LSTM based IoT Device Identification

Kahraman Kostas

Heriot-Watt University, Edinburgh EH14 4AS, UK

Another promising technique from deep learning methods is RNN. RNNs are used extensively in the analysis of sequential data such as video, audio and text. Network behaviours are also formed by the combination of sequential packets, and these packets follow a whole set of rules called "protocols". Because of this feature of network behaviour, many researchers have used methods such as RNN and its enhanced versions, LSTM and GRU, in their work [1–5].

Among these studies, Lopez-Martin et al.'s study [3] is particularly striking. In this study, CNN - RNN methods are used to classify IoT network traffic. In this study, TCP headers are used to obtain input data. For each flow, the first 20 packets are taken, and the six features of these packets are used (the source port, destination port, number of bytes in the packet payload, TCP window size (0 for UDP packets), interarrival time and direction of the packet). In this way, a 6x20 matrix time series is obtained for each network flow.

In our application, a strategy very similar to this method was followed. Since RNNs are used in the analysis of time series, we created consecutive sets of fingerprints to simulate the fingerprints of the devices into the time series. In the data used, there are 540 sessions in total. These sessions consist of 27 devices each producing 20 sessions.

By taking the first 12 packets of each session, a series of packets is created. The reason for choosing the first 12 packets is that this number creates the lowest limit in the dataset. The device, which produces the least number of network packets, produces 11 packets per session. Actually, this lower limit is 11, but instead of using a prime number like 11, the number 12, which is much more prone to division, was preferred. In this process, a 12th packet consisting of zeros was added to the session containing less than 12 packs. For session larger than 12, the packets after the 12th packet were ignored. As a result of this process, 540 matrices of 12X25 size were obtained. After that, data augmentation was performed on this dataset using the resampling and SMOTE methods.

In the learning step, among the many versions of RNN, 4 of them are used, whose names and brief descriptions are given below (Please see Figure 1 for the representation of architectures).

Vanilla LSTM: It is the simplest LSTM application. It consists of only one layer of LSTM [6].

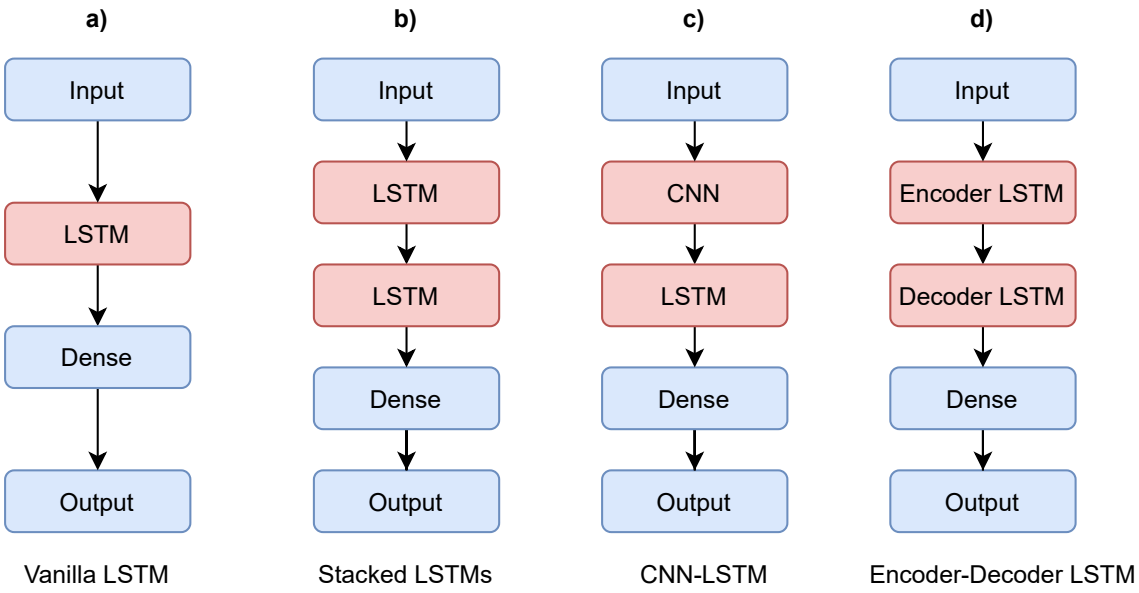

Fig. 1: The representation of architectures of LSTM types [6]

Stacked LSTM: It consists of multiple LSTM layers as hidden layer. It can also be called as deep LSTM [6].

CNN- LSTM: It consists of merging the CNN layer with the LSTM layer in the hidden layer. The CNN layer processes mxn size image / pseudo-image data. The results of this process are used as input in the LSTM layer [6].

Encoder-Decoder LSTM: It is a model consisting of combining two LSTM layers, one encoding and the other decoding. The Encode layer converts the inputs into vectors, and the decoding layer converts the vectors to the predicted output [6].

	CNN-LSTM	ED-LSTM	Stacked-LSTM	Vanilla-LSTM
Accuracy	0.763	0.750	0.740	0.769

Table 1: Comparison of 4 different types of LSTM Results

Table 1 shows the results of the applications of these 4 architectures, and Figure 2

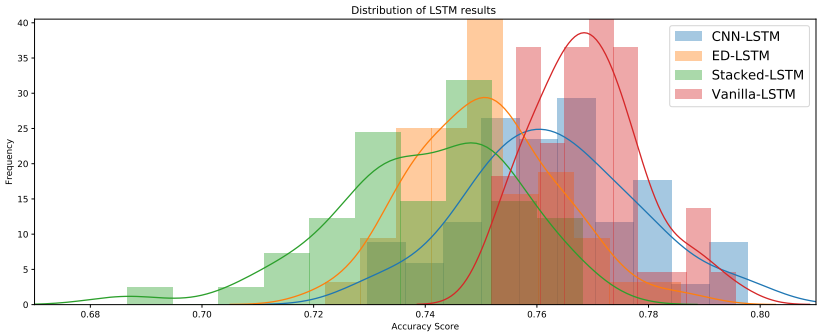

Fig. 2: Distribution of Accuracy Values

shows the distribution of these results. The results are the average of 50 repeats. All applications exceeded more than 0.70 accuracies, achieving notable success. Among the applications, Vanilla and CNN-LSTM have the highest accuracy, 0.769 and 0.763, respectively. The lowest value is the Stacked LSTM method. Statistical test results are available at Table 2 in Appendix G

References

1. H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computer Systems*, vol. 85, pp. 88–96, 2018.
2. N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.
3. M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.
4. N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2018.
5. J. Ortiz, C. Crawford, and F. Le, "Devicemien: network device behavior modeling for identifying unknown iot devices," in *Proceedings of the International Conference on Internet of Things Design and Implementation*, pp. 106–117, 2019.
6. J. Brownlee, *Long Short-term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning*. Machine Learning Mastery, 2017.

Appendix

Appendix G

4.2 RNN Implementation section Statistical test results

Test	P Value
$\text{ANOVA}_{\text{CNN-LSTM,ED-LSTM,Stacked-LSTM,Vanilla-LSTM}}$	1.32E-21
U_{test} (CNN-LSTM, ED-LSTM)	3.42E-05
U_{test} (CNN-LSTM, Stacked-LSTM)	8.02E-10
U_{test} (CNN-LSTM, Vanilla-LSTM)	0.007966986
U_{test} (ED-LSTM, Stacked-LSTM)	0.00066457
U_{test} (ED-LSTM, Vanilla-LSTM)	3.07E-11
U_{test} (Stacked-LSTM, Vanilla-LSTM)	1.16E-15

Table 2: Result of statistical tests of comparing LSTM architectures,U_{test}: Mann Whitney U Test The significance level is 0.05 for the ANOVA test and it is 0.0083 for the Mann Whitney U test