Application of Machine Learning Techniques for Detecting Anomalies in Communication Networks

Qingye Ding
qingyed@sfu.ca

Communication Networks Laboratory
http://www.ensc.sfu.ca/~ljilja/
School of Engineering Science
Simon Fraser University

SFU
SIMON FRASER UNIVERSITY

Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

Motivation

- The Internet is a critical asset of information and it contains multiple Autonomous Systems (ASes)
- An AS is a collection of Internet Protocol (IP) routing prefixes administrated by a single domain
- Border Gateway Protocol (BGP) plays an essential role in routing data between ASes
- Cyber attacks and threats significantly impact the Internet performance

Border Gateway Protocol

- Forwards IP traffic between Autonomous Systems (ASes)
- BGP 4: a standard for exchanging information among the Internet Service Providers (ISPs)
- Relies on the Transport Control Protocol (TCP) to establish a connection between routers
- Exchange the update message to advertise routing information:
  - an available route
  - withdraw multiple unavailable routes

Sample of BGP update message

Field	Value
TIME	2003 1 24 00:39:53
TYPE	BGP4MP/BGP4MP_MESSAGE AFI_IP
FROM	192.65.184.3
TO	193.0.4.28
BGP PACKET TYPE	UPDATE
ORIGIN	IGP
AS-PATH	513 3320 7176 15570 7246 7246
NEXT-HOP	192.65.184.3
ANNOUNCED NLRI PREFIX	198.155.189.0/24
ANNOUNCED NLRI PREFIX	198.155..0/24

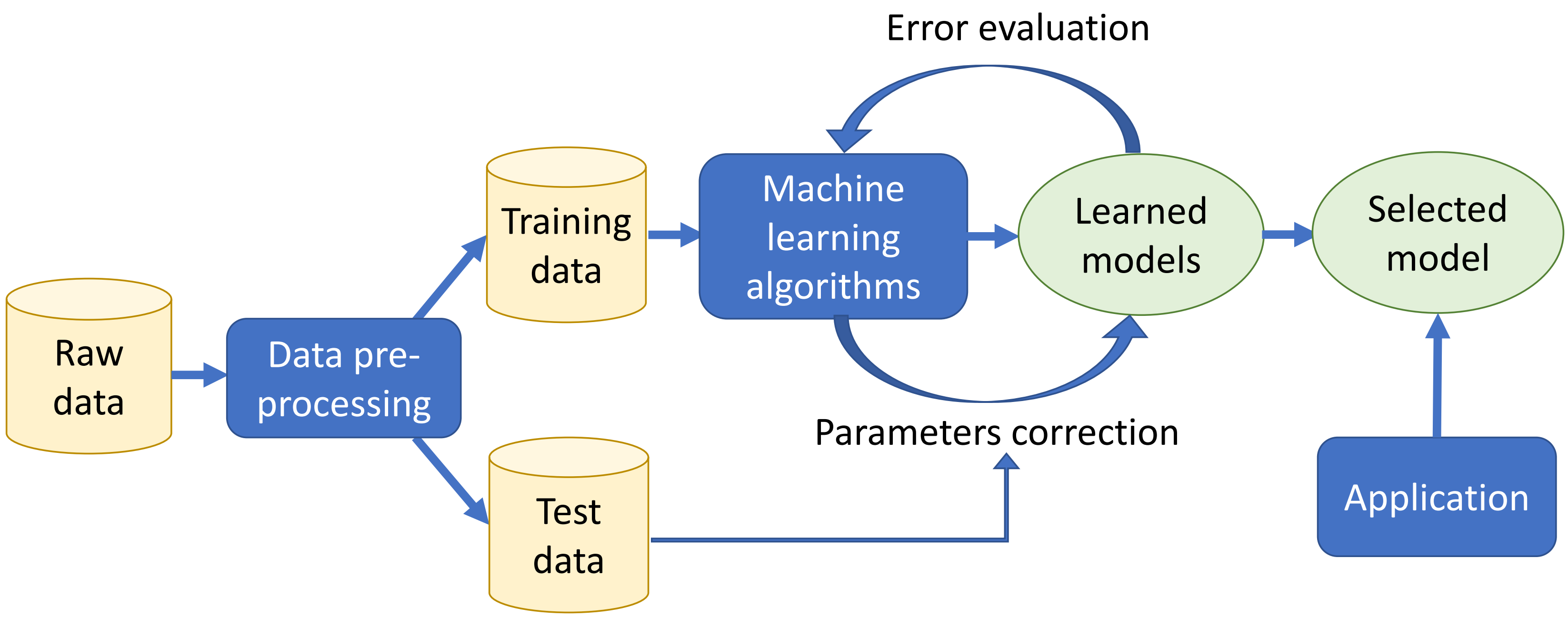
IGP: Interior Gateway Protocol

NLRI: Network Layer Reachability Information

Machine learning techniques

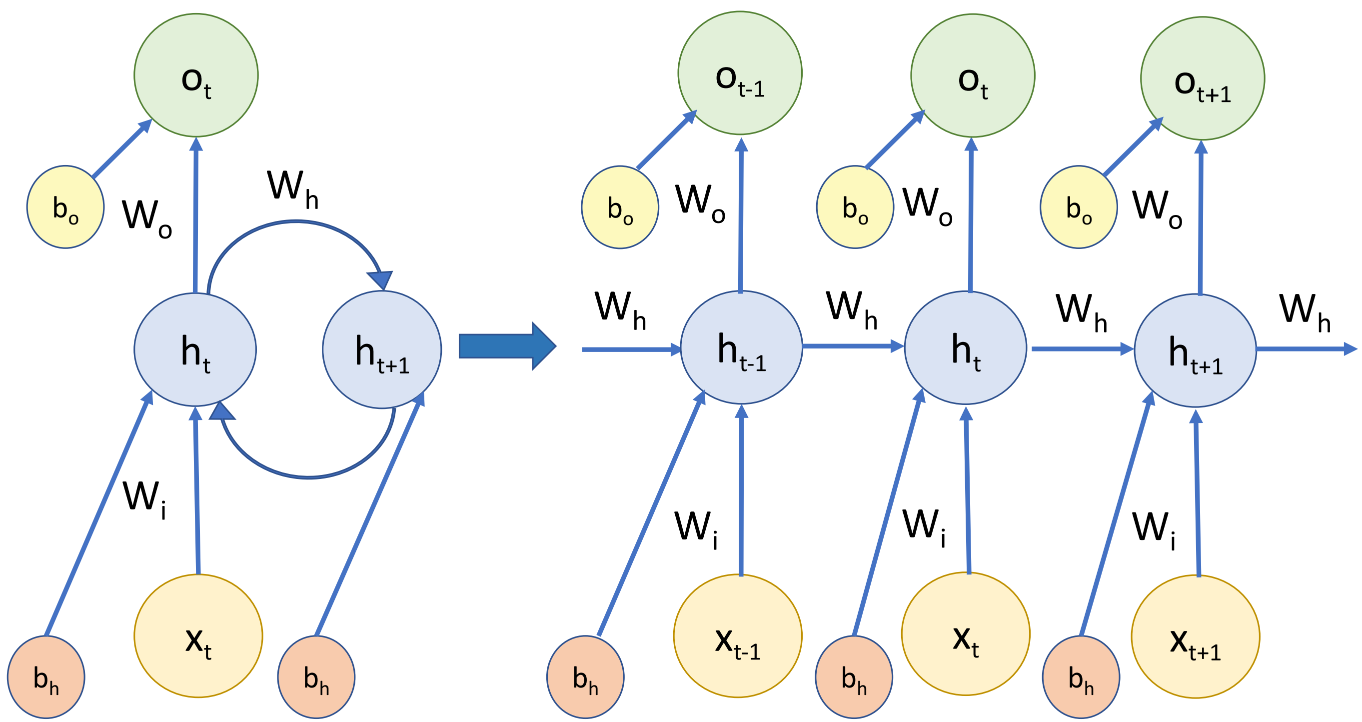
- Unsupervised learning:
  - aims to learn a function that represents the underlying structure from “unlabeled” data
  - motivation: labeled data is difficult to obtain
  - data clustering
- Supervised learning:
  - trains data based on the observation to predict labels for new events
  - Long Short-Term Memory, Support Vector Machine, Naïve Bayes, Decision Tree, and Extreme Learning Machine

Typical procedure for machine learning



Recurrent Neural Network: RNN

- Used for sequence recognition, pattern classification, and temporal prediction tasks



Research contributions

- View detection of BGP anomalies as a classification problem
- Apply Long Short-Term Memory algorithm to develop classification models
- Extract BGP features based on the attributes of BGP update messages
- Create balanced datasets by randomly reducing a subset of regular data points
- Improve classification results emanating from previous studies
- Show feasibility of LSTM for detecting BGP anomalies

Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

BGP anomaly: Slammer

- Attacked Microsoft SQL servers on January 25, 2003
- Generated random IP addresses and replicated itself
- The number of infected machines doubled approximately every 9 seconds
- The update messages consumed most of the routers' bandwidth causing routers to:
  - slow down
  - crash

BGP anomaly: Nimda

- Released on September 18, 2001
- Exploited vulnerabilities in the Microsoft Internet Information Services web servers for the Internet Explorer 5
- Three methods of propagation:
  - email messages
  - web browsers
  - file systems

BGP anomaly: Code Red I

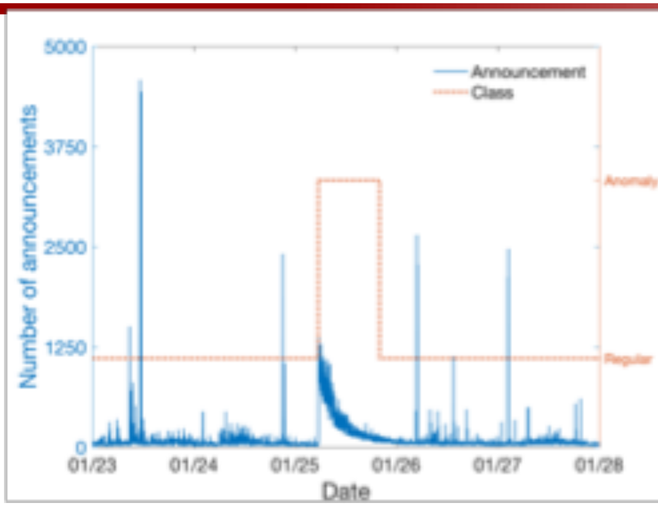
- Attacked web servers on July 19, 2001
- Affected approximately 500,000 IP addresses a day
- Searched for vulnerable servers and replicated itself
- Rate of infection was doubling every 37 minutes

BGP anomalies

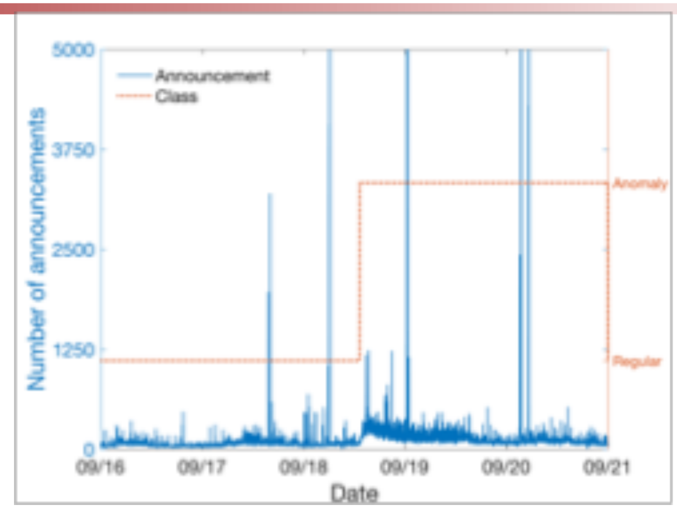
Dataset	Class	Date		Duration
		Beginning of the event	End of the event	(min)
Slammer	Anomaly	25.01.2003 at 5:31 GMT	25.01.2003 at 19:59 GMT	869
Nimda	Anomaly	18.09.2001 at 13:19 GMT	20.09.2001 at 23:59 GMT	3,521
Code Red I	Anomaly	19.07.2001 at 13:20 GMT	19.07.2001 at 23:19 GMT	600

GMT: Greenwich Mean Time

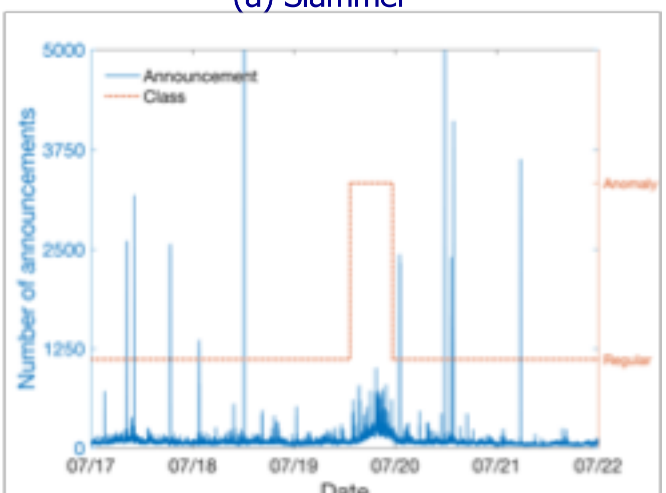
Number of announcements



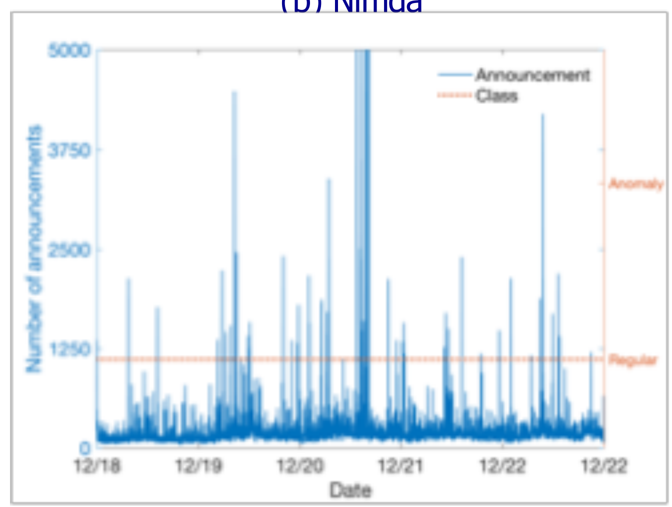
(a) Slammer



(b) Nimda



(c) Code Red I



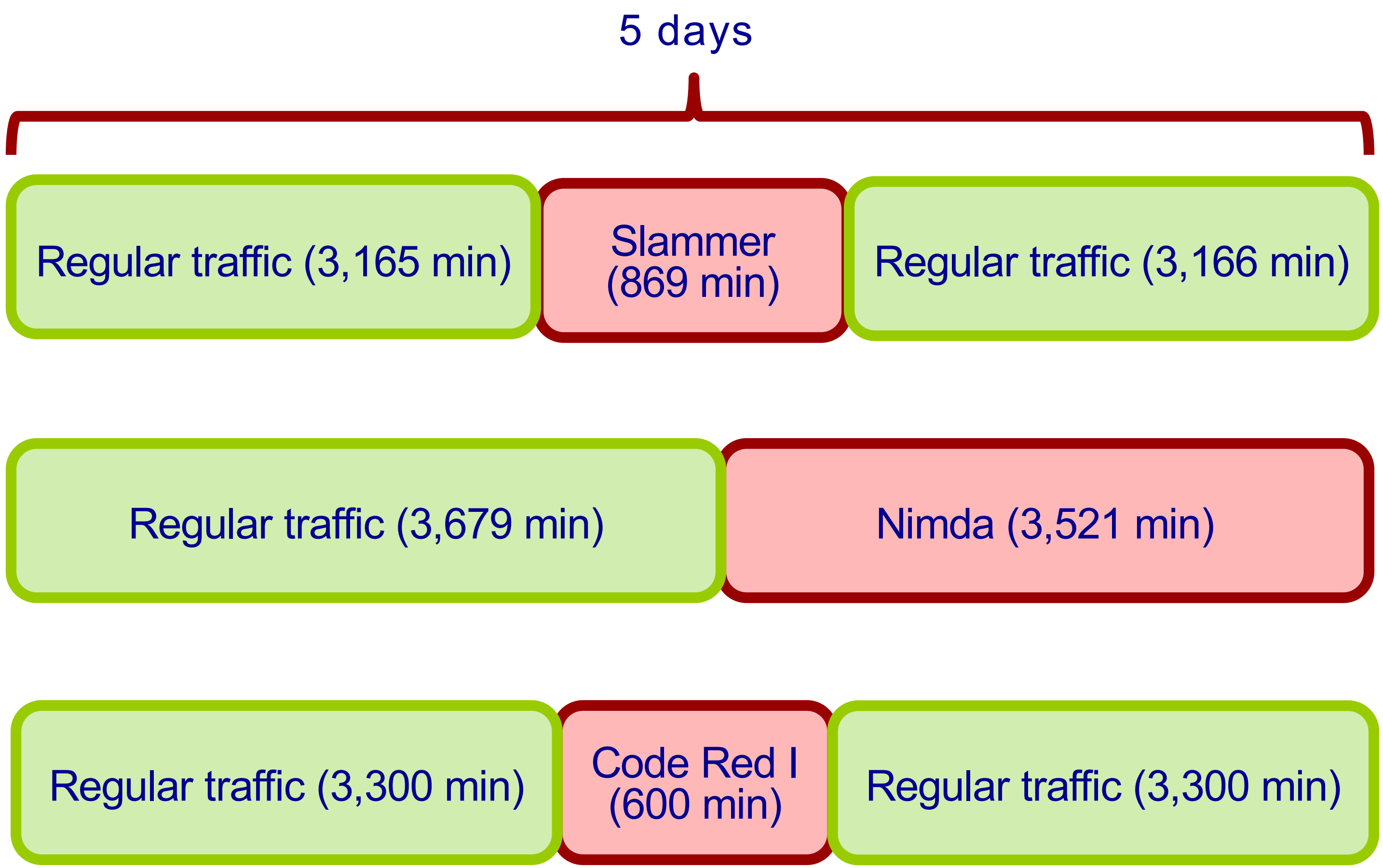
(d) Regular data

BGP datasets

- Réseaux IP Européens (RIPE) Network Coordination Centre:
  - Regional Internet Registry for Europe, Middle East, and parts of Central Asia
  - collects BGP update messages by the remote route collectors (rrc)
  - multi-threaded routing toolkit (MRT) binary format
  - AS 513 (rrc04, CIXP, Geneva, Switzerland)
- BCNET
  - Regular BCNET dataset
  - BCNET location in Vancouver, British Columbia, Canada

CIXP: CERN Internet eXchange Point

Collected data: unbalanced datasets



Collected data: balanced datasets

regular:anomaly = 1:1

Regular traffic (869 min) | Slammer (869 min)

Regular traffic (3,521 min) | Nimda (3,521 min)

Regular traffic (600 min) | Code Red I (600 min)

Pre-processing of the collected data

	Training dataset	Test dataset
1	Slammer and Nimda	Code Red I
2	Slammer and Code Red I	Nimda
3	Nimda and Code Red I	Slammer

- Datasets are concatenated to increase the size of training datasets

Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

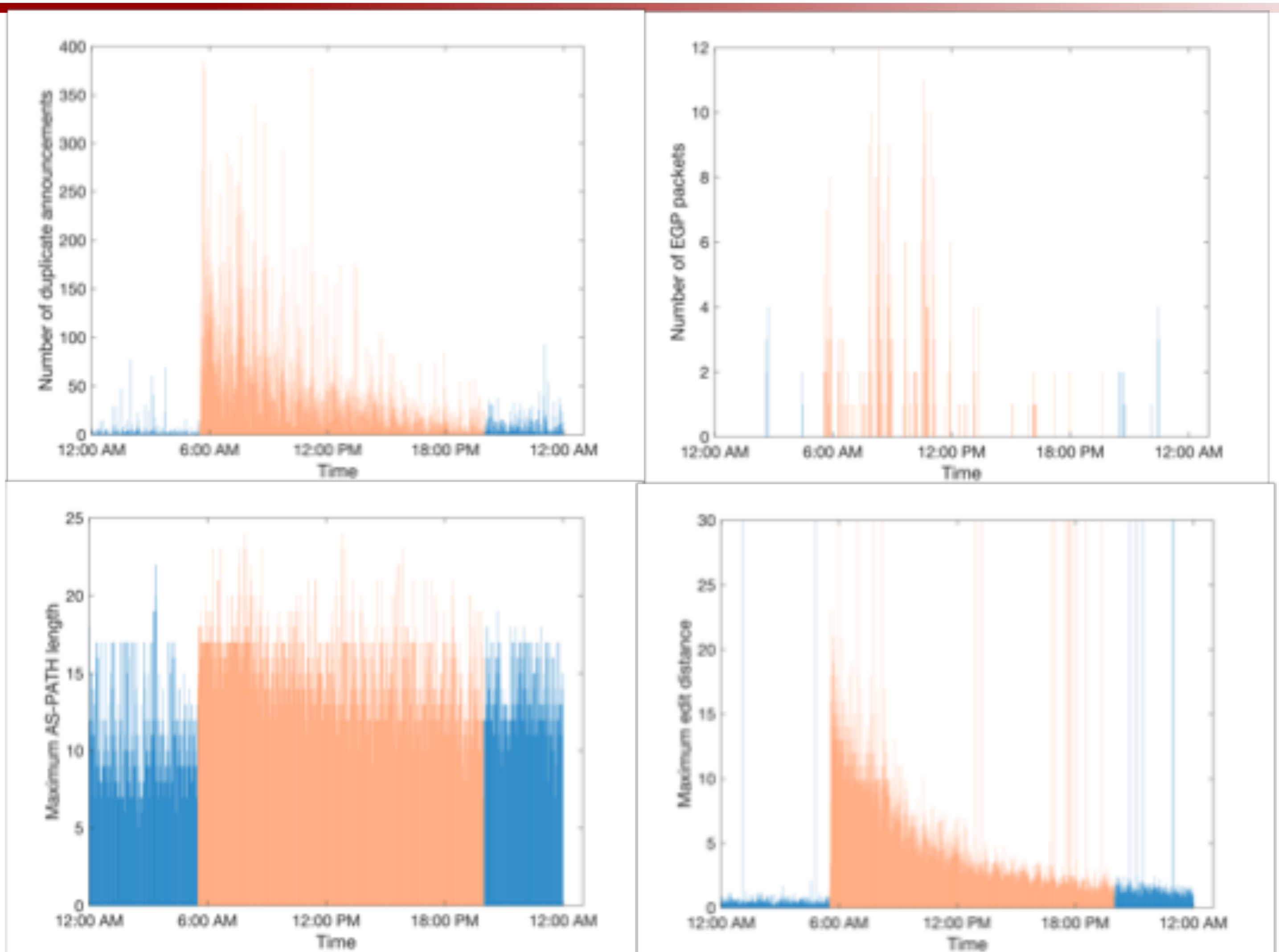
Feature extraction method

- Converted BGP update messages from MRT into American Standard Code for Information Interchange (ASCII) format
- Used LibBGPdump library on a Linux platform
- C# tool was used to extract features:
  - volume
  - AS-path

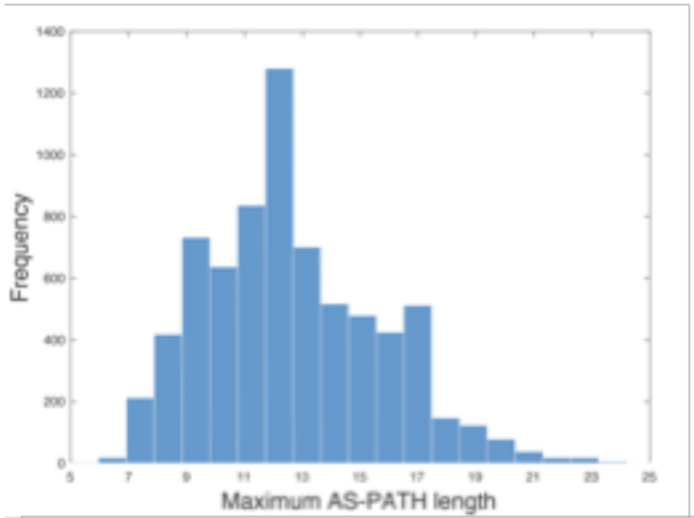
Extracted features

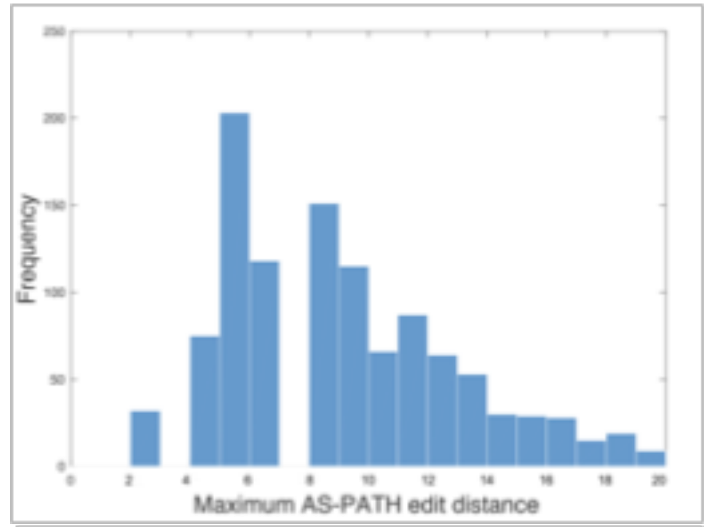
Feature	Name	Category
1	Number of announcements	*volume*
2	Number of withdrawals	*volume*
3	Number of announced NLRI prefixes	*volume*
4	Number of withdrawn NLRI prefixes	*volume*
5	Average *AS-path* length	*AS-path*
6	Maximum *AS-path* length	*AS-path*
7	Average unique *AS-path* length	*AS-path*
8	Number of duplicate announcements	*volume*
9	Number of duplicate withdrawals	*volume*
10	Number of implicit withdrawals	*volume*
11	Average edit distance	*AS-path*
12	Maximum edit distance	*AS-path*
13	Inter-arrival time	*volume*
14-24	Maximum edit distance $= n$, where $n = (7, ..., 17)$	*AS-path*
25-33	Maximum *AS-path* length $= n$, where $n = (7, ..., 15)$	*AS-path*
34	Number of Interior Gateway Protocol (IGP) packets	*volume*
35	Number of Exterior Gateway Protocol (EGP) packets	*volume*
36	Number of incomplete packets	*volume*
37	Packet size (B)	*volume*

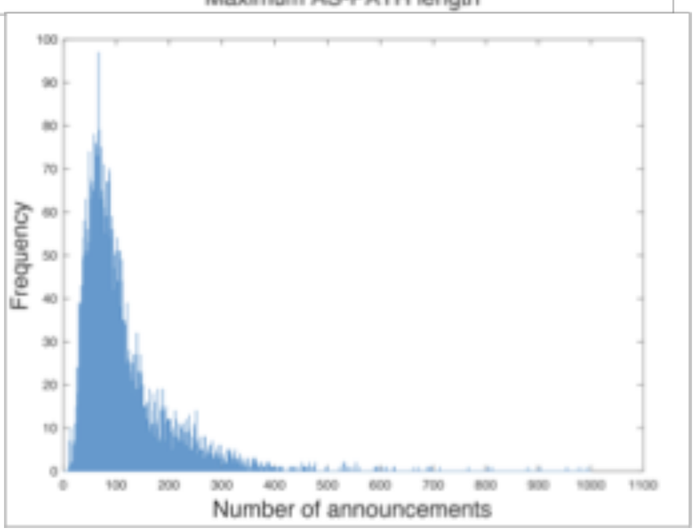
Volume and AS-path features: Slammer worm

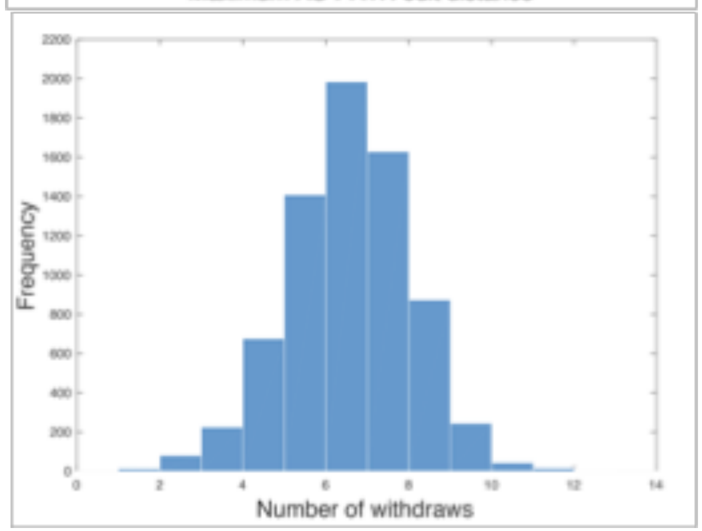


Distribution of features: Slammer worm









Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

Performance metrics

Confusion matrix:

	Predicted class	
Actual class	Anomaly (positive)	Regular (negative)
Anomaly (positive)	TP	FN
Regular (negative)	FP	TN

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{F-Score} = 2\times\frac{precision \times sensitivity}{precision + sensitivity}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{sensitivity (recall)} = \frac{TP}{TP + FN}$$

Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

Long Short-term Memory: LSTM

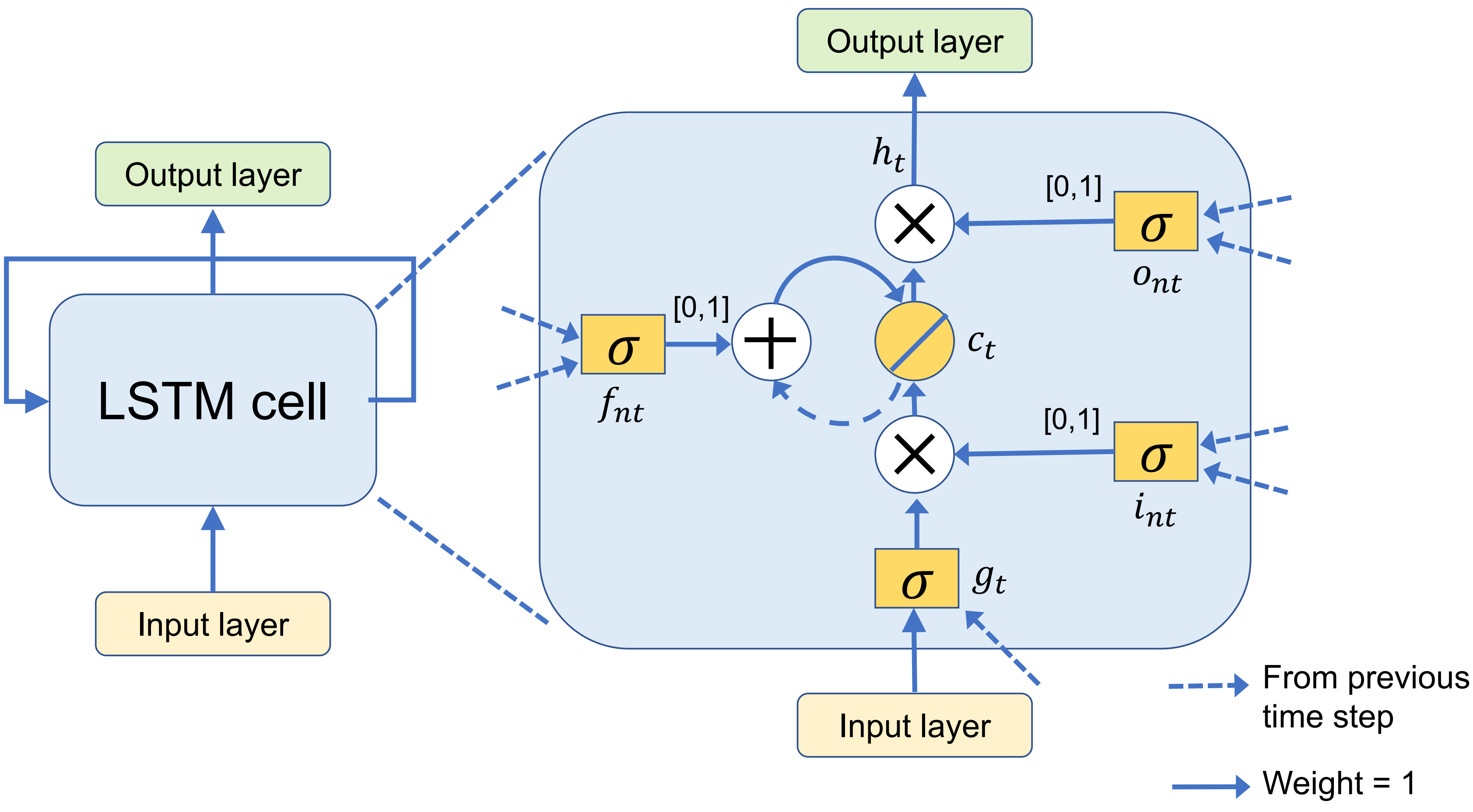
- A special form of the recurrent neural networks (RNNs):
  - LSTM cell (memory block)
- Connects time intervals (short-term memories) to form a continuous memory
- Overcomes long-term dependency
- Prevents vanishing gradient problems

LSTM cell: components

- Input node g_{nt}:
  - contains input information
- Input gate i_{nt}:
  - controls the information to be updated in the LSTM cell
- Internal state c_t:
  - stores the cell's memory
- Forget gate f_{nt}:
  - determines whether to remember or discard the memories
- Output gate o_{nt}:
  - filters and clears irrelevant memories

n: The n th LSTM cell

LSTM repeating module



LSTM components

Input node: $g_{nt} = tanh(U_{gn}h_{t-1} + W_g x_t + b_{gn})$

Input gate: $i_{nt} = \sigma(U_{in}h_{t-1} + W_i x_t + b_{in})$

Forget gate: $f_{nt} = \sigma(U_{fn}h_{t-1} + W_f x_t + b_{fn})$

Output gate: $o_{nt} = \sigma(U_{on}h_{t-1} + W_o x_t + b_{on})$

Internal state: $c_t = f_{nt} * c_{t-1} + i_{nt} * tanh(U_c h_{t-1} + W_c x_t + b_c)$

Output node: $h_t = o_{nt} * ReLU(c_t)$

ReLU: Rectified Linear Unit

LSTM components

- $tanh$: tangent activation function
- U_* and W_*: weight parameters
- h_{t-1}: hidden layer at the previous time step
- x_t: input at the current time step
- b_{*n}: bias of the *n*th LSTM cell
- σ: sigmoid activation function

Internal state: actions

Input gate	Forget gate	Actions
0	1	Keep the memory from the previous time step
1	1	Add the current information to the memory
0	0	Discard both current and past information
1	0	Overwrite the memory by current information

LSTM classification procedure

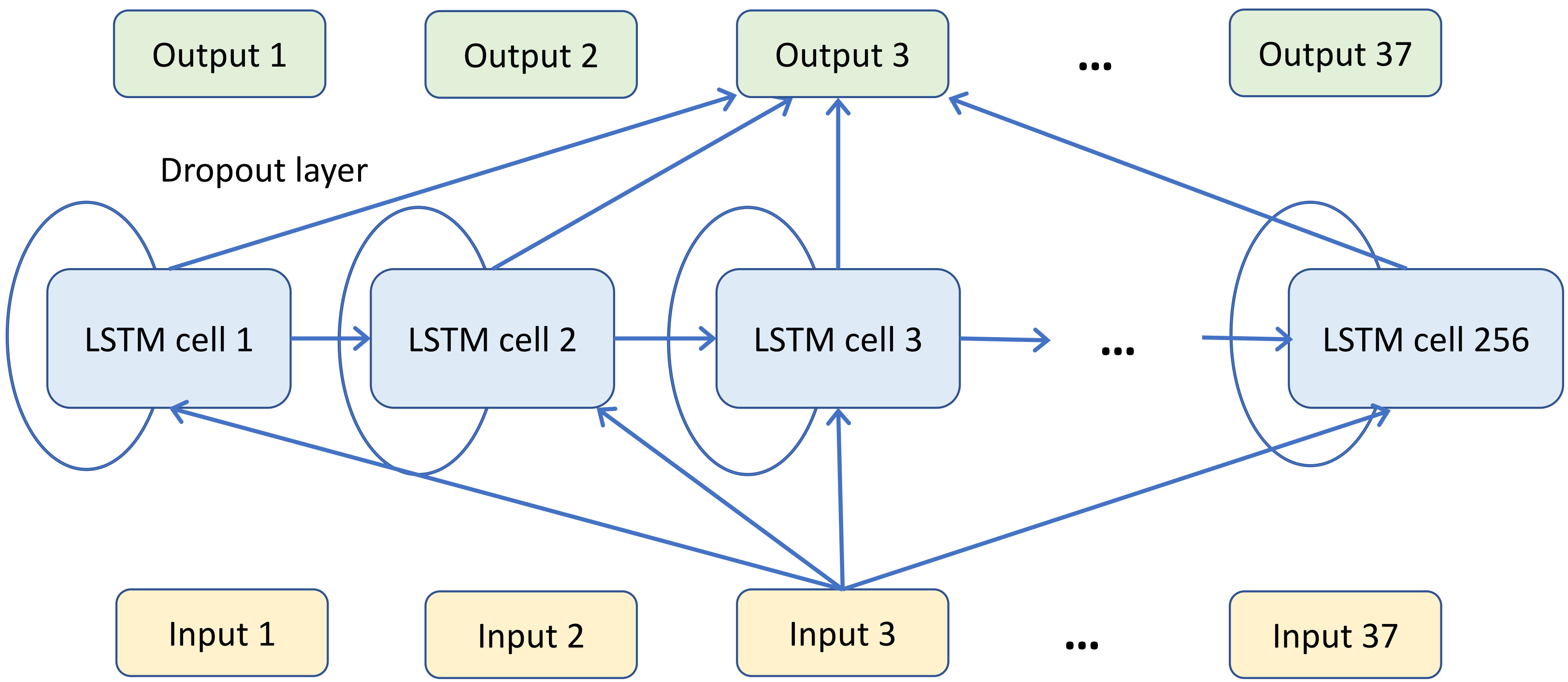
- Keras:
  - open source neural networks Application Program Interface (API) written in Python
  - enables fast experimentation with deep neural networks
  - runs on top of TensorFlow
- Import sequential model from Keras
- Normalize data points and scale their values within the range [0, 1]
- Replace anomaly labels by 0
- Length of time sequence: 20

- Keras: Deep Learning library for Theano and TensorFlow. [Online]. Available: https: //keras.io/ [Mar. 2018].
- TensorFlow. [Online]. Available: https://www.tensorflow.org [Mar. 2018].

LSTM classification procedure

- Adam optimizer
- Learning rate: 0.001
- Random seed: 77
- Batch size: 32
- Validation dataset: 20% of the original training dataset
- Epochs: 30

LSTM model: implementation



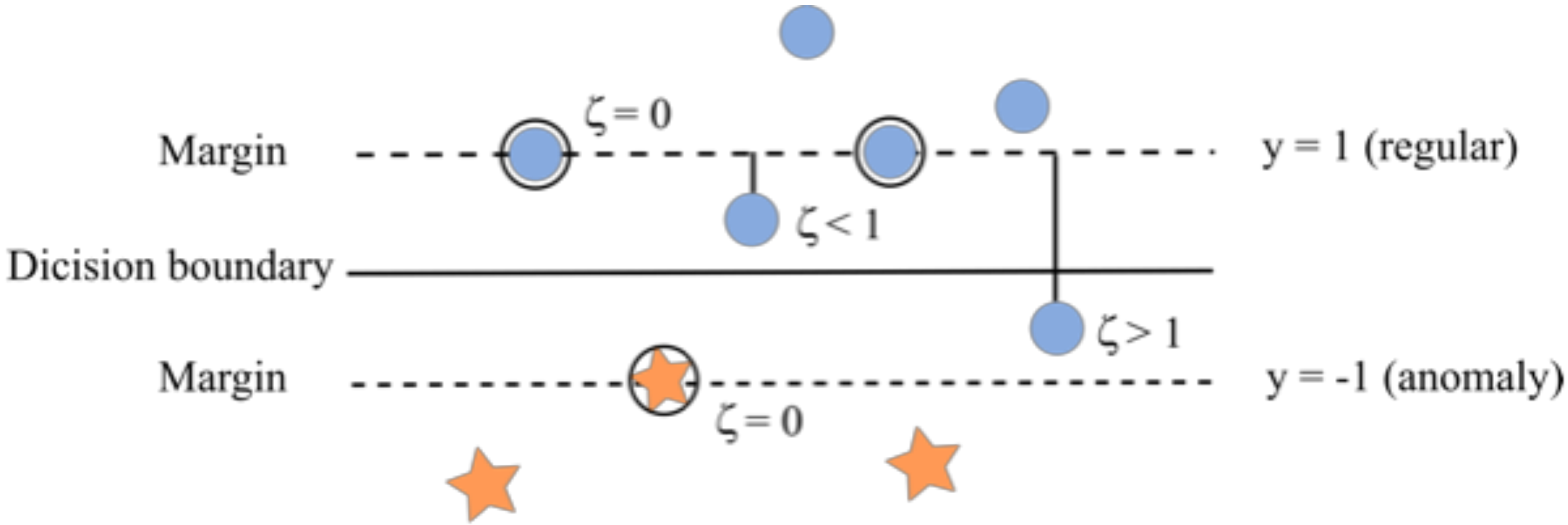
Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

Support Vector Machine: SVM

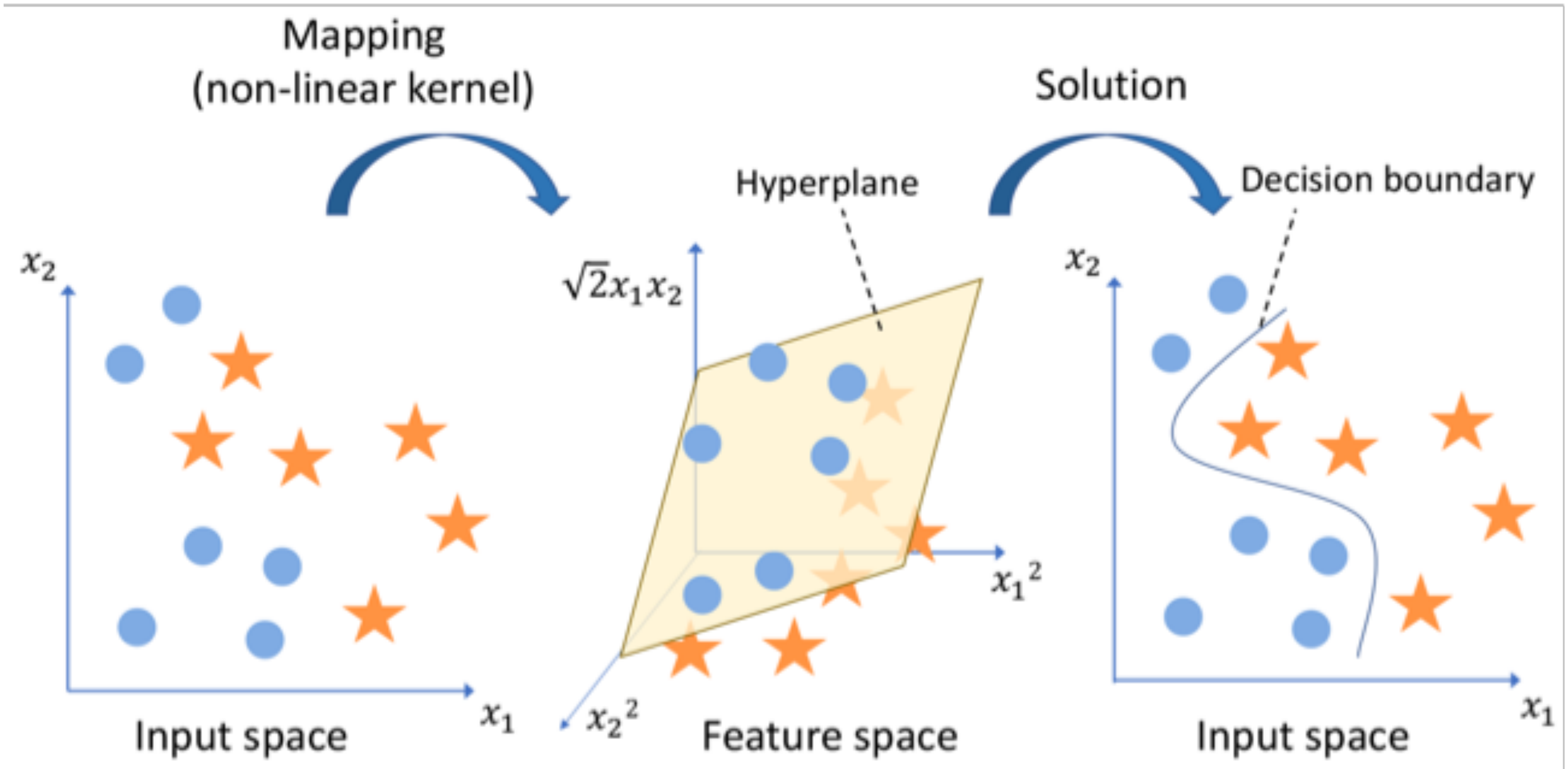
- Supervised learning algorithm used for classification and regression tasks
- Used as a binary classifier for detecting BGP anomalies
- Two types of SVM models:
  - hard-margin SVMs require each data point to be correctly classified
  - soft-margin SVMs allow some data points to be misclassified

Soft-Margin SVM



- Aims to find the maximum margin between both classes
- Support vectors determine the position of the decision boundary

Soft-Margin SVM: kernel function



Kernel function: $k(x_n, x_m) = \Phi(x_n)^\top \Phi(x_m)$

Naïve Bayes

- Used as supervised classifiers
- One of the most efficient machine learning classification techniques
- Assumes that features are conditionally independent for a given class
- Low complexity
- Trained effectively with smaller datasets
- Suitable for online real time detection of anomalies

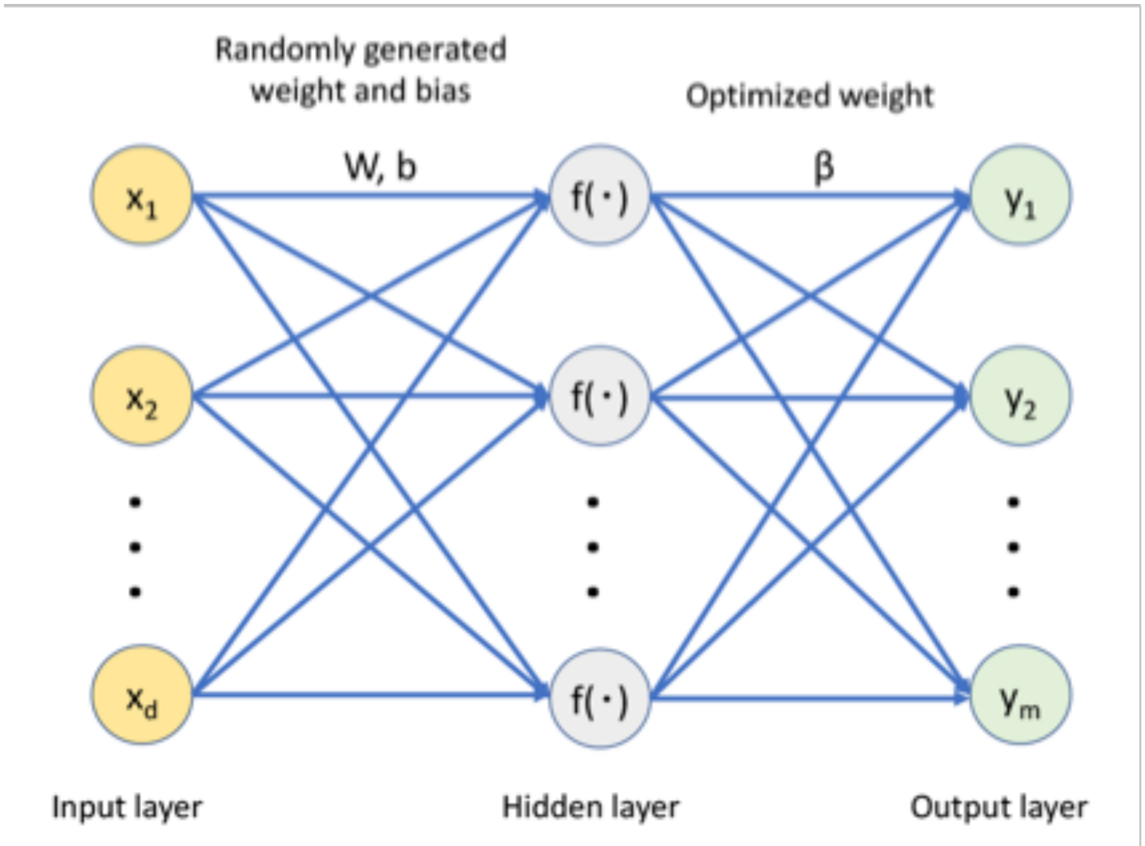
Decision Tree

- Used in data mining to predict the class labels
- A tree is “learned” by splitting the input dataset into subsets based on appropriate features:
  - root: source dataset
  - internal (non-leaf) node: input feature
  - tree branches: prediction outcomes
  - leaf node: class or class probability distribution
- Advantages:
  - does not require feature selection
  - does not require linear datasets
- Software tool: C5.0

Extreme learning machine: ELM

- Feed-forward neural network with single hidden layer
- Avoids the iterative tuning of the weights used in traditional neural networks
- Suitable for applications that require fast response and real-time predictions

ELM: architecture



Output: $$y_m = \sum_{i=1}^{k} \beta_i f(w_i x_d + b_i)$$

Training model	Test datasets			
	Accuracy (%)			F-Score (%)
	Code Red I	RIPE regular	BCNET	Code Red I
$LSTM_u1$	95.22	65.49	57.30	83.17
SVM_u1	78.65	69.17	57.22	39.51
Naïve $Bayes_u1$	82.03	82.99	79.03	29.52
Decision $Tree_u1$	85.36	89.00	77.22	47.82
ELM_u1	80.92	75.81	69.03	36.27
	Nimda	RIPE regular	BCNET	Nimda
$LSTM_u2$	53.94	51.53	50.80	11.81
SVM_u2	55.50	89.89	82.08	24.29
Naïve $Bayes_u2$	62.56	82.85	86.25	48.78
Decision $Tree_u2$	58.13	94.19	81.18	26.16
ELM_u2	54.42	96.15	91.88	13.72
	Slammer	RIPE regular	BCNET	Slammer
$LSTM_u3$	95.87	56.74	58.55	84.62
SVM_u3	93.04	73.92	59.24	75.93
Naïve $Bayes_u3$	83.58	84.79	81.18	51.12
Decision $Tree_u3$	95.89	89.42	77.78	84.34
ELM_u3	86.96	78.57	73.47	55.31

SVM, Naïve Bayes, Decision Tree, and ELM results have been reported in:

- Z. Li, Q. Ding, S. Haeri, and Lj. Trajković, “Application of machine learning techniques to detecting anomalies in communication networks: classification algorithms,” in *Cyber Threat Intelligence*, M. Conti, A. Dehghantanha, and T. Dargahi, Eds., Berlin: Springer, pp. 71-92, 2018.

Performance comparison: balanced datasets

Training model	Test datasets			
	Accuracy (%)			F-Score (%)
	Code Red I	RIPE regular	BCNET	Code Red I
LSTM$_b$1	56.43	60.48	62.78	26.59
	Nimda	RIPE regular	BCNET	Nimda
LSTM$_b$2	56.32	44.27	53.58	65.96
SVM$_b$2	69.26	51.81	44.86	72.32
	Slammer	RIPE regular	BCNET	Code Red I
LSTM$_b$3	82.98	55.00	48.20	58.54
SVM$_b$3	87.19	63.31	51.11	64.76

SVM results have been reported in:

- Z. Li, Q. Ding, S. Haeri, and Lj. Trajković, “Application of machine learning techniques to detecting anomalies in communication networks: classification algorithms,” in *Cyber Threat Intelligence*, M. Conti, A. Dehghantanha, and T. Dargahi, Eds., Berlin: Springer, to appear.

Performance comparison: unbalanced vs. balanced LSTM models

Training model	Test datasets			
	Accuracy (%)			F-Score (%)
	Code Red I	RIPE regular	BCNET	Code Red I
$LSTM_u1$	95.22	65.49	57.30	83.17
$LSTM_b1$	56.43	60.48	62.78	26.59
	Nimda	RIPE regular	BCNET	Nimda
$LSTM_u2$	53.94	51.53	50.80	11.81
$LSTM_b2$	56.32	44.27	53.58	65.96
	Slammer	RIPE regular	BCNET	Code Red I
$LSTM_u3$	95.87	56.74	58.55	84.62
$LSTM_b3$	82.98	55.00	48.20	58.54

Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

Discussion

- Sources of labeled anomalous datasets:
  - artificial datasets may not contain properties of the real-world data
  - RIPE and BCNET data were collected from deployed networks
- Selection of performance metrics:
  - Accuracy: ratio of correct predictions for the entire dataset
  - F-Score: more suitable because it emphasizes importance of the anomaly class
- Selection of appropriate machine learning approach:
  - application dependent
  - based on algorithm advantages and limitations

Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

Future work

- Optimize the LSTM performance:
  - use dropout technique in the input layer to learn independent representations of the dataset
- Tune hyperparameters to improve LSTM convergence:
  - number of LSTM cells
  - number of epochs
- Consider other LSTM architectures:
  - Gated Recurrent Unit (GRU): simplified LSTM
    - more efficient
    - requires smaller training dataset

Conclusions

- Classified anomalies in BGP traffic traces using a number of classification models
- Extracted features and created unbalanced and balanced datasets
- Compared the performance of LSTM models to SVM, Naïve Bayes, Decision Tree, and ELM classifiers
- Performance of classifiers is influenced by the employed datasets
- No single classifier performs the best across all used datasets
- Machine learning is a feasible approach to successfully classify BGP anomalies

Roadmap

- Introduction
- Border Gateway Protocol datasets
- Extraction of features from BGP update messages
- Performance metrics
- Long Short-Term Memory
- Comparison of classification algorithms
- Discussion
- Future work and conclusion
- References

References: BGP

- Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4),” RFC 1771, *IETF*, Mar. 1995. [Online]. Available: http://tools.ietf.org/rfc/rfc1771.txt [Mar. 2018].
- Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4),” RFC 4271, *IETF*, Jan. 2016. [Online]. Available: http://tools.ietf.org/rfc/rfc5271.txt [Mar. 2018].
- RIPE NCC: RIPE Network Coordination Center. [Online]. Available: http://www.ripe.net/data-tools/stats/ris/ris-raw-data [Mar. 2018].
- BCNET. [Online]. Available: http://www.bc.net [Mar. 2018].
- Bgpdump [Online]. Available: https://bitbucket.org/ripencc/bgpdump/wiki/Home [Mar. 2018].

References: Machine learning algorithms

- C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag, 2006, pp. 325–358.
- G. E. Hinton, S. Osindero, and Y-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Compt.*, vol. 18, no. 7, pp. 1527–1554, July 2006.
- S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.,* vol. 9, no. 8, pp. 1735–1780, Oct. 1997.
- F. A. Gers, J. Schimidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” *Neural Computation*, vol, 12, no. 10, pp. 2451–2471, Oct. 2000.
- L. Rokach and O. Maimon, “Top-down induction of decision trees classifiers–a survey,” *IEEE Trans. Syst., Man, Cybern., Appl. and Rev.*, vol. 35, no. 4, pp. 476–487, Nov. 2005.
- G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, pp. 489–501, Dec. 2006.

Publications:

- Q. Ding, Z. Li, S. Haeri, and Lj. Trajković, “Application of machine learning techniques to detecting anomalies in communication networks: datasets and feature selection algorithms,” in *Cyber Threat Intelligence*, M. Conti, A. Dehghantanha, and T. Dargahi, Eds., Berlin: Springer, pp. 47-70, 2018.
- Z. Li, Q. Ding, S. Haeri, and Lj. Trajković, “Application of machine learning techniques to detecting anomalies in communication networks: classification algorithms,” in *Cyber Threat Intelligence*, M. Conti, A. Dehghantanha, and T. Dargahi, Eds., Berlin: Springer, pp. 71-92, 2018.
- Q. Ding, Z. Li, P. Batta, and Lj. Trajković, “Detecting BGP anomalies using machine learning techniques,” in *Proc. IEEE Trans. Syst., Man, Cybern.*, Budapest, Hungary, Oct. 2016, pp. 3352–3355.
- P. Batta, M. Singh, Z. Li, Q. Ding, and Lj. Trajković, “Evaluation of support vector machine kernels for detecting network anomalies,” *IEEE Int. Symp. Circuits and Systems*, Florence, Italy, May 2018, pp. 1–4.
- H. Ben Yedder, Q. Ding, U. Zakia, Z. Li, S. Haeri, and Lj. Trajković, “Comparison of virtualization algorithms and topologies for data center networks,” *The 26th Int. Conf. Comput. Commun., Netw., 2nd Workshop on Netw. Security Anal. Automat.*, Vancouver, Canada, Aug. 2017.
- S. Haeri, Q. Ding, Z. Li, and Lj. Trajković, “Global resource capacity algorithm with path splitting for virtual network embedding,” in *Proc. IEEE Int. Symp. Circuits and Systems*, Montreal, Canada, May 2016, pp. 666–669.

Acknowledgements

- Chair:
  - Prof. Ivan V. Bajić
- Senior supervisor:
  - Prof. Ljiljana Trajković
- Supervisor:
  - Prof. Parvaneh Saeedi
- SFU examiner:
  - Prof. Qianping Gu

Thank you!

Questions?