International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 5835-5843

A Random Forest Estimator Combined With N-Artificial Neural Network Classifiers to Optimize Network Intrusion Detection

I. Lafram[1], N. Berbiche[2] and J. El Alami[3]

[1]PhD student, Mohamed V University, Rabat, Morocco.

[2]Research professor at LASTIMI laboratory and Professor in the Department of Computer Sciences Superior School of Technologies of Salé, Mohammed V University, Rabat, Morocco.

[3]Research Professor, LASTIMI Laboratory and Professor in the Department of Computer Sciences, Superior School of Technologies of Salé, Mohammed V University, Rabat, Morocco.

[1]Orcid: 0000-0002-2651-4515

Abstract

Information systems have become more complex and highly interconnected. While ensuring real-time connectivity, these systems encounter an increasing amount of malicious traffic. Hence the need to establish a defense method. One of the most common tools for network security is intrusion detection and prevention systems (IDPS). An IDS, while supervising the incoming traffic, tries to identify suspicious activities using either predefined signatures or pre-established user behavior. Signature and behavior based intrusion detection systems are unable to detect new attacks and fall down when facing small behavior deviations. To remedy this problem, many researchers have proposed different approaches for intrusion detection using machine learning techniques as a new and promising tool. Most of the proposed works focus on accuracy over latency and productivity and are tested on the outdated and much criticized kdd99 dataset [1].

In this paper, the authors present a two-level classification framework as a fast, scalable and much accurate traffic classification system, combining early network services identification using a Random Forest estimator and n-Artificial Neural Networks for packets classification. The performance of this model is evaluated on the relatively new proposed dataset of New Brunswick University, showing quick classification process with very high accuracy results.

Keywords: Intrusion Detection, Machine Learning, Traffic Classification, Artificial Neural Networks, Random Forest

NOMENCLATURE

IDS	Intrusion Detection System
IPS	Intrusion Prevention System
ANN	Artificial Neural Network
FNN	Feedforward Neural Networks
ML	Machine Learning
RFE	Random Forest Estimator
MLP	Multilayer perceptron
GI	Gini Index

INTRODUCTION

The rapid development of the internet and the expansion of information systems is a great achievement, but the world of highly connected computers has its harmful side: the damage caused by malicious purposes of hackers. A defender of a system is forced to assess the situation and develop a defense strategy. Intrusion detection systems are essential security tools implemented to detect network infractions. In the design of intrusion detection systems (IDS), there are two main approaches. The first defines signature-based IDS: intrusions are detected by looking for activities that correspond to known signatures of attacks. It requires the use of signature files that identify the intrusive activity. The second approach defines an anomaly-based system that detects intrusions by looking for normal behavior deviations [1].

The abnormal traffic behavior can be triggered by violating an event threshold frequency in a connection or by breaking a legitimate profile known as normal.

These static detection approaches, suffer from a set of disabilities and generally lead to inadequate detection systems [2]:

- Unable to detect new and unknown attacks
- Unable to detect variations of known attacks
- Require frequent updates, hence the loss of real-time detection
- Resources consuming …

As a remedy to these problems, machine learning techniques so-called, soft computing techniques, have been proposed to

break into intrusion detection domain. Soft computing describes a set of techniques of optimization and processing that are tolerant of imprecision and uncertainty [3].

The application of these techniques aims to train algorithms to learn from the existing data to be able to predict the class of each entry. To perform this process, researchers use publicly available datasets to train and evaluate detection models. The most widely used one is the kdd99 cup dataset [4] built based on the data captured in DARPA'98 IDS evaluation program [5]. This dataset lacks the very important characteristic of reflecting real-world traffic patterns and is outdated [6]. This paper presents a detection approach based on two-level of serial classification for fast detection and high accuracy. While traditional approach relies on entire packets inspection, we use a random forest estimator for early identification of the incoming traffic. Once the network service is identified, a specific artificial neural network classifies it. The model presented here has been trained and evaluated using the UNB (University of New Brunswick) Canadian Institute for Cyber security ISCX-IDS 2012 dataset [7].

This paper is organized as follows: the second section presents an overview of the techniques used. Some related works that have been achieved in the same context are discussed in the third section. We will present the methodology we followed in our study and our reasons for opting for a new dataset in the fourth section. The fifth section propounds the proposed detection model. Finally, the experimental results, which validate this approach, are provided in the last section.

TECHNIQUE OVERVIEW

A) Random forest estimator

Random forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees [8]

The strong point of random forests is their non-parametric nature which makes them one of the best families of classifiers [9]. Also, RF presents many advantages:

- It runs efficiently on large datasets
- It can handle thousands of input variables
- It gives estimates of what variables are important in the classification

RF provides an algorithm for estimating missing values [10]; Thus, greater classifier stability is achieved, as it makes it more robust when facing slight variations in input data and, at the same time, it increases classification accuracy [11].

A Random Forest classifier consists of a combination of classifiers

$$\{h(x, \Theta_k),\ k = 1, \dots.\} \quad (1)$$

x is the input vector

$\{\Theta_k\}$ are the independent and identically distributed random vectors [12]

Each classifier contributes with a single vote for the assignation of the most frequent class to the input vector (x).

$$\hat{C}_{rf}^{B} = majority\ vote\ \{\hat{C}_b(x)\}_1^B \quad (2)$$

$\hat{C}_b(x)$ is the class prediction of the b^{th} random forest tree.

RF increases the diversity by growing bootstrapped random trees using random features from different training data subsets [16]. Bootstrap aggregating is a technique used for training data creation by resampling randomly the original dataset with replacement [13].

By using a given combination of features, a decision tree is made to grow up to its maximum depth. [14].

When increasing the number of trees, the generalization error always converges and over-training is not a problem due to the "Feller's Strong Law of Large Numbers" [15].

A random forest estimator usually uses the Gini Index [16] as a measure for the best split selection, which measures the impurity of a given element with respect to the rest of the classes. For a given training dataset T, the Gini Index can be expressed as:

$$\sum \sum_{i \neq j} (f(C_i, T)/|T|)(f(C_j, T)/|T|) \quad (3)$$

$f(C_i, T)/|T|$ is the probability that a selected case belongs to class C_i

B) Feedforward Neural Networks

B.1) Information processing by an artificial neuron

A neuron is the basic processing unit of a neural network. It is connected to sources of information as input and returns output information.

The neuron receives a number of input information, each information is recovered by the neuron via its weight.

A weight is a coefficient w_i simply related to the information x_i. The i^{th} neuron receives the information $(w_i \times x_i)$. This data is passed to a neuron activation function to produce a final output of the neuron. [17]

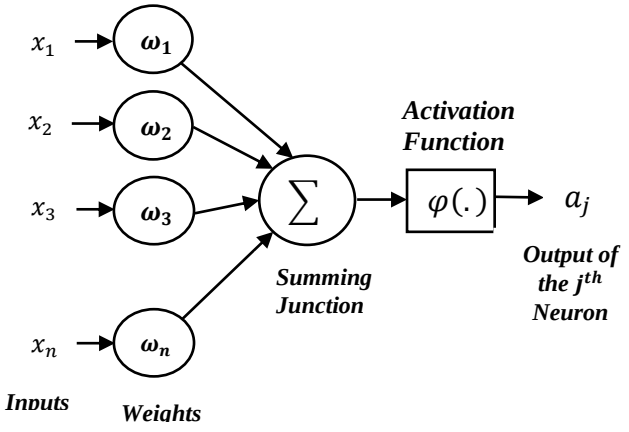


Figure 1: Schema of a single artificial neuron

We denote:

$\omega_{i,j}$ for ($1 <= i <= n$) and ($1 <= j <= p$), the weight connecting the information x_i and the neuron j

And

a_j the output of the j^{th} neuron, defined by the following equation:

$$\forall\, 1 \le j \le p : a_j = \varphi\left(\sum_{i=0}^{n} w_{i,j} \times x_i\right) \qquad (4)$$

The output a_j may become the stimulus for neurons in the next layer.

We used the sigmoid as the activation function given by:

$$\varphi(\gamma) = \frac{1}{(1 + e^{-\gamma})} \qquad (5)$$

In a neural network, activation function is used to produce a non-linear decision boundary via linear combinations of the weighted inputs [18]. The non-linear activation functions in the hidden layer neurons enable the neural network to be a universal approximator [19] [20].

B.2) Multilayer perceptron

The multilayer perceptron (MLP) belongs to feedforward neural networks (FFNN) structure, a basic type of artificial neural networks, capable of approximating generic classes of functions [18].

The FFNN architecture consists of an input layer, an output layer, and one or more hidden layers built of processors called neurons, which are fully interconnected with neurons in the subsequent layer using adaptable weighted connections.

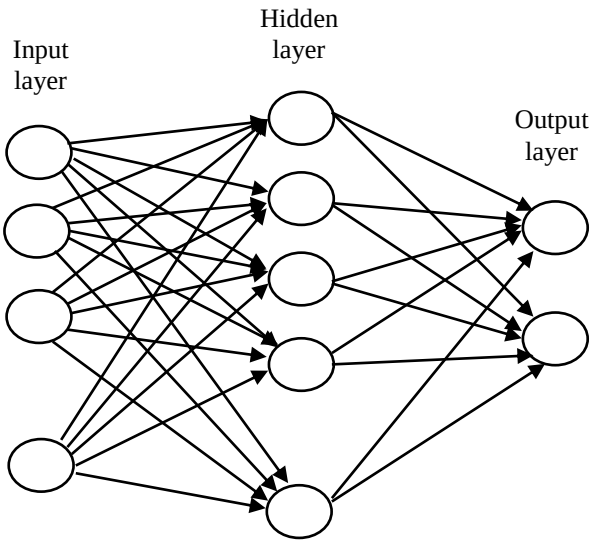


Figure 2: Architectural graph of MLP with one hidden layer

During the training (process of finding an optimal set of weight parameters ω to approximate the original problem behavior), training data is given as pairs of $(x_k, d_k), k = 1, \ldots, P$ where d_k is the desired outputs for inputs x_k of P training samples. The Backpropagation learning algorithm is applied for minimization of error function [21] defined by:

$$E = \frac{1}{2} \sum_{k \in T_r} \sum_{j=1}^{m} (y_j(x_k, \omega) - d_{jk})^2 \qquad (6)$$

$d_{jk} : j^{th}$ element of d_k

$y_j(x_k, \omega) : j^{th}$ neural network output for input x_k

T_r : A set of training data

RELATED WORKS

Most of the existing works are developed based on the flow lengths of the kdd99 cup dataset. A key issue concerning an intrusion detection system is the high dimensionality of data input vectors that have to be analyzed in order to identify network attacks [22]. This is because the higher is the dimensionality, the more time consuming is the model in terms of training and prediction setups.

Xiangmei Li [23] proposed an optimization of the Neural-Network-Based Multiple Classifiers Intrusion Detection System by adjusting the 41-dimensional input features. The multiple classifier intrusion detection system composed of DOS attacks sub-classifier, Probe attacks sub-classifier, R2L attacks sub-classifier and U2R attacks sub-classifier. Every sub-classifier is a neural network classifier designed to detect only one type of the attacks.

Results show that every adjusted sub-classifier is better in convergence precision, shorter in training time than the 41-features sub-classifier and the whole intrusion detection system is higher in the detection rate, and less in the false negative rate than the 41-features multiple classifiers intrusion detection system.

Although, the accuracy of the proposed system is optimized and showed better results, it didn't prove how fast the classification process was because that every sub-classifier should inspect the whole incoming traffic causing redundant and time-consuming process.

In 2006 Bernaille et al. [24] proposed a technique using an unsupervised ML (Simple K-Means) algorithm that classified different types of TCP-based applications using the first few packets of the traffic flow. The proposed method relies on the application's negotiation phase (usually a pre-defined sequence of messages) which characterizes every network application. The results of this work are inspiring for early detection of the traffic flow. However, it assumes that the classifier can always capture the start of each flow; missing the first few packets of the traffic may cause an ineffectiveness of the classifier. Also, with the use of unsupervised algorithm, the model faces the challenge of classifying an application when it does not dominate any of the clusters found.

Vladimir Bukhtoyarov et al. [25] proposed a neural network ensembles approach in which they joined many trained neural networks in order to combine outputs to get a solution for the classification problem. But the approach is hard to implement because of the complexity of the network topologies in real systems. Moreover, the effectiveness of the knowledge exchange between the ensemble members has not been proved.

METHODOLOGY

A) Dataset

A.1) The choice of the dataset

The most significant challenge an evaluation faces is the lack of appropriate public datasets for assessing anomaly detection systems." [26]

To measure the performance of any detection approach, we need to practice it and experiment with data that simulate real traffic of modern networks to an acceptable level.

Available datasets for machine learning in the field of network intrusion detection systems is limited. One of the few but at the same time, widely used datasets is the DARPA datasets (KDDcup99, NSL-KDD). Although, they are the most comprehensive existing datasets, they may not be a perfect representative of existing real networks and still suffer from the problems discussed by [6] and [27].

In our study, we wanted to test our approach on data close to the one generated in our network. For this, we opted for a relatively current IDS evaluation dataset containing real-world representative traffic data. The UNB ISCX 2012 dataset [7] generated by UNB (University of New Brunswick) and collected from modern complex networks.

UNB ISCX 2012 dataset has been generated in a physical testbed implementation using real devices that dynamically generate real traffic which reflects network traffic and intrusions [28].

The UNB ISCX 2012 Intrusion Detection Evaluation Dataset characteristics are:

- Realistic network and traffic
- Labeled dataset
- Total interaction capture
- Complete capture
- Diverse intrusion scenarios

A.2) Traffic composition of the UNB ISCX dataset

The traffic generated by UNB center is based on real traffic for HTTP, SMTP, FTP, SSH, IMAP and POP3 which is vital for the realism and effectiveness of the data set. Malicious activities were generated with multiple scenarios to make them sophisticated and hardly detectable [7].

The following diagram shows the composition of the generated traffic:

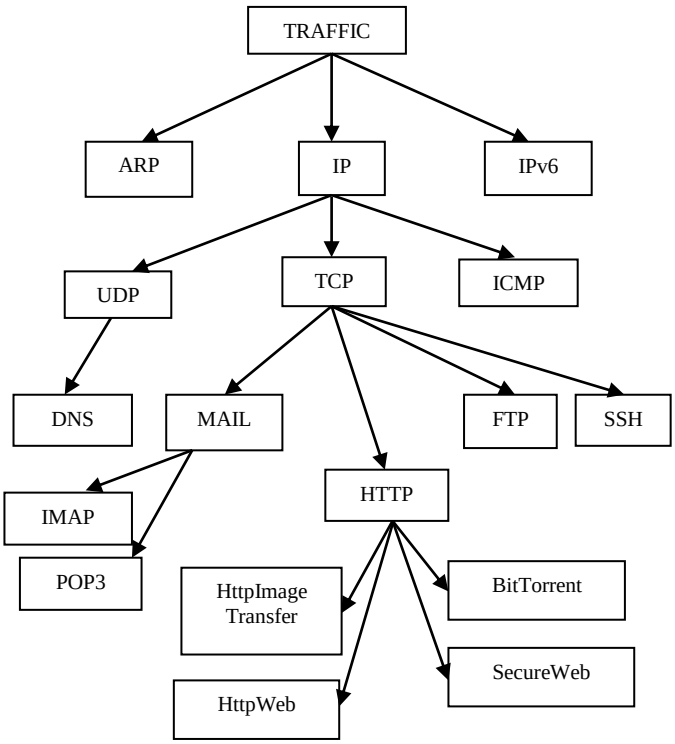


Figure 3: UNB dataset traffic composition diagram

Scenarios based on predefined user profiles were executed to generate normal traffic, besides various multi-stage attacks to mimic malicious behavior.

Our dataset is composed of the whole flows generated during the experiment on data collection at the UNB laboratory. Data was collected continuously during seven days and flows are distributed as follow:

Table 1. UNB-ISCX dataset major flows distribution

Type of flow	*Number of records*
TCP	1,941,454
UDP	498,032
ICMP	10,689
Normal	2,381,532
Attack	68,792
Total	2,450,324

The traffic observed in the dataset contains the internet most used services; this reflects the realism of the test bed network and shows that the experiment of generating a representative flow was conducted rigorously.

Traffic composition shows the network protocols and services most present in the dataset, the majority of it is IP traffic which contains mainly TCP packets as shown below.

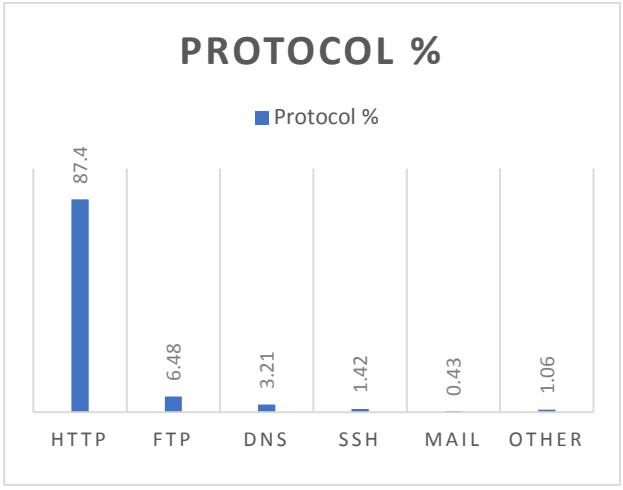


Figure 4: Protocols and applications percentage of the dataset

A.3) Feature Selection

The objective of attribute selection is to reduce the variables of the data that are irrelevant in order to improve the performance of the classifier making it faster and cost-effective. Our approach is based on a two-stage classification process, a random forest estimator for early traffic identification to recognize the network service and an appropriate neural network classifier to determine the packet nature. Thus, features will be selected depending on the classification stage.

a.1) First stage classification features

In our approach, at the classification first-stage, RFE evaluates the early traffic attributes by analyzing the basic features (table1) of packets header without inspecting the payload. These features listed below constitute the input vector of the random forest estimator which role is to identify the network service class of the incoming traffic.

Unlike traditional methods that use protocols-port application mapping and deep-packets inspection in order to determine packets signatures, random forest classifier relies on statistical pattern recognition to overcome some malicious techniques such as applications using ephemeral port number.

a.2) Early traffic identification

The nature of each network application allows classifying it into one of several discrete categories. Our technique uses training data, with samples of well-known traffic to allow the categorization of traffic using commonly-available information only. Basic features are the header first features listed below.

Table 2. Features selected for early traffic identification

Feature	Description
Protocol type	Connection protocol (e.g. udp, tcp)
Service	Destination service (e.g. telnet, ftp)
Flag	Status flag of the connection
Source bytes	Bytes sent from source to destination
Destination bytes	Bytes sent from destination to source
Port	Source/destination port number
Land	1 if connection is from/to the same host/port; 0 otherwise

The features listed below constitute the input vector of the random forest estimator which role is to classify the incoming traffic.

Being independent of packet payload inspection is a powerful aspect of our approach, which is robust to encryption, scalable and adaptive when new protocol join. The trained classifier

can be applied to determine the class of even unknown flows.

b) Second stage classification features

To choose features that will give us the best accuracy and the fastest prediction model, we used the recursive feature elimination algorithm where different subsets of features are evaluated and compared to each other in order to determine the best subset based on the model accuracy.

Depending on the first stage classification, the incoming packets are identified by their network service identifier. Incoming classes have their specificities and the input vector of every ANN-classifier of the second stage classification will be treated based on its service class.

The behavioral features (e.g. distribution of the size of packets, TCP window size, TCP flag bits and packet directions, are derived from the packet headers, the same source of information that is expected to be used by the routers of the internet [29].

Some applications use the IP layer encryption which makes traffic payload inspection impossible. Our approach is payload-independent so it robust to encryption. Traffic is classified by determining the similarity between it and groups formed of packets having similar traffic patterns.

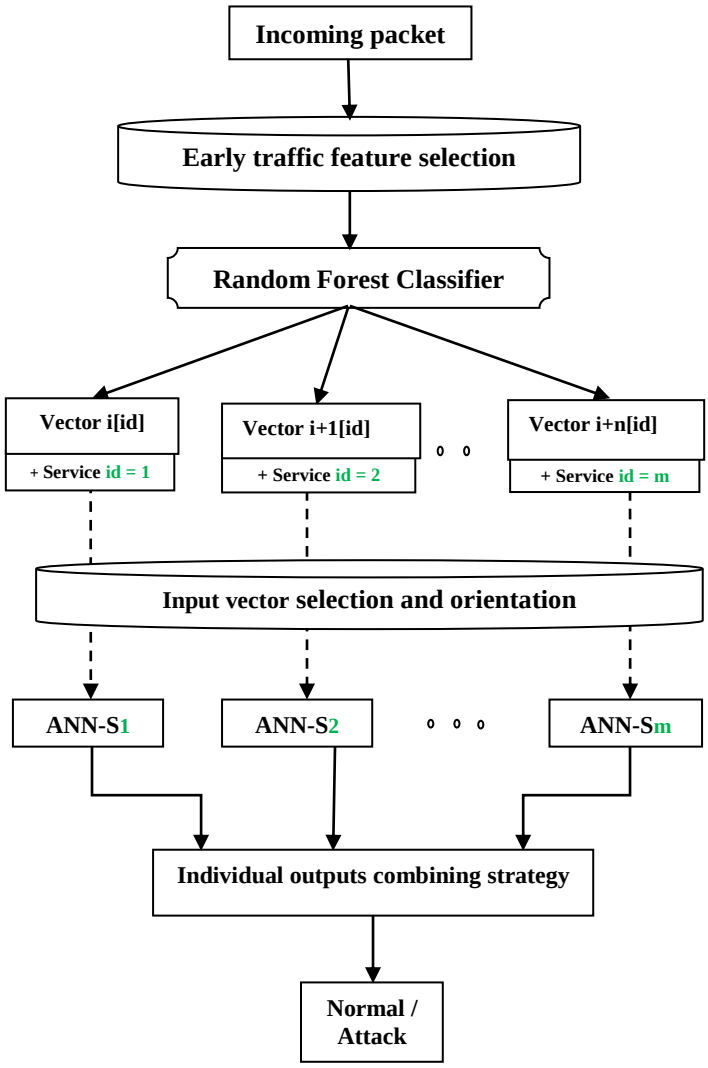
PROPOSED WORK

In this part, we will present a diagram and an algorithm to better illustrate our proposed approach for network traffic classification.

While most of the related existing studies propose a single machine learning classification model, we designed a two-level classification framework with n-artificial neural network classifier. The system obtained clearly shows an increase in performance: better accuracy and a reduced computation time.

A) The diagram

The incoming traffic passes by the random forest classifier which determinates its protocol service, and then the incoming vector is processed during feature selection. Based on its protocol, the incoming vector is processed by the appropriate artificial neural network which finally determines whether it is a normal or an attack traffic. The diagram below illustrates the classification framework:



g 8. Proposed classification framework

Figure 5: Proposed classification framework

Incoming packet: whole packet information passing by the network

Early traffic selection: selection of the early intercepted information of incoming packet (Table 1)

Random Forest classifier: The RF intercepts the early version of input vector for rapid identification of the service

Vector i[id]: the classified incoming vector with whole set of features and labeled with proper id (class protocol service identifier)

Input vector selection and orientation: Selection of best features subset for every incoming vector (e.g. eliminating common class features, adding the duration on real time…) then send the packet to its specific ANN.

Service Id: Every service has a given identifier from 1 to m.

ANN-SId: An artificial neural network trained only for one specific protocol service (e.g. HttpWeb, ftp, ssh…).

B) The algorithm

The algorithm is a three-step process:

- Rapid packets identification: by inspecting the early coming information of packets, the RF estimator classifies them into protocol service classes.

- Input vector selection and orientation: Once identified, the vector is set of the best subset of features and then directed to its proper ANN classifier.

- Final traffic classification: Finally, the incoming traffic get classified whether malicious or benign.

ALGORITHM: PACKETS CLASSIFICATION

INPUT: incoming packet Pi

OUTPUT: classified packet Pi(normal/attack)

```
BEGIN
/***/
 FOR i FROM 1 TO n DO
    FOREACH Pi DO
      /* rapid packets identification*/
      select first six attributes from packet header
      create an early input vector Vi
      /* Random Forest first level of classification*/
      RF classifier: Classify Vi
      set corresponding identifier to Vi
      Vi becomes Vi [id = network service identifier]
      /* Input vector selection */
      FOREACH Id FROM classId [1, m] DO
        select best features subset for Vi[id]
        Vi[id] becomes Ui[id]
        send Ui[id] to ANN-S[id]
      END
      /* Artificial neural network classification */
      classify Ui[id]
    END
    Return class (Pi)
 END
 /***/
 END
```

EXPERIMENTS AND RESULTS

The detection model has to classify the incoming traffic whether normal or attack. We aimed to optimize the existing ANN based intrusion detection systems that classify the incoming traffic directly by analyzing the whole packets.

First test is applied to the whole features of the packets. Second one, after applying early traffic identification with RFE for network service classification, a specific artificial neural network (ANN-Sid) is applied to a subset of best features selected for the given network service.

We tested our system on UNB ISCX dataset and the results below show the robustness of our work;

The most present protocol services are: Http-Image-Transfer, HttpWeb, ftp, SecureWeb, ssh, dns, BitTorrent, Imap, Pop3.

We tested our framework on these services and here are the results:

The detection time has decreased by applying two stage classification. The fact that the artificial neural network processes on one network service makes it faster and accurate. The training time is reduced because the ANN doesn't need to make computational operations to all other network services.

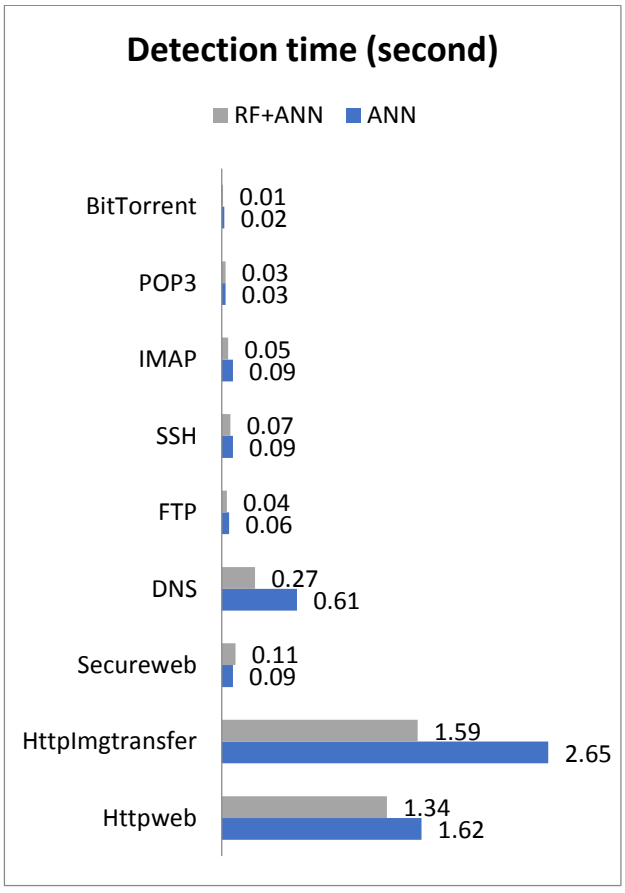


Figure 6: Detection time before and after two-stage classification

Accuracy percentage using one artificial neural network model to classify all data and the accuracy percentage using two-level classification model and service-specific artificial neural networks.

Table 3. Accuracy results before and after two-stage classification

	Accuracy %	
Service	ANN	RF+(ANN-Sid)
HttpWeb	61,31	96.62
HttpImageTransfer	58,11	99.97
SecureWeb	98,18	99.94
DNS	93,82	99.96
FTP	73.62	99.05
SSH	91.93	98.82
IMAP	90.82	98.82
POP3	65.74	99.72
BitTorrent	87.82	100

The accuracy of each specific artificial neural network is higher when applied to a single network service.

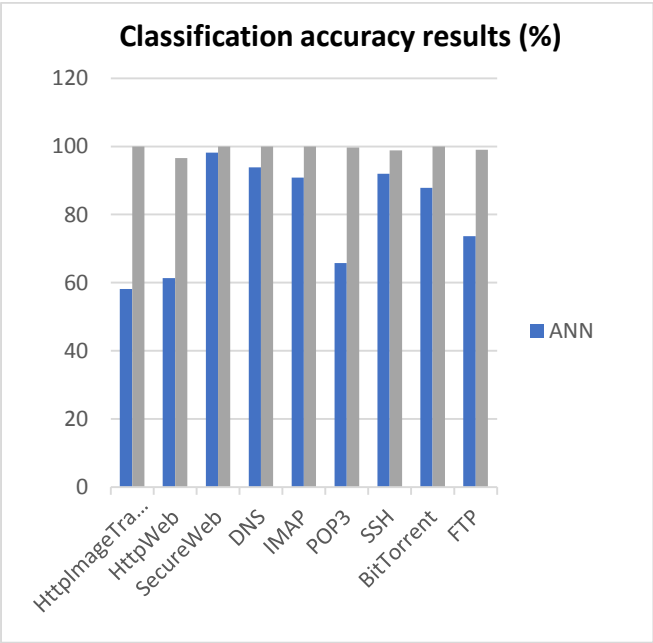


Figure 7: Accuracy results before and after two-stage classification

The random forest estimator has shown an accuracy of 99.99% on services classification, which makes the two stages classification much better in terms of accuracy and rapidity.

CONCLUSION

A new approach for intrusion detection based on the combination of artificial neural networks and Random forests was presented in this paper, showing the optimized performance of the elaborated model.

The proposed framework, in addition to these high performances, is tested on a more recent dataset and therefore more representative of current computer networks. This is a strong point of our study.

Of course, the proposed model can be optimized again, especially the part of the detection time that can be minimized to obtain a very powerful intrusion detection system in real time. In the future work, we will focus on this point.

CONFLICT OF INTERESTS

The authors declare no conflict of interests.

REFERENCES

[1] Axelsson S.: Intrusion detection systems: A taxonomy and survey. Technical Report No 99-15, Dept. of Computer Engineering. Chalmers University of Technology, Sweden, March 2000.

[2] A.Jelsiana Jennet, Dr. J Frank Vijay. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp.12635-12641.

[3] Piero P.Bonissone, "Soft computing: the convergence of emerging reasoning technologies," Soft Computing Journal, vol 1, no 1, pp. 6-18, Springer-Verlag 1997.

[4] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Costbased modeling for fraud and intrusion detection: Results from the jam project," discex, vol. 02, p. 1130, 2000.

[5] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," discex, vol. 02, p. 1012, 2000.

[6] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln

laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.

[7] Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security, 31(3): 357-374.

[8] https://en.wikipedia.org/wiki/Random_forest.

[9] Manuel Fernandez-Delgado, et al., Do we Need Hundreds of Classifiers to Solve Real World Classification Problems, Journal of Machine Learning Research 15 (2014) 3133-3181.

[10] Victor F. Rodriguez-Galiano et al, An assessment of the effectiveness of a random forest classifier for land-cover classification. ISPRS Journal of Photogrammetry and Remote Sensing 67(1):93–104 · January 2012.

[11] Breiman I, 2001. Random forests. Machine learning 45 (1), 5-32.

[12] Hastie T, Tibshirani, R, Friedman, J, 2009. Random forests, The Elements of statistical Learning, Springer, New York, pp. 587-604.

[13] Efron, B.; Tibshirani, R. (1993). An Introduction to the Bootstrap. Boca Raton, FL.

[14] Pal, M, Mather, P.M, 2003. An assessment of the effectiveness of decision tree methods for land cover classification. Remote sensing of Environment 86 (4), 554-565.

[15] Yuan Shih Chow, Cun-Hui Zhang. The annals of probability, 1986, Vol. 14, No. 3, 1088-1094.

[16] Breiman, L., 1984. Classification and Regression Trees. Chapman & Hall/CRC.

[17] http://alp.developpez.com/tutoriels/intelligence-artificielle/reseaux-de-neurones.

[18] Zhang, Q. J., F. Wang, and V. K. Devabhaktuni, ''Neural Network Structures for RF and Microwave Applications,'' *IEEE AP-S Antennas and Propagation Int. Symp*., Orlando, FL, July 1999, pp. 2576–2579

[19] Cybenko, G., ''Approximation by Superpositions of a Sigmoidal Function,'' *Math. Control Signals Systems*, Vol. 2, 1989, pp. 303–314.

[20] Hornik, K., M. Stinchcombe, and H. White, ''Multilayer Feedforward Networks are Universal Approximators,'' *Neural Networks*, Vol. 2, 1989, pp. 359–366.

[21] Rumelhart, D. E., G. E. Hinton, and R. J. Williams, ''Learning Internal Representations by Error Propagation,'' in Parallel Distributed Processing, Vol. 1, D.E. Rumelhart and J. L. McClelland, Editors, Cambridge, MA: MIT Press, 1986, pp. 318–362.

[22] Samira sarvari, et al., GA and SVM Algorithms for Selection of Hybrid Feature in Intrusion Detection Systems. IRECOS, Vol 10, No 3 (2015).

[23] 978-1-4244-5143-2 ©2010 IEEE

DOI**:** 10.1109/ITAPP.2010.5566641

Internet Technology and Applications, 2010 International Conference, 20-22 Aug. 2010.

[24] L.Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, Traffic classification on the fly. ACM Special Interest Group on Data Communication (SIGCOMM) Computer Communication Review, vol. 36, no. 2, 2006.

[25] WCCI 2012 IEEE World congress on computational intelligence. June 10-15-2012, Brisbane, Australia

[26] Sommer R, Paxson V. Outside the closed world: on using machine learning for network intrusion detection. In: Security and privacy, IEEE Symposium on; 2010. p. 305e16.

[27] Brown C, Cowperthwaite A, Hijazi A, Somayaji A. Analysis of the 1999 DARPA/Lincoln laboratory IDS evaluation data with netadhict. In: Proceedings of the second IEEE international conference on computational intelligence for security and defense applications. Piscataway, NJ, USA: IEEE Press; 2009. p. 67e73.

[28] Richard Zuech et al, 2015. "A New Intrusion Detection Benchmarking System". Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference.

[29] T.T.T.Nguyen, G.Armitage, Asurvey of techniques for Internet Traffic Classification using Machine Learning. 4th edition 2008 of IEEE.