BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 03/07 pp18–24
Volume 4, Number 1, March 2021
DOI: 10.26599/BDMA.2020.9020019

Mathematical Validation of Proposed Machine Learning Classifier for Heterogeneous Traffic and Anomaly Detection

Azidine Guezzaz*, Younes Asimi, Mourade Azrour, and Ahmed Asimi

Abstract: The modeling of an efficient classifier is a fundamental issue in automatic training involving a large volume of representative data. Hence, automatic classification is a major task that entails the use of training methods capable of assigning classes to data objects by using the input activities presented to learn classes. The recognition of new elements is possible based on predefined classes. Intrusion detection systems suffer from numerous vulnerabilities during analysis and classification of data activities. To overcome this problem, new analysis methods should be derived so as to implement a relevant system to monitor circulated traffic. The main objective of this study is to model and validate a heterogeneous traffic classifier capable of categorizing collected events within networks. The new model is based on a proposed machine learning algorithm that comprises an input layer, a hidden layer, and an output layer. A reliable training algorithm is proposed to optimize the weights, and a recognition algorithm is used to validate the model. Preprocessing is applied to the collected traffic prior to the analysis step. This work aims to describe the mathematical validation of a new machine learning classifier for heterogeneous traffic and anomaly detection.

Key words: anomaly detection; heterogeneous traffic; preprocessing; machine learning; training; classification

1 Introduction

Recent applications require relevant and robust techniques to treat large, heterogeneous, and unstructured data[1–3]. Data classification involves a set of methods for processing and organizing data into categories for their most effective use. These methods are used to not only analyze, extract, and interpret data but also control the systems that produce these big data[4–6]. Data classification is one of the research axes aimed at achieving an automatic resolution of real-world problems, such as the recognition of forms and classification and generalization using intelligent techniques for data processing and evaluation. Large dimensions and heterogeneity pose a challenge to data analysis methods[5, 6].

The rest of the paper is organized as follows. In Section 2, we provide an overview of some machine learning algorithms that are used for data classification. We evaluate their performances on the basis of parameters related to reliability and accuracy. In Section 3, a new classifier model for heterogeneous traffic based on a new machine learning algorithm is described. In Section 4, the mathematical validation of the new machine learning classifier for heterogeneous traffic and anomaly detection is detailed. In Section 5, the conclusion and future work are presented. The following notations are used in this work (see Table 1).

• Azidine Guezzaz is with Department of Computer Science and Mathematics, High School of Technology, Cadi Ayyad University, Essaouira 44000, Morocco. E-mail: A.GUZZAZ@gmail.com.
• Younes Asimi is with Department of Computer Science, High School of Technology, Ibn Zohr University, Guelmim 81000, Morocco.
• Mourade Azrour is with IDMS Team, Department of Computer Science, Faculty of Science and Technology, Moulay Ismail University, Errachidia 52000, Morocco.
• Ahmed Asimi is with Department of Computer Science and Mathematics, Faculty of Sciences Agadir, Ibn Zohr University, Agadir 80000, Morocco.
∗ To whom correspondence should be addressed.

Manuscript received: 2020-06-09; accepted: 2020-08-25

Table 1 Notations used in the study.

Notation	Meaning
f	Sigmoid function
$(X_i)_{i=1,\ldots,n}$	Present inputs
$X_i = (x_{i,j})_{j=1,\ldots,m}$	Present occurrences to input X_i
$W^{(0)} = (w_{i,0})_{i=1,\ldots,n}$	Initialize weights
$W_i = (w_{i,j})_{j=1,\ldots,m}$	Model weights initialized randomly and associated to input X_i
$w_{0,i}$	Initialize bias to 1 and associated with input X_i
a_i	Weight sum associated to input X_i
$y\,(a_i) = f\,(a_i)$	Output associated to input X_i
ε_i	Error associated to input X_i
$\varepsilon = \max\{\varepsilon_i, i = 1, \ldots, n\}$	Maximum error $(X_i)_{i=1,\ldots,n}$
$W_i^{(\mathrm{op})} = (w_{i,j}^{(\mathrm{op})})_{j=1,\ldots,m}$	Optimal system solution (training algorithm) for input X_i
$w_j^{(\max)} = \max\{w_{i,j}^{(\mathrm{op})}, i = 1, \ldots, n\}$	Maximum weights associated with input X_i
$a_i^{(\max)} = \sum\limits_{j=1}^{m} w_j^{(\max)} x_{i,j} + w_0^{(\max)}$	Maximum bias associated with input X_i
$a_i^{(\mathrm{op})} = \sum\limits_{j=1}^{m} w_{i,j}^{(\mathrm{op})} x_{i,j} + w_{0,i}^{(\mathrm{op})}$	Optimal weighted sum associated to input X_i
$W^{(\max)} = (w_j^{(\max)})_{j=1,\ldots,m}$	Maximum weights
$w_0^{(\max)} = \max\{w_{0,i}^{(\mathrm{op})}, i = 1, \ldots, n\}$	Maximum bias
$d = +1$	Normal output
$d = -1$	Abnormal output

2 Background

We present here a number of machine learning algorithms used for data classification, and we analyze their performances.

2.1 Machine learning methods

Machine learning is considered a subfield of artificial intelligence. A subfield of machine learning is automatic classification[4,7]. Automatic learning uses different algorithms to solve classification problems by grouping homogeneous classes of similar data objects. The classification is supervised learning that requires the use of a training set to train the decision rule and construct a classifier. The examples used in supervised training are regarded as complete because they contain the values of the variables and their classification. Training is a developmental task in which the behavior changes until a desired one is achieved through the optimization of weights. Examples are presented to establish new connections or modify existing ones. The calculated result is compared with the expected response in the output[2,3,7]. Preprocessing is often applied to the data presented in the training[8,9]. The choice of a training algorithm depends on the specification of the problem to be solved[1]. In fact, the classification of data becomes a highly useful discipline that helps to solve complex problems. Data are not always in a form that can be readily analyzed. Thus, the standardization of unstructured data requires additional work.

2.2 Data classification methods

The current requirements of data analysis and classification have significantly increased to address the need to make decisions on the basis of standardization and information extraction techniques[7,10–12]. As a machine learning tool, classification is the natural choice for performing prediction with discrete known outcomes. A classification method is a set of precise rules to classify objects on the basis of quantitative and qualitative variables characterizing these objects. Data classification is carried out for a variety of purposes, the most common of which is to support data security issues, especially in anomaly detection[1,3,9,13,14]. To be effective, a classification model should be simple enough that all employees can execute it properly.

2.2.1 Support vector machines

Support Vector Machines (SVMs) are the machine learning methods derived from the early work of Vapnik[4, 11]. They are supervised training techniques designed to solve problems of discrimination and regression. The principle is to transform input variables into a large characteristic space and then find a suitable hyperplane that models the data in space and separates them into two groups so that the linear boundary produces a maximum margin (see Algorithm 1).

2.2.2 Multilayer perceptron

A network of neurons is a set of mathematical methods used to model processes and recognize patterns[4, 10, 12]. A formal neuron is a processing unit of several inputs performing complex logic, arithmetic, or symbolic functions. The output corresponds to a weighted sum of inputs:

$$a_i = \sum_{i=1}^{n} w_i^{(j)} x_i^{(j)} + w_{0,i}.$$

The cumulative excitation exceeds the cumulative inhibition by an amount called the threshold. Multi-Layer Perceptron (MLP) is a neural network composed of successive feedforward layers that connect neurons by weighted links[12, 14]. The input layer is used to collect the input signals and the output layer provides responses. One or more hidden layers are added for transfer. MLP learning is performed by the error gradient propagation algorithm[14] (see Algorithm 2). The examples of the training base are shown successively to adjust the weights of the network by accumulating the calculated gradients. The training is stopped when the calculated error is less than a certain threshold.

2.2.3 *K*-nereast neighbour

K-Nearest Neighbor (KNN) classification is a supervised learning algorithm that is mainly used when the attributes are continuous[4–6], but it can be also modified to work with categorical attributes. The idea is to estimate the classification of new data examples on the basis of instances used in the training phase (see Algorithm 3).

Algorithm 1 SVM algorithm

1: Find the hyperplane as a solution to a constrained optimization problem.
2: Introduce the search for nonlinear separating surfaces by using a kernel that encodes the nonlinear transformation of the data.
3: Derive the equations on the basis of some scalar products by using the kernel and some database weights.

Algorithm 2 Backpropagation training

```
1: Data Base Algorithm (DBA): Training base.
   $X_i = (x_{i,j})_{j=1,\ldots,m}$ : Inputs;
   $C_i = (c_{i,j})_{j=1,\ldots,m}$ : Desired results for $X_i$;
   $W_i = (w_{i,j})_{j=1,\ldots,m}$ : Weights for $X_i$;
   $\theta_i$ : Calculated results;
   $\lambda_i$ : Training rate.
2: Begin: Calculate $(W_i)_{i=1,\ldots,n}$ for inputs $(X_i)_{i=1,\ldots,n}$.
       for $i$ from 1 to $n$ do
         Initialize randomly the wights
           Optimization of wights
             For $j$ from 1 to $m$ do
               $w_{i,j} = w_{i,j} + \lambda_i(c_{i,j} - \theta_i)x_{i,j}$;
             End For
       End For
3: End
```

The study of different classification methods in Refs. [4, 8, 9, 15, 16] allows us to perform an evaluation and comparison of these existing methods on the basis of some criteria related to accuracy (see Table 2).

3 A Novel Binary Classifier for Anomaly Detection

This section describes the proposed solutions to validate our new model of a heterogeneous traffic classifier based on a new machine learning classifier.

Algorithm 3 KNN algorithm

1: Fix a value of K that can be an integer.
2: Calculate the distance values between the test data samples and the training data by using Euclidean, Manhattan, or Hamming distance methods.
3: Sort in ascending order the distance values calculated previously.
4: Choose the optimal K rows from the sorted array.
5: Assign a class to the test point on the basis of the most frequent class of these rows.

Table 2 Performance of studied classification methods.

Method	Training type	Classification type	Nature of data	Convergence	Accuracy	Algorithm goal
SVM	Supervised	Linear	Small size	Fast	Average	Find the best hyper plane separator.
MLP	Supervised	Linear	Large size	Slow	High	Minimize the error between result and desired output.
	Unsupervised	Nonlinear	Incomplete			
KNN	Supervised	Linear	Small size	Fast	Average	Predict the values of new data points.

3.1 Proposed model

The work aims to propose a robust algorithm for the training and recognition of the events collected and integrated from network traffic[3, 8]. The incorporation of a new machine learning method based on MLP is suggested. The proposed model is easy to implement, is applicable to unstructured data, and presents high accuracy.

For classification problems, the number of classes to which the inputs belong should be determined. Each class is set to receive an input. In this problem, we predict a binary value, and our classifier model answers the following question with a "yes" or "no": Is the event extracted from the network traffic an intrusive activity or not? Two databases are utilized: One is used for training, and the other is used for testing and validation. In practice, no predefined rule exists with regard to sharing a database quantitatively between training and validation.

The validation of such a model requires static data with supervised methods or structured data presented in sequence. Our architecture is a neural network consisting of three layers, each of which has neurons directly linked to the neurons of the next layer. No direct connection exists between the input layer and the output layer. We add a single hidden layer to achieve our goal. The collection of traffic activities is performed using network sniffing tools. Heterogeneity is mainly represented at the level of the supported protocol, data source, version, and size[16, 17]. In addition, a packet may contain missing or noisy values. The total size of a package cannot exceed 65 535 bytes.

Significant formatting is required before analyzing and classifying traffic[9]. Normalization is also conducted to establish patterns of activities that facilitate the distinction between activities and the extraction of useful fields if necessary. We realize a particular coding for the enumeration of occurrences and adapt them to the model. The number of features must be fixed in advance.

The input layer receives the preprocessed occurrences successively. An occurrence is subdivided into a set of fields. Each field is received by a neuron. A weighted sum is calculated on the input values. A transfer function is applied to the calculated sum. A sigmoid function is implemented in the hidden layer:

$$f(x) = \frac{1}{1+\mathrm{e}^{-x}} \text{ for all } x \in \mathbb{R}.$$

Beyond this modeling, resolution requires the implementation of algorithms that are adapted to the nature of the problem. The proposed model is mathematical structuring used to represent significant aspects.

3.2 Description of solutions

The new design defines a supervised method using a three-layer perceptron. The training basis is intercepted in a period when the network is offline. Over this long period, the base will be the most significant. The collection is shared horizontally between the training and testing bases (see Fig. 1).

- 80% to implement the model (training basis).
- 20% to evaluate the model performances (test basis).

We can encode both targets in numerical variables. A successful event target can take a value of +1, and an intrusion takes a value of −1.

At this point, we have an optimization model containing modifiable variables that describe the problem, together with constraints representing the limits on these variables. We define a function that assigns a value to each iteration of adjustment and thus optimizes the model's weights. The cost function for minimization is $1 - y(a_i)$ with $a_i = \sum_{j=1}^{m} w_{i,j} x_{i,j} + w_{0,i}$ for $i = 1, \ldots, n$.

We refer to the training phase to adjust the weights of

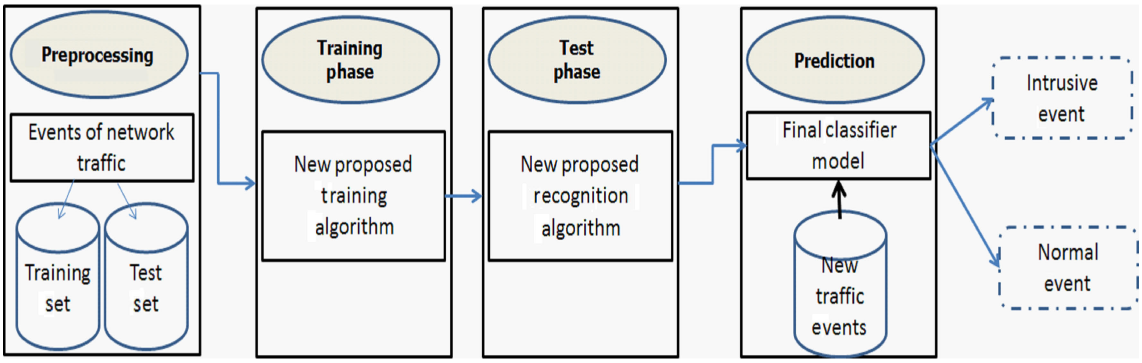

Fig. 1 Proposed classifier model.

the proposed model for reliable recognition.

Let $S=\{(X_i=(x_{i,j})_{j=1,\dots,m}, W_i^{(\text{op})}=(w_{i,j}^{(\text{op})})_{j=1,\dots,m}, w_{0,i}^{(\text{op})}, \varepsilon_i), i=1,\dots,n\}$ be the set of results obtained in the modeling of the training phase (Algorithm 4). ε_i is the calculated error tacked into account to ajust weights.

4 Mathematical Validation of Proposed Machine Learning Algorithm

Proposition 1: With the description of solutions mentioned in Section 3.2, we then have the following for all $i\in\{1,\dots,n\}$:

(1) $a_i^{(\max)}\geqslant a_i^{(\text{op})}$.

(2) $0<1-y(a_i^{(\max)})\leqslant\varepsilon_i$.

Proof 1:

(1) As $w_j^{(\max)}=\max\{w_{i,j}^{(\text{op})}, i=1,\dots,n\}\geqslant w_{i,j}^{(\text{op})}$ and $x_{i,j}\geqslant 0$ for all $i\in\{1,\dots,n\}$ and $j=1,\dots,m$, then $a_i^{(\max)}\geqslant a_i^{(\text{op})}$ for each $i\in\{1,\dots,n\}$.

(2) We have $1-y(a_i^{(\text{op})})=\varepsilon_i$ and $1>y(a_i^{(\text{op})})=f(a_i^{(\text{op})})>0$ for each $i\in\{1,\dots,n\}$.

Therefore, $0<y(a_i^{(\text{op})})=f(a_i^{(\text{op})})\leqslant f(a_i^{(\max)})=y(a_i^{(\max)})<1$ for each $i\in\{1,\dots,n\}$ bcecause f is an increasing function. Thereafter, $1-y(a_i^{(\max)})\leqslant 1-y(a_i^{(\text{op})})=\varepsilon_i$, which shows as Proof 1(2).

According to Proposition 1, we deduce the following definition to distinguish a normal occurrence of an intrusion for an input occurrence.

Definition: Let $K=(k_j)_{j=1,\dots,m}$ be an input occurrence and $a=\sum\limits_{j=1}^{m} w_j^{(\max)}k_j+w_0^{(\max)}$.

Algorithm 4 Training phase

```
Initialize the weights $W^{(0)}=(w_{i,j})_{j=1,\dots,m}$ such as $0\leqslant w_{i,0}\leqslant 10^{-3}$ and $w_{0,i}=1$ for $i=1,\dots,n$.
  For $i$ from 1 to $n$ do
1. Present the inputs: $X_i=(x_{i,j})_{j=1,\dots,m}$ and $a_i=\sum_{j=1}^{m} w_{i,j}x_{i,j}+w_{0,i}$;
2. Calculate $W_i^{(\text{op})}$, $\varepsilon_i\neq 0$ and $a_i^{(\text{op})}$ :
     For $k$ from 1 to $m$ do
          $\varepsilon_i=1-y(a_i)$;
       For $j$ from 1 to $m$ do
          $w_{i,j}=w_{i,j}+[1-y(a_i)]x_{i,j}$;
       End For
          $a_i=\sum_{j=1}^{m} w_{i,j}x_{i,j}+w_{0,i}$;
          $w_{0,i}=w_{0,i}+[1-y(a_i)]$;
     End For
3. End For
```

(1) K is a normal occurrence if there exists $i\in\{1,\dots,n\}$, such as $1-y(a)\leqslant\varepsilon_i$.

(2) K is an intrusion if $1-y(a)>\varepsilon_i$ for all $i\in\{1,\dots,n\}$.

Proposition 2: Let $K=(k_j)_{j=1,\dots,m}$ be an input occurrence and $a=\sum\limits_{j=1}^{m} w_j^{(\max)}k_j+w_0^{(\max)}$.

The following conditions are equivalent:

(1) K is normal information.

(2) $1-y\,(a)\leqslant\varepsilon$.

Proof 2:

(1) Proposition 2(1) ⇒ Proposition 2(2): According to the definition, there exists an $i\in\{1,\dots,n\}$, such as $1-y(a^{(\max)})\leqslant\varepsilon_i$. Thus, $1-y(a^{(\max)})\leqslant\varepsilon=\max\{\varepsilon_i, i=1,\dots,n\}$.

(2) Proposition 2(2) ⇒ Proposition 2(1): As $\varepsilon=\max\{\varepsilon_i, i=1,\dots,n\}$, then there exists an $i\in\{1,\dots,n\}$ such as $1-y(a^{(\max)})\leqslant\varepsilon_i$.

Corollary: Let $K=(k_j)_{j=1}^{m}$ be an input occurrence and $a=\sum\limits_{j=1}^{m} w_j^{(\max)}k_j+w_0^{(\max)}$. The following conditions are equivalent:

(1) K is an intrusion.

(2) $1-y(a)>\varepsilon$.

The proof of this corollary relies on definition and Proposition 2.

Propositions 1 and 2 allow us to model the training base through the following structure:

This optimization helps the new model to respond effectively by making fast and accurate detection decisions and by reducing the number of false positives and false negatives in current monitoring systems (see Algorithm 5).

Algorithm 5 Recognition phase

```
This phase involves validating our model by using the weights $(w_j^{(\max)})_{j=1,\dots,m}$ obtained in the training phase (Fig. 2).
New inputs $X=(x^{(j)})_{j=1,\dots,m}$, final output $d$, and activation state $a=\sum_{j=1}^{m} w_j^{(\max)}x^{(j)}+w_0^{(\max)}$;
            Classification of activities:
              if $(1-y(a)\leqslant\varepsilon)$ then
                   $d=1$ //Normal activity.
              else
                   $d=-1$ //Intrusion.
              End if
```

Obtained bias weight	Obtained weight	Caculated error
$W_0^{(\max)}$	$W^{(\max)}=(w_j^{(\max)})_{j=1}^{m}$	ε

Fig. 2 Trainning database structure.

5 Conclusion and Future Work

In this paper, we present a number of classification techniques used in supervised learning to solve different real problems. Most classifiers suffer from limitations due to the large dimensions and heterogeneity of data. The validation of the new classifier model based on the proposed machine learning algorithm is achieved on the basis of suggested solutions that guarantee an efficient and fast analysis. In our future work, we will design and validate an efficient intrusion detection system by using the proposed classifier to improve network monitoring and ensure accurate decision making.

References

[1] S. Y. Hao, J. Long, and Y. C. Yang, BL-IDS: Detecting web attacks using Bi-LSTM model based on deep learning, in *Security and Privacy in New Computing Environments*, J. Li, Z. L. Liu, and H. Peng, eds. Springer, 2019, pp. 551–563.

[2] Y. Zhou and P. C. Wang, An ensemble learning approach for XSS attack detection with domain knowledge and threat intelligence, *Comp. Secur.*, vol. 82, pp. 261–269, 2019.

[3] S. Rupam, A. Verma, and A. Singh, An approach to detect packets using packet sniffing, *Int. J. Comp. Sci. Eng. Surv.*, vol. 4, no. 3, pp. 21–25, 2013.

[4] L. Igual and S. Seguín, *Introduction to Data Science*: *A Python Approach to Concepts*, *Techniques and Applications*. Springer, 2017.

[5] O. K. Sahingoza, E. Buberb, O. Demirb, and B. Diri, Machine learning based phishing detection from URLs, *Expert Syst. Appl.*, vol. 117, pp. 345–357, 2019.

[6] S. Raschka and V. Mirjalili, *Python Machine Learning*. 2nd ed. Birmingham, UK: Packt Publishing, 2017.

[7] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, Machine learning: A review of classification and combining techniques, *Artif. Intell. Rew.*, vol. 26, no. 3, pp. 159–190, 2006.

[8] A. Guezzaz, A. Asimi, Y. Sadqi, Y. Asimi, and Z. Tbatou, A new hybrid network sniffer model based on pcap language and sockets (Pcapsocks), *Int. J. Adv. Comp. Sci. Appl.*, vol. 7, no. 2, pp. 207–214, 2016.

[9] A. Guezzaz, A. Asimi, Y. Asimi, Z. Tbatous, and Y. Sadqi, A global intrusion detection system using PcapSockS sniffer and multilayer perceptron classifier, *Int. J. Netw. Secur.*, vol. 21, no. 3, pp. 438–450, 2019.

[10] V. N. Vapnik, An overview of statistical learning theory, *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, 1999.

[11] F. Lauer and G. Bloch, Méthodes SVM pour l'identication, https://hal.archives-ouvertes.fr/file/index/docid/110344/filename/LauerBlochJIME06.pdf, 2006.

[12] M. Rochaa, P. Cortezb, and J. Nevesa, Evolution of neural networks for classification and regression, *Neurocomputing,* vol. 70, nos. 16–18, pp. 2809–2816, 2007.

[13] M. Idhammad, K. Afdel, and M. Belouch, Detection system of HTTP DDoS attacks in a cloud environment based on information theoretic entropy and random forest, *Hindawi Secur. Commun. Netw.,* vol. 2018, p. 1263123, 2018.

[14] A. Guezzaz, A. Asimi, M. Azrour, Z. Batou, and Y. Asimi, A multilayer perceptron classifier for monitoring network traffic, in *Big Data and Networks Technologies*, Y. Farhaoui, ed. Springer, 2020.

[15] Y. Farhaoui and A. Asimi, Performance method of assessment of the intrusion detection and prevention systems, *Int. J. Eng. Sci. Technol.*, vol. 3, no. 7, pp. 5916–5928, 2011.

[16] B. B. Yong, X. Liu, Q. C. Yu, L. Huang, and Q. G. Zhou, Malicious web traffic detection for internet of things environments, *Comp. Electr. Eng.*, vol. 77, pp. 260–272, 2019.

[17] M. ul-Hassan, M. A. Khan, K. Mahmood, and A. M. Shah. Analysis of IPv4 vs IPv6 traffic in US, *Int. J. Adv. Comp. Sci. Appl.*, vol. 7, no. 12, pp. 261–267, 2016.

Azidine Guezzaz received the MS degree in the field of computer science and distributed systems from Department of Mathematics and Computer Science, Faculty of Science, University Ibn Zohr, Agadir, Morocco in 2013. He received the PhD degree from Faculty of Science, University Ibn Zohr, Agadir, Morocco in 2018. He was a professor at the Technology High School and BTS in the period 2014–2018. He then joined Cadi Ayyad University in 2018 as an assistant professor. His main field of research interests are intrusion detection and prevention, computer and network security, and cryptography.

Younes Asimi received the PhD degree from Ibn Zohr University in 2015. He is currently an assistant professor in computer science at Ibn Zohr University since 2018. His research interests include authentication protocols, computer and network security, and cryptography.

Mourade Azrour received the PhD degree from Faculty of Sciences and Technologies, Moulay Ismail University, Errachidia, Morocco in 2019, and the MS degree in computer and distributed systems from Faculty of Sciences, Ibn Zouhr University, Agadir, Morocco in 2014. He currently works as a computer science professor at the Department of Computer Science, Faculty of Sciences and Technologies, Moulay Ismail University. His research interests include authentication protocol, computer security, Internet of Things, and smart systems. He is a scientific committee member of numerous international conferences. He is also a reviewer of various scientific journals, such as *International Journal of Cloud Computing* and *International Journal of Cyber-Security and Digital Forensics* (*IJCSDF*).

Ahmed Asimi received the PhD degree in number theory from the University Mohammed V Agdal in 2001. He is reviewer of *International Journal of Network Security* (*IJNS*). His research interest includes number theory, code theory, and computer cryptology and security. He is a full professor at the Faculty of Science Agadir, Ibn Zohr University, Morocco since 2008.