Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Automatic Detection of Malware-Generated Domains with Recurrent Neural Models

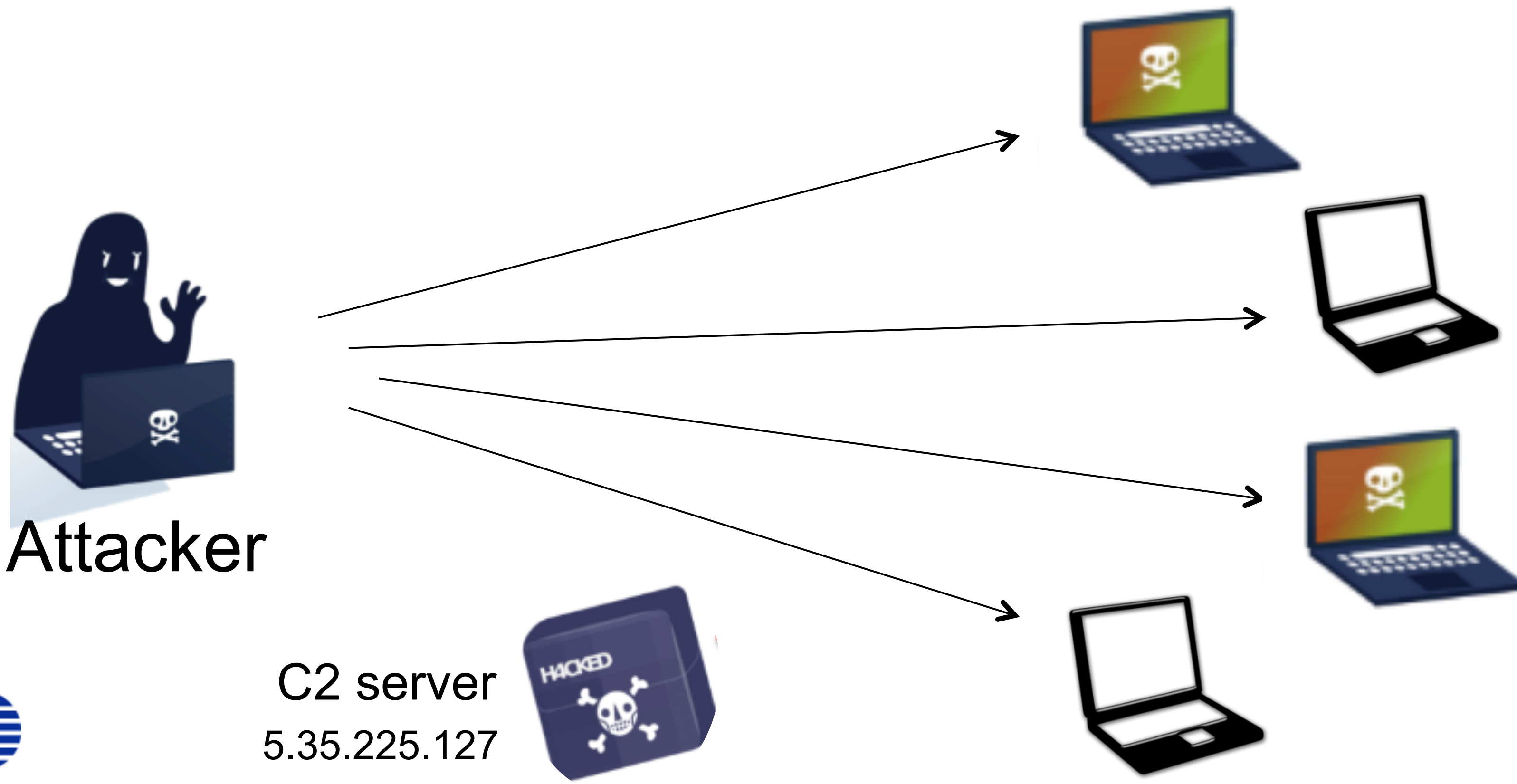
Pierre Lison
Norsk Regnesentral

Vasileios Mavroeidis
University of Oslo

NISK 2017, Oslo

Introduction

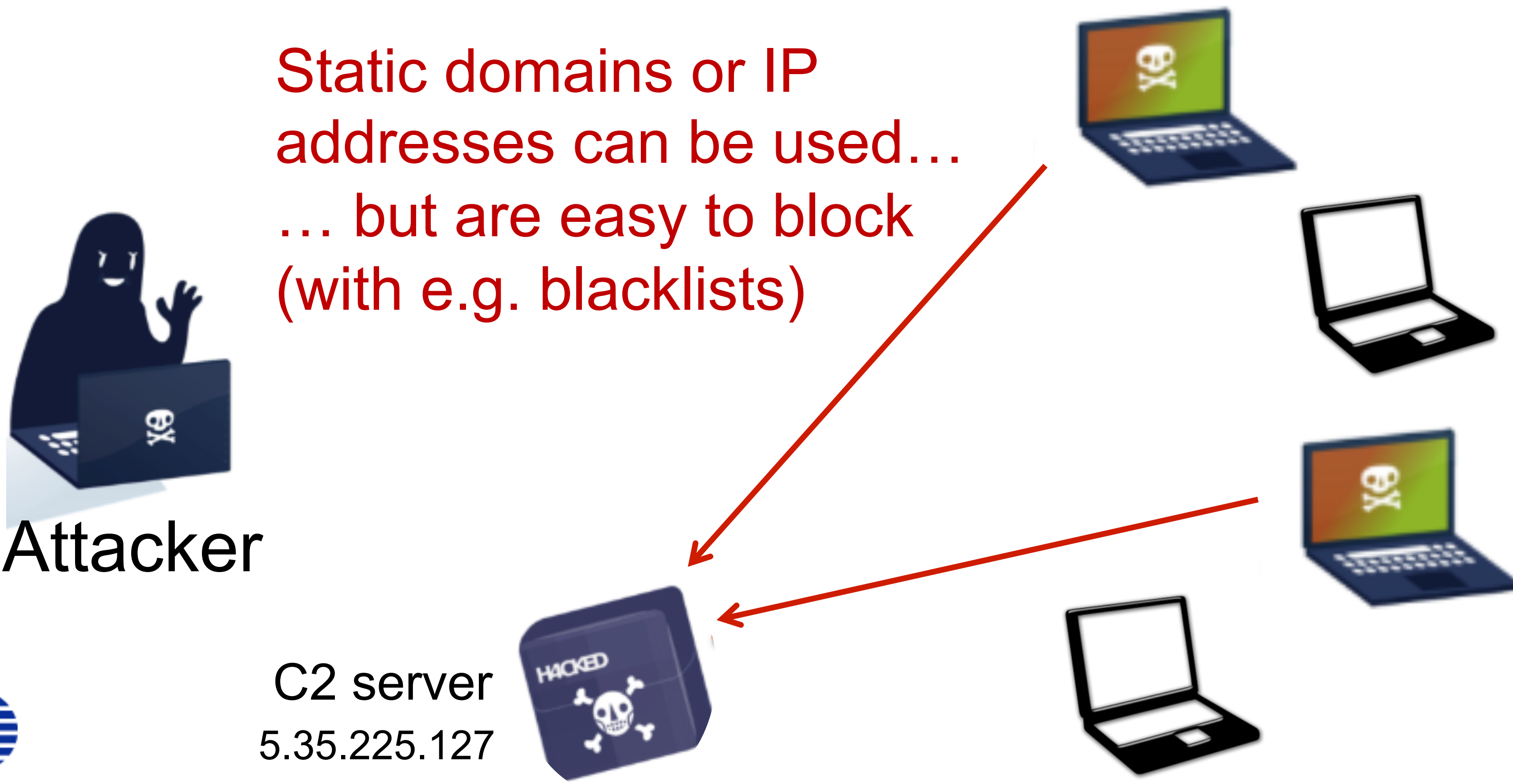
- Most malware must connect compromised machines with a *command and control* (C2) server for their operations



Introduction

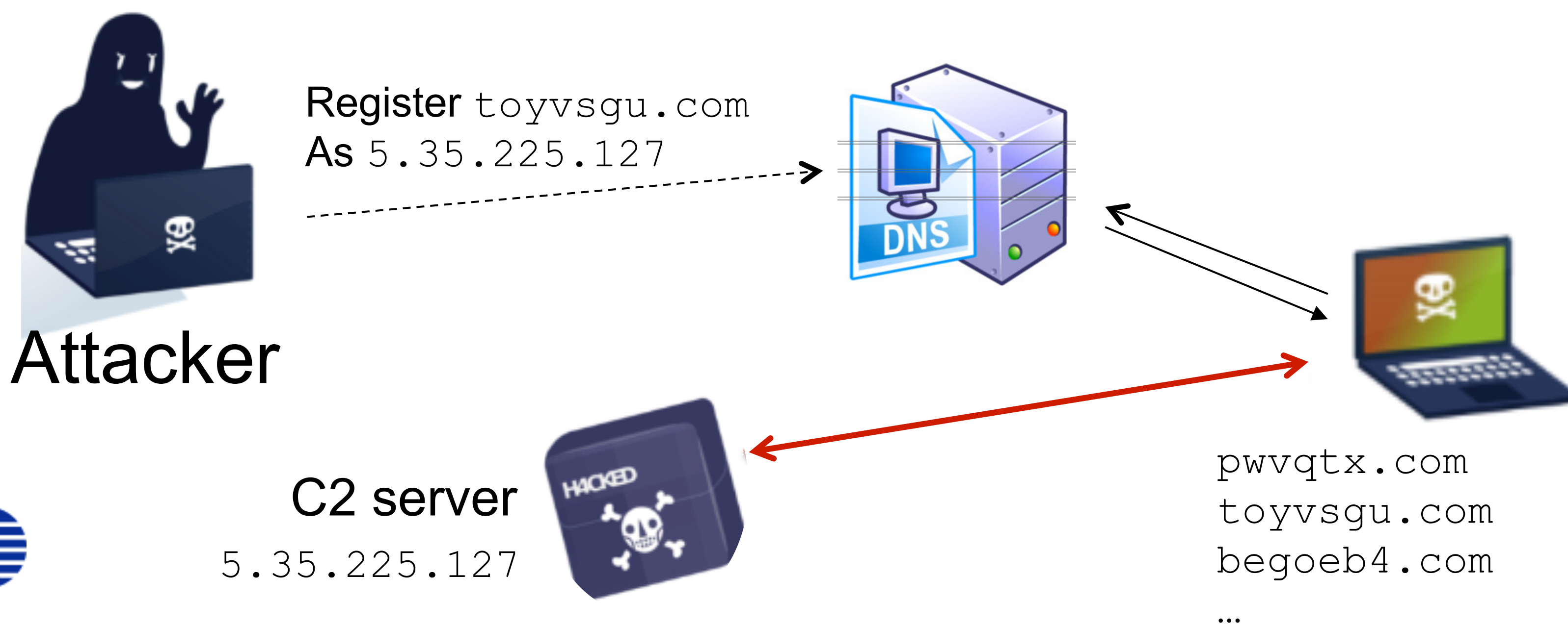
- Most malware must connect compromised machines with a *command and control* (C2) server for their operations

Static domains or IP addresses can be used…
… but are easy to block (with e.g. blacklists)



Introduction

- With domain-generation algorithms (DGA), compromised machines will attempt to connect to a large number of pseudo-random domain names…
- The attacker can then simply register a few of these artificial domains to establish a rendez-vous point



Introduction

- We present a *machine learning approach* to automatically detect domains generated by malware through DGA

Domain name `toyvsgu.com` → Recurrent neural net → DGA or not?

- The approach relies on a *recurrent neural network* trained on a large dataset of benign & malicious domains
- **Benefits**:
  - Can be used for real-time threat intelligence (no need for human intervention or external resources)
  - Purely data-driven: can adapt to new malware threats by regularly feeding new data to the model

Outline

1. **Domain-generating algorithms**
2. **Neural model**
   - Core model
   - Extensions
   - Training data
3. **Evaluation**
   - Experimental design
   - Results
   - Discussion

Domain-generating algorithms (DGAs)

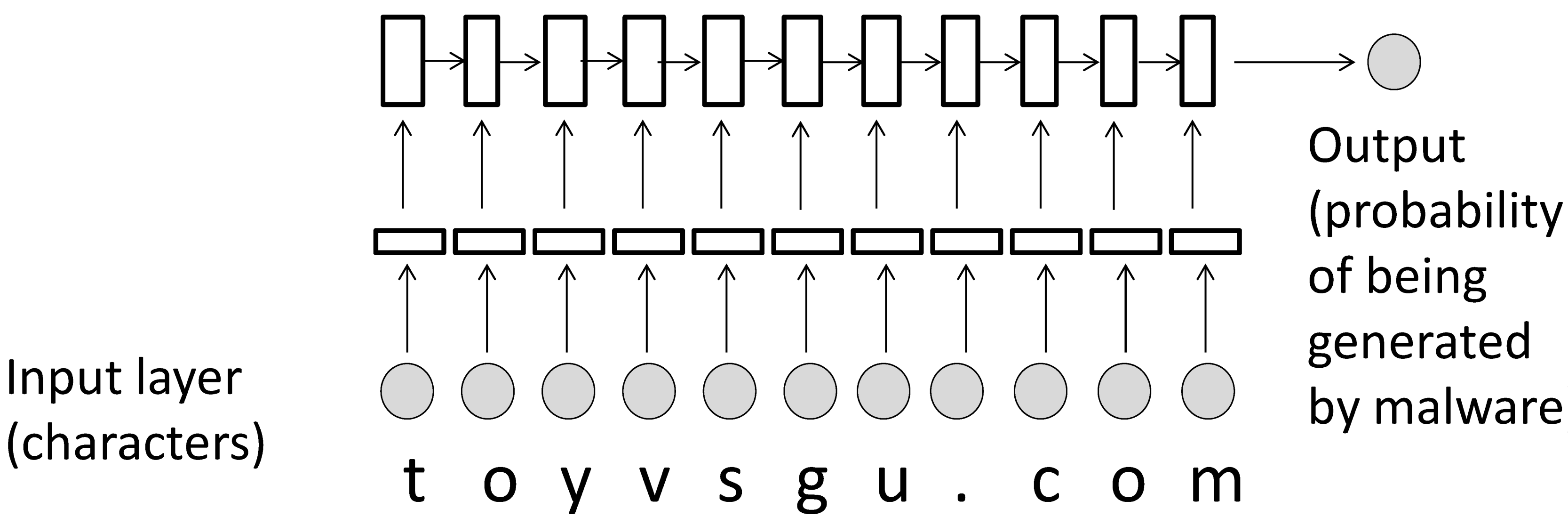
- DGAs are increasingly popular as C2 rendez-vous mechanism in botnets
  - First observed in the Kraken botnet (2008)
- DGAs can generate a large number of seemingly random domain names based on a *shared secret* (**seed**)
- Highly *asymmetric* situation:
  - Malicious actors only need to register a single domain to establish a C2 communication channel
  - While security professionals must control the full range of potential domains to contain the threat

Taxonomy of DGAS

- **Time dependence:**
  - Are the seeds fixed or are they only valid for a specific period (by including a time source in their calculation?)
- **Determinism:**
  - Are the seeds computed through a deterministic procedure, or do they include unpredictable factors (weather forecasts, stock markets prices, etc.)
- **Generation scheme:**
  - How are the domains generated from the seeds? Popular techniques include alphanumeric combinations, hash-based techniques, wordlists and permutations.

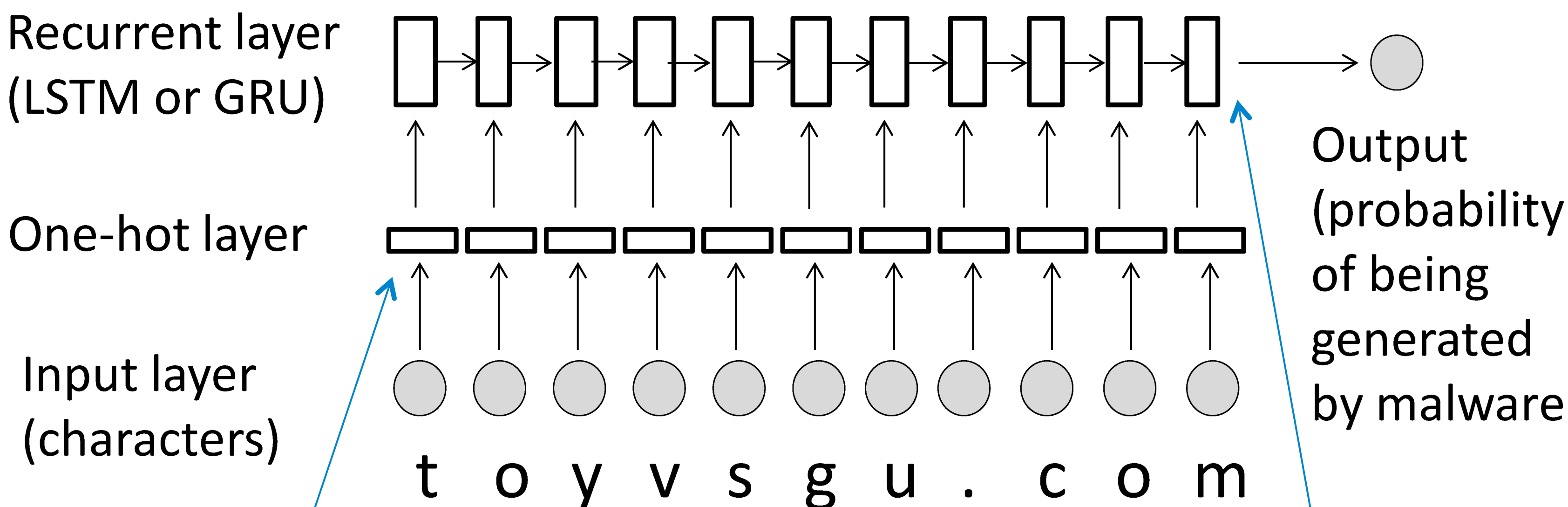
Detection of DGAs

- Most previous work relied on "shallow" machine learning models (such as Hidden Markov Models) to detect DGAs
- Our approach relies on **recurrent neural networks**
  - Ability to learn complex sequential patterns
  - Widely used in NLP tasks



Architecture

Recurrent layer builds up a representation of the character sequence as a dense vector



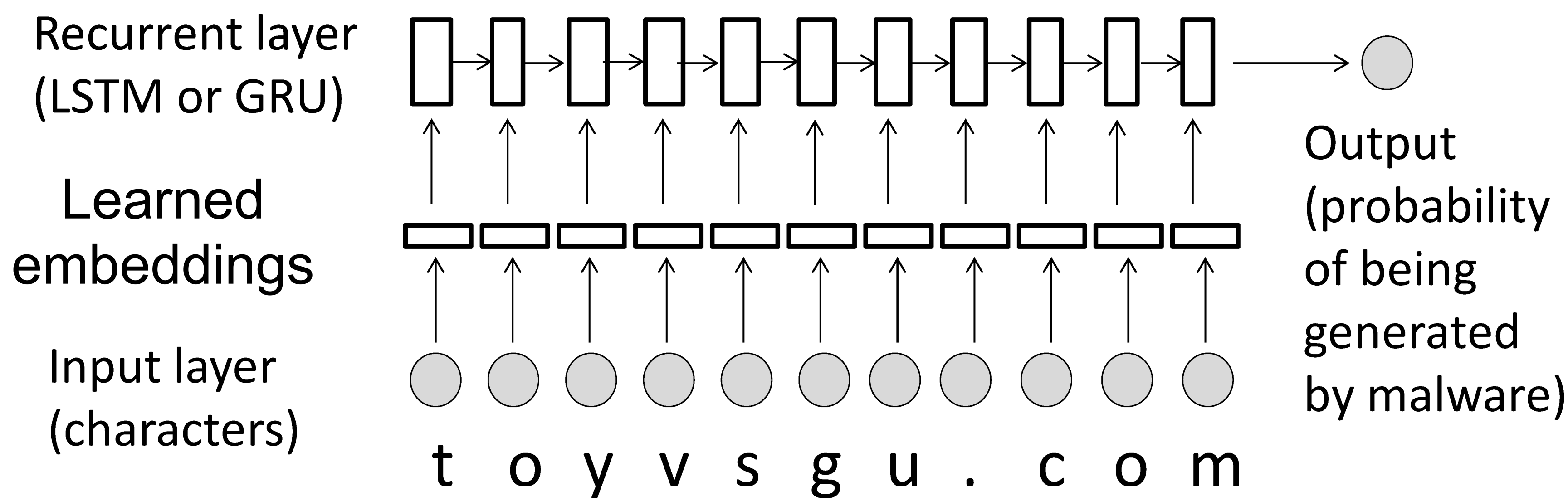
First layer encode each character as a "one-hot" vector

Domain name is fed to the neural network character by character

Final vector is used to predict whether the domain is DGA

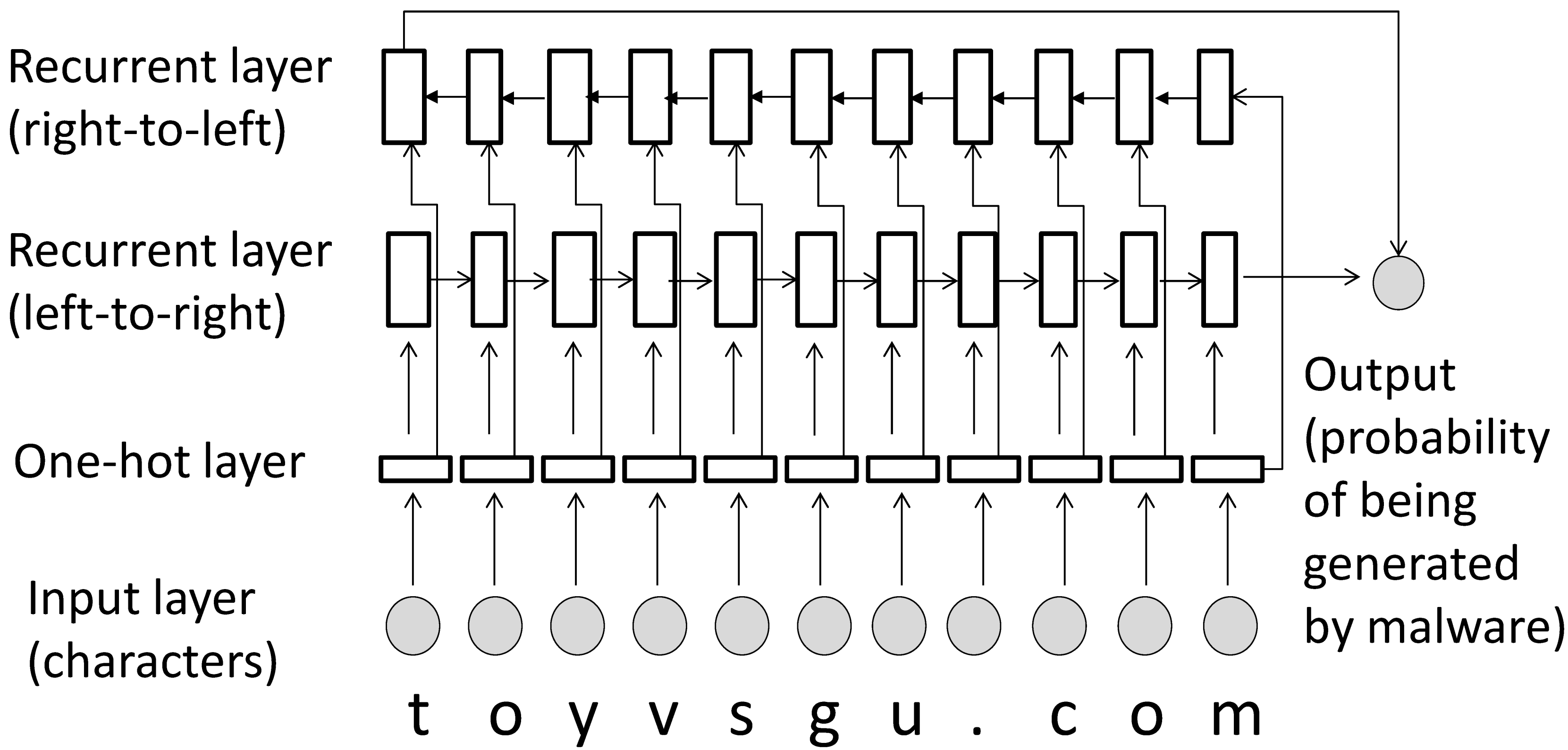
Extensions

- **Embeddings**
- Bidirectionality
- Hidden layer
- Multi-task learning



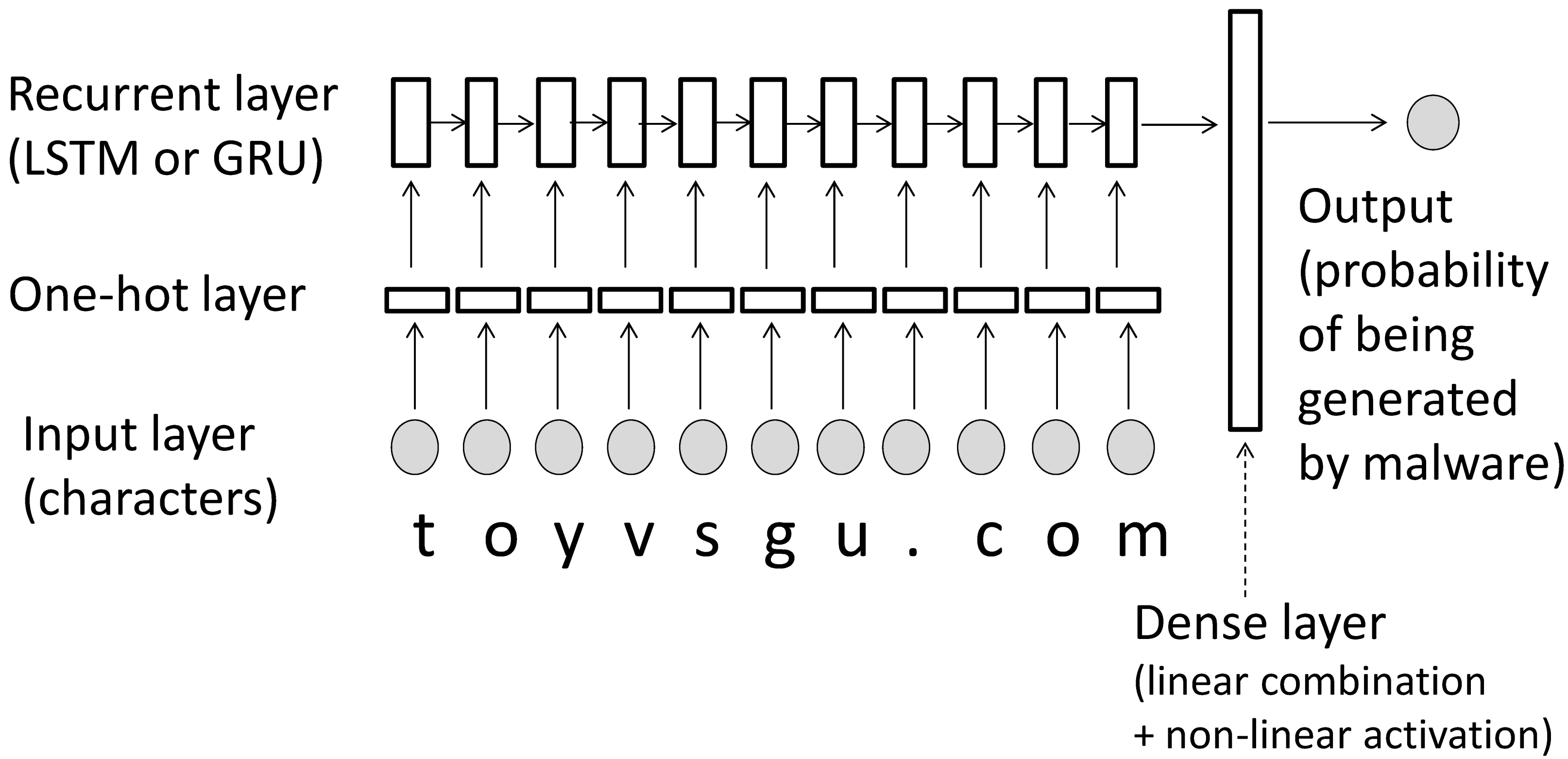
Extensions

- Embeddings
- **Bidirectionality**
- Hidden layer
- Multi-task learning



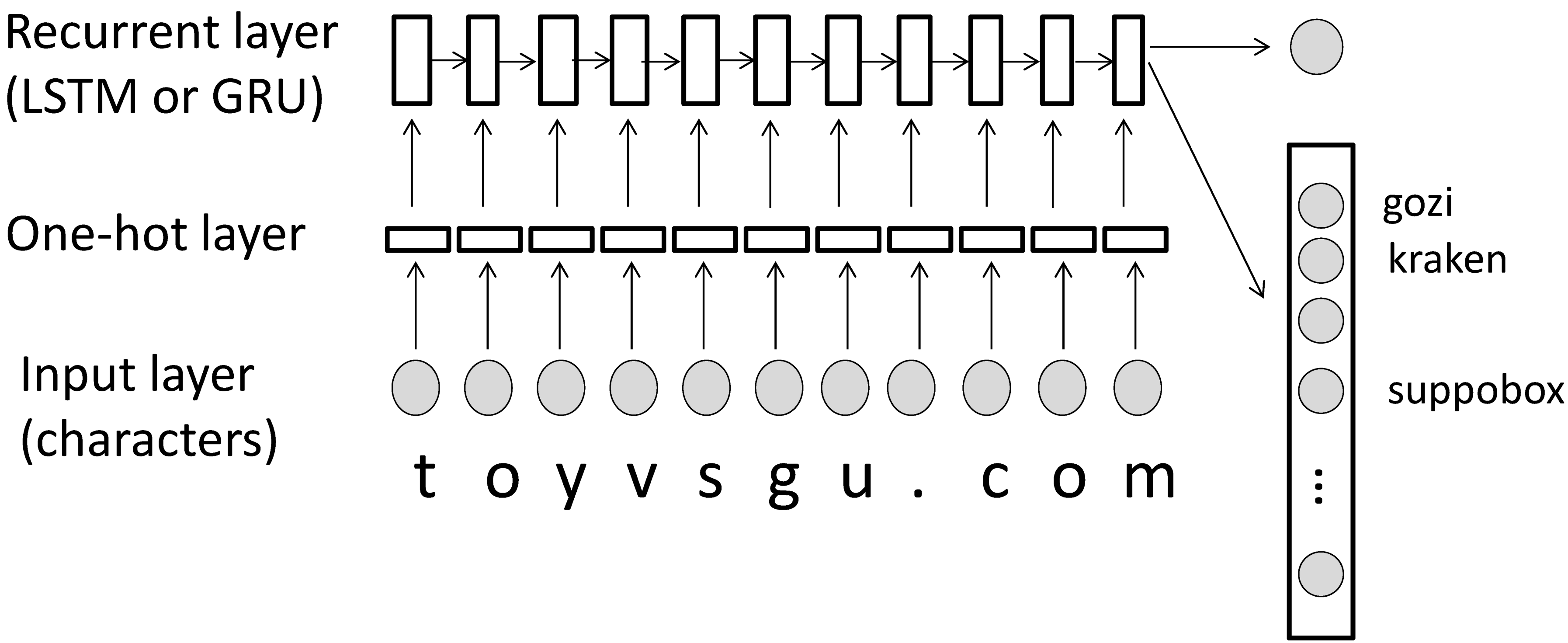
Extensions

- Embeddings
- Bidirectionality
- **Hidden layer**
- Multi-task learning



Extensions

- Embeddings
- Bidirectionality
- Hidden layer
- **Multi-task learning**



Data

- The parameters of the neural model must be estimated from training data
- **Negative examples** (benign domains):
  - Snapshots from the Alexa top 1 million domains
  - Total: over 4 million domains
- **Positive examples** (malware DGAs)
  - DGA lists from the DGArchive (63 types of malware)
  - Feeds from Bambenek Consulting
  - Domain generators for 11 DGAs
  - Total: 2.9 million domains

Data

Malware	Frequency				
bamital	40 240	gozi	105 631	ramdo	15 984
banjori	89 984	hesperbot	370	ramnit	90 000
bedep	15 176	locky	179 204	ranbyu	40 000
beebone	420	madmax	192	ranbyus	12 720
blackhole	732	matsnu	12 714	rovnix	40 000
bobax	19 288	modpack	52	shifu	4 662
conficker	400 000	murofet	53 260	simda	38 421
corebot	50 240	murofet$_w$	40 000	sisron	5 936
cryptolocker	55 984	necur	40 000	suppobox	41 014
cryptowall	94	necurs	36 864	sutra	9 882
dircrypt	11 110	nymaim	186 653	symmi	40 064
dnschanger	40 000	oderoor	3 833	szribi	16 007
downloader	60	padcrypt	35 616	tempedreve	453
dyre	47 998	proslikefan	75 270	tinba	80 000
ekforward	1 460	pushdo	176 770	torpig	40 000
emotet	40 576	pushdotid	6 000	tsifiri	59
feodo	192	pykspa	424 215	urlzone	34 536
fobber	2 600	pykspa2	24 322	vawtrak	1 050
gameover	80 000	qadars	40 400	virut	400 600
gameover_p2p	41 000	qakbot	90 000	volatilecedar	1 494
				xxhex	4400
				Total	2 925 168

Evaluation

- 10-fold cross validation on the full dataset
- **Baseline**: logistic regression on character bigrams
  - Toyvsgu.com → (to, oy, yv, vs, sg, gu, u., .c, co, om)
- Metrics: accuracy, precision, recall, F_1 score

$$\text{precision} = \frac{\text{\# correctly classified malware domains}}{\text{\# domains classified as malware by model}}$$

$$\text{recall} = \frac{\text{\# correctly classified malware domains}}{\text{\# actual known malware domains}}$$

$$F_1 \text{ score} = 2\frac{p \times r}{p + r}$$ (harmonic mean of the two)

Model selection

- The use of embeddings, bidirectional layers, and additional hidden layers did not improve the performance
- Multi-task learning (i.e. simultaneously learning to detect DGAs and to classify them) yielded the same results as networks optimised for these two tasks separately
  - The two tasks can use a shared latent representation
- The recurrent layer used GRU units with dimension=512
- Model trained on GPU with a batch size of 256, two passes and RMSProp as optimisation algorithm

Results

Area Under the Curve (AUC) of the ROC curve (see next slide)

- Detection

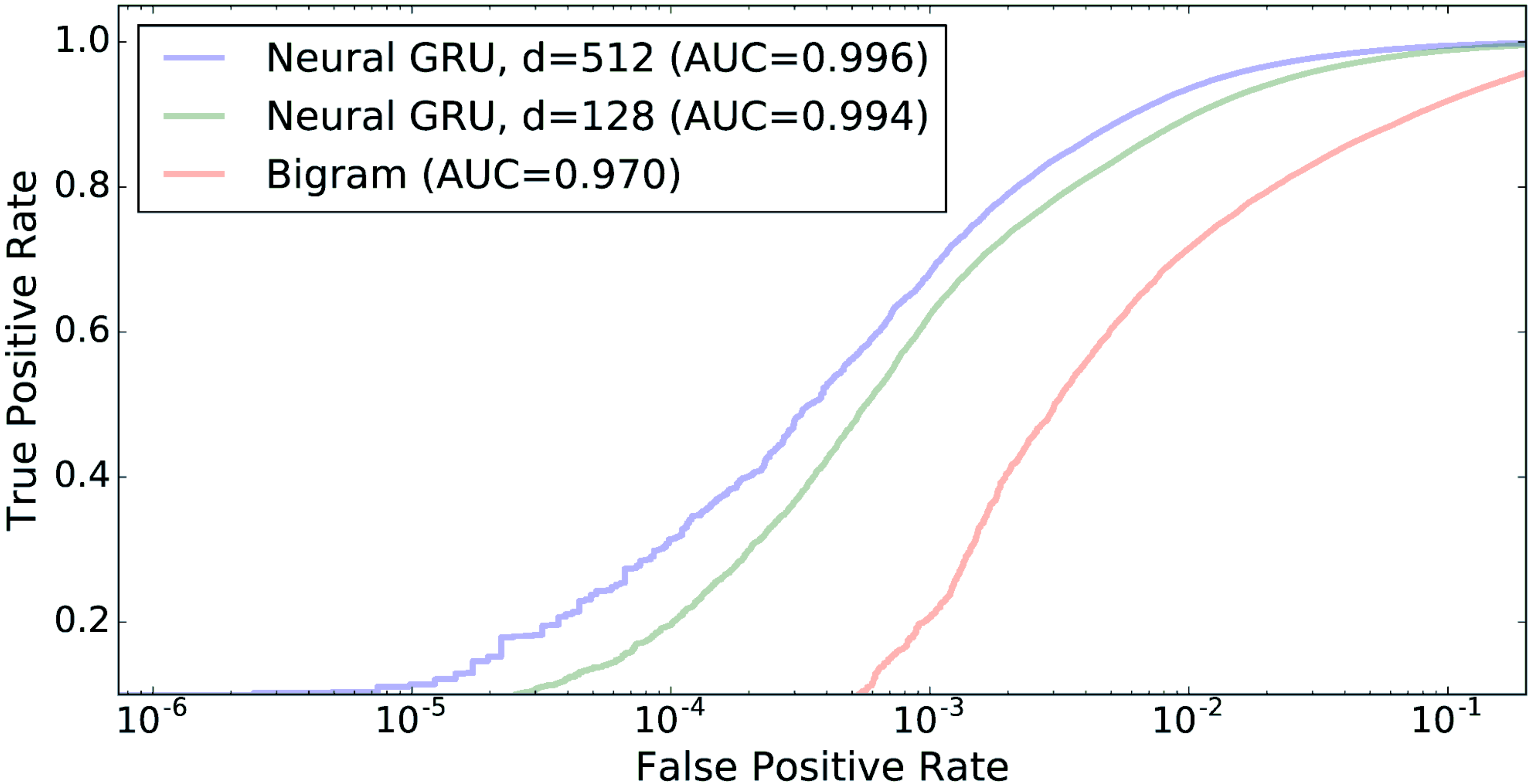
	Accuracy	Precision	Recall	F_1 score	ROC AUC
Bigram	0.915	0.927	0.882	0.904	0.970
Neural model	**0.973**	**0.972**	**0.970**	**0.971**	**0.996**

- Classification

	Accuracy	Precision		Recall		F_1 score	
		Micro	Macro	Micro	Macro	Micro	Macro
Bigram	0.800	0.787	0.564	0.800	0.513	0.787	0.522
Neural model	**0.892**	**0.891**	**0.713**	**0.892**	**0.653**	**0.887**	**0.660**

Micro: weighted averages over all classes
Macro: unweighted averages

ROC curve


Neural GRU, d=512 (AUC=0.996)
Neural GRU, d=128 (AUC=0.994)
Bigram (AUC=0.970)
True Positive Rate
0.2
0.4
0.6
0.8
1.0
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
False Positive Rate

Discussion

- See paper for detailed results for each malware family
- Neural model is also able to detect dictionary-based DGAs such as `suppobox` (recall of 93%, compared to only 12% for baseline) when given enough training examples
- Some DGAs still remain difficult to detect, such as `matsnu` (not enough training data to learn underlying wordlists)

Conclusion

- Data-driven approach to the detection of domain names generated by malware algorithms
- Recurrent neural architectures trained on a large dataset with millions of domain names
- Model can detect 93% of malware domains with a false positive rate of 1:100.
- **Current work**: integration of model as part of a larger architecture to detect cyber-threats in traffic data