Comparative performance analysis of classification algorithms for intrusion detection system

Mohammed Anbar*, Rosni Abdullah*†, Iznan H. Hasbullah*, Yung-Wey Chong*and Omar E. Elejla†
*National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Penang, Malaysia
Email: {anbar,rosni,iznan,chong}@nav6.usm.my
†School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia
Email: oeoe14_com063@student.usm.my

***Abstract*—The ability of an intrusion detection system (IDS) to accurately detect potential attacks is crucial in protecting network resources and data from the attack's destructive effects. Among many techniques available for incorporation into IDS to improve its accuracy, classification algorithms have been demonstrated to produce impressive and efficient results in detecting IPv4-based attacks but have not yet been investigated in IPv6-based attacks. This paper aims to present the result of a comparative analysis on the performance of three classifier algorithms, namely, decision tree, random forest, and k-nearest neighbor (*k*-NN), to detect an IPv6-based attack, specifically ICMPv6-based DoS flooding. The experimental results showed that there is no single best algorithm that outperforms others in all measured metrics. *k*-NN has the lowest false-positive outcome while RF has the lowest false-negative (missed attacks) percentage.**

Keywords: Decision tree, Intrusion detection system, k-nearest neighbor, Random forest,IPv6-based attack

I. Introduction

Computer networks have become an important dimension of modern organizations. Thus, ensuring that networks run at peak performance is treated as one of the key objectives of these organizations. To achieve this goal, an intrusion detection system (IDS) is typically deployed at the default gateway of the target network for the administrator to obtain a complete bird's-eye view of the entire network to monitor the overall network traffic activities and detect abnormal activities. IDSs detection mechanisms are classified into two main categories, namely, signature and anomaly-based detection mechanism. In a signature-based detection mechanism, IDS checks the incoming packet payload and system log files against a predefined signature stored in the IDS database. An IDS triggers an alert when a match is found. In an anomaly-based detection mechanism, a profiling program is created for normal network behavior to be used as a baseline. Deviation from this baseline is treated as an anomaly or a possible intrusion [1].

The ability of IDS to accurately detect potential attacks is crucial in protecting network resources and data from the attack's destructive effects. Many techniques are available for incorporation into IDS to improve its accuracy. One of the techniques that shows impressive and efficient results in detecting attacks is the classification algorithm.

The performance of classification algorithm is typically benchmarked against a common dataset. The KDD Cup 99 [2], NSL-KDD [3], and Kyoto 2006+ [4] datasets are some of the most common datasets used by researchers and security professionals to evaluate the performance of IDS against different types of IPv4-based attacks.

Since the advent of IP version 6 (IPv6) to supplant the earlier version of the protocol, IP version 4 (IPv4), the engine and algorithm of IDSs have to be updated or extended to detect IPv6-based attacks. The IPv6 packet structure is completely different from IPv4 [5], [6]. Accordingly, the dataset characteristics that represent IPv6-based attacks are different from those of any IPv4-based dataset. Thus, a dataset that properly represents different types of IPv6-based attacks is required to benchmark the capability and performance of particular IDSs to detect IPv6-based attacks.

In this paper, the effectiveness of the classifier algorithms on IDS is evaluated using a simulated dataset that consists of IPv6-based attacks. The rest of the paper is organized as follows: Section 2 presents a review of a decision tree (DT), k-nearest neighbor (k-NN) and random forest (RF) found in the literature. Section 3 discusses the experimental results. Section 4 concludes the present work.

II. Classification algorithms

A classification algorithm is a procedure for selecting a hypothesis from a set of alternatives that best fit a set of observations. Classifier algorithms are widely utilized in data mining. These algorithms have the ability to generate a solid prediction model based on a set of features during the training phase. These features belong to certain labelled classes. The generated prediction model is used later to predict new classes. In this section, a brief overview of DT, RF, and *k*-NN classification algorithms is presented. The selection of a classification algorithm depends on the requirements and the nature of the classification required.

A. Decision tree algorithm

DT is among the well-known machine learning techniques. A DT is a tree-like graph or model, or rather an inverted tree because it has its root at the top and it grows downwards [7]. This manner of representing the data makes DT meaningful and easy to interpret compared with other approaches. The goal is to create a classification model that predicts the value of a target attribute (usually called class or label) based on several input attributes of the example. DT has three basic

elements: (1) a decision node specifying a test attribute, (2) an edge or a branch corresponding to one of the possible attribute values (i.e., one of the test attribute outcomes), and (3) a leaf, also called an answer node, containing the class to which the object belongs. DT has two major phases: tree building and classification. The process of building a tree is based on a given training set. It involves selecting the "appropriate" test attribute for each decision node and defining the class label for each leaf. In the classification phase, classifying a new instance begins at the root of the decision tree, and then the attribute specified by this node is tested. The result of the test enables moving down the tree branch relative to the attribute value of the given instance. This recursive process is repeated until a leaf is encountered. The instance is then classified into the same class as the one characterizing the reached leaf. In general, the recursion stops when all examples or instances have the same label value, which is called pure subset. Recursion may also stop if most of the examples are of the same label value. However, other conditions can terminate the recursion process, such as when there are less than a certain number of instances or examples in the current subtree, when no attribute reaches a certain threshold, or when the maximal depth is reached. Pruning is a technique in which leaf nodes that do not add to the discriminative power of the decision tree are eliminated. This process is conducted to convert an over-specific or over-fitted tree to a more general form to enhance its predictive power in dealing with unseen datasets. Pruning comes in two types: pre-pruning and post-pruning. Pre-pruning is performed at the same time as the tree creation process, whereas post-pruning is conducted after the completion of the tree creation process.

B. Random forest algorithm

Random forests or random decision forests are an ensemble learning method employing a multitude of decision trees during training period. Each tree depends on the values of an independently sampled random vector with the same distribution for all trees in the forest. Once the trees are generated, the most popular class is voted. The definition of random forest according to Brieman is as follows: A random forest is a classifier consisting of a collection of tree structured classifiers h(x,θk), k = 1, ..., where θk are independent identically distributed random vectors, and each tree casts a unit vote for the most popular class at input x. The strength of individual trees in the forest and the correlation between them determines the generalization error of a forest of tree classifiers [8]. Random forests are a substantial modification of bagging technique that takes the average of a large collection of de-correlated trees. The performance of a random forest in many problems is comparable with boasting, but it is simpler to train and tune, thus making it very popular [9]. Although the mechanism of a random forest algorithm seems simple, it is still difficult to analyze, and many of its aspects remain unknown. Breiman (2000, 2004) attempted to investigate the factor behind the consistency of random forests, and Lin and Jeon (2006) managed to establish a connection between random forests and adaptive nearest neighbor methods [10].

C. K-nearest neighbor algorithm

The *k*-NN algorithm is a type of similarity-based classifier, in which prediction is based on the choice of a given test example and the training examples are strikingly similar to the target [11]–[13]. The training examples are described by n attributes, and each example represents a point in an n-dimensional space. Thus, all of the training examples are stored in an n-dimensional pattern space. During an encounter with an unknown example, *k*-NN searches the pattern space for the k training examples that are closest to the unknown example. These k training examples are the k "nearest neighbors" of the unknown example. "Closeness" is defined in terms of a distance metric, such as the Euclidean distance. The *k*-NN algorithm is among the simplest of all machine learning algorithms. An example is classified by a majority vote of its neighbors, and the example is assigned to the class most common among its k-nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the example is simply assigned to the class of its nearest neighbor. The same method is used for regression by simply assigning the label value for the example to be the average of the values of its k nearest neighbors. It is useful to weight the contributions of the neighbors in such a way that the nearer neighbors have more weighted to the average than to the one further away. This approach is the one adopted by several weighted *k*-NN methods. The neighbors are taken from a set of examples for which the correct classification (or the value of the label in the case of regression) is known. This training set is considered the one for the algorithm, although no explicit training step is required. The basic *k*-NN algorithm is composed of two steps. First, the k training examples that are closest to the unknown example are found. Secondly, the most commonly occurring classification for these k examples (or the average of these *k* label values in the case of regression) is taken.

III. Experimental result

This section aims to evaluate the effectiveness of the classifier algorithms in the performance of IDS. Section III-A describes the characteristic of the simulated dataset. The performance metrics have been highlighted in Section 3.2. The discussion of the experimental results has been presented in Sections 3.3.

A. Dataset

The effects of the three data mining classifiers (DT, RF, and *k*-NN) are evaluated on the performance of IDS using a simulated dataset available in [14]. The simulated dataset consists of regular packet trace files from various connection points to a network backbone. The file contains 19,998 different time-stamped row packets with different types of ICMPv6-based DoS [15] attacks triggered from the THC IPv6 toolkit [16]. In addition, the datasets consist of different sets of features, such as source IP, destination IP, ICMPv6 type,

and payload length, among others. In this experiment, only seven features, namely, source IP, destination IP, ICMPv6 type, payload length, hop limit, length, and payload length, are selected. The reasons for using a simulated dataset over other available datasets are as follows: (1) the IPv6 dataset, similar to the Ark IPv6 topology dataset, has the limitation of having specific traces and is not a comprehensive dataset of ICMPv6-based DoS flooding attack, (2) the source IP and destination IP features in the benchmark data, which play a significant role in the detection of ICMPv6-based DoS flooding attack, are removed from the dataset because of security and privacy issues; and (3) the lack of an available dataset used as a benchmark dataset for IPv6 networks. Dataset attribute values are normalized. Normalization is a process of ensuring that each attribute value in a dataset is suitable for further query and free from undesirable characteristics [17], [18]. For the purpose of the present study, source IP, destination IP, and time attribute values are normalized by converting each attribute into a unique integer value. The remaining packet size attribute value remains unchanged.

B. Performance metrics

The performance of IDS fitted with data mining classifiers was measured by calculating detection accuracy, precision, and recall percentage. Equation 1 is the standard formula that calculates detection accuracy, precision calculation uses Equation 2,and the formula for recall utilizes Equation 3. TableI describes the parameters in the accuracy metric equation.

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) * 100 \qquad (1)$$

PRECISION (P) is the proportion of attack cases that are correctly predicted relative to the predicted size of the attack class as calculated using the following equation:

$$Recall = \left(\frac{TP}{TP + FP}\right) * 100 \qquad (2)$$

RECALL (R) or TRUE POSITIVE RATE (TPR) is the proportion of correctly predicted attack cases to the actual size of the attack class as calculated using the following equation:

$$Accuracy = \left(\frac{TP}{TP + FN}\right) * 100 \qquad (3)$$

TABLE I
THE DECEPTION OF ACCURACY METRICS

Metric	Description
True positive (TP)	Number of samples correctly predicted as attack class
False positive (FP)	Number of samples incorrectly predicted as attack class
True negative (TN)	Number of samples correctly predicted as normal class
False negative (FN)	Number of samples incorrectly predicted as normal class.

TABLE II
HARDWARE SPECIFICATION

CPU	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.67GHz
Memory	5.00 GB
Operating System	Windows 7 (64 bit)

C. Experimental result discussion

This section presents and discusses the experimental result of applying the DT, RF, and *k*-NN classifier algorithms against the aforementioned simulated dataset (Section 3.1). The experiment was conducted using RapidMiner [19]. Table II shows the specification of the hardware used to perform the experiments.

First, the inter-rater agreement for the qualitative (categorical) items was measured. In statistics, inter-rater is the degree of agreement among raters. To measure, the kappa value [20] was calculated for each classifier. The result is presented in Figure 1.

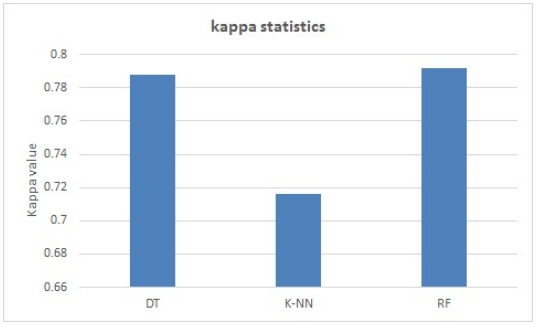

Fig. 1. Kappa values of each classifier

The kappa values for the classifiers range from 0.73 to 0.79, which indicates a substantial agreement among the classifiers. The performance of the classifier algorithms is based on the calculation of accuracy, precision, and recall as mentioned in subsection 3.2. Figure 2 shows the accuracy percentage of each classifier algorithm, and Figure 3 presents the precision and recall percentages.

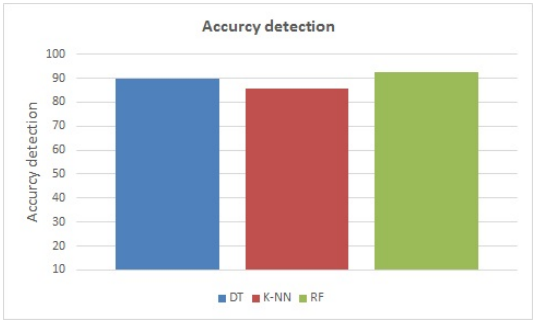

Fig. 2. Detection accuracy of each classifier

As shown in Figure 2, all tested classifiers achieved 85% and above in terms of detection accuracy, thus indicating that

TABLE III
k-NN PARAMETERS

K	1
Measure type	Mixed measures
Mixed measure	Mixed Euclidean distance

TABLE IV
DT AND RT PARAMETERS

Parameter	RD	DT
Number of trees	10	-
Criterion	Gain ratio	Gain ratio
Maximal depth	20	20
Confidence	0.25	0.25
Minimal gain	0.1	0.1
Minimal leaf size	2	2

most of the ICMPv6-based DoS attacks are properly detected. Figure 2 also shows that RF has the highest detection accuracy rate at 92.54% and outperforms the other classification algorithms. The accuracy of any classifier depends on the type of data, sample size, and data dimension. The result concludes that RF outperforms the other classifier algorithms in term of accuracy detection might be because of a small data size with a small set of features (seven features) was used. A high number of feature sets can have negative effects on the accuracy of RF, thus performing a feature reduction before using the dataset is highly recommended to evaluate the accuracy of the RF algorithm.

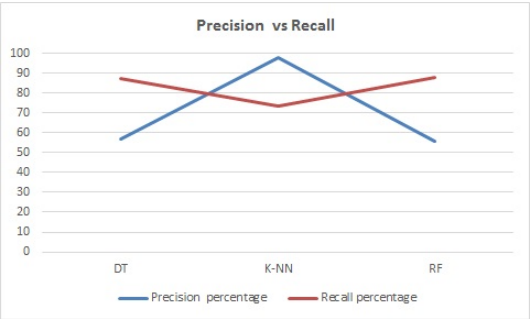

Fig. 3. Precision and recall percentages of each classifier

As shown in Figure 3, *k*-NN has the highest precision percentage (97.65%), which means that it has the lowest false-positive value. RF has the highest recall percentage (88.17%), which means that it has the lowest false-negative (missed attacks) percentage. The experimental results show that no single best algorithm can outperform the others in all situations. The percentage of accuracy, recall, and precision can be further increased or decreased by changing the classifier algorithm parameters. However, in these experiments, the default parameter set by RapidMiner was used. The exercise to determine the best or optimum parameters for the classifiers is left for future extensions of this work. Tables III and IV list the parameters used for *k*-NN as well as DT and RT, respectively.

IV. CONCLUSION

The IDS has become an integral part of any network to secure network resources and data from intruders. The effectiveness of IDS depends on robust classification algorithms. In this paper, the performance and effectiveness of three classifier algorithms in IDS were evaluated. The effectiveness of three classifier algorithms was benchmarked by running the evaluation with simulated datasets. Three performance metrics were used to evaluate the performance of classifier algorithms: (1) accuracy, (2) precision, and (3) recall. The experimental result shows that RF outperforms DT and *k*-NN in terms of accuracy detection. RF also has a lower false-negative percentage and *k*-NN has better results than the other two classifiers in terms of precision.

A potential extension of this work includes conducting experiments to determine the best and optimal parameters for the classifiers instead of using default values set by the tool. Comparison with other types of classifiers is also a possibility for future work.

ACKNOWLEDGMENT

This research is supported by Short Term Research Grant, Universiti Sains Malaysia (USM) No: 304/PNAV/6313272

REFERENCES

[1] R.-C. Chen, K.-F. Cheng, Y.-H. Chen, and C.-F. Hsieh, "Using rough set and support vector machine for network intrusion detection system," in *Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on*. IEEE, 2009, pp. 465–470.

[2] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*, 2009.

[3] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the jam project," in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, vol. 2. IEEE, 2000, pp. 130–144.

[4] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao, "Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation," in *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*. ACM, 2011, pp. 29–36.

[5] R. M. Saad, M. Anbar, S. Manickam, and E. Alomari, "An intelligent icmpv6 ddos flooding-attack detection framework (v6iids) using back-propagation neural network," *IETE Technical Review*, pp. 1–12, 2015.

[6] O. E. Elejla, M. Anbar, and B. Belaton, "Icmpv6-based dos and ddos attacks and defense mechanisms: Review," *IETE Technical Review*, pp. 1–18, 2016.

[7] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[8] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[10] G. Biau, L. Devroye, and G. Lugosi, "Consistency of random forests and other averaging classifiers," *Journal of Machine Learning Research*, vol. 9, no. Sep, pp. 2015–2033, 2008.

[11] D. Gadaleta, F. Pizzo, A. Lombardo, A. Carotti, S. E. Escher, O. Nicolotti, and E. Benfenati, "A k-nn algorithm for predicting oral subchronic toxicity in the rat," *Altex*, vol. 31, no. 4, pp. 423–432, 2014.

[12] O. Raevsky, V. Y. Grigor'ev, E. Liplavskaya, and A. Worth, "Prediction of acute rodent toxicity on the basis of chemical structure and physicochemical similarity," *Molecular Informatics*, vol. 30, no. 2-3, pp. 267–275, 2011.

[13] I. B. Stoyanova-Slavova, S. H. Slavov, B. Pearce, D. A. Buzatu, R. D. Beger, and J. G. Wilkes, "Partial least square and k-nearest neighbor algorithms for improved 3d quantitative spectral data–activity relationship consensus modeling of acute toxicity," *Environmental toxicology and chemistry*, vol. 33, no. 6, pp. 1271–1282, 2014.
[14] O. E. B. B. M. A. A. Alnajjar, "A reference dataset for icmpv6 flooding attacks," *International Journal of Engineering and Applied Sciences*, vol. 11, no. 3, pp. 476–481, 2016.
[15] M. Anbar, R. Abdullah, R. M. Saad, E. Alomari, and S. Alsaleem, "Review of security vulnerabilities in the ipv6 neighbor discovery protocol," in *Information Science and Applications (ICISA) 2016*. Springer, 2016, pp. 603–612.
[16] M. Heuse, "Thc ipv6 attack tool kit," urlhttps://www.thc.org/thc-ipv6/", 2016, online; accessed 8 August 2016.
[17] Z. Liu *et al.*, "A method of svm with normalization in intrusion detection," *Procedia Environmental Sciences*, vol. 11, pp. 256–262, 2011.
[18] E. F. Codd, "Further normalization of the data base relational model," *Data base systems*, pp. 33–64, 1972.
[19] A. Naik and L. Samant, "Correlation review of classification algorithm using data mining tool: Weka, rapidminer, tanagra, orange and knime," *Procedia Computer Science*, vol. 85, pp. 662–668, 2016.
[20] A. J. Viera, J. M. Garrett *et al.*, "Understanding interobserver agreement: the kappa statistic," *Fam Med*, vol. 37, no. 5, pp. 360–363, 2005.

IEEE COPYRIGHT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

TITLE OF PAPER/ARTICLE/REPORT/PRESENTATION/SPEECH (hereinafter, "the Work"):

COMPLETE LIST OF AUTHORS:

IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book):

Copyright Transfer

The undersigned hereby assigns to the Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to the above Work, and any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work. The undersigned hereby warrants that the Work is original and that he/she is the author of the Work; to the extent the Work incorporates text passages, figures, data or other material from the works of others, the undersigned has obtained any necessary permissions. **See reverse side for Retained Rights and other Terms and Conditions.**

Author Responsibilities

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements of IEEE Policy 6.4, including provisions covering originality, authorship, author responsibilities and author misconduct. The full policy may be viewed at http://www.ieee.org/about/whatis/policies/p6-4.xml. Authors are advised especially of IEEE Policy 6.4.1B(k): "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE Policy 6.3.B: "It shall be acknowledged that statements and opinions given in work published by the IEEE are the expression of the authors. Responsibility for the content of published papers rests upon the authors, not IEEE."

General Terms

- The undersigned represents that he/she has the power and authority to make and execute this assignment.
- The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
- In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall become null and void and all materials embodying the Work submitted to the IEEE will be destroyed.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.

(1) _____ 20/10/2016

Author/Authorized Agent for Joint Authors **Date**

U.S. Government Employee Certification (where applicable)

This will certify that all authors of the Work are U.S. government employees and prepared the Work on a subject within the scope of their official duties. As such, the Work is not subject to U.S. copyright protection.

(2) _____ _____

Authorized Signature **Date**

(Authors who are U.S. government employees should also sign signature line (1) above to enable the IEEE to claim and protect its copyright in international jurisdictions.)

Crown Copyright Certification (where applicable)

This will certify that all authors of the Work are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties. As such, the Work is subject to Crown Copyright and is not assigned to the IEEE as set forth in the first sentence of the Copyright Transfer Section above. The undersigned acknowledges, however, that the IEEE has the right to publish, distribute and reprint the Work in all forms and media.

(3) _____ _____

Authorized Signature **Date**

(Authors who are British or British Commonwealth Government employees should also sign line (1) above to indicate their acceptance of all terms other than the copyright transfer.)

rev. 121302

IEEE COPYRIGHT FORM *(continued)*

RETAINED RIGHTS/TERMS AND CONDITIONS

1. Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.

2. Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.

3. Authors/employers may make limited distribution of all or portions of the Work prior to publication if they inform the IEEE in advance of the nature and extent of such limited distribution.

4. In the case of a Work performed under a U.S. Government contract or grant, the IEEE recognizes that the U.S. Government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract/grant so requires.

5. For all uses not covered by items 2, 3, and 4, authors/employers must request permission from the IEEE Intellectual Property Rights office to reproduce or authorize the reproduction of the Work or material extracted verbatim from the Work, including figures and tables.

6. Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. The IEEE Intellectual Property Rights office must handle all such third-party requests.

INFORMATION FOR AUTHORS

IEEE Copyright Ownership

It is the formal policy of the IEEE to own the copyrights to all copyrightable material in its technical publications and to the individual contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and, at the same time, to facilitate the appropriate re-use of this material by others. The IEEE distributes its technical publications throughout the world and does so by various means such as hard copy, microfiche, microfilm, and electronic media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various compendiums, collective works, databases and similar publications.

Author/Employer Rights

If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IEEE assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

Reprint/Republication Policy

The IEEE requires that the consent of the first-named author and employer be sought as a condition to granting reprint or republication rights to others or for permitting use of a Work for promotion or marketing purposes.

PLEASE DIRECT ALL QUESTIONS ABOUT THIS FORM TO:
Manager, IEEE Intellectual Property Rights Office, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331.
Telephone +1 (732) 562-3966