32ND USENIX
SECURITY SYMPOSIUM

An Input-Agnostic Hierarchical Deep Learning Framework for Traffic Fingerprinting

Jian Qu
Xi'an Jiaotong University

Xiaobo Ma
Xi'an Jiaotong University

Jianfeng Li
Xi'an Jiaotong University

Xiapu Luo
Hong Kong Polytechnnic University

Lei Xue
Sun Yat-sen University

Junjie Zhang
Wright State University

Zhenhua Li
Tsinghua University

Li Feng
Southwest Jiaotong University

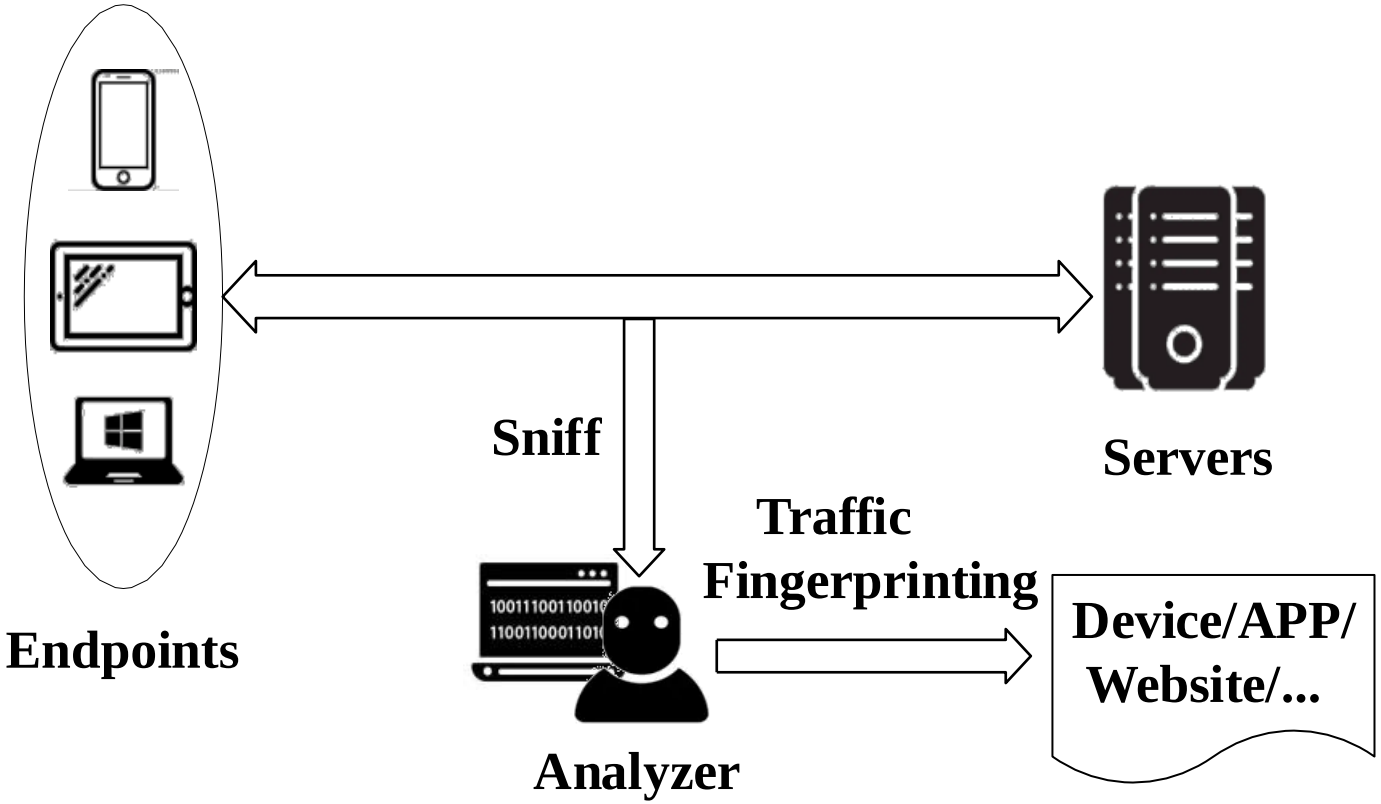
Xiaohong Guan
Xi'an Jiaotong University

August 2023

Outline

- **Background and Problem Description**
- System Design
- Evaluation
- Conclusions

Background and Problem Description



Related work

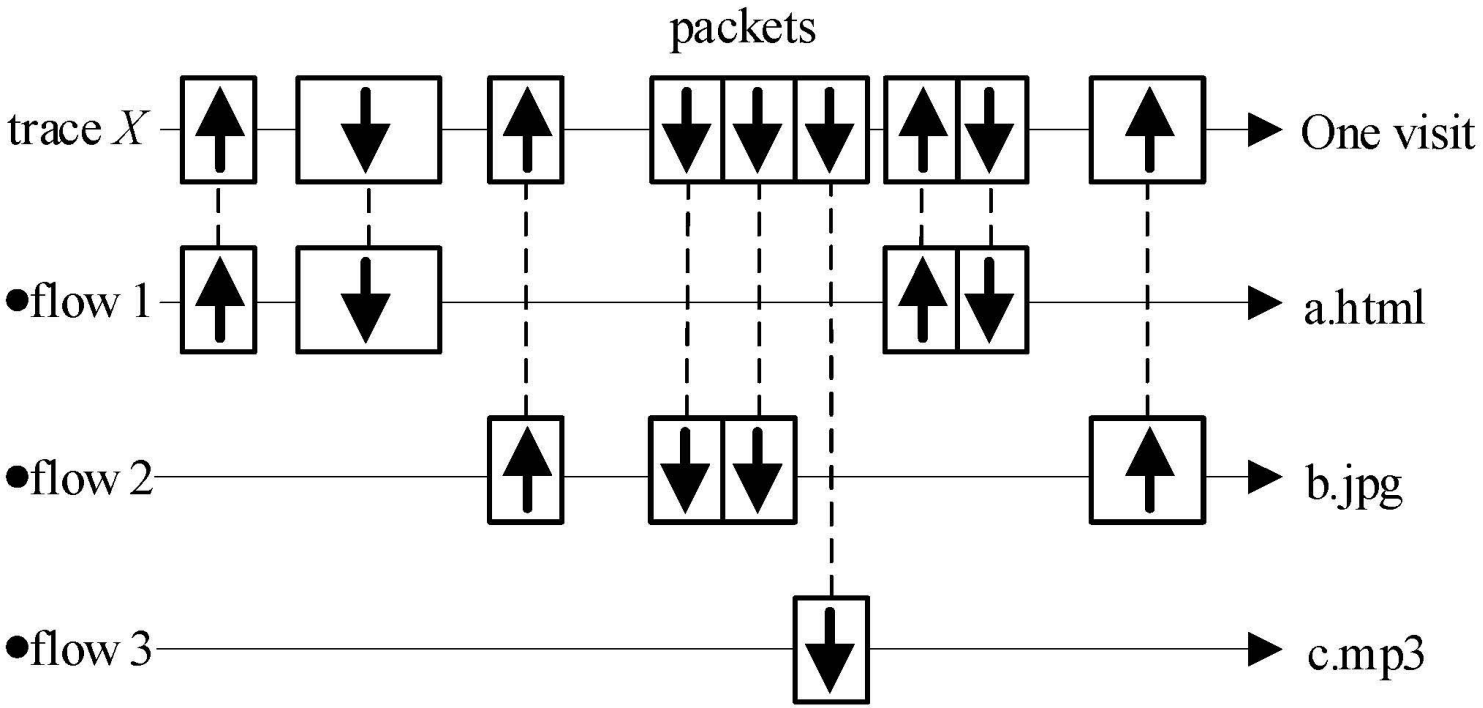
Website Fingerprinting
Shen et al. [ACM ARES, 2019]
Di Martino et al. [IEEE ICC, 2019]

Application Fingerprinting
App-Net [INFOCOM WKSHPS, 2020]
FOAP [USENIX Security, 2022]

Internet of Things Fingerprinting
Ma et al. [IEEE INFOCOM, 2020]
IoTFinder [EuroS&P, 2020]

……

Motivation



Related work

Feature-based traffic fingerprinting
k-fingerprinting [USENIX Security, 2016]
Shafiq et al. [The Journal of Supercomputing, 2019]

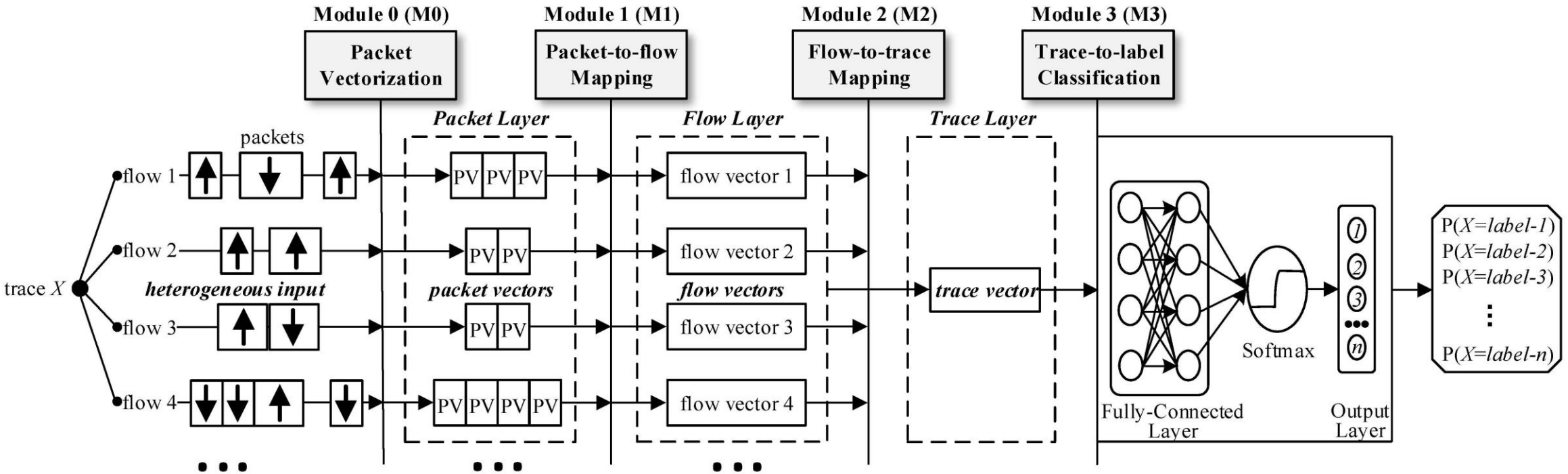
Deep learning-based traffic fingerprinting
Deep fingerprinting [ACM CCS, 2018]
Var-cnn [PETS, 2019]
SHAME [ACM WPES, 2021]

……

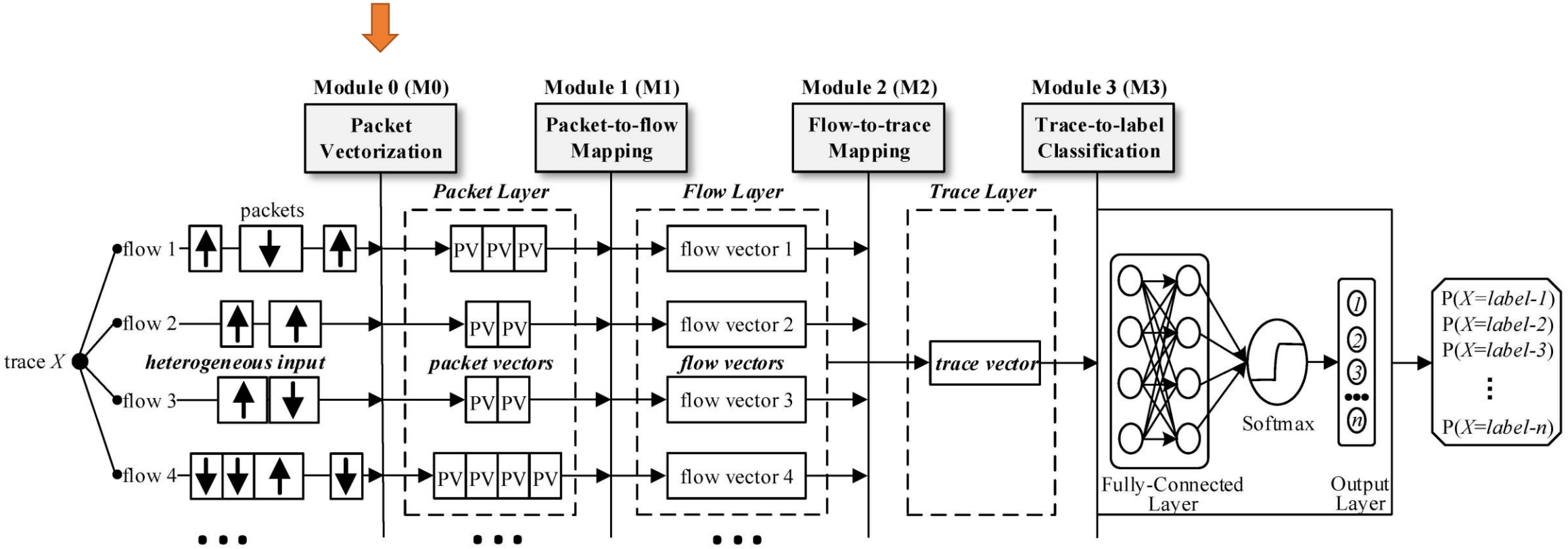
Outline

- Background and Problem Description
- **System Design**
- Evaluation
- Conclusions

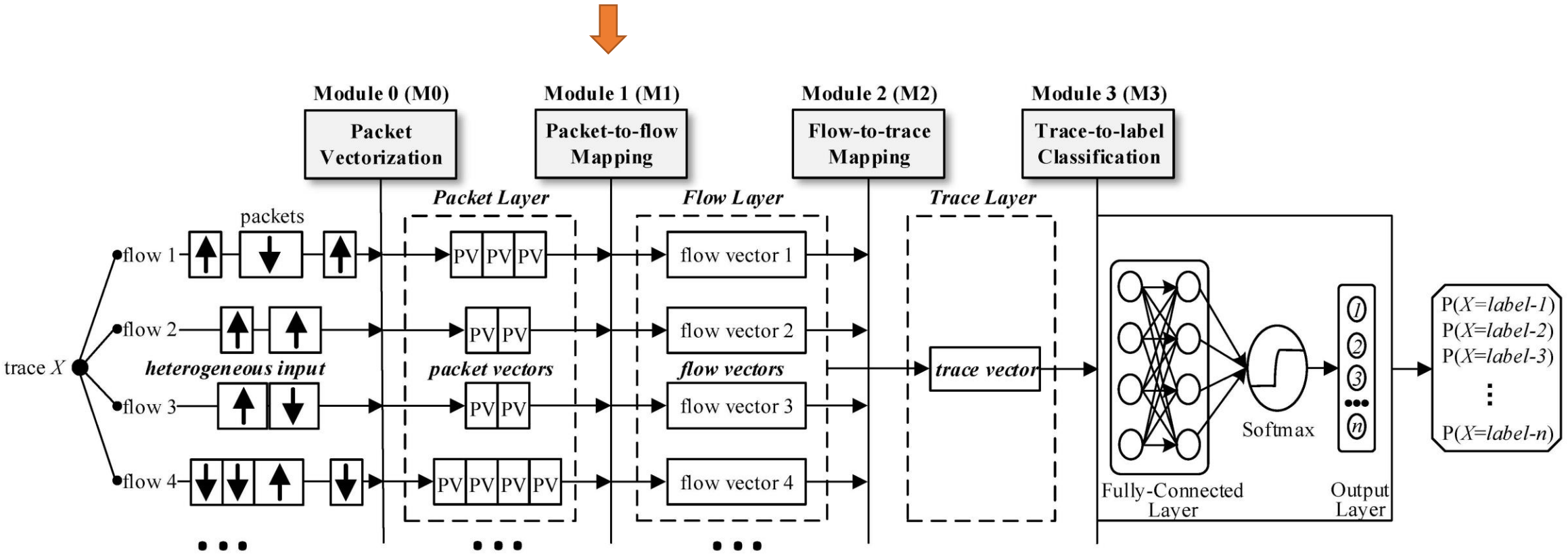
System Design



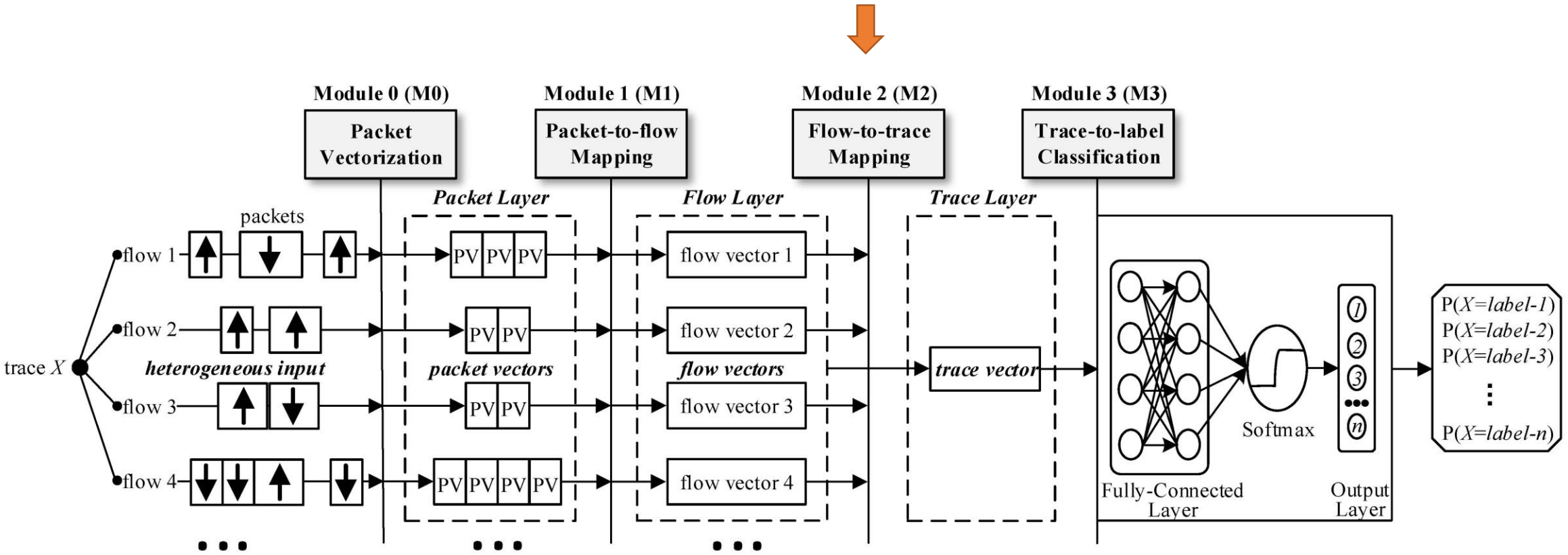
System Design



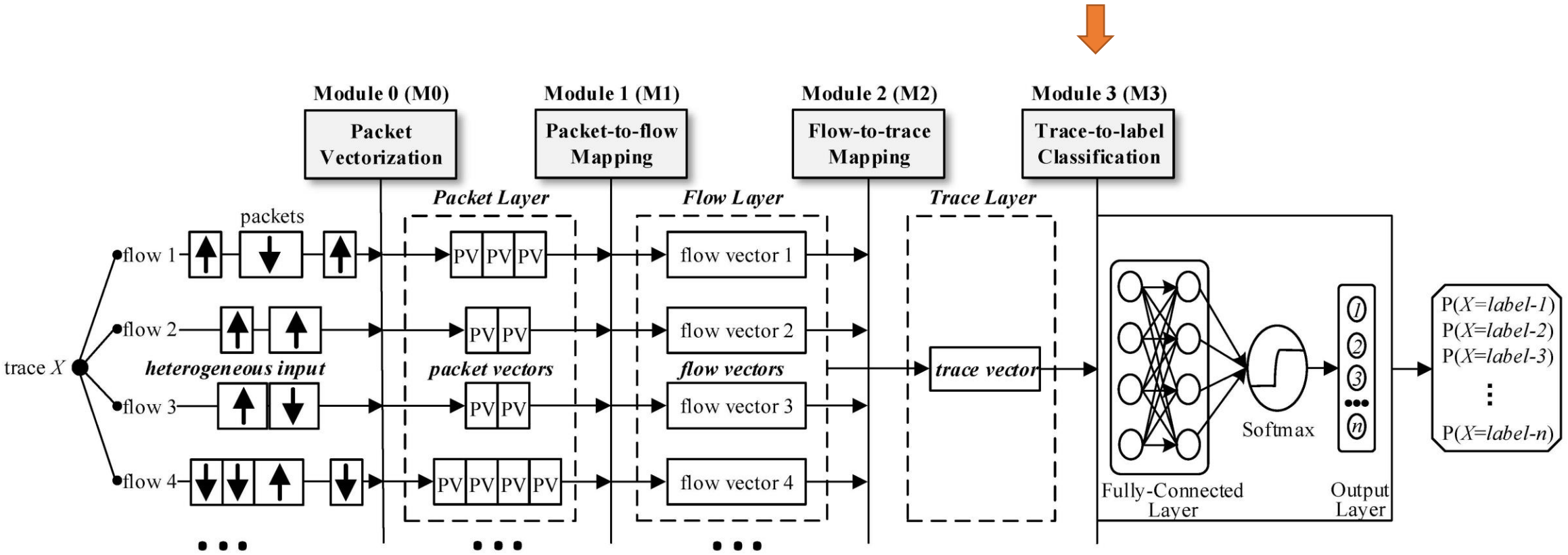
System Design



System Design



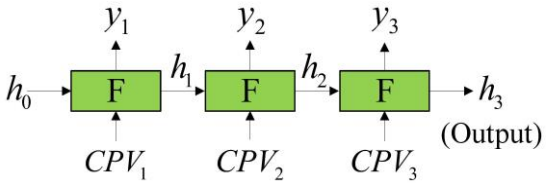
System Design



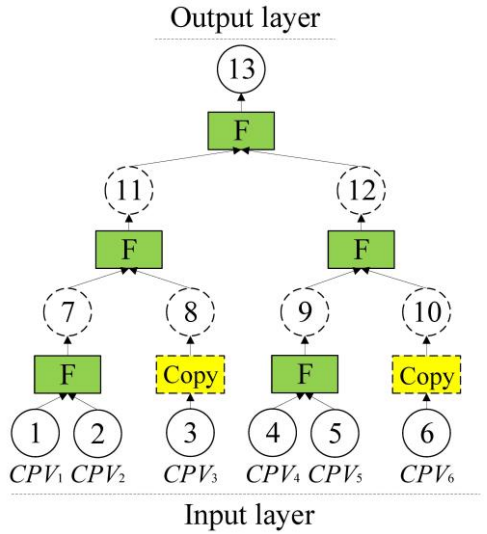
System Design

NN Structure in Packet-to-flow (M1) Mapping and Flow-to-trace (M2) Mapping

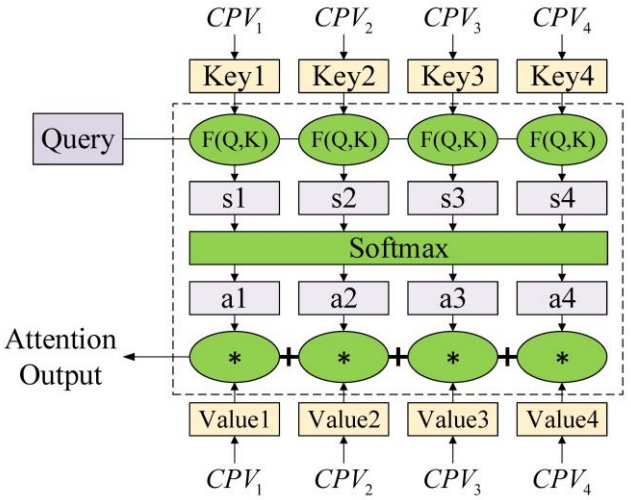
(1) Chain-structured



(2) Tree-structured



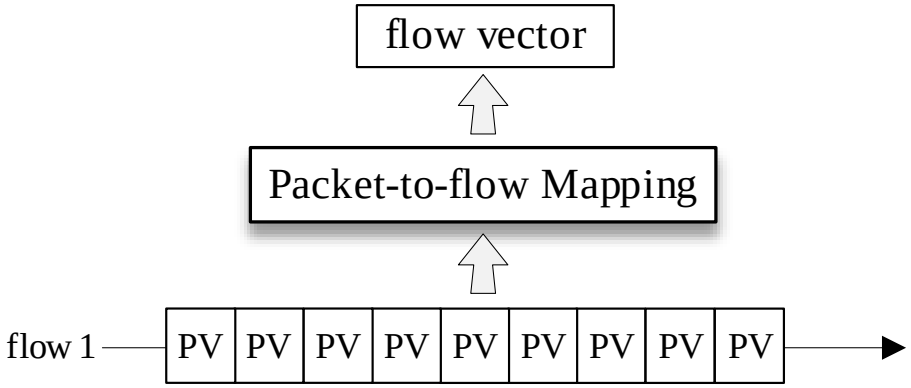
(3) Attention-structured



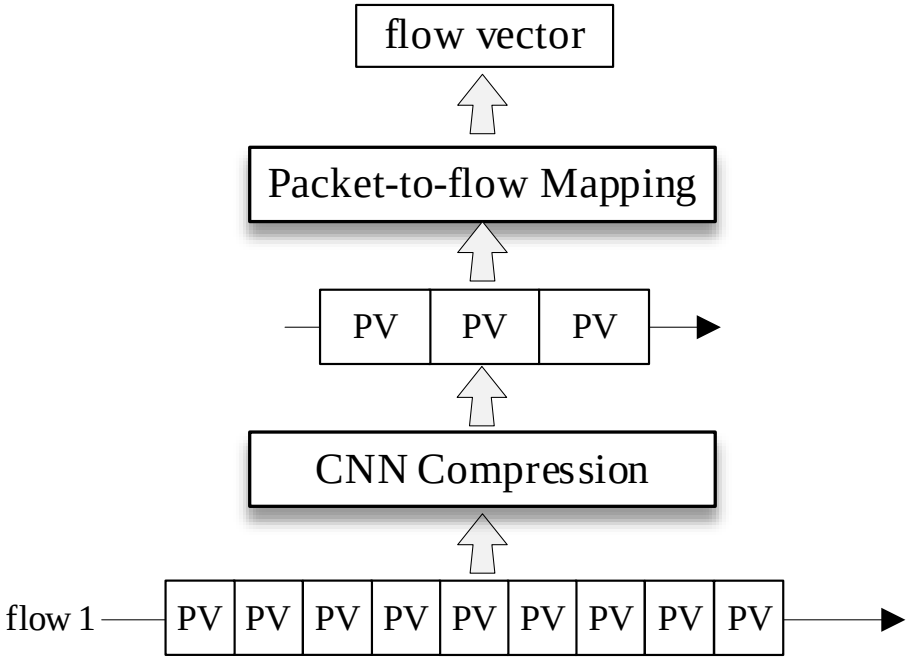
(4) Hybrid (uses multiple neural network structures)

System Design

Use CNN Compression to Speed up Training



(a) Without CNN Compression



(b) With CNN Compression

System Design

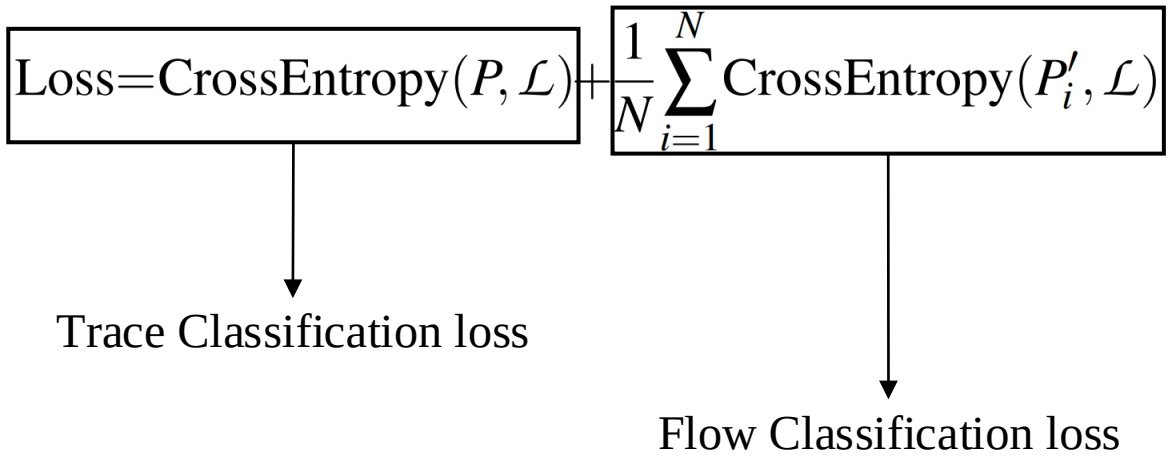
Techniques to Handle Overfitting

- Early Stopping
- Weight Decay
- Dropout
- Batch Normalization
- Auxiliary Loss
- Data Enhancement

System Design

Techniques to Handle Overfitting

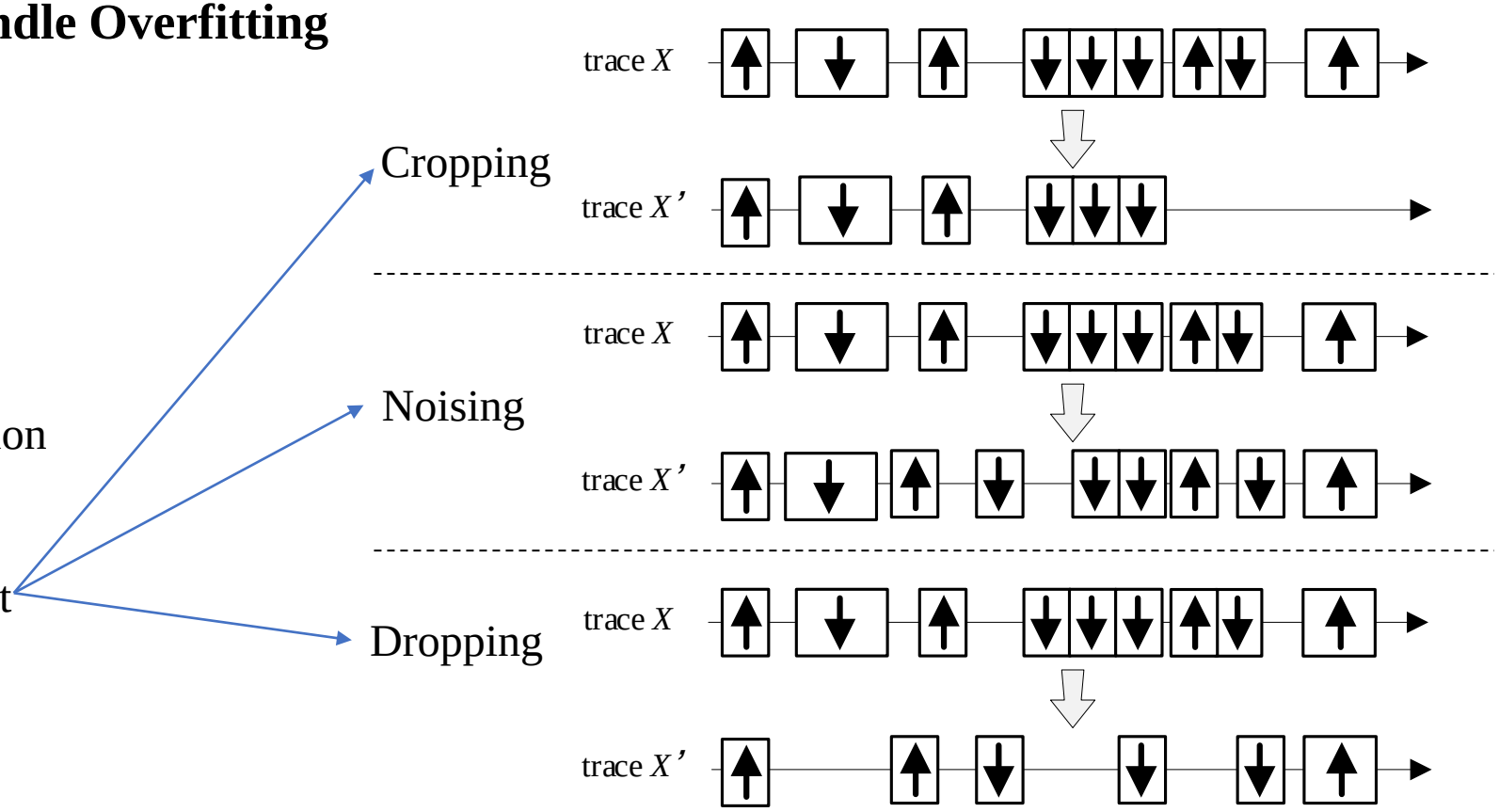
- Early Stopping
- Weight Decay
- Dropout
- Batch Normalization
- Auxiliary Loss
- Data Enhancement



System Design

Techniques to Handle Overfitting

- Early Stopping
- Weight Decay
- Dropout
- Batch Normalization
- Auxiliary Loss
- Data Enhancement



System Design

Techniques to Handle Overfitting

- Early Stopping
- Weight Decay
- Dropout
- Batch Normalization
- Auxiliary Loss
- Data Enhancement
- **Hybrid**

Outline

- Background and Problem Description
- System Design
- **Evaluation**
- Conclusions

Evaluation

Performance comparison with the SOTA methods.

Dataset	Our F1-score	SOTA method	SOTA F1-score
UAV [4]	0.928 (±0.023)	K-Means +RF [4]	**0.957**
SWF	**0.996** (±0.003)	RF+LCS [43]	0.982
KWF	**0.977** (±0.009)	PSC+ET [44]	0.974
IDI [30]	**0.940** (±0.040)	RF [45]	0.91
ISD [31]	**0.984** (±0.022)	CCR-ELM [46]	0.961

◆ Our method effectively fingerprint traffic across multiple tasks.

Datasets

- User Activities (UAV)
- IoT Device Identification (IDI)
- Intrusion Detection (ISD)
- Keyword Searching (KWS)
- Shadowsocks Website Fingerprinting (SWF)

Evaluation

Macro F1-scores using different neural network structures

M1 module	UAV [4]	SWF	KWS	IDI [30]	ISD [31]
Attention	**0.925** (±0.019)	**0.995** (±0.004)	0.977 (±0.010)	**0.968** (±0.024)	0.990 (±0.010)
Chain	0.922 (±0.029)	0.992 (±0.006)	**0.979** (±0.014)	0.954 (±0.033)	0.995 (±0.008)
Tree	0.923 (±0.023)	0.993 (±0.001)	0.976 (±0.010)	0.937 (±0.037)	0.986 (±0.014)
Hybrid	0.920 (±0.025)	0.993 (±0.003)	0.974 (±0.020)	0.940 (±0.040)	**0.997** (±0.005)

M2 module	UAV [4]	SWF	KWS	IDI [30]	ISD [31]
Attention	0.912 (±0.031)	0.991 (±0.003)	**0.979** (± 0.009)	0.924 (±0.055)	0.992 (±0.008)
Chain	**0.920** (±0.027)	**0.994** (±0.004)	0.950 (±0.014)	**0.963** (±0.030)	0.982 (±0.022)
Tree	0.907 (±0.027)	0.987 (±0.006)	0.806 (±0.039)	0.848 (±0.041)	0.992 (±0.012)
Hybrid	0.920 (±0.025)	0.993 (±0.003)	0.974 (±0.020)	0.940 (±0.040)	**0.997** (±0.005)

- Hybrid structures should be adopted for stable Macro F1-scores.

Evaluation

Macro F1-scores using different solutions to handle overfitting.
H-* removes method * from the hybrid solution.

Solutions		UAV [4]	SWF	KWS	IDI [30]	ISD [31]
No Handling		0.918 (±0.027)	0.956 (±0.025)	0.204 (±0.275)	0.721 (±0.137)	0.998 (±0.005)
Pure	ES	0.913 (±0.034)	0.953 (±0.024)	0.203 (±0.275)	0.723 (±0.134)	0.995 (±0.010)
	WD	0.922 (±0.022)	0.945 (±0.027)	0.391 (±0.391)	0.720 (±0.066)	0.998 (±0.005)
	DO	0.922 (±0.024)	0.736 (±0.161)	0.052 (±0.132)	0.277 (±0.105)	0.994 (±0.008)
	BN	0.922 (±0.021)	0.994 (±0.003)	0.970 (±0.011)	0.859 (±0.211)	0.992 (±0.011)
	AL	0.919 (±0.023)	0.990 (±0.005)	0.869 (±0.010)	0.833 (±0.070)	0.992 (±0.008)
	DE	0.921 (±0.022)	0.980 (±0.016)	0.140 (±0.261)	0.789 (±0.071)	0.995 (±0.010)
Hy-brid	H-ES	**0.932** (±0.023)	0.996 (±0.002)	0.820 (±0.076)	0.940 (±0.049)	0.998 (±0.005)
	H-WD	0.918 (±0.026)	0.995 (±0.004)	0.974 (±0.007)	0.933 (±0.037)	0.989 (±0.016)
	H-BN	0.917 (±0.024)	0.991 (±0.008)	0.872 (±0.033)	0.848 (±0.080)	0.994 (±0.008)
	H-AL	0.916 (±0.021)	0.993 (±0.003)	0.970 (±0.015)	0.944 (±0.040)	0.995 (±0.007)
	H-DE	0.924 (±0.024)	0.996 (±0.004)	0.973 (±0.005)	**0.958** (±0.026)	0.990 (±0.011)
	H	0.928 (±0.023)	**0.996** (±0.003)	**0.977** (±0.009)	0.935 (±0.041)	**0.998** (±0.004)

◆ Hybrid solutions should be adopted for high Macro F1-scores.

Evaluation

Macro F1-scores when confronted with hierarchy unawareness deep learning methods

Method	UAV [4]	SWF	KWS	IDI [30]	ISD [31]
HA-1.1	0.835 (± 0.024)	0.964 (± 0.012)	0.388 (± 0.365)	0.816 (± 0.072)	0.811 (± 0.057)
HA-1.2	0.906 (± 0.022)	0.966 (± 0.013)	0.927 (± 0.021)	0.840 (± 0.112)	0.914 (± 0.063)
HA-2	0.556 (± 0.025)	0.800 (± 0.022)	0.204 (± 0.012)	0.749 (± 0.018)	0.872 (± 0.029)
Ours	**0.928** (± 0.023)	**0.996** (± 0.003)	**0.977** (± 0.009)	**0.940** (±0.040)	**0.997** (±0.005)

- HA-1.1: Treat a trace consisting of multiple flows as a sample, **without** distinguishing between flows.
- HA-1.2: Treat a trace consisting of multiple flows as a sample, **with** distinguishing between flows.
- HA-2: Treat each flow of a trace as a sample, and classifying it into different trace labels.

◆ Hierarchy awareness is important.

Outline

- Background and Problem Description
- System Design
- Evaluation
- **Conclusions**

Conclusions

- We take the first step to designing an input-agnostic hierarchical deep learning framework to seamlessly land deep learning onto traffic fingerprinting.
- Our framework successfully applies in various fingerprinting tasks where SOTA methods rely on handcrafted features and deep learning is not easily applicable.
- We proposed techniques to handle overfitting and analyzed real-world factors that affect performance.
- Code available at https://github.com/shashadehuajiang/trace_classifier

Thank you!

Feel free to contact with any questions:

qj904154277@stu.xjtu.edu.cn