Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

An Improved Deep Dense CNN-LSTM Based Malware Identification and Classification

Lingaraj Sethi [1*], Prashanta Kumar Patra [2]
[1] *Biju Patnaik University of Technology, Rourkela, Odisha, India*
[2] *SOA University, Bhubaneswar, Odisha, India*

Abstract:- Malware identification and classification are receiving greater attention these days as a result of the growing number of attacks on financial and industrial networks. Malware categorization is very tough owing to the exponential increase in the quantity and variety of dangerous files. To have a strong malware defence and post-attack recovery mechanism in place, hostile files must be classified based on their goal, activity, and danger. Malware categorization is an undivided issue that is theoretically NP hard because to the NP hardness of the halting problem. However, as malware has gotten more sophisticated and complicated, traditional tactics have proven more useless. In this paper, we introduced a unique malware classification approach based on convolutional neural networks. CNN had a slightly higher accuracy of 89.7 percent, and when these two were combined (CNN+LSTM), we achieved an accuracy of 92.01 percent.

Keywords*:* cyber security; malware classification; convolutional neural networks.

1. Introduction

The internet has now become an important and unavoidable aspect of everyone's life. People utilise the internet for a variety of reasons, including social networking, banking, communication, shopping, and so on. Computers and the Internet are becoming more common and indispensable in everyday life. Malware (short for malicious software) is one of the most serious Internet risks. Malware is an abbreviation for malicious software that is designed to do damage. It is described as specifically created programmes designed to execute destructive operations. Malware assaults are becoming one of the most serious dangers to computer security. Malware assaults may impair a person's or organization's daily usage of computer systems, steal personal or secret information, damage data, or irritate users. Malware is divided into families, and the behaviour of malware in one family varies from that of malware in another. Finding the most effective malware detection method has become a key challenge in cyber security. Malware categorization is a critical and difficult subject in information security. Machine learning models, which may be trained on characteristics such as opcode sequences, API calls, and byte n-grams, among others, are used in modern malware classification approaches. Many machine learning algorithms have been presented to address the aforementioned issues [1–5]. Machine learning techniques, as opposed to signature-based methods, try to construct a data-driven model for malware identification based on given characteristics. Malware writers use a range of tactics and strategies when building code to mask their identity. As a consequence, the most difficult component is identifying the malware family or kind. Traditional antivirus software has a difficult time keeping up with the vast amount of malware that arises every day. To address this issue, computer scientists and antivirus corporations have started to use machine learning models. To recognise malicious software, academic researchers and developers have developed machine learning classifiers such as neural networks and logistic regression [6–7].

In recent decades, cloud-based applications have become more popular in a variety of application domains for a wide range of objectives involving a large number of human activities. Unfortunately, the number of cyber-attacks against cloud systems is rising on a daily basis. However, despite improvements in Artificial Intelligence research allowing for the resolution of many parts of the issue, malware categorization jobs remain difficult. A range of antimalware programmes, such as McAfee and Kingsoft, are available to protect users against the impacts of malware [8]. This anti-malware programme offers a security mechanism by detecting malware using signature-

based approaches [9]-[10]. There is a database in this approach that holds a certain signature formed by malware, therefore if an application contains that signature, it is identified as malware. However, it may modify its behaviour after a few minutes, causing the signature to change and no longer match, in which case it is classified as innocuous even though it is malware. To acquire control, attackers might modify the packaging of malware. This is a shortcoming of signature-based techniques [11]. Heuristic approaches are presented to work, which are based on simple guidelines offered by security specialists. However, if such procedures are carried out manually, they are unlikely to equal the pace of malware preparation [12]. To address this issue, there is a growing trend of employing automated malware categorization algorithms. To address this issue, there is a strong movement toward creating automated malware categorization techniques based on deep learning algorithms. These devices can identify unknown malware as well as known malware. The detection system is divided into two parts: feature extraction, which is the most significant, and classification, which is the second. The main objective for this study is to use deep learning technology in conjunction with the self-attention mechanism to capture rich context and semantic information. The following are the primary contributions of this work:

- We suggested a byte-level CNN-LSTM model to investigate relevant aspects in binary executables' one-dimensional structure. The experimental findings suggest that our CNNLSTM model can produce promising outcomes by reducing bit/byte-level sequences to lower sizes.
- We presented a bit-level CNN-LSTM mode in addition to the byte-level 1D CNN model by expanding bytes into bits from byte sequences. Our investigations suggest that we can improve performance by augmenting the information from the byte sequences using bit expansion.
- In terms of the number of multiply-add operations, our suggested CNN-LSTM achieves better or equivalent outcomes with less computational cost than CNNs.
- Our trials give thorough experimental data for CNN-LSTM models with various resizing lengths. The findings might serve as a reference for future CNN-LSTM-based malware classification systems that use alternative scaling lengths. The remainder of the paper is structured as follows: Section II reviews comparable work, and Section III explains our methodology. Section IV describes our empirical evaluation benchmarking platform and dataset, as well as insights into the outcomes gained. Section V summarises the article and suggests future research.

2. Objectives

The primary objective of the research is to Model hyperparameter tuning reveals best-practice parameters, and the ensemble confusion matrix delves into classification efficacy. The research also aims to analyze comparing the proposed approach to current methods show that it is superior at detecting malware. Finally, the research work

Proposes suggestions for a safe environment to deploy the model and for frequent updates to address shifting cybersecurity threats.

3. Methods

A convolutional neural network (CNN), on the other hand, is intended to cope with local structure. When substantial information is not local, a convolutional layer cannot be expected to perform effectively. Because of the smaller amount of weights, CNNs can train convolutional layers significantly more effectively than fully connected layers.

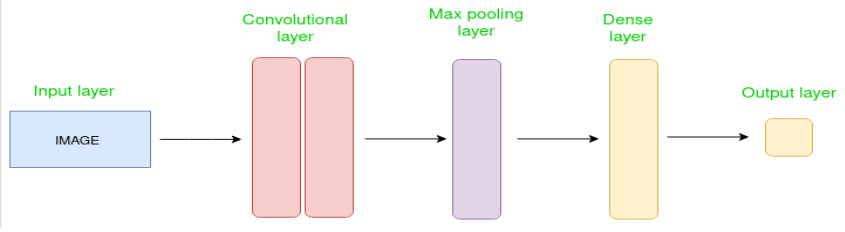

Fig.1 Architecture of CNN

Long-term memory (LSTM) networks are a kind of RNN architecture that is intended to cope with longterm dependencies. That is, LSTM can cope with long "gaps" between the emergence of a feature and the moment at which the model requires it. Due to vanishing gradients, this is often not achievable with simple vanilla RNNs. The primary distinction between an LSTM and a typical vanilla RNN is that an LSTM has an extra information flow channel. In other words, in addition to the concealed state, there is a cell state that may be utilised to effectively retain information from earlier phases. During backpropagation, the cell state is intended to act as a gradient "highway." In this manner, the gradient may "flow" considerably farther back with less risk of disappearing (or exploding) along the way. As an aside, the LSTM architecture is one of the most commercially effective learning approaches ever invented.

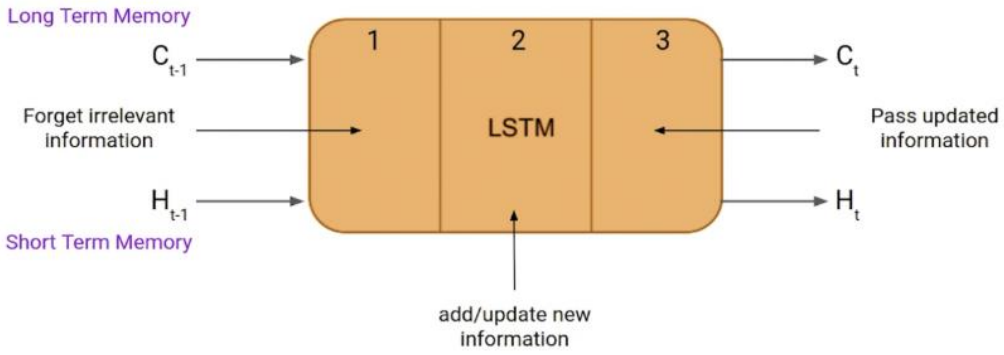

Fig.2 LSTM architecture

LSTMs are used in a variety of applications, including Google Allo [18], Google Translate [19], Apple's Siri [20], and Amazon Alexa [21]. BiLSTM models are LSTM extensions that analyse data in both forward and backward directions in two independent LSTM layers.

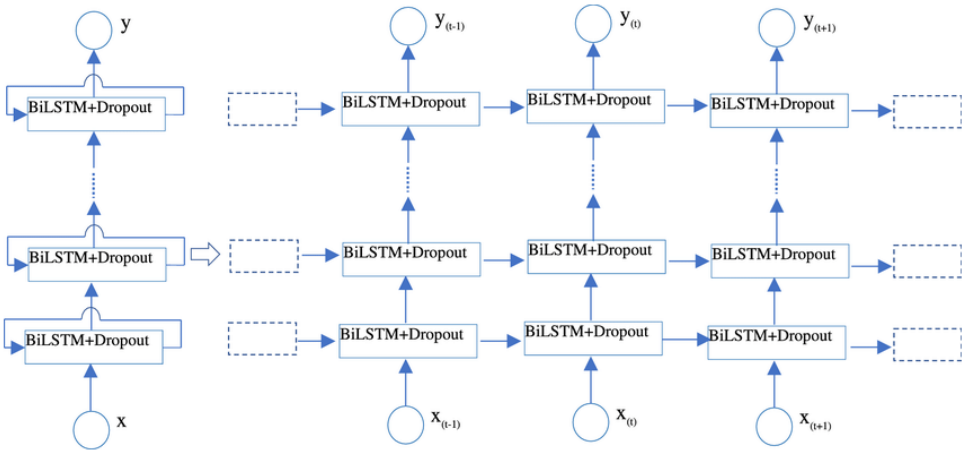

Fig. 3 Bi-LSTM

The forward layer processes the input similarly to a regular LSTM, whereas the backward layer processes the same data in reverse order [22]. The initial layer and the point of entry into a neural network is the input layer.

- Dropout Layer: During training, it adds chaos to the network by randomly breaking the amount of connections between neurons from one layer to the next. Overfitting is decreased as a result, enabling models to generalise more effectively. This usually results in increased model correctness throughout assessment.
- LSTM Layer: Implements a single LSTM layer with all forward and backward propagation techniques
- A wrapper layer that enables RNN layers to construct bidirectional models. Instead of building two distinct RNN layers for forward and backward direction and concatenating the results, the bidirectional wrapper layer performs it all in one layer.

- Dense Layer: A single fully linked vanilla neural network layer is implemented.
- Embedding Layer: This layer is in charge of converting positive integers into vectors of floating point values.
- Conv1D Layer: A one-dimensional implementation of the convolutional neural network layer.
- MaxPooling1D Layer: This layer implements the maximum pooling operation in a single dimension. Algorithm 1 depicts the pseudo-code for constructing this design. It is simple to translate such pseudo code into real implementation using high-level domain-specific deep learning technologies like Keras. In practise, fine-tuning the model parameters is even more difficult and time intensive.

Algorithm 1: Hybrid Classification Component

Steps:

1. Add Embedding to Input Layer
2. CNN applied to Embedding Layer
3. LSTM applied to the input
4. Dropout applied to LSTM
5. Dense Layer adder to dropout
6. Activation Layer applied to dense
7. Train using RNN
8. Testing of model

The overall workflow of the model is described in Fig.4.

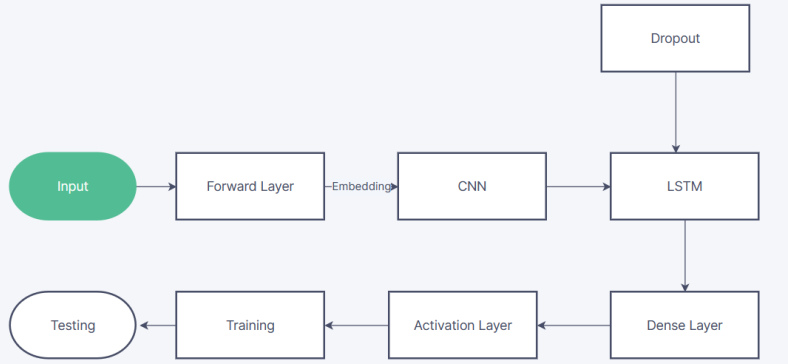

Fig. 4 Overall workflow of the model

4. Results

In this part, we assess the efficacy of our proposed framework on the formention dataset. We begin by introducing the dataset and the statistical outcomes of preprocessed samples. The model is then trained to assess the results after we display the training settings. Furthermore, several assessment criteria are presented. Finally, we provide the experimental findings in detail in comparison to a previous study baseline. As previously stated, this study makes use of a publicly accessible dataset released on Kaggle by Microsoft for the Malware Classification Challenge (BIG 2015) [23]. Anti-malware industry professionals have classified this information into nine distinct groupings. The competition on Kaggle includes different training and assessment datasets. The performance of our approach is evaluated exclusively by conducting hold-out validation on the Kaggle training dataset. The graph depicts performance variances in terms of the F1-score measure for several models under 30 repetitions. We can observe that at 80% of training data amount, performance variance (with 20% test set) is quite little. We were able to obtain more than 97 percent accuracy with 99 percent sensitivity, 95 percent specificity, and 97 percent

precision. Other models may attain an accuracy of approximately 90% with a sensitivity of 96 percent, specificity of 80%, and precision of 92 percent in certain repetitions.

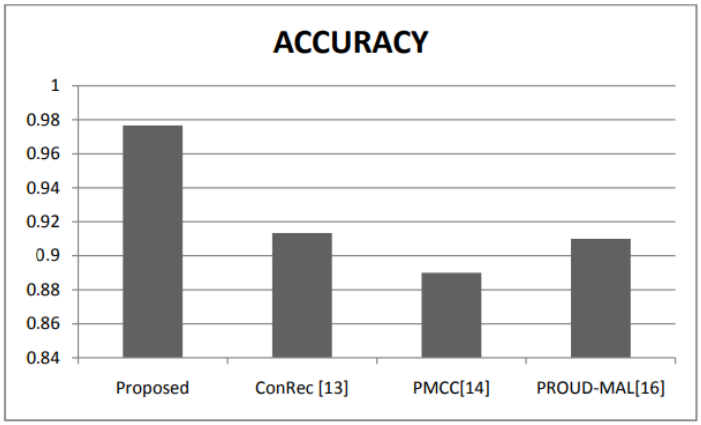

Fig.3 Comparison of accuracy

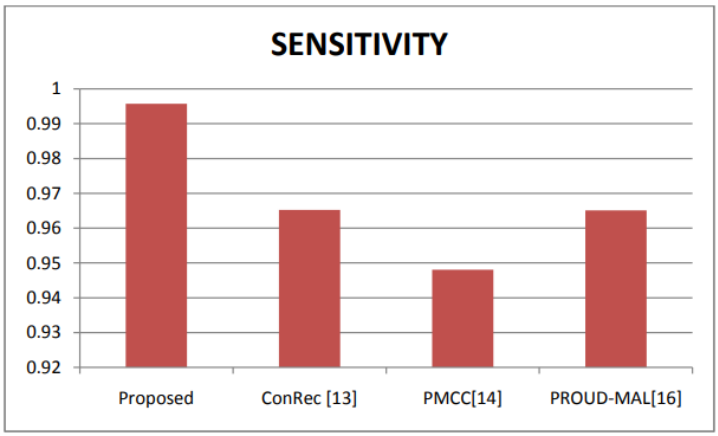

Fig.4 Comparison of Sensitivity

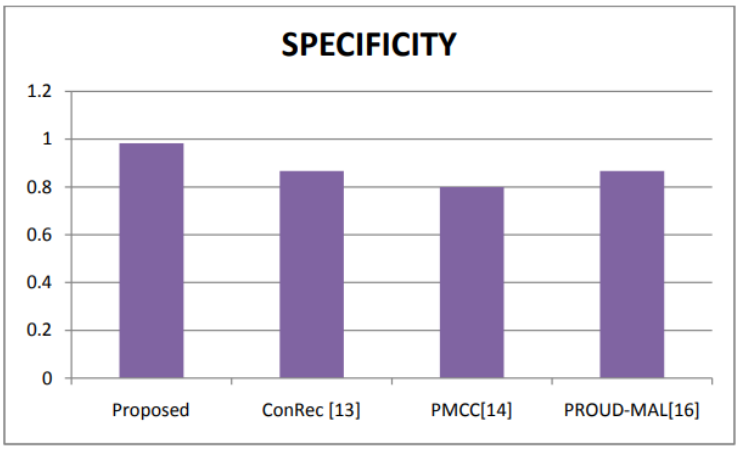

Fig. 5 Comparison of Specificity

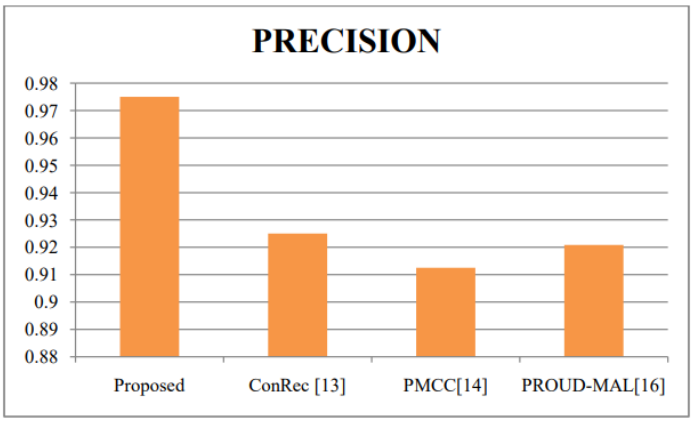

Fig.6 Comparison of Precision

5. Discussion

We developed a system and approach for benchmarking malware classification using deep neural networks in this study. The results clearly show that extracting features using CNN and classification with LSTM delivers the best performance for Malware classification. When just compiled files are used to categorise malware programmes, an accuracy of 97.4 percent is attained. A innovative strategy for categorising assembly files using a basic LSTM network is provided in this study. Furthermore, suggested methodologies avoid the need of domain-specific techniques such as feature engineering, reverse engineering, disassembly, and others previously employed for malware classification.

Refrences

[1] Primiero, Giuseppe, Frida J. Solheim, and Jonathan M. Spring. "On malfunction, mechanisms, and malware classification." Philosophy & Technology 32 (2019): 339-362.

[2] Saeed, Mariwan Ahmed Hama. "Malware in computer systems: Problems and solutions." IJID (International Journal on Informatics for Development) 9, no. 1 (2020): 1-8.

[3] Christopher Elisan (5 September 2012). Malware, Rootkits & Botnets A Beginner's Guide. McGraw Hill Professional. pp. 10–. ISBN 978-0-07-179205-9

[4] Imtithal A Saeed, Ali Selamat and Ali M A Abuagoub. Article: A Survey on Malware and Malware Detection Systems. International Journal of Computer Applications 67(16):25-31, April 2013.

[5] Zahra Bazrafshan et al.., A survey on heuristic malware detection techniques, 5th Conference on Information and Knowledge Technology (IKT), 13, 2013, 113-120.

[6] Imtithal A. Saeed, & Ali M. A. Abuagoub, A Survey on Malware and Malware Detection Systems, International Journal of Computer Applications, 67 (16), 2013, 25-31.

[7] Gil Tahan, Lior Rokach, & Yuval Shahar, Mal-ID: Automatic Malware Detection Using Common Segment Analysis and Meta-Features, Journal of Machine Learning Research, 1, 2012, 1-33.

[8] Levy H.M. Saroiu S., Gribble S.D. Measurement, and analysis of spyware in a university environment. Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI '04), San Francisco, CA, March 2004

[9] Jyoti Landage, Prof. M. P. Wankhade, Malware and Malware Detection Techniques, A Survey, International Journal of Engineering Research and Technology (IJERT), 2 (12), 2013, 61-68

[10] D. Evett. (2006) "More malware-adware, spyware, spam, and spim". [Online]. Available: http://aic.gov.au/media_library/publications/htcb/htcb011.pdf. [Accessed: 15- Dec- 2015].

[11] www.mcafee.com. (2005) "Potentially Unwanted Programs Spyware and Adware". [Online]. Available: http://www.mcafee.com/us/resources/white-papers/wp-potentiallyunwanted-programs-spyware-adware.pdf. [Accessed: 15- Dec2015].

[12] DuPaul, N. Common Malware Types: Cybersecurity 101. OCTOBER 12, 2012 3/21/2018]; Available from: https://www.veracode.com/blog/2012/10/common-malware-types-cybersecurity-101.

[13] Zhang Xiaolei. The diagnosis and prevention of computer virus [M]. Beijing: China Environmental Science Press, 2008
[14] La Polla, M., F. Martinelli, and D. Sgandurra, A survey on security for mobile devices. IEEE communications surveys & tutorials, 2013. 15(1): p. 446-471.
[15] Sharma, Vishrut. "An analytical survey of recent worm attacks." IJCSNS 11, no. 11 (2011): 99-103.
[16] Hoque, Nazrul, Dhruba K. Bhattacharyya, and Jugal K. Kalita. "Botnet in DDoS attacks: trends and challenges." IEEE Communications Surveys & Tutorials 17, no. 4 (2015): 2242-2270.
[17] Ammar Ahmed E. Elhadi, Mohd Aizaini Maarof, & Bazara I. A. Brry, Improving the Detection of Malware Behavior Using Simplified Data Dependent API Call Graph, International Journal of Security and Its Applications, 7 (5), 2013, 29-42.
[18] Muni Prashneel Gounder, Mohammed Farik. New Ways To Fight Malware, International Journal of Scientific & Technology Research, ISSN 2277-8616, Volume 6, Issue 06, June 2017
[19] Manju Khari, Chetna Bajaj. Detecting Computer Viruses, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), ISSN: 2278 – 1323, Volume 3 Issue 7, July 2014.
[20] Karim, Asif, Sami Azam, Bharanidharan Shanmugam, Krishnan Kannoorpatti, and Mamoun Alazab. "A comprehensive survey for intelligent spam email detection." IEEE Access 7 (2019): 168261-168295.
[21] International Standards Organization (ISO). ISO/IEC 17799 information technology security techniques: code of practice for information security management. Geneva: ISO; 2005.
[22] Whitman M. Enemy at the gate: threats to information security. Communications of the ACM 2003;46:91–5.
[23] Stacey TR, Helsley RE, Baston JV. Identifying information security threats. Information Systems Security 1996;5:50–9.
[24] Loch KD, Carr HH, Warkentin ME. Threats to information systems: today's reality, yesterday's understanding. MIS Quarterly 1992;16: 173–86.
[25] Solms RV, Solms BV. From culture policies. Computers & Security 2004;23:275–9.
[26] Erbschloe M. Trojans, worms, and spyware: a computer security professional's guide to malicious code. Butterworth-Heinemann; 2004.
[27] Mitnick KD, Simon WL. The art of deception: controlling the human element of security. Wiley & Sons; 2002.
[28] Wagner A, Dübendorfer T, Plattner T, Hiestand R. Experiences with worm propagation simulations. In: Proceedings of the 2003 ACM Workshop on Rapid Malcode. Washington, DC; 2003.
[29] Sanok D. An analysis of how antivirus methodologies are utilized in protecting computers from malicious code. In: Proceedings of the 2nd Annual Conference on Information Security Curriculum Development. Kennesaw, GA; 2005.
[30] Brownlow M. E-mail and website statistics. E-mail Marketing Reports; May 3, 2008.
[31] Kienzle D, Elder M. Recent worms: a survey and trends. In: Proceedings of the 2003 ACM Workshop on Rapid Malcode. Washington, DC; 2003.
[32] National Institute of Standards and Technology (NIST). History of worms. Available from; 1994.
[33] Jakobbson M, Myers S. Phishing and countermeasures: understanding the increasing problem of electronic identity theft. WileyInterscience; 2006.
[34] Tiauzon S. TrendLabs malware blog. Available from; 2006.
[35] Ramzan R. Drive-by-pharming in the wild. Symantec. Available from; 2008.
[36] Tepper M. The rise of social software. netWorker 2003;7:19–23.
[37] Narain R. Rogue's advertisement pushes scareware to NYTimes.com readers. Threat Post: Kaspersky Lab Security News Service; 2009.
[38] Jagatic T, Johnson N, Jakobsson M, Menczer F. Social phishing. Communications of the ACM 2007;50:94–100.
[39] Dang H. The origins of social engineering. McAfee Security Journal; 2008. Fall
[40] Andreas Moser, Christopher Kruegel, and Engin Kirda, Limits of Static Analysis for Malware Detection, Secure Systems Lab Technical University Vienna.