Genetic & Evolutionary Feature Selection for Author Identification of HTML Associated with Malware

Henry C. Williams, Joi N. Carter, Willie L. Campbell, Kaushik Roy, and Gerry V. Dozier

***Abstract*—Malicious software, also known as malware, is a huge problem that costs consumers billions of dollars each year. To solve this problem, a significant amount of research has been dedicated towards detecting malware. In this paper, we introduce a genetic and evolutionary feature selection technique for the identification of HTML code associated with malware. We believe that there may be an association between malware and the HTML code that it is embedded in. Our results show that this technique outperforms previous techniques in terms of recognition accuracy as well as the total number of features needed for recognition.**

***Index Terms*—Authorship classification, biometrics, feature extraction, genetic and evolutionary computation (GEC), malware.**

I. Introduction

It is estimated that in the US alone, malicious software (malware) costs consumers billions of dollars each year [1]. In an effort to reduce this cost, a significant amount of research has been dedicated to detecting malware [2]-[5]. The primary goal of these detection techniques involves determining if a software sample has malicious intent or not. This process is typically done by analyzing the behavior or structure of software [2]-[5].

While malware detection has improved significantly, the application of author identification [18] would be effective and complementary with existing approaches. This research should not be mistaken as identity verification [14]. The intention is to identify authorship of source code, not to verify the identity of that author. The goal of this research is to improve author identification techniques to classify an HTML sample as being associated with malware or not.

The remainder of this paper is as follows. Section II provides a background on behavioral biometrics [9], uni-gram based feature extraction [13], and genetic and evolutionary feature selection [20]. Section III presents our methodology and Section IV describes our experiments. Our results are provided in Section V, a discussion is provided in Section VI, and our conclusions and future work are presented in Section VII.

Manuscript received October 30, 2013; revised January 5, 2014. This work was supported in part by the National Science Foundation (NSF), Army Research Laboratories (ARL), Lockheed Martin, and the Science & Technology Center: Bio/Computational Evolution in Action Consortium (BEACON).

H. C. Williams, J. N. Carter, W. L. Campbell, K. Roy, and G. V. Dozier are with the Computer Science Department, North Carolina Agricultural and Technical State University, Greensboro, NC 27411 USA (e-mail: hcwillia@aggies.ncat.edu, jncarte1@aggies.ncat.edu, wlcampbe@aggies.ncat.edu, kroy@ncat.edu, gvdozier@ncat.edu).

II. Background

A. Behavioral Biometrics

Biometrics is the area of research devoted to identifying individuals using physiological or behavioral characteristics [9]. A typical biometric system consists of 4 modules: a sensor, a feature extraction module, a matching/decision module, and a database module. When an individual is authenticated with a biometric system, the sensor module takes a sample of the individual. Depending on the biometric modality being used, the sample could be in the form of an image, a recording, etc. The feature extraction module then extracts discriminant features from the sample. These features are usually represented in a feature vector. The matching module compares the newly acquired feature vector to feature vectors previously enrolled in the database.

Behavioral biometrics is a subset of biometrics and includes modalities such as signature, keystroke, voice, gait, etc [10]. Unlike physiological biometrics, behavioral biometrics deals with how an individual acts [11]. The collection of behavioral biometrics is generally cost effective [12]. This makes it ideal for use in identifying authors of HTML code associated with malware.

B. Author Identification

Authorship analysis is the process of identifying the author of an anonymous text, or text whose authorship is not clear [13]. The most extensive use of authorship analysis falls in literature with studies such as the Federalists papers and Shakespeare's work, but has made a leap into the modern era with source code [16]. Since source code has the potential to be unique based on its' author, it becomes appropriate to try and inspect the elements that make it unique from author to author. As described by [17], the key to identifying the author of harmful code is by evaluating the appropriate body of code and identifying appropriate features for comparison. Features such as the use of white space, indentation, variable names, and the levels of readability all contribute to the authenticity of an author [18].

This paper applies the concept of author identification, a subset of authorship analysis, to HTML code associated with malware samples. The process of author identification involves comparing the features of a sample to another sample [15], [20]. Previous research in this area has resulted in a variety of ways to extract and evaluate features from a particular sample. In [15], the feature extraction technique extracted 170 style-based features including, but not limited to, the number of blank lines, the average sentence length, and the total number of function words. When dealing with source code there are less stylometric features and more lexical features to focus on [21]. In this case an N-Gram

based feature extraction technique may be more suitable since this approach is able to capture a trace of style, lexical information, punctuation and capitalization [19].

C. Genetic and Evoluationary Feature Selection

Genetic and Evolutionary Feature Selection (GEFeS) and Genetic and Evolutionary Feature Selection and Weighting (GEFeWS) [20] are feature selection techniques used to evolve a near-optimal/optimal subset of features in order to maximize accuracy and minimize computational complexity (as measured in feature comparisons for matching). In order to perform feature selection, a genetic algorithm is used to evolve a set of feature weights. A feature weight is a real value between zero and one. In the set of feature weights, there is a weight that corresponds to each value in the feature vector. A feature mask (FM) is then created from this set of feature weights.

To create a feature mask (FM), a masking threshold is used. A masking threshold is a value used to determine if a feature will contribute in the matching process. In GEFeS if the feature weight is above the masking threshold, the FM value is set to 1.0. Otherwise, the FM value is set to 0.0 and the corresponding feature will not contribute in the matching process. Similarly for GEFeWS, if a feature's weight is below the masking threshold the FM value is set to 0.0; however if the feature's weight is above the masking threshold the weight remains as is in the FM.

III. Feature Extraction

A. Uni-Gram Style Feature Extraction

Research has shown, that the frequency of individual characters within a document can be one of the most effective identifiers for Author Identification [13].

The first step in character uni-gram feature extraction is to count the frequency of each character and the total number of characters in a sample. Once these frequencies have been counted, each character frequency is then divided by the total number of characters. A feature vector is then created for each sample in the dataset. An example of the types of characters that can be used in uni-gram FE is shown in Fig. 1.

(space)	!	“	#	$	%	&	‘	(	)
*	+	,	-	.	/	0	1	2	3
4	5	6	7	8	9	:	;	<	=
>	?	@	A	B	C	D	E	F	G
H	I	J	K	L	M	N	O	P	Q
R	S	T	U	V	W	X	Y	Z	[
\	]	^	_	`	A	b	C	d	e
f	g	H	I	j	K	l	M	n	o
p	q	R	S	t	U	v	W	x	y
z	{	|	}	~					

Fig. 1. The subset of Unicode characters used in our experiments.

B. Stylometric and Structural Features

Stylometric and structural feature extractors can calculate style-based features that may include, but are not limited to, word length frequency distribution, the total number and frequency distribution of function words, total number of words, total number of distinct words, and total number of characters [15], [20].

One such feature extractor is described by O. de Vel *et al.* in [15]. The described feature extractor produces feature vectors with a total of 170 style marker attributes and 21 structural attributes. A list of the stylometric features proposed by O. de Vel *et al.* is shown in Fig. 2. Their proposed structural features are specific to email content such as quoted text position when replying, HTML tag frequency, and greeting/salutation acknowledgments.

Stylometric Features
Number of blank lines/total number of lines
Average sentence length
Average word length(number of characters)
Vocabulary richness i.e., V/M
Total number of function words/M
Function words (122)
Total number of short words/M
Count of hapax legomena/M
Count of hapax legomena/V
Number of characters in words/C
Number of alphabetic characters in words/C
Number of upper-case chars/C
Number of digit characters in words/C
Number of white space characters/C
Number of space characters/C
Number of space characters/white space characters
Number of tab spaces/C
Number of tabs spaces/number of white spaces
Number of punctuations/C
Word length frequency distribution/M (30)

Fig. 2. Stylometric features proposed by O. de Vel *et al.* [15]

IV. Experiments

For our experiments, we use a dataset of 116 HTML samples. This dataset consists of 58 HTML samples associated with known malware, and 58 samples from known legitimate news websites. We then divide each of the samples into 3 equal instances and perform feature extraction on each of the sections, resulting with a set of 348 feature vectors. These feature vectors are stored in a file together each on a separate line. Preceding each feature vector is a value we call an ID. Each of the feature vectors is labeled with two separate IDs, one signifies if the sample is associated with malware and the other is a unique ID given to each full size HTML sample. When a full size sample is divided into thirds each third retains the unique ID of the original full size sample.

These feature vectors are then divided into a probe set and gallery set. The probe set represents the inputs into the system. The gallery set represents the database of the system. The first instance of each sample is put in the probe set. The last two instances are put in the gallery set. Each instance in the probe set is then compared to all of the instances in the gallery set.

To compare two instances, the Manhattan distance of the feature vectors are calculated. The Manhattan distance is calculated by taking the distance between two vectors. In this instance we take the distance between each instance in

a probe set and each instance in a gallery set. The formula for weighted Manhattan distance is shown in Equation (1), where w is the set of weights, v_1 is one feature vector and v_2 is the other. While performing feature selection with GEFeS each weight w_i is either a one or a zero.

$$d_m = \Sigma\, w_{i*} \mid v_{1,i} - v_{2,i} \mid \quad (1)$$

After a probe instance has been compared to each gallery instance, the gallery instance that is closest to the probe is considered its mate. If the gallery instance came from a different sample than the probe instance, an error is recorded.

Separate sets of experiments were performed to target our two specific goals. One set of experiments aims to match the IDs signifying the feature vectors association with malware. If a feature vector in the probe set has an ID signifying that it is associated, but matches a feature vector in the gallery set that is not associated with malware an error is recorded and vice versa. We refer to these experiments as malware association experiments. Our other set of experiments aims to match the ID given to each full length HTML sample. When a feature vector in the probe set matches a feature vector in the gallery set, with a different ID than its own, an error is recorded. We refer to these experiments as source identification experiments.

V. Results

These results were obtained by using X-TOOLS to perform feature selection. X-TOOLSS is a suite of genetic and evolutionary computations (GECs) that are used to find the optimal or near optimal solution. X-TOOLSS uses GECs to evolve a population of candidate solutions (CS) and assign them a fitness. In this research, GEFeS uses a Steady State Genetic Algorithm (SSGA) and an Estimation of Distribution Algorithm (EDA) to evolve a feature mask (FM) and select the most significant features. A steady-state genetic algorithm (part of X-TOOLSS [8]) is used with a population size of 20, uniform crossover, Gaussian mutation with a 20% mutation range, and binary tournament selection. An EDA is used with a population size of 20 and 5 elites. Feature selection was performed 30 times. The two GECs were limited to 500 function evaluations for each run. We then average the accuracy and percentage of features used for each run.

The results for these experiment are provided in Table I-IV. The top of each table shows the feature extraction technique used and the baseline accuracy that was obtained with 100% of the features from that dataset. The first column shows the algorithm that was used in the experiment. The second column in each table describes the percentage of accuracy obtained when the best features from the validation set were applied to the test set. This column is formatted as best accuracy followed by average accuracy in parenthesis. The fourth column describes the percentage of features that remained after training. The final column describes the Equivalency Class (EC) of each algorithm as it applies to the % of features kept by that algorithm.

In Table I we show the results on an source identification experiment using the O. de Vel style feature extractor. These results show significant improvement over the baseline accuracies while using less than 50.00% of the total features. The best accuracy was achieved by the GEFeWS – EDA algorithm. This gave us accuracies of 42.46% accuracy while using 49.41% of the features. This algorithm however was in the second EC based on percentage of features. The lowest percentage of features used by an algorithm was shown using GEFeS – SSGA with only 37.90%.

In Table II we show the results on a source identification experiment using the uni-gram style feature extractor. These results show an improvement over the baseline accuracies while using less than 63.00% of the total features. The best accuracy was achieved by the GEFeS – EDA algorithm with 51.01% accuracy and only 62.98% of the features. However, GEFeS – SSGA was in a class of its own based on feature reduction with only 46.91% of the features needed.

TABLE I: Source Identification Results

Algorithm	**O. de Vel Style Feature Extractor**		
	Test Set Baseline: 13.04%		
	Accuracy %	***Average % Features***	***EC of % features***
GEFeS – SSGA	52.17 (41.45)%	37.90%	1
GEFeS – EDA	56.52 (42.17)%	42.80%	1
GEFeWS – SSGA	56.52 (41.88)%	41.45%	1
GEFeWS – EDA	60.87 (42.46)%	49.41%	2

Source: identification Results using the O. de Vel style feature extractor

TABLE II: Source Identification Results

Algorithm	**Uni-Gram Style Feature Extractor**		
	Test Set Baseline: 39.13%		
	Accuracy %	***Average % Features***	***EC of % features***
GEFeS – SSGA	60.87 (47.54)%	46.91%	1
GEFeS – EDA	65.22 (51.01)%	62.98%	2
GEFeWS – SSGA	60.87 (46.09)%	56.11%	2
GEFeWS – EDA	65.22 (49.13)%	60.84%	2

Source: identification results using the uni-gram style feature extractor

TABLE III: Malware Association Results

Algorithm	**O. de Vel Style Feature Extractor**		
	Test Set Baseline: 65.22%		
	Accuracy %	***Average % Features***	***EC of % features***
GEFeS – SSGA	82.61 (73.19)%	38.19%	1
GEFeS – EDA	91.30 (74.64)%	35.14%	1
GEFeWS – SSGA	86.96 (74.93)%	42.96%	1
GEFeWS – EDA	86.96 (75.80)%	41.35%	1

Malware association results using the O. de Vel style feature extractor

In Table III we show the results on a malware association experiment using the O. de Vel style feature extractor.

These results show an improvement over the baseline accuracies while using less than 42.00% of the total features. The best accuracy was achieved by the GEFeWS – EDA algorithm. This gave us accuracies of 75.80% accuracy while using only 41.35% of the features. All four algorithms were in the same EC based on percentage of features, but it can be seen that GEFeS – EDA has the lowest average using only 35.14%.

In Table IV we show the results on an malware association experiment using the uni-gram style feature extractor. These results show a slight improvement over the baseline accuracies while using less that 57.00% of the total features. The best accuracy was achieved by the GEFeWS – SSGA algorithm. This gave us accuracies of 74.20% accuracy while using 49.37% of the features. This algorithm is in the same EC, based on percentage of features used, as GEFeS – SSGA which used the fewest features on average.

In all cases we show that these feature extractors work well on classifying the source of HTML code and its association with malware. We also show that the four algorithms we applied can improve accuracy as well as reduce the number of features needed for classification.

TABLE IV: MALWARE ASSOCIATION RESULTS

Algorithm	**Uni-Gram Style Feature Extractor**		
	Test Set Baseline: 69.57%		
	Accuracy %	***Average % Features***	***EC of % features***
GEFeS – SSGA	91.30 (70.43)%	39.23%	1
GEFeS – EDA	86.96 (73.04)%	50.98%	2
GEFeWS – SSGA	91.30 (74.20)%	49.37%	1
GEFeWS – EDA	86.96 (73.04)%	56.88%	2

Malware association results using the O. de Vel style feature extractor

VI. DISCUSSION

There are tools available to alert consumers of potentially malicious websites. The following commercial malware detection tools classify domains using a variety of characteristics. Web of Trust (WOT) [22] works by aggregating the opinions of a global community of millions of users to form a reputation score. McAfee Threat Center [23] uses data obtained by webs spam tests, download tests, and IP address reputations. Cisco Lookup [24] uses data from location, network owners, IP address, and email reputation. Google Safe Browsing [25], like our research, tests a number of individual pages on a given domain to determine if malicious content is present. In addition they test if this malicious content is being downloaded and installed without user consent or acts as an intermediary to infect other sites. McAfee Site Advisor [26] uses aggregate data from a number of users as well as data about downloads and links to malicious content. AVG Threat Lab [27] also aggregates data from a community of users. This data includes experience, tags, popularity, timeline of previous malicious content, linked websites, and a map of where, in the world, a user was when they detected malicious content. URLVoid [28] collects data about domain connection including HTTP header size, download data size, and transfer speed. URLVoid also collects data about IP address reputation, global traffic, social activity related to the domain, and checks a number of online blacklists to see if the domain was blacklisted.

We used these tools to see how accurately the classified the instances of our test set. Their accuracy can be compared to our association experiments since the goal of the test is to classify malicious intent. The results of this are shown in Table V.

TABLE V: COMMERCIAL TOOL CLASSIFICATION RESULTS

WOT	**McAfee Threat Center**	**Cisco Lookup**	**Google Safe Browsing**	**McAfee Site Advisor**	**AVG Threat Lab**	**URL Void**
69.57%	91.30%	91.30%	65.22%	47.83%	52.17 %	95.65%

Test set classification using commercial tools

VII. CONCLUSIONS AND FUTURE WORK

Our results show that GEFeS, in all cases, had better performance in terms of reducing features; however in all but one instance GEFeWS had a higher average accuracy. Our results also show that before feature selection a uni-gram based feature extractor is better at identifying the authors of HTML code associated with malware than the O. de Vel style feature extractor as well as identifying if the HTML sample is associated with malware. After feature selection both algorithms prove to classify at nearly the same accuracy. This proves that both extractors have features contributing to classification and features that hinder the process. Our results also show that our accuracy on the association experiments can compete with several commercial solutions. URLVoid [28] had the highest accuracy at 95.56% and we came very close to that. We did, however, tie in accuracy with McAfee Threat Center [23] and Cisco Lookup [24] while beating the other commercial tools we tested [22], [25]-[27]. An important distinction we would like to make is that our algorithm does not rely on a community of users and it can search each individual HTML page on a domain. This may be crucial if a respected domain has been infected on a single page. The only other tool we tested that can search individual HTML pages is Google Safe Browsing [25]. In the future we would like to combine the two feature extractors in hopes of increasing accuracy. We would also like to increase the size of our dataset.

ACKNOWLEDGMENT

The authors would like to thank the ARL and the NSF for their support of this research.

REFERENCES

[1] Consumer Reports. (June 2011). [Online]. Available: http://www.consumerreports.org/cro/magazine-archive/2011/june/electronics-computers/state-of-the-net/online-exposure/index.htm

[2] C. Willems, T. Holz, and F. Freiling, "Toward automated dynamic malware analysis using CWSandbox," *Security and Privicy*, IEEE, vol. 5, no. 2, pp. 32-39, March-April 2007.

[3] U. Bayer, C. Kruegel, and E. Kirda, "TTAnalyze: A tool for analyzin malware," in *Proc. 15th Annual Conference on the European Institute for Computer Antivirus Research (EICAR)*, 2006.

[4] D. Brumley, I. Jager, T. Avgerinos, and E. Schwartz, "BAP: A binary analysis platform," in *Proc. 2011 Conference on Computer Aided Verication*, G. Gopalakrishnan, S. Qadeer, Eds., vol. 6806, pp. 463-469. Springer, Heidelberg, 2001.
[5] Computer Economics. (2005). Malware Report: The impact of malicious code attacks. [Online]. Available: http://www.computereconomics.com/article.cfm?id=1090
[6] Symantec. Internet Security Threat Report, Trends from Jan. 2005 to Jun. 2005. (September 2005). [Online]. 3. Available: http://www.symantec.com/enterprise/threatreport/index.jsp
[7] X-TOOLSS. [Online]. Available: http://nxt.ncat.edu/
[8] A. K. Jain and A. Ross, *Handbook of Biometrics*, pp. 1-22, 2008.
[9] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics*, vol. 14, no. 1, January 2004.
[10] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Communications of the ACM*, vol. 43, no. 2, pp. 91-98, February 2000.
[11] R. V. Yampolskiy, *Computer Security, from Password to Behavioural Biometrics*, New Academic Publishing, London, UK, 2008.
[12] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan,"N-gram-based detection of new malicious code," in *Proc. the 28th Annual International Computer Software and Applications Conference - Workshops and Fast Abstracts*, vol. 2, IEEE Computer Society, Washington, DC, USA, 2002, pp. 41-42.
[13] R. Forsyth, "Short substrings as document discriminators," in *Proc. ACH-ALLC Conferenc*e, 1997.
[14] N. McCombe, "Methods of author identification," Final Year Project, CSLL, May 2002.
[15] O. de Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *SIGMOD Record*, vol. 30, no. 4, pp. 55-64, 2001.
[16] H. Spafford and S. Weeber, "Software forensics: Can we track code to its authors?," *Computers & Security*, vol. 12, no. 6, pp. 585-595, 1993.
[17] P. Sallis, S. MacDonell, G. MacLennan, A. Gray, and R. Kilgour, "Identified: Software authorship analysis with case-based reasoning," in *Proc. the Addendum Session of the Fourth International Conference on Neural Information Processing (ICONIP'97)*, Dunedin, New Zealand, pp. 53 – 56, 1998.
[18] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for Information Science and Technology*, Dec. 16, 2008.
[19] A. Kumar and D. Zhang, "Biometric Recognition using Feature Selection and Combination," in *Proc. AVBPA*, New York, USA, July 2005, pp. 813-822.
[20] A. Alford, J. Adams, J. Shelton, K. Bryant, J. C. Kelly, and G. Dozier, "Analyzing the cross-generalization ability of a hybrid genetic & evolutionary application for multibiometric feature weighting and selection," in *Proc. the 2012 Genetic and Evolutionary Computation Conference*, 2012.
[21] J. Kothari *et al.*, "A probabilistic approach to source code authorship identification," in *Proc. IEEE Fourth International Conference* on *Information Technology*, 2007.
[22] Web of Trust. [Online]. Available: http://www.mywot.com/
[23] McAfee Threat Center. [Online]. Available: http://www.mcafee.com/threat-intelligence/domain/popular.aspx/
[24] Cisco Lookup. [Online]. Available: http://www.senderbase.org/lookup
[25] Google Safe Browsing. [Online]. Available: https://developers.google.com/safe-browsing/
[26] McAfee Site Advisor. [Online]. Available: http://www.siteadvisor.com/
[27] AVG Threat Labs. [Online]. Available: http://www.avgthreatlabs.com/website-safety-reports/
[28] URLVoid. [Online]. Available: http://www.urlvoid.com/

Henry C. Williams was born in Maryland on December 15, 1986, who received an associates of applied science in network technologies from Guilford Technical Community College in 2004. Henry is now completing his bachelor's degree in computer science at North Carolina Agricultural and Technical State University (NCAT), expected to graduate May 2014.

He has worked as a supplemental instructor at NCAT and is currently working in the CASIS research group at NCAT in Greensboro, NC. He held internships at Carnegie Mellon University, as a research assistant in Robotics, and at Lawrence Livermore National Labs, as a Cyber defender. He is currently researching malware classification with computer learning.

Mr. Williams is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM).

Joi N. Carter was born in Augsburg, Germany on January 13, 1992, and moved to North Carolina for college. She is a senior working on a Bachelor of Science degree in computer science from North Carolina A&T State University in Greensboro, NC. She is scheduled to obtain this degree in May 2014.

During the summer of 2012 she interned with the EMC Corporation, in Frankiln Massachusets, as a software engineer and application developer where she developed software to support systems testing. In the summer of 2011 she interned at the Northrop Grumman Corporation in Linthieum, MD, as a cross platform intern. She spent this tie designing interactive applications for the Microsoft Surface. Her research focus was blog author identification for the the Center for Advanced Studies in Identity Sciences at NC A&T SU.

Willie L. Campbell was born in Fort Sill, Oklahoma on July 12, 1988, and moved to North Carolina just after birth. He received a Bachelor of Science degree in computer science from North Carolina A&T State University in Greensboro, NC in 2011 and is currently pursuing a Master of Science degree in computer science from North Carolina A&T State University in Greensboro, NC.

During the summer of 2007, upon completion of his first semester of school, he interned with the Department of Transportation Highway Division in Nashville, NC as an engineer assistant where he surveyed all the road in the three surrounding counties as well as overseeing various traffic project being done. The following summer of 2008, he interned with the Department of Transportation IT Division in Raleigh, NC as a Business and Technical Analyst where he worked with a project manager learning the Software Development Lifecycle, and worked with different members of the project team on various documents used in the project. The summer of 2009 he interned with SAS Institute in Cary, NC as a student tester where he used automation software to create test suites to test an internal application. Currently he is doing research for the Center for Advanced Studies in Identity Sciences at NC A&T SU in the area of Evolving Feature Extractors for Detecting Malware Associated with HTML Code.

Kaushik Roy received his PhD from Concordia University, Montreal, QC, Canada in 2011 in Computer Science. He also completed his MS degree in computer science from the Concordia University in 2006 and B.Sc. degree in computer science from University of Rajshahi, Bangladesh in 2001. Kaushik Roy is currently an assistant professor at the Department of Computer Science, and Assistant Director of the Center for Advanced Studies in Identity Sciences (CASIS), North Carolina A&T State University, USA. Previously, he worked as a postdoctoral fellow in the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada during 2011-2012. He also taught at Rajshahi University of Engineering and Technology (RUET) as a lecturer of the Department of Computer Science and Engineering during 2001-2004. He is also the recipient of several fellowships and awards including the prestigious NSERC Visiting Fellowship, FQRNT B3 (Postdoctoral), NSERC (Doctoral) and FQRNT B2 (Doctoral). His research interests include Biometrics, cyber identity, game theory, information fusion, computer vision, machine learning, and pattern recognition. He has published 1 book, 2 book chapters, 11 journal articles and 38 conference articles.

Gerry Vernon Dozier is a professor and the chair of the Computer Science Department at North Carolina A&T State University. He is the director of the Center for Advanced Studies in Identity Sciences (CASIS), as well as the PI for the Center for Cyber Defense (recognized by the National Security Agency and the Department of Homeland Security as a Center for Academic Excellence in Information Assurance Education). During Gerry's tenure as chair, the department has seen an increase in extramural funding and research publications as well as the establishment of a Ph.D. program. He has also lead in the development of an undergraduate research program where approximately 20% of the undergraduate students are active participants in funded research projects. Under Gerry's leadership, the NSF Alliance for the Advancement of African American Researchers in Computing (A4RC, www.a4rc.org) experienced a threefold increase (from 6 to 20) in the number of participating universities. A4RC was effective in increasing the number of African-American recipients of advanced degrees in Computer Science.

Gerry has published over 130 conference and journal publications. He has served as an Associate Editor of the IEEE Transactions on Evolutionary Computation and the International Journal of Automation & Soft Computing. Gerry is also a member of the Editorial Board for the International Journal of Intelligent Computing & Cybernetics. His research interests include: Artificial & Computational Intelligence, Genetic, Evolutionary, and Neural Computing, Biometrics, Identity Sciences, Cyber Identity, Distributed Constraint Reasoning, Artificial Immune Systems, Machine Learning and Network Intrusion Detection. Gerry earned his Ph.D. from North Carolina State University.