-facto interdomain incremental routing protocol
  - Facilitates routing of Internet Protocol (IP) traffic
  - Used to establish connection between autonomous systems (ASes)
    - ASes: networks of routers managed by single administrative domain
  - Transport Control Protocol (TCP) connection - port 179
  - Routes data between ASes using optimal path

RFC 1771 - A border gateway protocol 4 (BGP-4). [Online]. Available:https://datatracker.ietf.org/doc/html/rfc1771. Accessed: Apr. 2023.

Background

- **Autonomous Systems (ASes):**
  - Group of networks of routers that are managed by a single administration
  - Perform packet delivery and assist with network connectivity
  - Internet: composed of various ASes

What is an autonomous system? | What are ASNs?. [Online]. Available: https://www.cloudflare.com/learning/network-layer/what-is-an-autonomous-system/. Accessed: Apr. 2023.

Border Gateway Protocol

- **BGP messages:**
  - **Open**
  - **Keepalive**
  - **Update:**
    - BGP update message contains protocol status and configurations
    - fields are extracted to obtain critical information about the network connectivity
  - **Notification**

RFC 1771 - A border gateway protocol 4 (BGP-4). [Online]. Available:https://datatracker.ietf.org/doc/html/rfc1771. Accessed: Apr. 2023.

Border Gateway Protocol

- **BGP anomalies:**
  - Worms: Code Red, Nimda, Slammer
  - Ransomware attacks: WannaCrypt, WestRock
  - Link failures: Moscow, Pakistan blackout

RFC 1771 - A border gateway protocol 4 (BGP-4). [Online]. Available:https://datatracker.ietf.org/doc/html/rfc1771. Accessed: Apr. 2023.

Overview of Related Work

- **BGP:** prone to attacks due to its inherent trust mechanism
- Difficult to differentiate between BGP anomaly and reliability:
  - Example: measures taken by Internet Service Providers (ISPs) to optimize utilization of the network resources may be falsely perceived as an anomaly
- BGP anomalies may be detected using:
  - Cryptographic based prevention, anomaly mitigation, mitigation of unstable route propagation, and anomaly detection techniques

RFC 1771 - A border gateway protocol 4 (BGP-4). [Online]. Available: https://datatracker.ietf.org/doc/html/rfc1771. Accessed: Mar. 15, 2023.
B. Al-Musawi, P. Branch, and G. Armitage, "BGP anomaly detection techniques: a survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 377–396, 2017.

Overview of Related Work

- Various intrusion detection approaches proposed in the literature:
  - IDSs may make assumptions about data
  - Computational costs should be considered when selecting security solutions based on applications
  - Unsupervised machine learning (*k*-means) approaches:
    - assume regular data points are prevalent with high probability

V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, July 2009.
K. P. Murphy, Machine Learning: A Probabilistic Perspective. Cambridge, MA, USA: The MIT Press, 2012.

Overview of Related Work

- Short training time for machine learning based models is desired:
  - Facilitate real-time anomaly detection, scalability when using large datasets, computational effectiveness
- Models generated using various machine learning algorithms:
  - Light gradient boosting machine (LightGBM), gated recurrent unit (GRU), bidirectional GRU (Bi-GRU), broad learning system (BLS)
  - BGP, NSL-KDD, CIC datasets
- BLS models obtained effective performance using shorter training time

Z. Li, A. L. Gonzalez Rios, and Lj. Trajkovic, "Machine learning for detecting anomalies and intrusions in communication networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2254–2264, July 2021.

C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2019.

Introduction

- **Cyberattacks:**
  - Diminish the integrity, confidentiality, availability of resources to legitimate users
  - Viruses, worms, botnets, denial of service (DoS), distributed denial of service (DDoS), ransomware attacks
- **Worms:**
  - Malicious programs that contaminate a system by self propagation and replication
  - World's first known worm Morris: Nov. 2, 1988
  - Code Red, Nimda, Slammer

The Mechanisms and Effects of the Code Red Worm, Sans Institute. [Online]. Available: https://bit.ly/300jRDO. Accessed: Apr. 2023.

Worms

- **Code Red:** Exploited vulnerable Microsoft Internet information service (IIS) servers by overflowing an **unchecked buffer**
  - Spread the worm by probing IP addresses
- **Nimda:** Vulnerabilities of Internet Explorer 5 running on IIS web servers
  - Propagated via infected attachments using email messages, websites, shared network drives
- **Slammer:** Exploited buffer overflow vulnerability in the MS Structured Query Language (SQL) servers
  - Propagated using User Datagram Protocol (port 1434)

Responding to the Nimda Worm: Recommendations for Addressing Blended Threats, Symantec, Cupertino, CA, USA. [Online]. Available: https://bit.ly/43v97On. Accessed: Apr. 2023.
Attack of Slammer Worm - a Practical Study, SANS Institute. [Online]. Available: https://bit.ly/3MtDwqe. Accessed: Apr. 2023.

Worms

- **Code Red:**
  - Overflows an unchecked buffer:
    - attacker gains complete control of the server and executes any code
    - code gained system-level access and generated a list of random IP addresses using a random seed generator
  - Spread the worm by probing these IP addresses
  - Generated IP addresses using the random seed generator are unpredictable
  - Infected thousands of vulnerable systems

The Mechanisms and Effects of the Code Red Worm, Sans Institute. [Online]. Available: https://bit.ly/300jRDO. Accessed: Apr. 2023.

Ransomware Attacks

- Malicious software that encrypts victims' data (partially or completely)
- Attacker masks as law enforcement or authoritative figure to gain ransom as fine for illicit criminal cyber activities:
  - Threatens the victim to pay ransom or otherwise sensitive data will be leaked
- Categories:
  - Ransomware-as-a-service (RaaS), cryptoworm, automated active adversary
  - Locker, crypto, scareware
- **WannaCrypt, WestRock**

N. A. Hassan, *Ransomware Revealed*. Berkeley, CA, USA: Apress, 2019, pp. 3–28.

Ransomware Attacks

- **WannaCrypt (WannaCry): May 2017**
  - ExploitBlue: Cyberattack tool developed by U.S National Security Agency (NSA) was leaked by hackers (April 14, 2017)
  - Server Message Block version 1 (SMBv1) vulnerability in MS Windows 7:
    - application layer protocol provides shared access to files and printers
- **WestRock: USA manufacturing company**
  - Targeted in January 2021
  - Information (IT) and operational (OT) technology systems were targeted
  - Resulted in major shipments and production delays

How Ransomware Attacks, SophosLabs. [Online]. Available: https://bit.ly/3GtR5ly. Accessed: Apr. 2023.
WestRock Provides Update on Ransomware Incident. [Online]. Available: https://bit.ly/3KLHEQM. Accessed: Apr. 2023.

Roadmap

- Motivation and Introduction
- **Dataset Description**
- Dimension Reduction and Clustering
- Feature Selection
- Feature Analysis Based on Goodness of Fit Test
- Performance Evaluation
- Conclusion and References

Datasets

- Generated using **BGP update** messages
- Collection sites:
  - **Réseaux IP Européens (RIPE)**:
    - Routing Information Service (RIS) project initiated by RIPE Network Coordination Centre (NCC)
  - **Route Views**:
    - project at the University of Oregon

RIPE Network Coordination Centre: About us. [Online]. Available: https://www.ripe.net/about-us/. Accessed: Apr. 2023.
RIPE NCC. [Online]. Available: https://www.ripe.net . Accessed: Apr. 2023.
University of Oregon Route Views project. [Online]. Available: http://www.routeviews.org . Accessed: Apr. 2023.

BGP Datasets

- Datasets contain regular (two days prior and two days after the attack) and anomalous (days of the attack) data
- Each row represents one minute of collected data
- 37 extracted features:
  - Volume and AS-Path
- Binary classification:
  - Regular: 0
  - Anomaly: 1
- Training and test datasets contain: 60% and 40% of the anomalies, respectively

Border Gateway Protocol Routing Records from Réseaux IP Européens (RIPE) and BCNET. [Online]. Available: http://ieee-dataport.org/1977. Accessed: Apr. 2023.

RIPE: BGP Datasets

Dataset	Regular (training)	Anomaly (training)	Regular (test)	Anomaly (test)	Collection Date Start 00:00:00	Collection Date End 23:59:59
Code Red	3,679	361	2,921	239	17.07.2001	21.07.2001
Nimda	3,673	827	3,635	474	16.09.2001	21.09.2001
Slammer	3,210	530	3,121	339	23.01.2003	27.01.2003
WannaCrypt	2,880	3,420	2,880	2,340	10.05.2017	17.05.2017
WestRock	2,952	6,008	2,880	4,000	21.01.2021	31.01.2021

BGP Features

Feature Number	Name	Category
1	Number of announcements	volume
2	Number of withdrawals	volume
3/4	Number of announced/withdrawn NLRI prefixes	volume
5/6/7	Average/maximum/average unique AS-path length	AS-path
8/10	Number of duplicate announcements	volume
9	Number of implicit withdrawals	volume
11/13	Maximum/average edit distance	AS-path
12	Arrival rate	volume
14-23/24-33	Maximum AS-path length/edit distance	AS-path
34/35/36	Number of IGP/EGP/incomplete packets	volume
37	Packet size	volume

Roadmap

- Motivation and Introduction
- Dataset Description
- **Dimension Reduction and Clustering**
- Feature Selection
- Feature Analysis Based on Goodness of Fit Test
- Machine Learning Approaches
- Performance Evaluation
- Conclusion and References

Dimension Reduction

- Robustness of machine learning models rely on the quality of data
- Selecting features that do not capture relationships between input data may lead to poor classification results
- Unbalanced datasets: classifier may be biased towards the majority class
- Redundancies in datasets increase training time, computational complexity of models, and memory usage
- Dimension reduction are employed to eliminate irrelevant features:
  - Transforming features
  - Selecting subset of data

Dimension Reduction using PCA

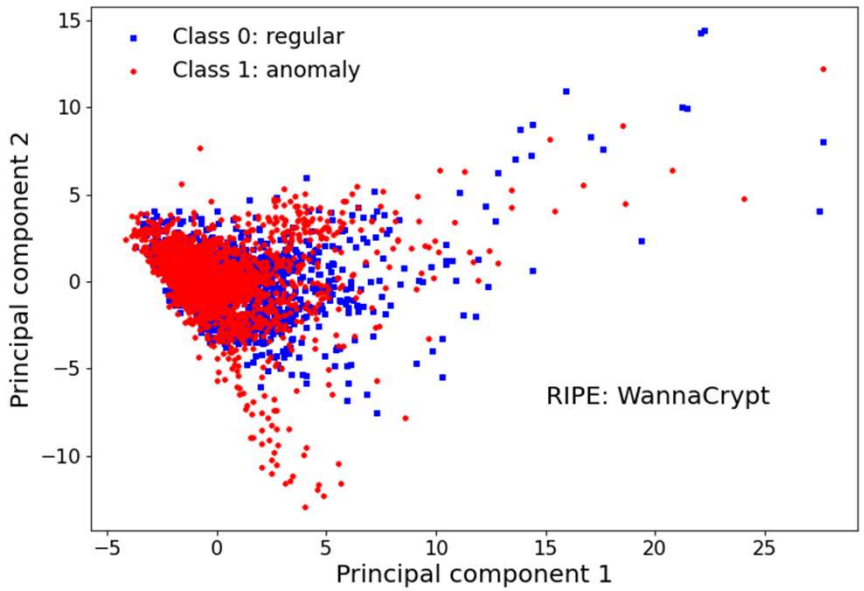
- **Principal component analysis (PCA):**
  - Data are transformed to orthogonal components (principal components)
  - Employed to eliminate linearly dependent features
  - Data are normalized using z-score: mean = 1; std dev = 0;
  - Selected 10 principal components to preserve approximately 70 % variance

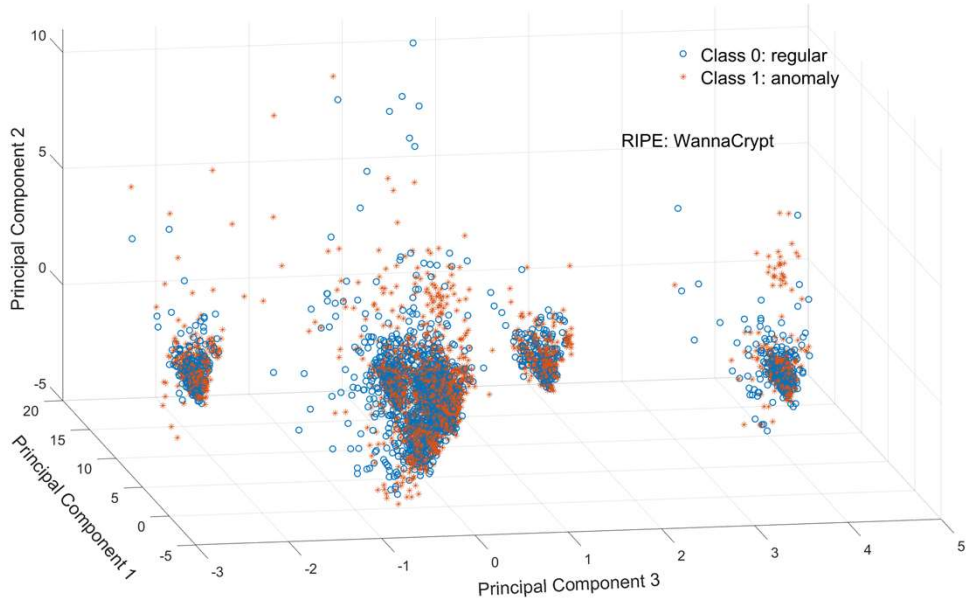
Dataset	Variance (%)
Code Red	69.58
Nimda	68.46
Slammer	67.70
WannaCrypt	71.70
WestRock	69.02

A. Zheng, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Alice Zheng and Amanda Casari., 1st ed. O'Reilly, 2018.

Principal Component Analysis

- RIPE WannaCrypt data: Scatter plot of principal components



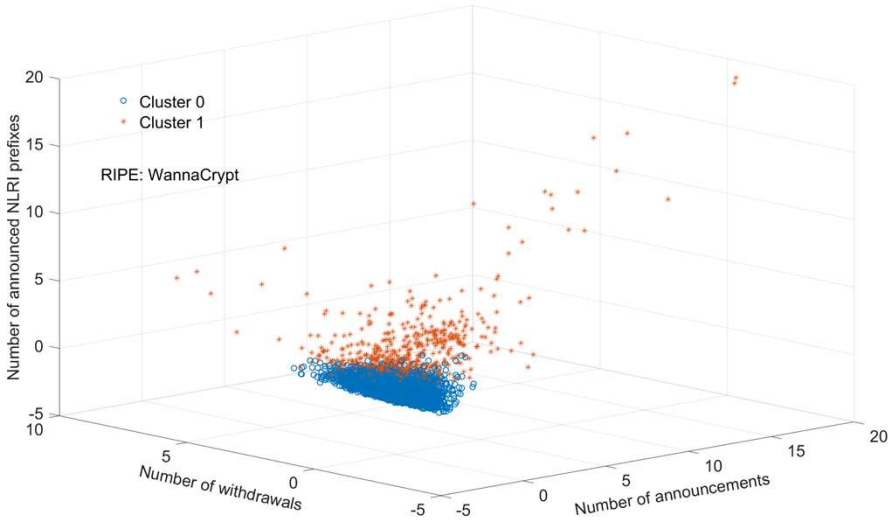


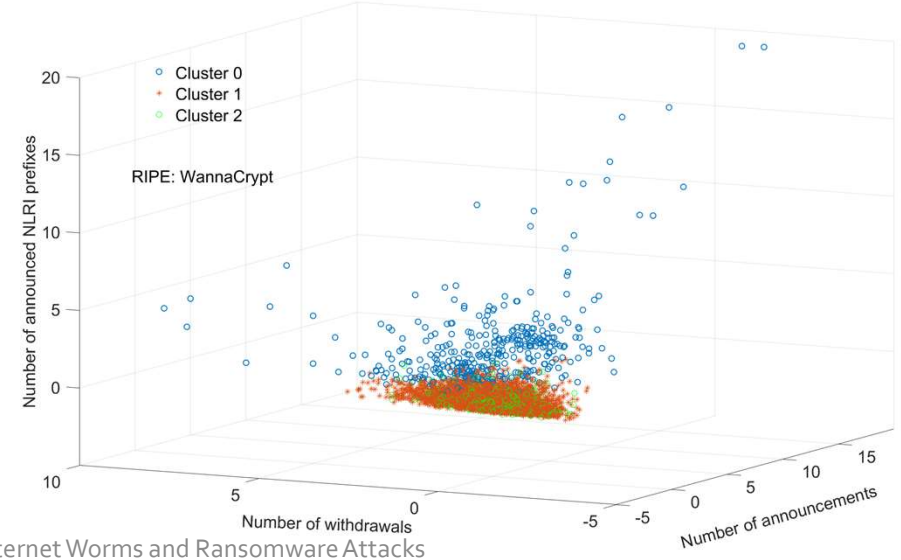
Clustering: *k*-Means

- ***k*-Means:**
  - Unsupervised iterative machine learning algorithm
  - Based on Euclidean distance
  - Intra-cluster: average distance between each data point and the centroid
  - Inter-cluster: average distance between clusters
- **Steps:**
  - *k* centroids are randomly initialized
  - Data points are assigned to the nearest clusters
  - Minimize their intra-cluster distance from the centroid (inertia)
  - Centroids are re-calculated

k-Means: WannaCrypt

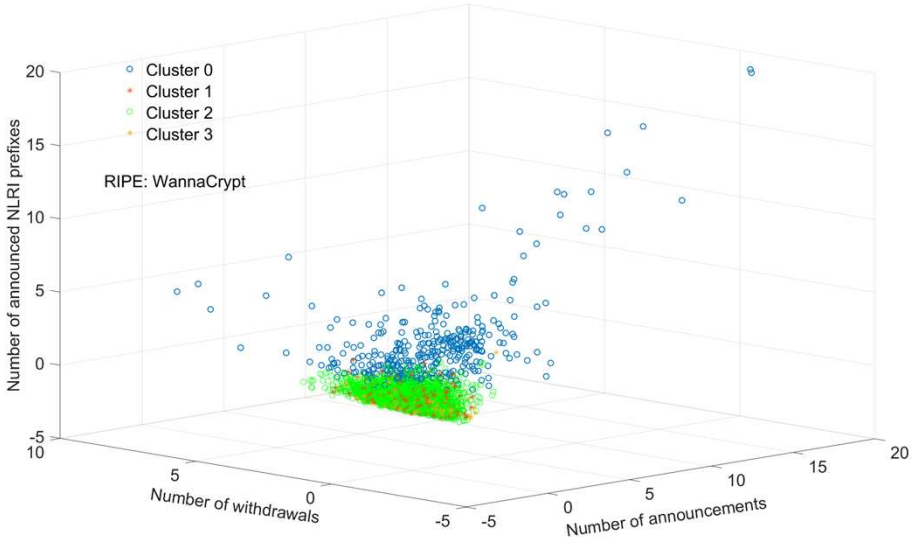
- **RIPE WannaCrypt data scatter plots:** number of clusters *k*
  - Multiple clusters: *k* = 2, 3
  - Silhouette coefficient: measure of separation between clusters

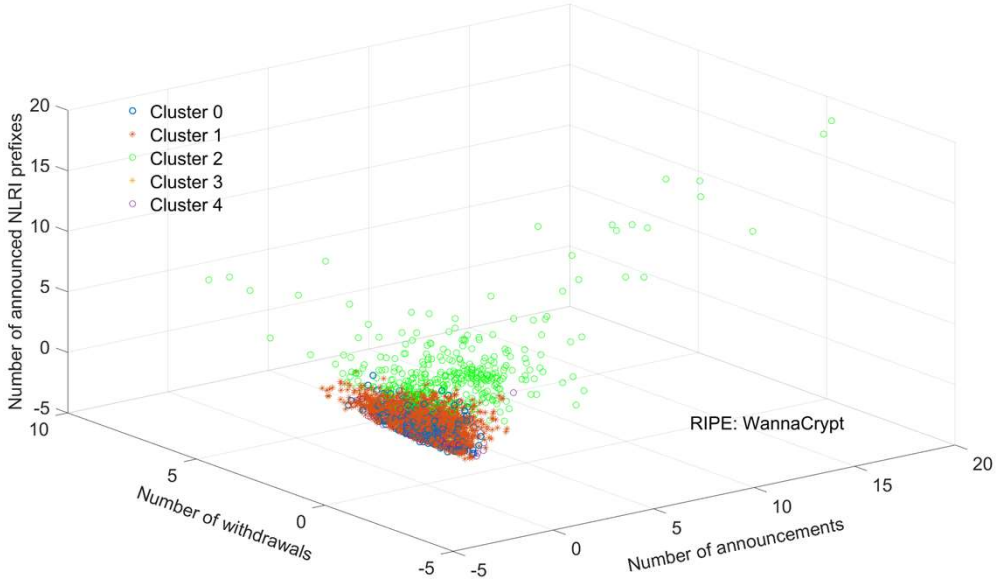




k-Means: WannaCrypt

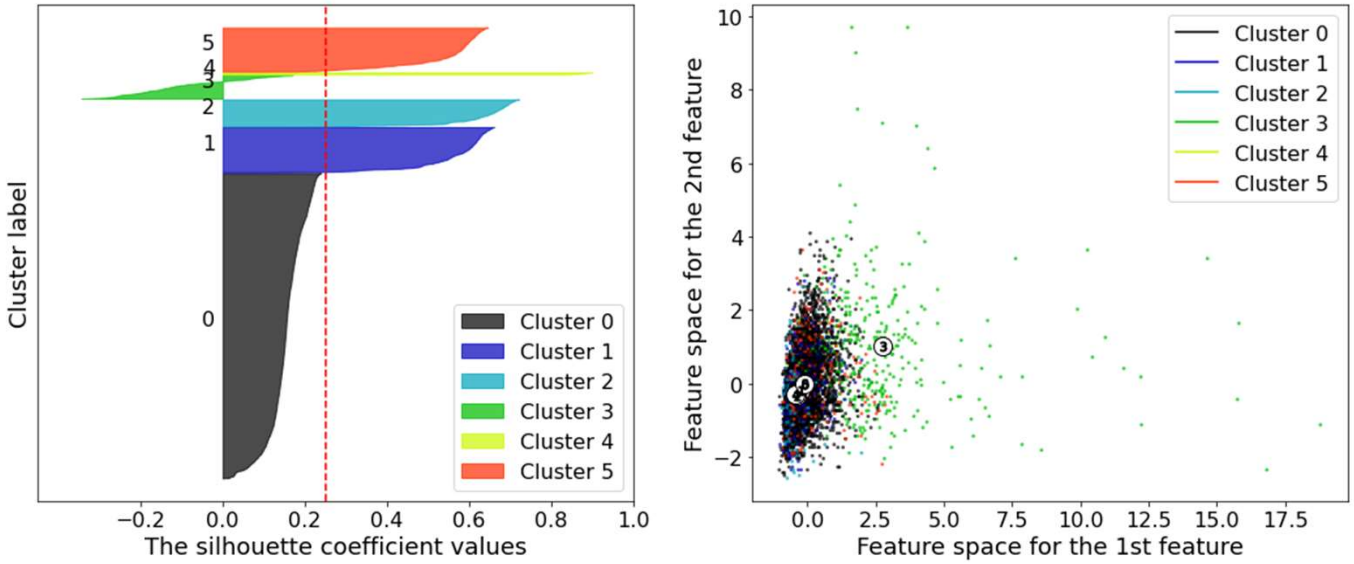
- **RIPE WannaCrypt data scatter plots:** number of clusters *k*
  - Multiple clusters: *k* = 4, 5





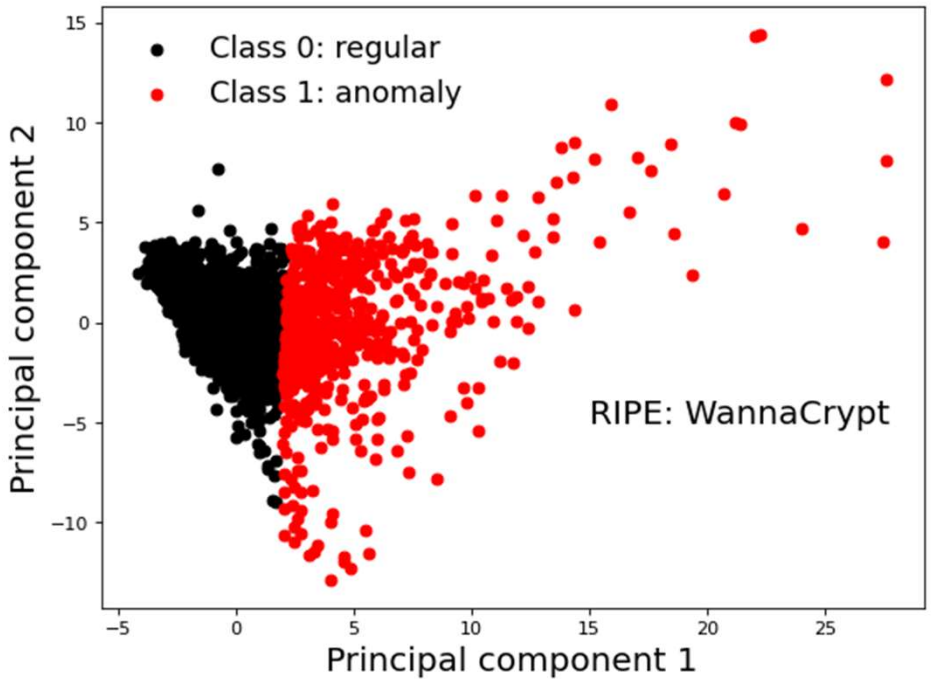
k-Means: Silhouette Coefficient

- Silhouette coefficient = $\frac{b-a}{\max(b,a)}$:
  - a: inter-cluster distance; b: intra-cluster distance



Cluster Refinement: PCA

- Silhouette coefficient increases:
  - 0.3393, *k*-means
  - 0.5222, *k*-means with PCA transformed features



C. Ding and X. He, "K-means clustering via principal component analysis," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 29.

Roadmap

- Motivation and Introduction
- BGP Datasets
- Dimension Reduction and Clustering
- **Feature Selection:**
  - Correlation: Pearson, Spearman
  - Supervised machine learning: Random Forests, Extra-Trees
- Feature Analysis Based on Goodness of Fit Test
- Performance Evaluation
- Conclusion and References

Feature Selection

- **Statistical approaches:**
  - Correlation:
    - Pearson (ρ): linear relationships
    - Spearman (r_s): non-linear relationships
    - $\rho/r_s \in [-1, 1]$: +1: strong-positive; -1: strong-negative; and o no relationship
- **Supervised machine learning:**
  - Random forests
  - Extra-trees

K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: The MIT Press, 2012.
L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Jan. 2001.
P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Mach. Learn., vol. 63, no. 1, pp. 3–42, Apr. 2006.

Feature Selection: Correlation

- **Pearson Correlation:**
  - Captures linear relationships
  - $Cov(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$
    - *X* and *Y* are column vectors that consist of n samples (n ≠1)
    - *i*: index variable
    - x_i and y_i: ith elements of vectors *X* and *Y*
    - $\bar{x}$ and $\bar{y}$: mean values of vectors *X* and *Y*

Feature Selection: Correlation

- **Pearson Correlation Coefficient:**
  - Correlation = $\frac{Cov(X,Y)}{\sigma_X \sigma_Y}$

    X and *Y:* vectors with *n* elements:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

  - ρ ∈ [−1, 1]
    - +1: strong positive linear relationship
    - 0: no linear relationship
    - -1: strong negative linear relationship

Feature Selection: Correlation

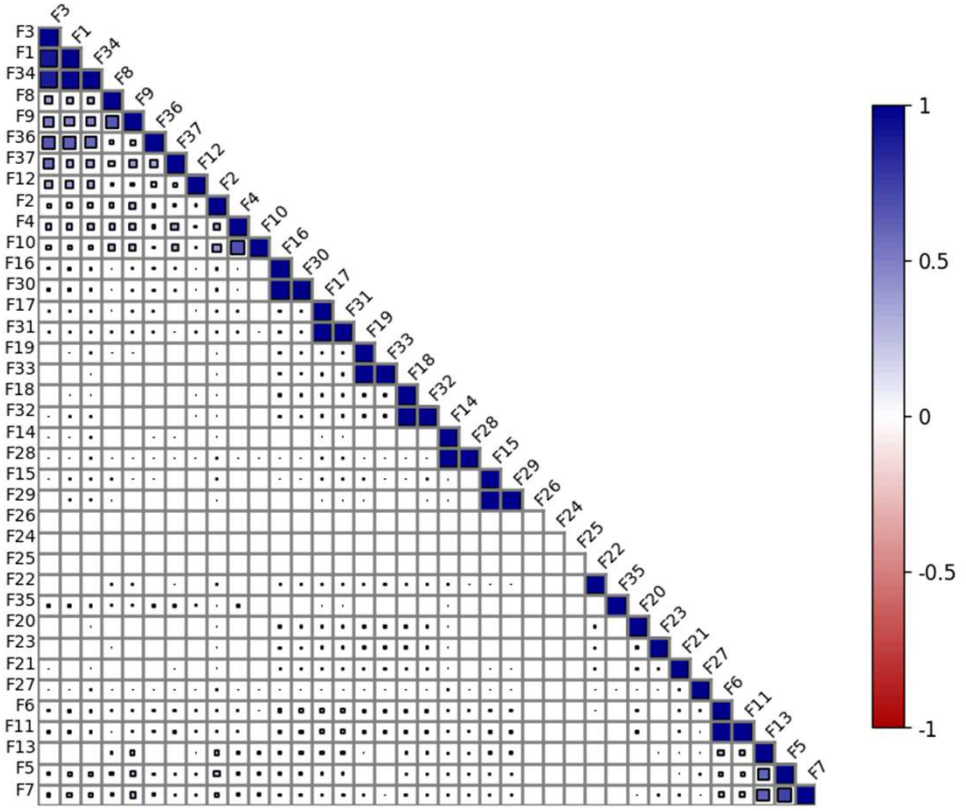
- **Spearman Correlation Coefficient:**
  - Measures rank correlations (degree of similarity)
  - $r_s(X, Y) = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$,

    X and Y: column vectors of size n that consist of ranked values.

    d_i: i^{th} difference between the ranks of the i^{th} values of X and Y
  - $r_s \in [-1, 1]$
    - +1: strong positive non-linear relationship
    - 0: no non-linear relationship
    - -1: strong negative non-linear relationship

Feature Selection: Correlation

- **Pearson and Spearman Correlation:**
  - Highly correlated features (ρ/ $r_s \geq 0.9$) were identified
  - Feature from highly correlated pairs were removed
  - Example: remove F34



Feature Selection: Random Forests

- Multiple decision trees are used to form predictions
- Bootstrap aggregation (bagging) generates:
  - Multiple uncorrelated decision trees
  - Selecting samples with replacements (bagging)
  - Models are trained in parallel
- Random approach: selected subset of features and threshold values for splitting
- Quality of a split is measured based on *Gini impurity*
  - Preferred splits lead to reduced *Gini impurity*
- Each classifier makes a prediction
- Outcome with a majority vote is selected as the output

L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Jan. 2001.

Feature Selection: Extra-Trees

- **Extremely Randomized Trees (Extra-Trees):**
  - Derived from random forest
  - Faster execution time
  - Each decision tree is trained using a complete dataset without sampling
  - Split point for each decision tree is selected randomly
  - Feature scores selected based on *Gini importance*

P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Mach. Learn., vol. 63, no. 1, pp. 3–42, Apr. 2006.

Roadmap

- Motivation and Introduction
- BGP Datasets
- Dimension Reduction and Clustering
- Feature Selection
- **Feature Analysis Based on Goodness of Fit Test**
- Machine Learning Approaches
- Performance Evaluation
- Conclusion and References

Feature Analysis

- Machine learning heavily depend on data
- Data distributions: mathematically represented using probability distribution functions (PDFs)
- Characteristics defined by:
  - Mean, standard deviation, skewness, kurtosis
- BGP features are analyzed to estimate the best fitting distributions:
  - Analyze skewness of worms and ransomware datasets

Feature Analysis: Goodness of Fit Test

- **Goodness of Fit Kolmogorov – Smirnov (K – S) Test:**
  - Compares the reference probability distributions with the sampled data distribution
  - Calculate the difference D_n between the cumulative distribution functions (CDFs):

$$\max |F_n^1(x) - F_n^2(x)|$$

    - $F_n^1(x)$ and $F_n^2(x)$ are CDFs of random variable x
    - selected reference probability distributions compared with distribution of sampled data
    - two data distributions are selected

Feature Analysis: Goodness of Fit Test

- **Probability distributions selected to estimate BGP features:**
  - Gaussian (normal), exponential, gamma
  - Heavy-tailed:
    Weibull, Rayleigh, Burr, t Location-Scale, log-normal, and log-logistic
- Traffic traces in communication networks are often heavy-tailed

Y. Dodge, *The Concise Encyclopedia of Statistics*.
New York, NY: Springer New York, 2008, pp. 283–287.
N. T. Thomopoulos, *Statistical Distributions Applications and Parameter Estimates*.
Cham, Switzerland: Springer Nature, 2017.
A. Alzaatreh, C. Lee, and F. Famoye, "A new method for generating families of continuous distributions,"
METRON, vol. 71, pp. 63–79, June 2013.

Machine Learning: Classification

- **Support Vector Machine (SVM)**
- **Deep learning networks**
  - Long-Short Term Memory (LSTM)
- **Learning rate scheduling**
- **Ensemble learning:**
  - Boosting: Gradient Boosting Decision Trees (GBDT)

Support Vector Machine (SVM)

- **Supervised machine learning:**
  - Generates decision boundary (hyperplane) to separate data points into distinct classes
  - Decision surface is generated to maximize the distance (margin) between the closest data points belonging to distinct classes
- **Polynomial, Gaussian radial basis function, sigmoid kernels**
- **Regularization parameter is used to modify the decision boundary:**
  - Hard-margin: high regularization parameter (prone to overfitting)
  - Soft-margin: low regularization parameter

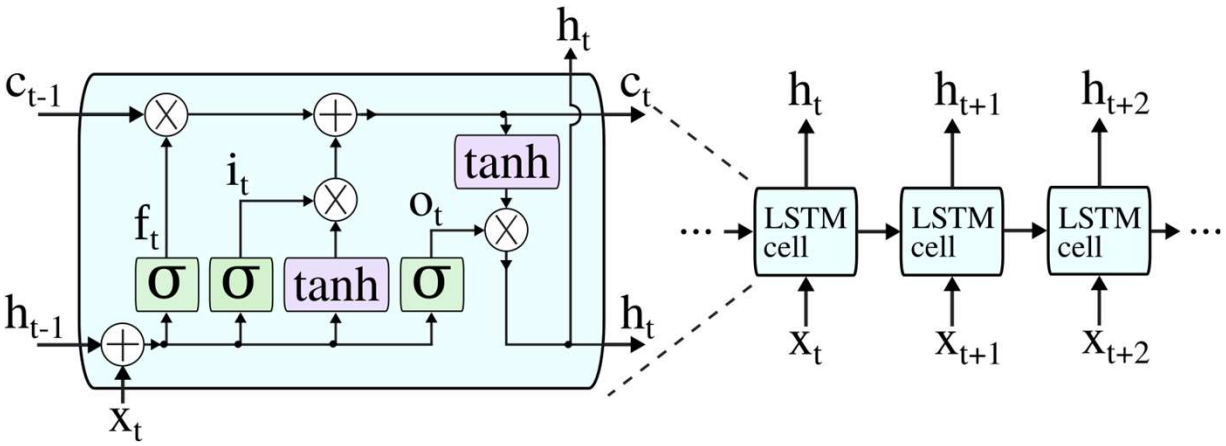
Deep Learning Networks

- **Deep Learning Networks:**
  - Widely advocated for their performance with large datasets
  - Examples: multi layer perceptron (MLP), recurrent neural networks (RNNs)
- **RNNs:**
  - Perform better using sequential data
  - Suffer from vanishing or exploding gradient problem
  - Long-Short Term Memory (LSTM)
  - Gated Recurrent Unit (GRU)

I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning.
Cambridge, MA, USA: The MIT Press, 2016.
Graves, "Long short-term memory," in *Supervised Sequence Labelling with Recurrent Neural Networks*.
Berlin, Heidelberg: Springer, 2012, pp. 37–45.

Deep Learning Network: LSTM

- **Long-Short Term Memory (LSTM):**
  - Does not suffer from vanishing gradient problem
  - Effectively learns time sequences
  - Three gates:
    - input
    - forget
    - output



S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Oct. 1997.

LSTM Cell: Equations

- Input (i_t) = $\sigma(W_{ii}x_t + b_{ii} + U_{hi}h_{t-1} + b_{hi})$
- Forget (f_t) = $\sigma\left(W_{if}x_t + b_{if} + U_{hf}h_{t-1} + b_{hf}\right)$
- Output (o_t) = $\sigma(W_{io}x_t + b_{io} + U_{ho}h_{t-1} + b_{ho})$
  - x^t : input at time t
  - W and U: respective weights
  - h^t : output at t
  - b: bias vectors
- Current cell state: $c_t = f_t \otimes c_{t-1} + i_t \otimes tanh(W_{ic}x_t + b_{ic} + U_{hc}h_{t-1} + b_{hc})$
- Output: $h_t = o_t \otimes tanh(c_t)$
  - $\otimes$: elementwise multiplication

S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Oct. 1997.

Learning Rate

- Machine learning models are trained to obtain the lowest possible loss:
  - Analytical: least-squares
  - Iterative: gradient descent
    - learning rate: an optimizer hyperparameter
- Desired loss function is differentiable and convex
- Loss function be multi-modal and contain saddle points:
  - Fast learning rate: may skip the optimal minima
  - Slow learning rate: may get stuck in a saddle point or very slowly converge to the desired minima
  - Ideal learning rate: gradually converges to the loss minima

I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts using statistical probability distribution," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France: OpenReview.net, Apr. 2017.

Learning Rate

- Optimizer hyperparameter for gradient-based iterative approaches:
  - Rate at which model hyperparameters are updated to reduce loss
  - Important to control the rate of updating hyperparameters:
    - avoid skipping the optimal loss minima or getting stuck in saddle points
- Dynamically update learning rate instead of using a constant value:
  - Beneficial to initialize the learning rates to high values to obtain suitable model weights
  - Then fine-tune model by slowly reducing the learning rate in order to converge gradually to a minima
- Cosine annealing: a periodic dynamic learning rate scheduler

I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts using statistical probability distribution," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France: OpenReview.net, Apr. 2017.

Learning Rate Scheduler

- Pre-defined static or dynamic functions employed to update the learning rate hyperparameter
- Dynamically update learning rate instead of using a constant value
- Beneficial to initialize the learning rates to high values to obtain suitable model weights
- Model is then fine tuned by slowly reducing the learning rate in order to converge gradually to minimum
- Cosine annealing: a periodic dynamic learning rate scheduler

Attention Mechanism

- Length of the sequence: a challenge when dealing with sequential and temporal data
- For long sequences:
  - Gradient-based learning algorithms suffer from vanishing gradient
  - Past information of long sequences is not retained
- Encoder-decoder model:
  - Last encoder hidden state is encapsulated into a context vector of fixed length
  - Decoder uses context vector to generate decoded output
  - Fixed-length context vector does not encode information of earlier input sequences
- Overcomes shortcoming of fixed-length context vector

Attention Mechanism

- Selective information from a sequence is used instead of the entire sequence:
  - Uses weighted combinations of hidden states of the encoder
  - Emphasizes the contribution of the local sequences
- Context vector in encoder-decoder model is of fixed length:
  - Context vector: last encoder hidden state
  - Long sequences: information of earlier input sequences is lost

Attention mechanism in deep learning, explained.
[Online]. Available: https://bit.ly/3GWI7ug. Accessed: Apr. 2023.
A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. ukasz Kaiser, and I. Polosukhin,
"Attention is all you need," *in Proc. Advances Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan,
and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

Attention Mechanism

- Alignment scores: $e_{t,i} = a(s_{t-1}, h_i)$

  s_{t-1} = previous decoder hidden state

  h_i = hidden encoder state

- Attention weights: $\alpha_{t,i} = softmax(e_{t,i})$
- Context vector (variable length) at time t:

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} h_i$$

T = number of hidden states

Attention mechanism in deep learning, explained. [Online]. Available: https://bit.ly/3GWI7ug. Accessed: Apr. 2023.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. ukasz Kaiser, and I. Polosukhin, "Attention is all you need," *in Proc. Advances Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

Ensemble Learning

- Branch of machine learning that combines multiple independent models called weak (base) learners to generate a final high performing model
- Final model has lower bias and variance
- Three variants:
  - Bootstrap aggregation (Bagging)
  - Boosting
  - Stacking

X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers Comp. Sci.*, vol. 14, pp. 241–258, Apr. 2020.

Ensemble Learning: Bagging

- Bootstrapping re-sampling technique: uniformly samples data using replacement to generate training datasets
- Data point might appear multiple times in a given training dataset
- Selected bootstrap samples are then used to independently train multiple base learners in parallel
- Employs majority vote to obtain final output

Ensemble Learning: Boosting

- Sequentially combines models to generate an optimal model
- Heavier weights are assigned to the misclassified data points
- Next model is generated by using these weighted data points for training (emphasis on misclassified data points)
- Boosting: adaptive and gradient
- Example: Gradient boosting decision tree algorithms employ decision trees (base learners) and are variants of gradient boosting machines (GBM):
    - eXtreme gradient boosting (XGBoost): asymmetrically level-wise
    - Light gradient boosting (LightGBM): asymmetrically leaf-wise
    - Categorical boosting (CatBoost): symmetrically

GBDT Algorithms

- Use decision trees as base learners
- Predicted output $\hat{y}$ of GBDT model:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

K: number of estimators, f_k: kth decision tree, and x^i : ith row vector of input data matrix *X*

$$\hat{y}_i^{(k)} = \hat{y}_i^{(k-1)} + f_k(x_i)$$

$\hat{y}_i^{(k)}$: predicted output during the kth iteration

$\hat{y}_i^{(k-1)}$: predicted output during the previous iteration

GBDT Algorithms

- Objective function of GBDT model:

$$L^{(k)} = \sum_{i=1}^{N} l\left(y_i, \hat{y}_i^{(k)}\right) + \Omega(f_k)$$

  - l(.): loss function
  - y_i: expected label of the i^{th} input
  - $\Omega(f_k)$: optional regularization term

GBDT Algorithms

- XGBoost
- LightGBM
- CatBoost

T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: a highly efficient gradient boosting decision tree," in *Proc. Int. Conf. Neural Inform. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 3146–3154.

L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Proc. Int. Conf. Neural Inform. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 6639–6649.

GBDT: XGBoost

- Scalable tree boosting algorithm that optimizes the loss function by adding the regularization term
- Loss function can be optimized using iterative approach due to presence of functions as parameters
- Histogram-based splitting using each feature and then selects the optimal split
- Asymmetric trees that grow level-wise (depth)

T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

GBDT: XGBoost

- L^2 regularization term $\Omega(.)$ added to reduce the loss function:

$$\Omega(f_k) = \gamma T + \frac{1}{T}\lambda\|\omega\|^2$$

  - L^2: regularization term $\Omega(.)$
  - γ and λ: regularization coefficients
  - T: number of leaves in the tree
  - ω: leaf weights

GBDT: LightGBM

- Histogram-based Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) employed to enhance training speed
- Requires lower memory than XGBoost algorithm
- Employs histogram-based approach to discover faster the best splitting point for each feature:
  - Feature histograms created by bundling together mutually exclusive features
  - GOSS technique sorts training data in descending order based on the absolute gradient values
  - GOSS relies on using data points with high gradient values
  - Employs asymmetric leaf-wise growth

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: a highly efficient gradient boosting decision tree," in *Proc. Int. Conf. Neural Inform. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 3146–3154.

GBDT: CatBoost

- Handles categorical features
- Ordered target statistic approach:
  - Ordered boosting employs permutations to train and evaluate decision trees using different sets of samples
  - Weighted Minimal Variance Sampling (MVS) technique: each point selected at least once
- Trees are grown level-wise using the same splitting criteria for a given tree level (oblivious trees):
  - Less prone to overfitting

L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Proc. Int. Conf. Neural Inform. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 6639–6649.

Performance Metrics

- Confusion Matrix
- Precision
- Sensitivity
- Accuracy
- F-Score

Accuracy vs. F1-Score.
[Online]. Available: https://bit.ly/3ofqoe2. Accessed: Apr. 2023.

Performance Metrics

- **Confusion matrix:**
  - **True positive (TP)**: anomalous data point being classified as an anomaly
  - **False positive (FP)**: regular data point being classified as an anomaly
  - **True negative (TN)**: regular data point being classified as regular
  - **False negative (FN)**: anomalous data point being classified as regular

		Predicted Class	
		Regular	Anomaly
Actual Class	Regular	TN	FP
	Anomaly	FN	TP

Performance Metrics

- **Precision:**
  - Measures the correctly identified positive from all predicted positive cases

$$\frac{TP}{TP + FP}$$

  - Useful when the costs of false positives is high
- **Sensitivity (recall):**
  - Measures the correctly identified positive from all actual positive cases

$$\frac{TP}{TP + FN}$$

  - Useful when the cost of false negatives is high

Performance Metrics

- **Accuracy**:
  - Measure of the correctly identified cases:

$$\frac{TP + TN}{TP + \mathrm{TN} + FP + FN}$$

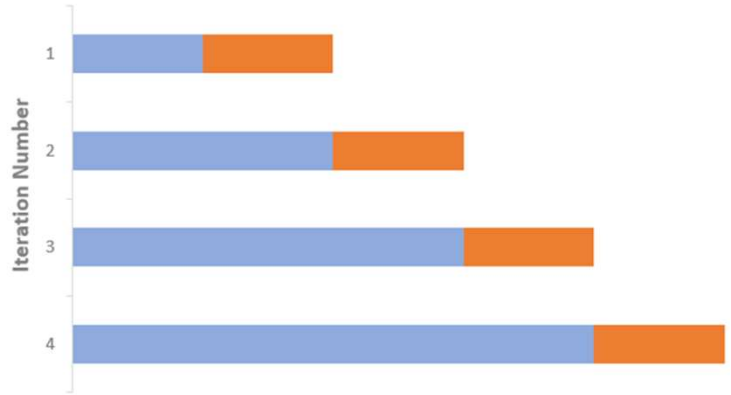
  - Used when all classes are equally important
- **F-Score**:
  - Harmonic mean of precision and sensitivity:

$$\frac{2(precision \times sensitivity)}{precision + sensitivity}$$

  - Better for unbalanced datasets
  - Better measure of incorrectly classified cases than accuracy

Cross-Validation

- Time series split cross-validation:
  - Variation of *10*-fold cross validation
  - Training (blue) and test (orange) datasets
  - Successive training datasets are concatenated over time
  - Maintains time sequence of sequential data



Roadmap

- Motivation and Introduction
- BGP Datasets
- Dimension Reduction and Clustering
- Feature Selection
- Feature Analysis Based on Goodness of Fit Test
- **Performance Evaluation:**
  - Feature Selection
  - Goodness of Fit Test
  - Classification: SVM, LSTM, GBDT
- Conclusion and References

Feature Selection: Random Forests

- 10-fold time-series split cross-validation experiments performed based on accuracy or F-Score

Dataset	Feature numbers in order of importance	No. of estimators
Code Red	34, 1, 3, 4, 12, 36, 9, 37, 8, 10, 2, 5, 11, 6, 7, 13	10
Nimda	1, 4, 34, 3, 12, 36, 9, 37, 8, 10, 6, 2, 11, 13, 7, 5	110
Slammer	1, 34, 36, 12, 4, 3, 10, 2, 8, 9, 13, 37, 6, 11, 7, 5	210
WannaCrypt	4, 8, 10, 3, 9, 2, 1, 34, 36, 37, 12, 6, 11, 13, 35, 7	90
WestRock	36, 1, 8, 3, 9, 34, 10, 37, 4, 11, 2, 6, 12, 22, 5, 13	180

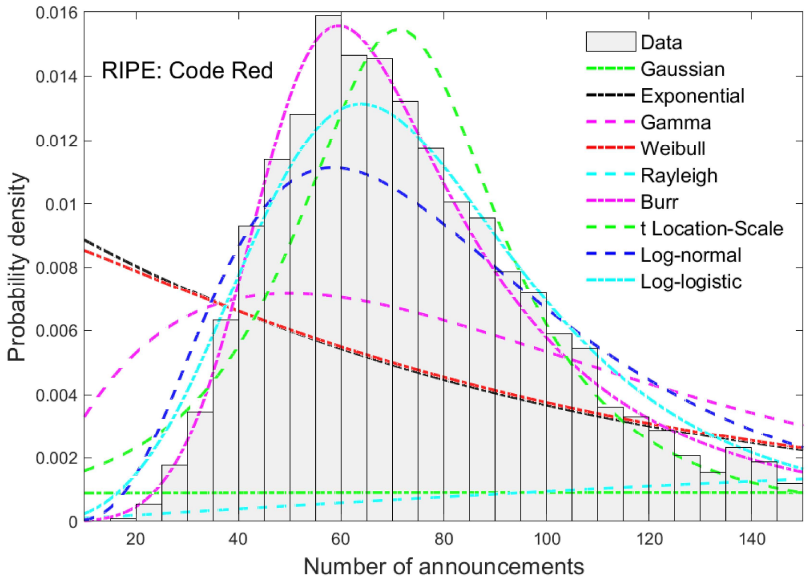
Feature Selection: Extra-Trees

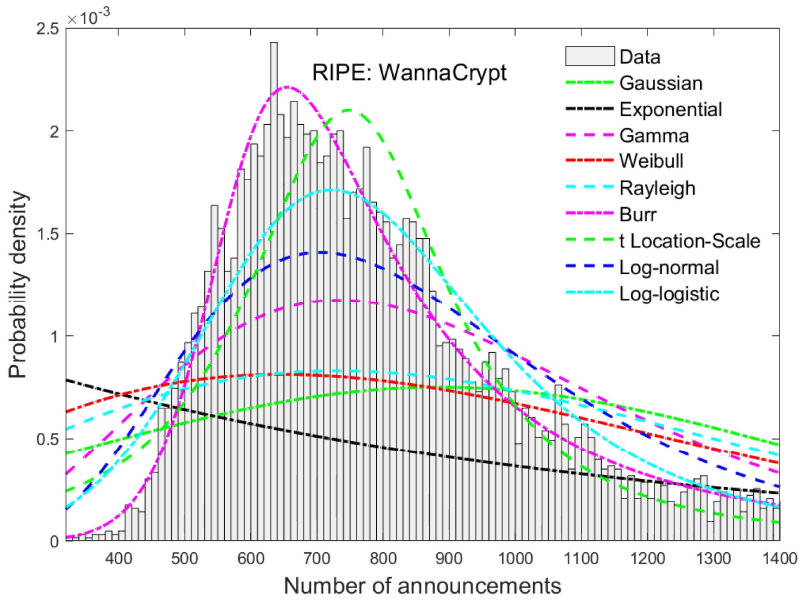
- 10-fold time-series split cross-validation experiments performed based on accuracy or F-Score
- Model hyperparameters: Number of estimators: 500; maximum tree depth: 20

Dataset	Feature numbers in order of importance
Code Red	34, 1, 4, 3, 12, 2, 9, 37, 36, 8, 10, 13, 5, 7, 35, 6
Nimda	1, 34, 3, 4, 9, 36, 12, 37, 8, 23, 10, 2, 13, 7, 11, 5
Slammer	36, 1, 9, 34, 10, 8, 3, 4, 2, 20, 11, 12, 6, 13, 5, 7
WannaCrypt	4, 8, 2, 3, 10, 37, 1, 34, 36, 9, 12, 13, 35, 11, 6, 7
WestRock	8, 9, 3, 37, 2, 1, 36, 34, 10, 4, 12, 35, 13, 6, 11, 7

Goodness of Fit Test

- **Nine probability distributions were fitted to the data:**
  - Top 10 important BGP features selected using extra-trees





Goodness of Fit Test

- Suitable candidates: Burr, *t* location-scale, log-normal, and log-logistic PDFs selected based on visual inspection
- Evaluate statistic measures:
  h, *p*-value, *k*, and *c*

Distribution	**α = 0.10**	**α = 0.05**	**α = 0.01**
Burr:	*p*-value = 0.292473, k = 0.015371		
F3: Code Red			
h	**0**	**0**	**0**
c	0.019214	0.021325	0.025565

Goodness of Fit Test

- Features and distributions with accepted null hypothesis
- Highlighted are accepted distributions based on high *p*-value ≥ α = 0.05 for respective features

Dataset	Distribution	Feature
Code Red	Burr	**F34, F1, F3, F9**, F37
Nimda	Burr/**Log-normal**/ Log-logistic	**F9**
Slammer	Burr	F3
WannaCrypt	Burr	**F4, F3, F10, F1, F34**, F36, F9
WestRock	Burr	**F9**, F4

Goodness of Fit Test

- **Code Red and WannaCrypt** datasets:
  - Common features (F34, F1, and F3) accepted by Burr distribution indicating similarities between the two datasets
  - WannaCrypt, being a cryptoworm, propagates through a network using similar self-replication and self-propagation techniques employed by worms
- **Worm datasets**: no common features with same accepted null hypothesis
- **Code Red and WestRock** datasets: feature F9 follows the Burr distribution
  - Number of implicit withdrawals (F9) is the number of newly advertised AS-paths for the already announced NLRI prefixes
  - Indicates that during these attacks traffic may have been re-routed through desired AS-paths by the attacker

NLRI: Network layer reachability information

Classification: SVM, LSTM, GBDT

- **SVM**
- **LSTM:**
  - Learning rate scheduling
  - Attention mechanism
- **GBDT:**
  - Classification using PCA transformed data
  - Classification using feature selection techniques:
    - Pearson correlation
    - Spearman correlation
    - random forests
    - extra-trees

Classification: SVM

- Linear kernel:
  - **Best F-Score**: 73.54 % using **WestRock** dataset
  - o F-Score using Code Red dataset:
    - not linearly separable
- SVM models generated using high regularization coefficient values result in high F-Scores and are not considered:
  - May suffer from overfitting due to hard-margin kernels

Classification: LSTM

- **Using learning rate scheduling:**
  - WannaCrypt: accuracy: 68.06 % (65.48 %*); F-Score: 66.27 % (63.22 % *)

Parameter	Published	Improved
Length of sequence	100	100
No. of epochs	30	30
No. of hidden nodes	$FC_1, FC_2, FC_3 = 64, 32, 16$	$FC_1, FC_2, FC_3 = 64, 2, 16$
Dropout rate	0.4	0.4

- **Using attention mechanism:**
  - WestRock: Best F-Score 76.92 %

Z. Li, A. L. Gonzalez Rios, and Lj. Trajković, "Detecting internet worms, ransomware, and blackouts using recurrent neural networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Toronto, Canada, Oct. 2020, pp. 2165–2172.

GBDT Classification and PCA

- **GBDT** models: offer short training time
- Experiments are performed to evaluate performance of GBDT model:
  - Various feature selection approaches
  - Normalized and unnormalized data:
    - high performance using unnormalized data
- **Dimension reduction using 10 PCA components:**
  - 10-fold time series split cross-validation experiments:
    - best F-Score: **WestRock** dataset:
      - **XGBoost**: 71.81 %

GBDT Classification and Pearson Correlation

- Feature selection using **Pearson correlation**:
  - 10-fold time series split cross-validation experiments based on F-Score
  - **Best F-Score**:
    - **Code Red XGBoost** model: 80.65 %
    - **WestRock LightGBM** model: 70.94 %
    - LightGBM model offers shorter training time
    - high sensitivity: detects true positives at a higher rate

GBDT Classification and Spearman Correlation

- Feature selection using **Spearman correlation**:
  - 10-fold time series split cross-validation experiments based on F-Score
  - **Best F-Score:**
    - **Code Red CatBoost** model: 76.58 %
    - **WestRock XGBoost** model: 72.04 %
- **Code Red LightGBM** model:
  - Model did not learn data properties due to highly unbalanced data

GBDT Classification and Random Forests

- Feature are selected using **random forests**:
  - 10-fold time series split cross-validation:
    - based on accuracy or F-Score
  - **Best F-Scores**:
    - **Code Red CatBoost** model (37 features): 81.30%
    - **WestRock CatBoost** model (8 features): 66.90%
- **Code Red LightGBM model:**
  - Did not learn data properties due to highly unbalanced data

GBDT Classification: Extra-Trees

- Feature selection using **extra-trees:**
  - Based on accuracy or F-Score:
    - grid search: exhaustive ad hoc approach
    - 10-fold time series split cross-validation: reliable approach
  - **Best F-Scores**:
    - **CatBoost Code Red** model (37 features): 81.30%
    - **XGBoost WestRock** model (8 features): 72.96%

Roadmap

- Motivation and Introduction
- BGP Datasets
- Dimension Reduction and Clustering
- Feature Selection
- Feature Analysis Based on Goodness of Fit Test
- Performance Evaluation
- **Discussion and Conclusion**
- References

Discussion

- Machine learning model obtain high accuracy using worm datasets
- Considered worms and ransomware datasets have been collected decades apart and employ different attack mechanisms
- Anomalous network activities were easier to observe in the early development of the Internet
- Internet expansions, increased digital presence, device connectivity, and malicious activity have impacted traffic behavior and have made detection of anomalous activities challenging
- While increased traffic volume was easily observed during the worm attacks, the distinction between regular and anomalous traffic during the ransomware attacks is less evident

Conclusion

- **Intrusion detection systems** should employ machine learning models:
  - Short training time:
    - scalability
    - deployment in real-time environments
- Machine learning intrusion detection techniques offer effective solutions to identify and detect cyberattacks
- Various dimension reduction and feature selection techniques are applied to identify important features and enhance the performance of the machine learning models

Conclusion

- **Dimension Reduction:**
  - PCA – transformed data to contain 10 PCA components
  - **Best F-Score**: **WestRock** dataset
- **Clustering:**
  - **PCA and *k*-means**: Enhanced separation between anomaly and regular class data using **WannaCrypt** dataset

Conclusion

- **Goodness of Fit K-S test**:
  - Number of features follow heavy-tailed probability distributions
  - Underlying similarities between Code Red and WannaCrypt dataset
- **SVM models** using **linear kernels:** offer high F-Scores when generated using WestRock dataset
- Performance of best performing **LSTM model** was enhanced by using learning rate scheduling
- **Dynamic learning rates** offer higher F-Score than constant learning rates
- **GBDT models**: offer effective performance using extra-trees to select important features

References: Tools

- **Python:** https://pypi.org
- **Pandas:** https://pandas.pydata.org/
- **MATLAB:** https://www.mathworks.com/
- **Google Colab:** https://colab.research.google.com/

References: BGP and Anomaly Detection

- **RFC**: A Border Gateway Protocol 4 (BGP-4): https://datatracker.ietf.org/doc/html/rfc4271
- **Collection Site:**
  - RIPE NCC: https://www.ripe.net/analyse
- **IEEE DataPort:** Border Gateway Protocol (BGP) datasets: https://ieee-dataport.org/open-access/border-gateway-protocol-bgp-routing-recordsreseaux-ip-europeens-ripe-and-bcnet
- **Surveys:**
  - V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, July 2009.
  - B. Al-Musawi, P. Branch, and G. Armitage, "BGP anomaly detection techniques: a survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 377–396, 2017.

References: Feature Selection and Analysis

- L. Breiman, "Random forests," *Mach. Lear.*, vol. 45, no. 1, pp. 5–32, Jan. 2001.
- P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.
- F. J. Massey, "The Kolmogorov-Smirnov test for goodness of fit," *J. American Statistical Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.

References: Machine Learning

- I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: The MIT Press, 2016.
- K. P. Murphy, *Probabilistic Machine Learning: An introduction*. Cambridge, MA, USA: The MIT Press, 2022.
- C. Cortes and V. Vapnik, "Support-vector networks," *J. Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sept. 1995.
- S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Oct. 1997.
- D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, May 2015.

References: Machine Learning

- T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
- G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: a highly efficient gradient boosting decision tree," in *Proc. Int. Conf. Neural Inform. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 3146–3154.
- L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Proc. Int. Conf. Neural Inform. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 6639–6649.
- I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts using statistical probability distribution," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France: OpenReview.net, Apr. 2017.


Thank you!