The effectiveness of classification algorithms on IPv6 IID construction

Clinton Carpene*, Michael N. Johnstone and Andrew J. Woodward

School of Computer and Security Science,
ECU Security Research Institute,
Edith Cowan University,
270 Joondalup Drive, Joondalup,
Western Australia, 6027, Australia
Email: c.carpene@ecu.edu.au
Email: m.johnstone@ecu.edu.au
Email: a.woodward@ecu.edu.au
*Corresponding author

Abstract: This study assessed the effectiveness of classifying IPv6 interface identifier (IID) address construction using machine learning algorithms. It was observed that IID construction can be reliably determined through the usage of assisted machine learning algorithms such as the naïve Bayesian classifiers (NBC) or artificial neural networks (ANNs). It was also observed that the NBC classification, whilst more efficient, was less accurate than the use of ANN for classifying interface identifiers. Training times for an unoptimised ANN were seen to be far greater than NBC, which may be a considerable limitation to its effectiveness in real world applications (such as log or traffic analysis). Future research will continue to improve the classification training times for ANN situations, potentially involving general-purpose computing on graphics processing units (GPGPU) systems, as well as applying the techniques to real world applications such as IPv6 IDS sensors, honeypots or honeynets.

Keywords: IPv6; privacy; machine learning; IID construction; artificial neural networks; ANNs; naïve Bayesian classifiers; NBC.

Reference to this paper should be made as follows: Carpene, C., Johnstone, M.N. and Woodward, A.J. (2017) 'The effectiveness of classification algorithms on IPv6 IID construction', *Int. J. Autonomous and Adaptive Communications Systems*, Vol. 10, No. 1, pp.15–22.

Biographical notes: Clinton Carpene is a PhD candidate in Computer Science and Security. His PhD topic seeks to address the specific problem of locating networked nodes in an IPv6 environment. The benefits of the research are to improve machine-learning systems used for classification, as well as improve visibility over IPv6 networks. He has also performed research in a number of other topics including security issues in mobile devices, hard drives and the smart grid.

Michael N. Johnstone gained his MSc and PhD degrees from Curtin University in 1997 and 2008, respectively. He is a Senior Lecturer at Edith Cowan University (ECU) in Western Australia where he teaches secure programming and software engineering. He is a member of the Security Research Institute at ECU. His research interests include secure development methodologies (for mobile applications), wireless sensor networks (for military systems) and cloud

security (for e-health data sharing). He has been a Contractor for private industry, government and research organisations and has held various roles including programmer, systems analyst, project manager and network manager before moving to academia. He is a member of the Australian Computer Society.

Andrew J. Woodward gained his BSc (Hons) and PhD degrees from Edith Cowan University in 1995 and 2004, respectively. He is an Associate Professor and Head of School of Computer and Security Science at Edith Cowan University (ECU) in Western Australia. He is a member of the Security Research Institute at ECU. His research interests include wireless network security and digital forensics. He has been a contractor for private industry, government and research organisations and has held various roles including IT manager and network security analyst before moving to academia. He is a member of the Australian Computer society.

This paper is a revised and expanded version of a paper entitled 'The effectiveness of classification algorithms on IPv6 IID construction' presented at 5th Symposium on Cyberspace Safety and Security, Zhangjiajie, China, 13–15 November 2013.

1 Introduction

Internet protocol version 6 (IPv6) (Deering and Hinden, 1998), proposed to address the limited IPv4 address space, is being increasingly implemented globally. Since its ratification in 1998, IPv6 adoption rates have slowly increased. In addition to social media services, which hold significant private information about their users, there is increasing attention being paid to internet traffic at the Layer 3 (IP) level. These data are used by organisations to target advertising to those browsing the web, but data can also be mined to determine other information about a user. There are frequent reports of issues with social media privacy settings leading to users having their personal information made available to the wider internet. Reports of these types of privacy issues have raised concerns in the wider community about technology in general, and more specifically whether the exploitation of data made available by users of these services is creating privacy problems for them.

With more services becoming IPv6 enabled, and the eventual transition to the protocol moving closer, security aspects of the protocol become increasingly important. One aspect of protocol security that has been identified is the makeup of the IPv6 addresses themselves (Carpene and Woodward, 2012). As the address space is so large, it is possible that addresses could be used to fingerprint users or devices.

The introduction of the IPv6 protocol brings with it a number of means by which a computer or user can derive the host portion of its unique, 128 bit IPv6 address(es). These methods, discussed in Carpene and Woodward (2012), include using stateless address autoconfiguration (SLAAC) (Narten et al., 2007b), dynamic host control configuration protocol for IPv6 (DHCPv6) (Droms et al., 2003), or by manual assignment. Each of these methods listed can be examined for various footprints, which can be used to deduce how the addresses were constructed. This can provide further insight for network traffic monitoring systems (such as IDS/IPS, honeypots, firewalls and

log analysis systems). Machine learning algorithms have previously been applied to cyber security issues (Johnstone and Woodward, 2012), and in this study we tested the capabilities of two artificial intelligence classification algorithms, naïve Bayesian classification and artificial neural networks (ANN), in determining the means of interface identifier (IID) address construction (Haykin, 2009).

2 IID construction

Under IPv6 there are two main categories of IPv6 IID construction; stateless and stateful approaches. Stateless approaches (such as SLAAC) allow a networked device to generate an IID without intervention. Alternately stateful means (such as DHCPv6 or manual assignment) require third party intervention, either from a human or server.

SLAAC is an address assignment policy defined by RFC 4862 (Narten et al., 2007b). Under SLAAC, hosts self-configure a unique IPv6 address, without the requirement for manual intervention or DHCP. Under the scheme, the host is provided with the first 64 bits of the address, which is known as the network prefix. The host then derives the IID portion represented as a modified extended unique identifier 64 bit (EUI-64). Once generated, an IID is statically assigned to the device perpetually, which can also incur unwanted device tracking and raising potential privacy concerns. Further, part of the process used to derive a modified EUI-64 address involves incorporating the device's MAC address. The MAC address can then be simply reverse-engineered from the IPv6 address.

To address the above concerns, privacy extensions were introduced into SLAAC by RFC 4941 (Narten et al., 2007a). Privacy extensions define a method of generating a temporary IID by cryptographic means in order to prevent traceability of a device, and obfuscate the MAC address inside Layer 3 transmissions. The temporary IIDs are designed to be short-lived (e.g., the duration of a network session), after which a new IID will be generated. Due to the cryptographic process used to determine the IID resulting in a pseudo-random address string, the privacy extended address is detectable.

DHCPv6 lets a server allocate IP addresses to networked devices from an allocation pool specified by an administrator. The allocation scheme is simple, based on a first-come, first-served method, from the address pool. Consequently, the addresses are allocated in a linear fashion. Manual assignment involves an administrator allocating addresses to devices according to some network policy. Generally, the purpose of manual assignment is to ensure device addresses are easily memorable. These results in little variation of addresses configured manually (Carpene and Woodward, 2012).

It has been discussed that these methods of IID generation can be distinguished from each other through various footprints (Carpene and Woodward, 2012). As an example, a DHCPv6 server will generally distribute addresses sequentially from a starting point until the pool is exhausted (Droms et al., 2003). This means that IIDs will follow a standard pattern, with minimal changes until the last quartet. In the case of SLAAC, standard padding (the hexadecimal string 0xfffe) is imposed at bits 24 to 48 of the IID. In the case of SLAAC with privacy extensions, it can be assumed that the IID will be a pseudo-random string of hexadecimal characters (since the means for creating privacy extensions involves performing an MD5 hash function against the IID). The above demonstrates that IIDs can be classified into three schemes, which are used within this

study, SLAAC, SLACC w/Privacy Extensions and Manual Assignment/DHCPv6. If these address construction schemes could be reliably classified, then a privacy issue would exist since the information can provide insight into a network's configuration.

3 Assisted machine learning

To test the hypothesis of whether address construction schemes can be classified reliably using automated means, potentially suitable machine-learning algorithms (naïve Bayesian classifier and ANN) were selected and employed (Johnstone and Woodward, 2012).

A NBC assumes that the determining attributes of a class are independent of one another. This means that using such an approach is swift and requires little training, making this algorithm particularly suited to machine learning, and especially classification problems (Zhang, 2004). The assumption may not be true in practice (and often is not the attributes that determine an object are frequently dependent on one another) but can be accurate despite this simplification.

An ANN is an algorithm that attempts to mimic the behaviour of neurons inside the human brain. A single network node may have multiple weighted inputs and the node only fires the output when the sum of the inputs reaches a predefined value. The activation function (which determines when the output 'fires' or is activated) can take one of two forms, a threshold or sigmoid function. In this study, we use a variant of the latter, a hyperbolic tangent function. Neural networks appear to show promise as an effective and efficient machine-learning algorithm (Freeman and Skapura, 1991) and may be especially suitable for the problem space described in this study.

As the problem to be solved is essentially one of classification, algorithms for both ANNs and NBCs can provide legitimate solutions, as both can be used for classification purposes. A key assumption of ANNs is that the input variables are linearly separable, although the input can be non-parametric. We adhere to this assumption by analysing the IPv6 addresses at a bitwise level. Bayesian methods are not nearly so sensitive to overlapping input as ANNs and as the data did not conform to a Gaussian distribution, both multinomial and Bernoulli methods were tested.

4 Experimental method

14,340 cases (IPv6 addresses) were collected for training purposes. A test dataset, comprised of a further 16,456 IPv6 addresses were also collected. The IPv6 address data collected for the study were extracted from public information sources, such as DNS AAAA records and passive DNS providers. The collected IPv6 addresses were classified into the three categories; SLAAC; SLAAC w/Privacy Extensions; and DHCPv6 or Manual Assignment. These classifications were calculated for each address in both datasets. The purpose of this is twofold, primarily the classification provides the machine learning algorithms with data by which to train, and secondly, provides the machine learning algorithms with a reference point during the testing phase.

The test system was a 2.13 GHz Intel Core 2 Duo MacBook Air with 4 GB of RAM. We used Python 2.7 programs to implement both algorithms. The testing used the Scikits-learn libraries (in particular the sklearn.naïve_bayes module). The Scikits-learn library includes a Python-based implementation of the naïve Bayes classification (NBC)

algorithms (multinomial and Bernoulli). For the ANN, we used a backpropagation algorithm, modified from (Graupe, 2007).

In order to conduct the training and testing, the dataset has to be transformed into usable inputs for the NBC and ANN. The sklearn's NBC algorithms expect multidimensional arrays as inputs. It was decided that the most effective method of training the algorithm would be to convert the data from the collapsed, 128 bit IPv6 address format into raw binary, since a standardised format for the data reduces margin for error (Table 1). Once converted to a 64 character binary represented string, the data was again converted into an acceptable input for the NBC. This was accomplished by creating an array of the data with each bit a separate item in the array. The same bit array was used to feed input into the ANN. The structure of the ANN was a three-layer network with 64 input nodes, eight nodes in the hidden layer and a three-node output layer (thus representing the scheme described in Section 2).

Table 1 Collapsed 128 bit IPv6 addresses converted to 64 bit exploded hexadecimal and binary IID representations

IPv6 address	*Exploded IID*	*Binary IID*
2607:f1c0:1000:60b8 :fb97:6127:6d65:10	fb97612 76d650010	1111101110010111011000010010011101101101011001010000000000010000
2001:288:2400:2::27	0000000 000000027	0000000000000000000000000000000000000000000000000000000000100111
2001:e41:db5e:edfe::1	0000000 000000001	0000000000000000000000000000000000000000000000000000000000000001
2a00:15f8:a000:5:1:14:0:8a1d	0001001 400008a1d	0000000000000001000000000010100000000000000010001010000011101
2607:f298:1:107::8af:684f	0000000 008af684f	0000000000000000000000000000000000000000100010101111011010000100111
2404:6800:4008:2::15	0000000 000000015	0000000000000000000000000000000000000000000000000000000000010101
2a02:100e:befc:8cd1: 0:203:11:1	0000020 300110001	0000000000000000000000010000000110000000000010001000000000000001
2a01:3b0:a00:5:a0b9:cdf3::	a0b9cdf 300000000	1010000010111001110011011111100110000000000000000000000000000000
2a00:1158:0:300:4086::1	4086000 000000001	0100000010000110000000000000000000000000000000000000000000000001

Note: Binary IID values are used as input to the machine learning algorithms.

5 Results and discussion

5.1 Testing a NBC

There are three main event models, which can be used when applying a NBC to a dataset for the purposes of classification, and these are dependent upon the distribution of the data. For the purposes of classifying discrete datasets, multinomial and Bernoulli distributions are often employed. When applying the Bernoulli NBC to the test dataset, it performed reasonably well, with a 94.94% success rate. The training and testing time

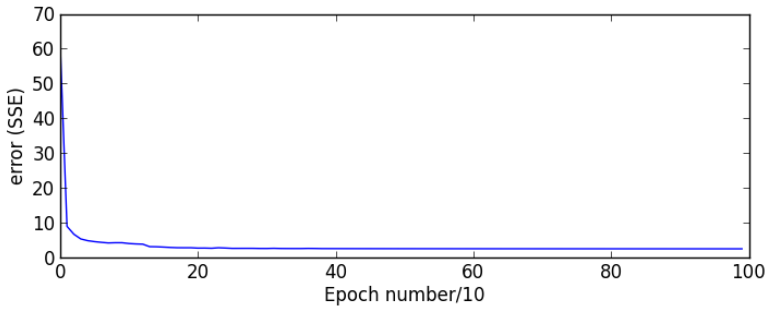
were 0.391 and 0.449 seconds, respectively. The multinomial NBC performed less satisfactorily with a success rate of 90.67% with training and testing times of 0.331 and 0.375 seconds, respectively.

5.2 Testing an ANN

The second machine learning algorithm to be tested was an ANN. The data described in the previous section were input to an ANN with a single hidden layer.

The performance of the fully-trained ANN was excellent, with a test time of 0.375 seconds and a success rate of 96.4% with a hidden layer consisting of eight nodes. The time required to train the network was, however, significant (29,398 seconds, or approximately eight hours). Figure 1 shows the error plot for the dataset used in these experiments. The error converges quickly (after 100 iterations) which suggests that it is not necessary to train with 14,340 cases. This would be beneficial given the eight hour training time. Note that once trained, however, the algorithm takes less than one second to test 16,456 addresses. Therefore, it can be concluded that the ANN algorithm is effective in solving this classification problem. A discussion of efficiency follows in the next section.

Figure 1 Error variation in ANN training for IPv6 address information (see online version for colours)



5.3 Factors affecting the performance of the ANN

Several factors influence the performance of the algorithm, including the number of cases analysed in training and the number of iterations.

Given the long training time and the well-behaved nature of the data, we evaluated the relationship between the number of training cases and the time taken for training as well as considering the effect of over-training. We found that over the range 100–400 training cases there was a linear relationship between the number of cases and the training time. This allowed us to use a least squares estimator to determine training time a priori. We were able to reduce the training set from 14,340 cases to 400 but with some loss of accuracy (93.25%, cf. Table 2). The key benefit was the reduction in training time to 776 seconds (13 minutes vs. 8 hours in the original dataset). Given the results shown in Figure 1, we considered the proposition that the network was being over-trained as the error term appeared to be constant past 100 iterations. Therefore, we

reduced the number of iterations from 1,000 to 100, with a concomitant reduction in training time (89 seconds, no loss of accuracy).

We found that adding more neurons to the hidden layer increased the training time of the system considerably and that the relationship between the number of hidden nodes and the training time to be linear (at least over the range of interest). Conversely, reducing the number of nodes in the hidden layer increased training speed, but at the expense of the error term. What was important here was that whilst increasing the size of the hidden layer increased training time (an undesirable outcome), it also allowed the system to come to an equilibrium with respect to the error more swiftly (a desirable outcome). Thus, there was a balance between the training time, the number of neurons in the hidden layer and the speed of convergence to an error minimum.

5.4 Comparing NBC and ANN

A training and testing regime consisting of 14,340 training and 16,456 testing samples, with 1,000 test iterations, was used. A comparison of results for the same dataset between the NBC and ANN algorithms is presented in Table 2. From this, it can be seen that the ANN algorithm was the most reliable classifier, with a false-negative rate of 3.6%. If training time is considered to be an important factor, then the Bernoulli NBC is the preferred algorithm, although the false-negative rate was 5.06%.

Table 2 Summary of algorithm performance for the ANN and NBCs used to classify IPv6 address information

Algorithm	*Training time (sec)*	*Test time (sec)*	*Percent classified correct*
ANN-backpropagation	29,398.108	0.375	96.40%
NBC-Bernoulli	0.391	0.449	94.94
NBC-Multinomial	0.331	0.376	90.67

6 Conclusions

We tested three machine-learning algorithms to determine whether they could aid in analysing and classifying previously collected IPv6 addresses. The aim was to determine whether either an NBC or ANN would aid in reliably determining the host portion of an IPv6 address. All algorithms performed well with the test dataset, and were reasonably accurate, with the ANN performing the better of the three. The results demonstrate that classifying IID generation schemes is not only possible, but can be performed reliably and quickly. This has significant implications for the privacy of network traffic, devices and ultimately the end users of those devices. Currently, IDS and IPS systems use IP addresses for a variety of purposes, including geo-location. In IPv6 the lack of NAT, and focus on unique public addresses makes an IID an important characteristic for analysis. The next step will be to investigate the minimum number of samples required to achieve optimal accuracy and speed of each algorithm. Further research will determine whether the comparatively long learning time for ANN can be reduced through the use of general purpose graphics programming units (GPGPUs), and will include applying these techniques to honeypots, log analysis and intrusion detection systems. The research thus far has shown promise in the application of machine learning to this problem.

References

Carpene, C. and Woodward, A. (2012) 'Exposing potential privacy issues with IPV6 address construction', in Valli, C. and Woodward, A. (Eds.): *The 10th Australian Information Security Management Conference*, Edith Cowan University, Perth, Western Australia, pp.44–50.

Deering, S. and Hinden, R. (1998) *Internet Protocol, Version 6 (IPv6) Specification*, December, RFC 2460 (Draft Standard), Updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946, 7045, 7112, Internet Engineering Task Force, IETF [online] http://www.ietf.org/rfc/rfc2460.txt.

Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C. and Carney, M. (2003) *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, RFC 3315, July.

Freeman, J.A. and Skapura, D.M. (1991) *Neural Networks: Algorithms, Applications, and Programming Techniques (Computation and Neural Systems Series)*.

Graupe, D. (2007) *Principles of Artificial Neural Networks*, World Scientific Publishing Co. Pte. Ltd., University of lllinois, Chicago, USA.

Haykin, S.S. (2009) *Neural Networks and Learning Machines*, Vol. 3, Prentice Hall, New York.

Johnstone, M.N. and Woodward, A. (2012) 'Towards effective algorithms for intelligent defense systems', in *Cyberspace Safety and Security*, pp.498–508, Springer.

Narten, T., Draves, R. and Krishnan, S. (2007a) *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, September, RFC 4941 (Draft Standard), Internet Engineering Task Force, IETF [online] http://www.ietf.org/rfc/rfc4941.txt.

Narten, T., Thomson, S. and Jinmei, T. (2007b) *IPv6 Stateless Address Autoconfiguration*, September, RFC 4862 (Draft Standard), Internet Engineering Task Force, IETF, [online] http://www.ietf.org/rfc/rfc4862.txt.

Zhang, H. (2004) 'The optimality of naive Bayes', *A. A.*, Vol. 1, No. 2, p.3.