Smart Spying via Deep Learning: Inferring Your Activities from Encrypted Wireless Traffic

Tao Hou†, Tao Wang‡, Zhuo Lu† and Yao Liu†
†University of South Florida, Tampa, FL, USA, {taohou@mail., zhuolu@, yliu@cse.}usf.edu
‡New Mexico State University, Las Cruces, NM, USA, taow@nmsu.edu

***Abstract*—Wireless networks nowadays are ubiquitous in our daily life. Due to the open channel nature, wireless networks are vulnerable to eavesdropping attacks. Though wireless conversation can be encrypted against eavesdropping, attackers can still infer a user's activities via traffic analysis on encrypted data. Nevertheless, previous inference methods usually have the limitation that they can only achieve a relatively high accuracy in a specific domain. In this paper, we propose a smart spying strategy that can infer a user's activities of multiple domains with a higher accuracy. We also develop a prototype tool on top of this strategy to conduct experiments. The evaluation results show our strategy works effectively in activity inference on encrypted data, with an accuracy rate as high as 99.17%.**

***Index Terms*—Wireless Networking; Eavesdropping; Security Attack; Deep Learning; Neural Networks; Activity Inference.**

I. Introduction

Wireless networks nowadays are ubiquitous in our daily life. In wireless networks, a transmitter broadcasts wireless signals over the air, and any receiver within the transmitter's power coverage is able to receive the signal and decode the data from the signal. Such an open channel nature makes wireless communication vulnerable to eavesdropping attacks [1]–[6]. Once the transmitted data is obtained by an attacker, it can further decode the message to infer the user's sensitive activities (e.g., voice or video chatting) or even steal the user's private information (e.g., password and personal information), which is carried in the eavesdropped data.

A simple yet efficient method to prevent such information leakage on wireless communication is to encrypt the transmitted data [3], such that it is difficult for the eavesdropper to decode useful information. Researchers have proposed multiple encryption schemes to prevent sensitive information leaked to eavesdroppers [7]–[10]. However, data encryption does not stop the attacker from exploring new ways to spy on users. Through traffic analysis [11], [12] on the patterns or statistic results of side-channel information, attackers can successfully infer a user's activities on encrypted data [13]–[17].

However, these attacks usually only utilized the statistic results of features from a specific domain to perform activity inference. Consequently, they can only achieve a relatively high accuracy of user activity inference in a corresponding domain (e.g., APP usages [13], [14], spoken phrases [15], motions and behaviors [16], [17]). In this paper, we aim to overcome this limitation and improve the inference accuracy by proposing a smart spying strategy. This strategy can infer a user's activities of multiple domains with a higher accuracy on the encrypted data. The core idea behind this smart spying strategy is: 1) besides the statistic results of side-channel information, encoding the encrypted data to improve the data representativeness; 2) developing a fusion Deep Neural Network (DNN) model which integrates multiple traditional neural networks to improve the learning abilities. On top of this strategy, we develop a prototype tool called, SS-Infer (i.e., Smart Spying-Infer), which can efficiently infer a user's activities in real-world scenarios.

Specifically, we develop a mechanism to encode the encrypted data. In this way, our system can capture the characteristics concealed in the data payload, which are ignored by previous methods. Moreover, we design a classification model that involves multiple neural networks, and takes the advantage of multiple concatenated hidden layers to achieve a high classification accuracy. In particular, we first utilize the Convolutional Neural Network (CNN) to learn the spatial dependency features among the encoded data; and then adopt the Long Short-Term Memory (LSTM) to learn the temporal dependency features on the results from the first step. Finally, we combine the spatial-temporal features from previous steps with the flow features directly extracted from network traffic to improve the classification accuracy. With this proposed architecture, our evaluation results show that SS-Infer can achieve a classification accuracy rate as high as 99.17% when identifying a user's real-world activities.

The remainder of this paper is organized as follows. In Section II, we discuss the preliminaries and related work. In Section III, we introduce the system design of SS-Infer. Next, we present and discuss the experimental setups and results in Section IV. Finally, we discuss and conclude this paper in Section V.

II. Preliminaries and Related Work

In what follows, we briefly describe the background knowledge and related work of activity inference.

A. Wireless Eavesdropping

Eavesdropping attacks in wireless networks usually fall in two categories [4]: (1) passive eavesdropping and (2) active eavesdropping. Passive eavesdropping is also known as non-evasive eavesdropping, where the adversary only monitors and intercepts the data traffic between a transmitter and a receiver without any interference towards the transmit signals [18]. On

the other hand, an active eavesdropping adversary may actively intercept, interfere or even modify the transmit signals in favor of its eavesdropping performance [19]. Activity inference usually belongs to passive eavesdropping.

B. Transmission Encryption

To defend against the eavesdropping attacks, multiple security protocols have been proposed to preserve the confidentiality of the data traffic during the transmissions. For example, the Wi-Fi Protected Access II (WPA2) protocol has been widely adopted in modern wireless routers to secure the wireless communication at the MAC layer [20], [21]. Further, application layer encryption is also widely adopted to defend against eavesdropping. Examples of application layer encryption include Hypertext Transfer Protocol Secure (HTTPS), Pretty Good Privacy (PGP), Message Security Protocol (MSP) and etc [22].

C. Inference on Encrypted Data

Although it is difficult to directly decode the encrypted data, an eavesdropper can still infer a user's activities by analyzing different patterns of the network traffic [13]–[17]. For example, the work in [14] presents a hierarchical classification system to identify a user's online activities. The study in [15] proposes a spoken phrase classifier over the encrypted VoIP conversation. In [13], the authors analyze the data traffic of android APPs to detect and track a user's specific actions.

However, previous works only analyze features of a user's activities from a specific domain, they cannot be applied in activity inference of multiple domains. In this work, we attempt to design a smart spying strategy to overcome this limitation and further improve the inference accuracy. Towards this goal, 1) besides the statistics results of side-channel information, we encode the encrypted data to improve the data representativeness; 2) we develop a fusion DNN model which integrates multiple traditional neural network models to improve the learning abilities.

III. System Design

In this section, we first briefly introduce the architecture of SS-Infer, then present the technical details of each component.

A. Overview

Different network activities (e.g., browsing a website, streaming a video, making a VoIP call) indicate different behaviors in the level of traffic flows, rather than the behaviors of single packets. Normally, a semantically complete flow of traffic is the data transmitted during an active connection between two entities. We consider exclusively TCP/IP traffic flows in this paper, as they widely exist in today's networks and can be easily extended to other traffic types. As Figure 1 shows, there is a connection between Entity 1 and Entity 2, a traffic flow is transmitted through the connection. Usually, a traffic flow is composed of multiple data packets in transmission. We consider a wireless eavesdropper that can intercept data packets of different connections. According to the TCP/IP protocol, the intercepted packets will be then aggregated and grouped into traffic flows of different connections [23]. We further assume that the transmitted data is encrypted to prevent wireless eavesdropping attacks such that the eavesdropper cannot infer the user's activities by directly decoding the intercepted traffic.

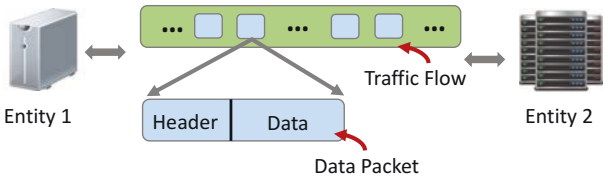

Fig. 1. Network traffic flow.

In this paper, we develop SS-Infer to infer a user's activities from encrypted wireless traffic data. The core idea behinds SS-Infer is to infer activities on top of a deep learning based traffic classifier, which is a fusion DNN model to integrate multiple traditional neural networks to improve the learning abilities. Moreover, other than traditional methods which only learn statistic information, SS-Infer can also learn spatial and temporal dependencies from the encrypted data to improve the information representativeness. SS-Infer aims to guarantee a high accuracy rate and maintain a relatively low computational cost.

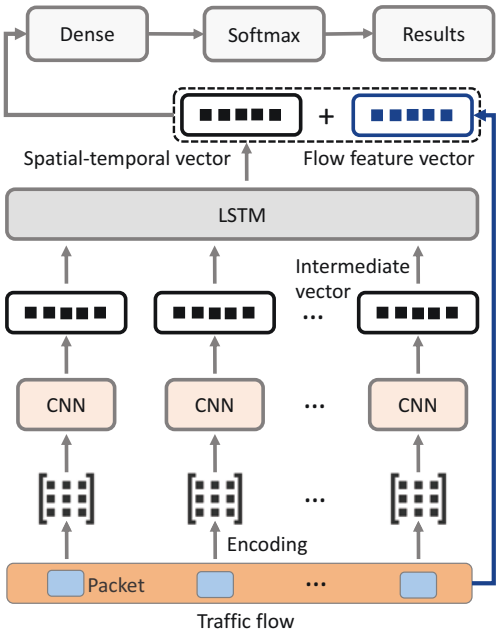

Fig. 2. System design of SS-Infer.

As Figure 2 shows, SS-Infer integrates both the internal layers of CNN and LSTM, such that it can capture not only spatial dependencies for the data in each packet, but also temporal dependencies among different packets. Besides, it also directly extracts flow features from the entire network traffic as additional features. Finally, SS-Infer makes the

TABLE I
FLOW FEATURES.

Category	Feature	Description
Header Information	Source Port	Port number at source
	Destination Port	Port number at destination
	Source Address	IP address of source
	Destination Address	IP address of destination
Statistical Information	Forward Inter-arrival Time	(mean, min, max, std) Inter-arrival time for forward packets in a traffic flow
	Backward Inter-arrival Time	(mean, min, max, std) Inter-arrival time for backward packets in a traffic flow
	Packet Length	(mean, min, max, std) Number of bytes for packets in a traffic flow
	Active Time	The time a flow was active
	Idle Time	The time a flow was idle
	Out of Order Count	The total number of packets that arrive destination out of order in a traffic flow
	Bytes per Second	The number of bytes transmitted per second in a traffic flow
	Packets per Second	The number of packets transmitted per second in a traffic flow

classification based on the combination of spatial-temporal vector and flow feature vector.

B. Learning Spatial-temporal Features

1) Data Encoding: Though encrypted, the data payload in a packet still contains information indicating a user's activities. In particular, the spatial and temporal correlation relationships among different packets can both contribute to the classification for different activities. SS-Infer learns these features to improve the data representativeness. Specifically, we adopt One-Hot Encoding (OHE) to represent the encrypted data, and we only take into consideration of packets with data payload larger than 300 bytes. The OHE vector is the binary code of each byte, i.e., an 8-dimensional vector. To improve performance, only the first 100 bytes, middle 100 bytes and last 100 bytes of a packet are encoded.

2) Learning Intermediate Vector through CNN: CNN is designed with the ability to learn the spatial dependencies. Here, we adopt internal layers of CNN to extract the spatial dependencies from each packet as the intermediate vector. This vector is the inputs of LSTM. Assume filter $\mathbf{w}$ works with a window size of s, $\mathbf{m}_i$ is the i-th generated feature, $\mathbf{c}_i$ is the i-th column of the data encoding matrix, $\mathbf{b}$ is a bias, and f is ReLUs. We get:

$$\mathbf{m}_i = f(\mathbf{w} \centerdot \mathbf{c}_{i:i+s-1} + \mathbf{b}), \tag{1}$$

Then, a max-over-time pooling operation is applied to the feature map $\mathbf{m} = [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_{300-h+1}]$ to get the intermediate vector:

$$\hat{\mathbf{m}} = max\{\mathbf{m}\}. \tag{2}$$

3) Learning Spatial-temporal Vector through LSTM: LSTM is suitable for learning temporal features, especially for the long-term temporal dependencies. We apply internal layers of LSTM after the convolution layers to learn the spatial-temporal dependencies. It takes the intermediate vector of each packet in order as inputs. Assume there are p valid packets in a flow, the input sequence is therefore denoted as $\{\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_2, ..., \hat{\mathbf{m}}_p\}$. Through a series of transitions by a set of adaptive multiplicative gates in these internal layers of LSTM, we get the output vector $\{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_p\}$, which is the spatial-temporal vector.

C. Extracting Flow Features

In addition to spatial-temporal features, we also extract flow features directly from the traffic flow to improve the data representativeness. There are two categories of flow features, the header information and the statistical information. We list the used features and their descriptions for both two categories in Table I.

- **Header Information:** This category of features includes the fields in the header of a data packet (e.g., source/destination port, source/destination address). Even when the data is encrypted, the header information in network layer and transport layer are still available. Such information is useful for activity inference. For instance, if the connection is established between a user and an IP belonging to YouTube, it is easy to infer that the user was watching streaming videos.
- **Statistical Information:** These features can be calculated from the entire traffic flow, such as the max (or min, or average) inter-arrival time between two packets, the max (or min, or average) packet length, total number of packets that arrive at the destination out of order. These statistical features are easy to calculate even when data is encrypted.

D. Final Classification

The vector used for final classification is the combination of the spatial-temporal vector and the flow feature vector. SS-Infer inputs the final vector to a dense layer and a softmax layer as shown in Figure 2 to perform the final classification.

IV. EVALUATION

In this section, we first introduce the experimental setups, then we evaluate the performance of the SS-Infer system and analyze the influence of size of flow features.

A. Experimental Setups

In our experiments, we use a high-performance workstation with four NVIDIA GeForce RTX 2080 GPUs, one 14-core Intel Core i9-9940X CPU and 128GB Memory to perform the wireless traffic classification on top of TensorFlow [24]. We use the UNB ISCX Network Traffic Dataset [25] for evaluation. It is captured from real world networks and the traffic flows are labeled manually as ground truth. Meanwhile, the UNB ISCX Network Traffic Dataset also contains the original encrypted data, which is needed for the classification in SS-Infer. The dataset includes seven types of encrypted traffic. Table II shows the details of the traffic types and the associated activities.

TABLE II
LIST OF TRAFFIC TYPES AND THE ASSOCIATED ACTIVITIES.

Traffic	Activity
Web Browsing	Browsing webs through Firefox and Chrome
Email	Sending/receiving email through SMTP/POP3/IMAP
Chat	Online chat through Skype, AIM, ICQ, etc
Streaming	Video streaming by watching Vimeo and Youtube
File Transfer	File transfer through FTP using Filezilla
VoIP	Voice call through Facebook, Skype and Hangouts
P2P	P2P download through uTorrent and Transmission

We adopt a 10-fold cross-validation to train and evaluate the performance of the SS-Infer system. The accuracy rate in the evaluation is defined as:

$$\text{Accuracy rate} = \frac{\text{Number of correctly classified flows}}{\text{Total number of flows}}.$$

B. Classification Performance

We first evaluate the classification performance of SS-Infer. In particular, we consider three classification models: Model1 is a neural network only using the spatial-temporal features for classification; Model2 is a neural network only using the flow features for classification, and Model3 combines the spatial-temporal features and the flow features, which is adopted in SS-Infer.

Figure 3 shows the evaluation results of the three different models. We can see that when only using spatial-temporal features or flow features to infer a user's activities, the accuracy rates are 93.63% and 85.26% for Model1 and Model2, respectively. Though the accuracy rate is already relatively high for activity inference, our SS-Infer design (i.e. Model3) can achieve a more accurate result, with an accuracy rate being as high as 99.17%.

C. Size of Flow Features

We also evaluate the impact on SS-Infer's performance when the number of flow features changes. The results are shown in Table III. This evaluation exhibits the changes of the accuracy rate and the computational time when we increase the number of flow features from 0 to the maximum of 21. Note that we define the computational time as the multiple of the baseline computational time (which is measured as the real time duration for the computation when the number of feature is 0).

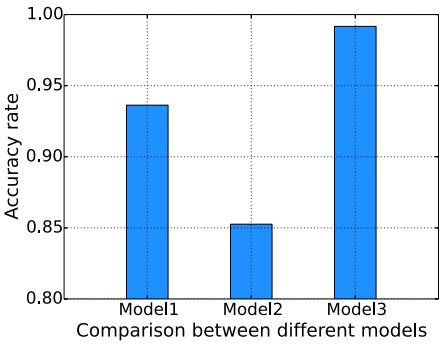

Fig. 3. Classification performance.

TABLE III
THE ACCURACY RATE AND COMPUTATIONAL TIME FOR DIFFERENT NUMBERS OF FLOW FEATURES.

Number	0	5	10	15	21
Accuracy (%)	93.36	98.53	99.10	99.15	99.17
Time	1.00	1.02	1.05	1.10	1.18

As Table III shows, the accuracy rate increases quickly until the number reaches 10, and the corresponding accuracy rate achieves 99.10%. After that point, the improvement of accuracy rate is relatively slight with the increase of the number of flow features, while the computational time continues increasing when the number increases. These results indicate that 10 flow features is a good choice for the tradeoff between the classification accuracy rate and the computational time. However, a powerful adversary can always choose to use all flow features to achieve the best accuracy rate in user activity inference.

V. CONCLUISION

In this paper, we design a smart spying strategy, named SS-Infer, which can accurately and efficiently infer a user's activity from encrypted wireless traffic. Comparing with other methods, SS-Infer has a stronger learning ability by integrating the advantages of multiple neural networks, and a better data representativeness by learning both spatial-temporal features and flow features. The evaluation results show that SS-Infer can achieve an accuracy rate as high as 99.17% in user activity classification.

ACKNOWLEDGEMENT

This work at USF was supported in part by NSF CNS-1553304 and CNS-1717969.

REFERENCES

[1] Yuanyu Zhang, Yulong Shen, Hua Wang, Jianming Yong, and Xiaohong Jiang. On secure wireless communications for iot under eavesdropper collusion. *IEEE Trans. on Automation Science and Engineering*, 2015.

[2] Tao Wang, Yao Liu, Qingqi Pei, and Tao Hou. Location-restricted services access control leveraging pinpoint waveforming. In *ACM CCS*, 2015.

[3] Song Fang, Tao Wang, Yao Liu, Shangqing Zhao, and Zhuo Lu. Entrapment for wireless eavesdroppers. In *IEEE INFOCOM*, 2019.

[4] Hong-Ning Dai, Hao Wang, Hong Xiao, Xuran Li, and Qiu Wang. On eavesdropping attacks in wireless networks. In *IEEE CSE and EUC and DCABES*, 2016.

[5] Shyamnath Gollakota and Dina Katabi. Physical layer wireless security made fast and channel independent. In *IEEE INFOCOM*, 2011.

[6] Tao Wang, Yao Liu, Tao Hou, Qingqi Pei, and Song Fang. Signal entanglement based pinpoint waveforming for location-restricted service access control. *IEEE Trans. on Dependable and Secure Computing*, 2016.

[7] Daniel E Lewis. Multi-level encryption access point for wireless network, 2003. US Patent 6,526,506.

[8] Jianming Zhu and Jianfeng Ma. A new authentication scheme with anonymity for wireless environments. *IEEE Trans. on Consumer Electronics*, 2004.

[9] Nachiketh R Potlapally, Srivaths Ravi, Anand Raghunathan, and Ganesh Lakshminarayana. Optimizing public-key encryption for wireless clients. In *IEEE ICC*, 2002.

[10] Prasidh Ramabadran, David Malone, Sidath Madhuwantha, Pavel Afanasyev, Ronan Farrell, John Dooley, and Bill O'Brien. A novel physical layer encryption scheme to counter eavesdroppers in wireless communications. In *IEEE ICECS*, 2018.

[11] Diala Naboulsi, Marco Fiore, Stephane Ribot, and Razvan Stanica. Large-scale mobile traffic analysis: a survey. *IEEE Communications Surveys & Tutorials*, 2015.

[12] Mauro Conti, Qian Qian Li, Alberto Maragno, and Riccardo Spolaor. The dark side (-channel) of mobile devices: A survey on network traffic analysis. *IEEE Communications Surveys & Tutorials*, 2018.

[13] Qinglong Wang, Amir Yahyavi, Bettina Kemme, and Wenbo He. I know what you did on your smartphone: Inferring app usage over encrypted data traffic. In *IEEE CNS*, 2015.

[14] Fan Zhang, Wenbo He, Xue Liu, and Patrick G Bridges. Inferring users' online activities through traffic analysis. In *ACM WISEC*, 2011.

[15] Charles V Wright, Lucas Ballard, Scott E Coull, Fabian Monrose, and Gerald M Masson. Spot me if you can: Uncovering spoken phrases in encrypted voip conversations. In *IEEE Symposium on Security and Privacy*, 2008.

[16] Mauro Conti, Luigi V Mancini, Riccardo Spolaor, and Nino Vincenzo Verde. Can't you hear me knocking: Identification of user actions on android apps via traffic analysis. In *ACM CODASPY*, 2015.

[17] Tao Jiang, Helen J Wang, and Yih-Chun Hu. Preserving location privacy in wireless lans. In *ACM MobiSys*, 2007.

[18] Dzevdan Kapetanovic, Gan Zheng, and Fredrik Rusek. Physical layer security for massive mimo: An overview on passive eavesdropping and active attacks. *IEEE Communications Magazine*, 2015.

[19] Sulei Wang, Zhe Chen, Yuedong Xu, Qiben Yan, Chongbin Xu, and Xin Wang. On user selective eavesdropping attacks in mu-mimo: Csi forgery and countermeasure.

[20] Shen Bai, Yi-jun WANG, and Zhi Xue. Research on security of wpa/wpa2 protocol [j]. *Information Security and Communications Privacy*, 2012.

[21] Randall K Nichols, Panos Lekkas, and Panos C Lekkas. *Wireless security*. McGraw-Hill Professional Publishing, 2001.

[22] Behrouz A Forouzan. *Cryptography & network security*. McGraw-Hill, Inc., 2007.

[23] Behrouz A Forouzan. *TCP/IP protocol suite*. McGraw-Hill, Inc., 2002.

[24] TensorFlow. https://www.tensorflow.org, 2019.

[25] Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of encrypted and vpn traffic using time-related features. In *ICISSP*, 2016.