

Network intrusion detection in big datasets using Spark environment and incremental learning

Abdelwahed Elmoutaoukkil, Mohamed Hamlich, Amine Khatib, Marouane Chriss

Laboratory of Complex Cyber-Physical Systems, National School of Arts and Crafts, Casablanca, University Hassan II, Casablanca, Morocco

Article Info

Article history:

Received Nov 23, 2023

Revised Mar 24, 2024

Accepted Apr 17, 2024

Keywords:

Big data

Incremental learning

Internet of things

Intrusion detection system

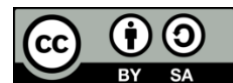
Machine learning

Streaming linear discriminant analysis

ABSTRACT

Internet of things (IoT) systems have experienced significant growth in data traffic, resulting in security and real-time processing issues. Intrusion detection systems (IDS) are currently an indispensable tool for self-protection against various attacks. However, IoT systems face serious challenges due to the functional diversity of attacks, resulting in detection methods with machine learning (ML) and limited static models generated by the linear discriminant analysis (LDA) algorithm. The process entails adjusting the model parameters in real time as new data arrives. This paper proposes a new method of an IDS based on the LDA algorithm with the incremental model. The model framework is trained and tested on the IoT intrusion dataset (UNSW-NB15) using the streaming linear discriminant analysis (SLDA) ML algorithm. Our approach increased model accuracy after each training, resulting in continuous model improvement. The comparison reveals that our dynamic model becomes more accurate after each batch and can detect new types of attacks.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Abdelwahed Elmoutaoukkil

Laboratory of Complex Cyber-Physical System, National School of Arts and Crafts Casablanca

University Hassan II

Casablanca, Morocco

Email: elmoutaoukkilabd@gmail.com

1. INTRODUCTION

An intrusion detection system (IDS) [1], [2] is an instrument or software application that surveys the network and the system for cynical activities and warns the system or network administrator. There are two kinds of IDS, network-based IDS, and host-based IDS. A host-based IDS keeps track of individual host machines and gives notification to the user if suspicious activities like deleting or modifying a system file, undesired configuration changes, or unnecessary sequence of system calls are detected [3]. Generally, a network-based intrusion detection system (NIDS) [4] is kept at network points like a gateway or routers to detect intrusions in the network traffic. Lately, artificial intelligence (AI) has been used in the field of cybersecurity, and to achieve high-performance IDS, machine learning (ML) techniques can be used for all types of detection techniques. ML is a subclass of AI used in computers with the ability to learn without being calculated.

ML algorithms can be supervised or unsupervised, the first category builds a mathematical model of a data set that contains both the desired inputs and outputs. The data is known as training data and consists of a set of training examples. Each training example has one or more inputs and outputs. The second category takes a set of data containing only inputs and finds a pattern in the data, such as grouping or clustering data points. The algorithms, therefore, learn from test data that has not been labeled, classified, or categorized [5].

In our research work, we have proposed a framework in which a feature reduction algorithm is used for eliminating the less important features. Then we applied supervised data mining techniques on the UNSW-NB15 network dataset for fast, efficient, and accurate detection of intrusion in the Netflow records by leveraging the power of Spark. In this paper, we have used principal component analysis (PCA) and streaming linear discriminant analysis (SLDA) algorithms to analyze the performance of the proposed framework. We focused on three performance indicators: accuracy, training time, and duration of prediction.

2. METHOD

Agrawal and Agrawal [6] have surveyed abnormality detection with data mining techniques to detect intrusions. They have classified the anomaly detection techniques with three features: clustering-based techniques, classification-based techniques, and hybrid techniques. Their comprehensive analysis sheds light on the diverse approaches utilized in the ongoing pursuit of enhancing cybersecurity measures.

Buczak and Guven [7] made a survey that describes the application of data mining and ML techniques to detect known and unknown attacks. They established a clear difference between ML and data mining. Their study underscores the growing significance of these approaches in preventing and detecting threats within the complex landscape of modern cybersecurity.

Haija *et al.* [8] have done a new inclusive discovery scheme that evaluates five supervised ML classifiers: logistic regression, decision trees, linear/quadratic discriminant, naïve Bayes, and ensemble boosted trees. Port scanning attacks involve attackers sending packets with various port numbers to scan for accessible services and identify open or weak ports in a network, necessitating the development of multiple detection and prevention techniques. Through performance comparison using the PSA-2017 dataset, the logistic regression model demonstrated superior results with 99.4% accuracy, 99.9% precision, 99.4% recall, 99.7% F-score, and a detection overhead of 0.454 μ Sec, highlighting its effectiveness and enhanced attack discovery speed compared to existing models.

Haija *et al.* [9] have introduced an advanced self-reliant system designed to detect mutations of internet of things (IoT) cyber-attacks using a deep convolutional neural network (CNN) and CUDA-based Nvidia-Quad GPUs for parallel computation. This innovative approach achieved exceptional attack classification accuracy, exceeding 99.3% for binary classifiers and 98.2% for multi-class classifiers. They highlight the substantial growth and impact of the IoT technology, while also addressing its vulnerability to cyber-attacks due to the limitations in computation, storage, and communication capacity of endpoint devices such as thermostats and home appliances.

Dhanya *et al.* [10] have surveyed various ML-based techniques applied to UNSW-NB15 dataset. They compared the naïve Bayes algorithm with proposed probability-based supervised ML algorithms using a reduced UNSW NB15 dataset. Their examination offers valuable insights into the efficacy of different ML approaches when applied to this dataset, contributing to the ongoing discourse on cybersecurity methodologies.

Alsulami *et al.* [11] have proposed a predictive ML model to detect and classify network activity in an IoT system. Specifically, the model distinguishes between normal and anomaly network activity. Furthermore, it classifies network traffic into five categories: normal, Mirai attack, denial of service (DoS) attack, Scan attack, and man-in-the-middle (MITM) attack. Five supervised learning models were implemented to characterize their performance in detecting and classifying network activities for IoT systems.

Haija [12] introduces an innovative and versatile top-down framework for enhancing intrusion detection and classification within IoT networks through the utilization of non-conventional ML techniques. The article puts forth a novel architecture that offers the flexibility to adapt and apply to intrusion detection and classification tasks involving various IoT cyber-attack datasets, such as the CICIDS dataset and MQTT dataset. More specifically, this newly proposed system comprises three distinct subsystems: the feature engineering (FE) subsystem, the feature learning (FL) subsystem, and the detection and classification (DC) subsystem.

Haija *et al.* [13] conducted a study that develops and evaluates machine-learning-based Darknet traffic detection systems (DTDS) in IoT networks, utilizing six supervised machine-learning techniques. Their research highlights the application of these techniques to address vulnerabilities in IoT infrastructures due to limited endpoint device capabilities. The study underscores the effectiveness of bagging ensemble techniques (BAG-DT) in achieving superior accuracy and lower error rates, demonstrating significant improvements over existing DTDS models by 1.9% to 27%.

The majority of researchers employing ML in the cybersecurity domain concentrate on assessing the performance of models generated using diverse ML algorithms. However, these performances, such as accuracy and prediction time, do not exhibit temporal variations with the emergence of new attacks, ultimately rendering these models outdated over time. This research underscores two significant aspects: firstly, the effectiveness of intrusion detection methods employing incremental algorithms, such as SLDA, in identifying

emerging attack patterns, and secondly, the progressive enhancement of their accuracy over time. The UNSW-NB15 dataset has been curated for experimentation within the Spark tool environment.

2.1. Description of UNSW-NB 15 dataset

The UNSW NB15 dataset has carved its place as a cornerstone in the field of network security research. Numerous researchers have harnessed the potential of this dataset to unravel the complexities of network behavior, intrusion detection [14], and the ever-evolving landscape of cyber threats. UNSW NB15 stands as a testament to the pivotal role that data plays in comprehending and safeguarding against cyber-attacks.

For the evaluation of the performance and effectiveness of our IDS, we required a comprehensive dataset that contains both normal and abnormal behaviors. A lot of research has been done using older benchmark data sets like KDDCUP 99 and NSLKDD but these datasets do not offer realistic output performance. The reason is that KDDCUP 99 has lots of redundant and missing records in the training set. So these datasets are not comprehensive representations of modern low footprint attack environments. UNSW-NB 15 dataset [15] was created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). It contains both real modern normal activities and synthetic contemporary attack behaviors [16]. UNSW-NB15 dataset is available in comma-separated values (CSV) file format. There are 175,341 records in the training set and 82,332 records in the testing set with all different 9 types of attacks and normal records. There are 49 attributes or features with 10 class values in this dataset. All records are divided into two major categories of the records - normal and attack. The attack categories as shown in Table 1 are again subdivided into 9 categories. Attack types are fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms [17].

Table 1. List of attacks UNSW NB-15 dataset

Category	Training set	Testing set
Normal	56000	37000
Generic	40000	18871
Exploits	33393	11132
Fuzzers	18184	6062
DoS	12264	4089
Reconnaissance	10491	3496
Analysis backdoor	2000	677
Backdoor	1746	583
ShellCode	1133	378
Worms	130	44
Total instances	175,341	82,332

2.2. System modeling

2.2.1. Intrusion detection systems architecture

Creating a general model for all IDS with a fixed accuracy is not a viable solution. Different users have different models and the model must be updated on the fly with terrain data. However, the training model consumes a lot of resources, which hinders the promotion and practicality of incremental models on resource-constrained devices. The proposed system allows changing the model parameters, as shown in Figure 1. When the data stream arrives, SLDA can infer each upcoming sample and update its parameters by exploiting the benefits of online learning.

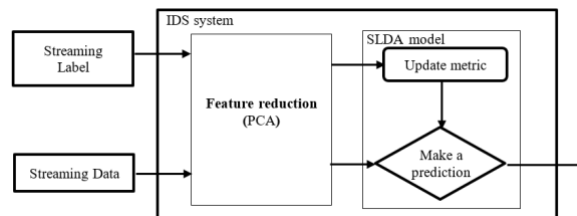


Figure 1. The building blocks of our IDS system

2.2.2. Model training process

Initially, we structured the database into separate training and testing segments. Subsequently, we implemented the PCA technique to reduce dimensionality. The training phase involved feeding our models with

data batches, mimicking a consistent data stream. This approach enabled us to monitor the accuracy progression post each training iteration. The proposed framework for online intrusion detection is shown in the Figure 2.

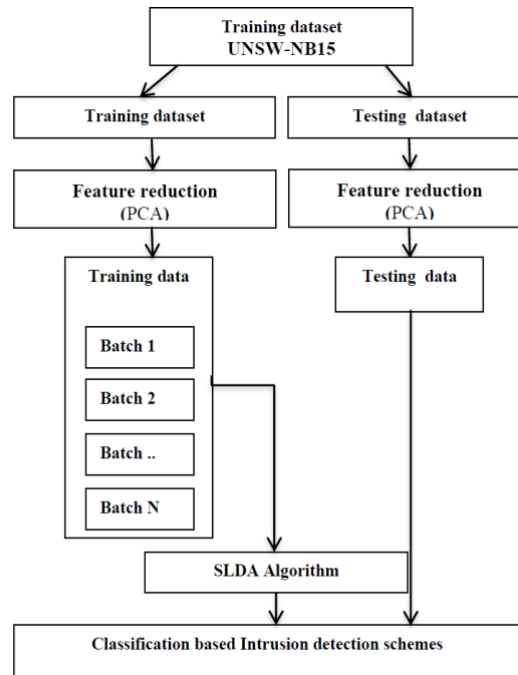


Figure 2. Proposed framework for intrusion detection

2.3. Streaming linear discriminant analysis algorithm

Linear discriminant analysis (LDA) is a widely embraced method in the fields of computer vision and pattern recognition due to its versatility in both dimensionality reduction and classification tasks. Numerous studies [18], [19] have demonstrated its effectiveness. LDA involves seeking a linear data transformation that optimally separates classes while reducing the dimensionality of the data [20].

The typical implementation of the LDA technique requires that all samples be available in advance [21]. However, there are situations where the complete data set is not available and the input data is observed as a flow. In this case, the LDA feature extraction should have the ability to update the computed LDA features by observing the new samples without running the algorithm on the entire data set. For example, in many real-time applications such as mobile robotics, online facial recognition, and IoT networks, it is important to update the extracted LDA features as soon as new observations are available.

Simply observing new samples is a streaming LDA algorithm, and this idea has been widely studied over the past two decades. Chatterjee and Roychowdhury [22] proposed an incremental self-organizing LDA algorithm to update LDA features. Demir and Ozmehmet [19] proposed local online learning algorithms to update LDA features incrementally using error correction and Hebbian learning rules. Later, Ghassabeh *et al.* [21] derived fast incremental algorithms to update LDA functionality by observing new samples. Let be the following prediction equation:

$$y_t = W \cdot z_t + b \quad (1)$$

Where $z_t \in \mathbb{R}^d$ is a vector, $Wz_t \in \mathbb{R}^k \times d$ and $b \in \mathbb{R}^k$ are updated online parameters, and k is the total number of classes.

SLDA stores one mean vector per class $\mu_k \in \mathbb{R}^d$ with an associated number $c_k \in \mathbb{R}$ and a single matrix of shared covariance $\Sigma \in \mathbb{R}^d \times d$. When a new data point (z_t, y) arrives, the average vector and the associated counter are updated as:

$$\mu_{(k=y,t+1)} = \frac{c_{(k=y,t)} + z_t}{c_{(k=y,t)} + 1} \quad (2)$$

$$c_{(k=y,t+1)} = c_{(k=y,t)} + 1 \quad (3)$$

with μ as the mean of the class y has the moment t and $c_{(k=y,t)}$ is the associated counter. For SLDA with an online variable variance, we use the following update:

$$\Sigma_{t+1} = \frac{t\Sigma_t + \Delta_t}{t+1} \quad (4)$$

Δ_t is calculated as:

$$\Delta_t = \frac{t(z_t - \mu_{(k=y,t)})(z_t - \mu_{(k=y,t)})^T}{t+1} \quad (5)$$

We use (1) and W_k to calculate the prediction. The columns of W is given as:

$$W_k = \Lambda \mu_k \quad (6)$$

with Λ are determined by:

$$\Lambda = [(1-\epsilon)\Sigma + \epsilon I]^{-1} \quad (7)$$

b_k is updated as:

$$b_k = -\frac{1}{2}(\mu_k \cdot \Lambda \mu_k) \quad (8)$$

2.4. Big data processing tools: Apache Spark

The Apache Spark environment is a robust tool in the realm of big data, offering an open-source distributed cluster-computing framework. It boasts a comprehensive ML library known as MLlib, specifically designed for ML classifiers. Apache Spark excels in in-memory processing and provides support for multiple programming languages, including Java, Scala, Python, SQL, and R, with particularly strong compatibility with Python.

2.4.1. Components of the Spark ecosystem:

The Spark ecosystem comprises several key components, including Spark core component, Spark SQL, Spark streaming, Spark MLlib, Spark GraphX, and SparkR. Each component serves a distinct purpose, facilitating different aspects of big data processing and analytics. In this specific work, we focus on using Spark MLlib, a scalable ML library that encompasses various ML algorithms. Spark MLlib's capabilities enable efficient implementation and execution of ML tasks on large datasets, highlighting its significance within the Spark ecosystem for advanced data analysis and predictive modeling [23], [24].

2.4.2. Features of Apache Spark:

Some notable features of Apache Spark include [25]:

- Rapid processing: Apache Spark boasts high-speed data processing capabilities, performing approximately 100 times faster in memory and 10 times faster on disk compared to traditional methods.
- Dynamic: it facilitates the development of parallel applications within Spark, thanks to the availability of around 80 high-level operators.
- In-memory computation: Apache Spark supports in-memory computation, enabling enhanced processing speeds by keeping data in memory for quicker access and analysis.

Overall, Apache Spark offers a powerful and versatile environment for processing large-scale data and performing ML tasks efficiently.

2.5. Performance of machine learning model

The performance of a ML model is a critical aspect that determines its effectiveness in making accurate predictions. Evaluating the performance involves a set of metrics that provide insights into how well the model can classify or predict data points. Key measures include accuracy, which indicates the proportion of correct predictions; sensitivity (or recall), which measures the ability to identify true positives; specificity, which assesses the identification of true negatives; precision, which evaluates the accuracy of positive predictions; and the F1 score, which balances precision and recall. These metrics collectively offer a comprehensive

understanding of the model's strengths and weaknesses, guiding improvements and ensuring the model's reliability in practical applications.

- To evaluate the accuracy or superiority of our classifiers in predicting the class labels of tuples, we use several key metrics. These classifier evaluation measures include accuracy, sensitivity (or recall), specificity, precision, and F1 score. The accuracy of a classifier on a given test set is determined by the percentage of true positives and true negatives out of all correctly classified instances.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Here, TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

- Training time: measurement of the time taken by the algorithm to train a new batch.
- Prediction time: measurement of time taken by a classifier to predict new data.

3. RESULTS AND DISCUSSION

Table 2 and Figure 3 illustrates the evolution of the accuracy of a machine-learning model across 12 training batches. The initial accuracy stands at 9.85% after the first batch. Subsequently, as the model is exposed to more training data and learning iterations, its accuracy gradually improves throughout these batches. It becomes evident that the model undergoes significant improvement.

The most remarkable aspect of this training process is the substantial leap in accuracy, culminating in an impressive rate of 96.37 % after the last (the 12th) batch. This indicates that the model has made significant progress in its ability to make accurate predictions as it learns from the data. When we compare the results of our approach to previous studies based on traditional learning techniques, IDS based on the progressive learning model showed superior adaptability and efficiency. Notably, the step-by-step approach achieved similar or higher accuracy levels while significantly reducing computational resources and training time.

Table 2. Performance evaluation using SLDA on dataset–1 and dataset–2 of (UNSW NB-15 dataset)

	Duration of training (Sec)	Duration of prediction (Sec)	Accuracy (%)
Batch 1	0.0779	0.0229	9.85
Batch 2	0.0929	0.0109	18.88
Batch 3	0.0829	0.0099	28.18
Batch 4	0.1159	0.0089	37.01
Batch 5	0.0849	0.0079	49.39
Batch 6	0.0829	0.0069	52.60
Batch 7	0.0849	0.0059	56.63
Batch 8	0.0829	0.0049	62.40
Batch 10	0.0969	0.0059	69.48
Batch 11	0.0889	0.002997	76.28
Batch 12	0.0879	0.002998	96.37

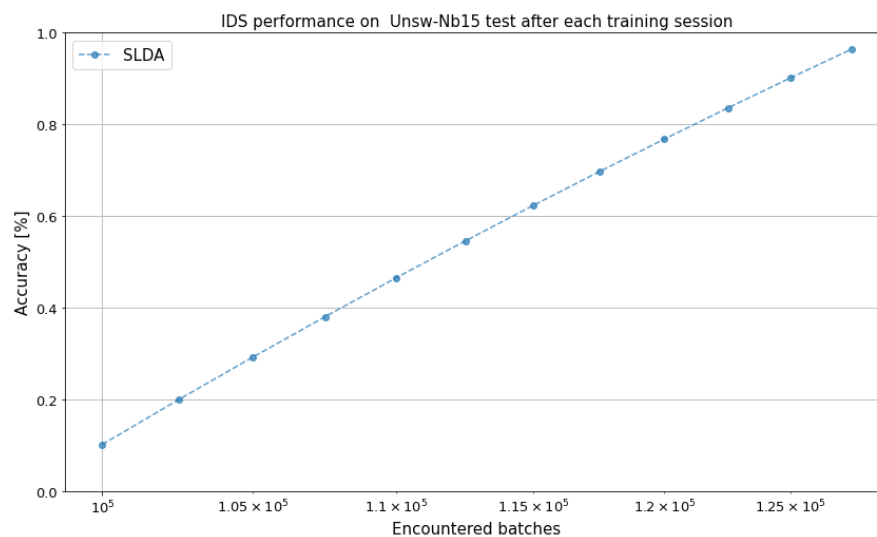


Figure 3. IDS performance on UNSW-NB15 test after each training session

4. CONCLUSION

The paper proposed a framework that was fast and effective for intrusion detection. We have used the UNSW NB-15 dataset for performance evaluation of the proposed framework by applying feature reduction using PCA and classification algorithm SLDA. It is found that using SLDA is more efficient compared to author static algorithms especially when the input data is observed as a flow as is the case in our subject where the precision of the model improves during the arrival of the new batch of data. It can be concluded that this approach is better, faster, and more efficient when used on Apache Spark.

5. RECOMMENDATIONS FOR FUTURE WORK

Several suggestions for future research endeavors could be explored to expand upon this study. These additional recommendations are outlined as follows: the proposed algorithm has the potential to be fine-tuned and utilized for various real-world applications that necessitate image recognition and classification. This includes domains such as medical imaging, biomedical analysis, and handwriting recognition applications. The proposed system could be integrated into an IoT device to offer intrusion detection services for IoT networks. Further investigation into this proposed IDS could delve into aspects such as power consumption, memory utilization, communication protocols, and computational complexity, particularly when implemented on low-power IoT nodes with small-scale system components, such as battery-operated or energy-efficient devices.




REFERENCES

- [1] K. Albulayhi, Q. A. A. -Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Applied Sciences*, vol. 12, no. 10, May 2022, doi: 10.3390/app12105015.
- [2] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "The architecture of a network level intrusion detection system," Department of Computer Science, University of New Mexico, Albuquerque, Mexico, Aug. 1990, doi: 10.2172/425295.
- [3] S. Axelsson, "Intrusion detection systems: a survey and taxonomy," *CiteSeerX*, pp. 1–27, 2000, doi: 10.1.1.1.6603.
- [4] G. Vigna and R. A. Kemmerer, "NetSTAT: A network-based intrusion detection system," *Journal of Computer Security*, vol. 7, no. 1, pp. 37–71, Jan. 1999, doi: 10.3233/JCS-1999-7103.
- [5] M. Azhari, A. Abarda, B. Ettaki, J. Zerouaoui, and M. Dakkon, "Higgs boson discovery using machine learning methods with Pyspark," *Procedia Computer Science*, vol. 170, pp. 1141–1146, 2020, doi: 10.1016/j.procs.2020.03.053.
- [6] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015, doi: 10.1016/j.procs.2015.08.220.
- [7] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.
- [8] Q. A. A. -Haija, E. Saleh, and M. Alnabhan, "Detecting port scan attacks using logistic regression," in *2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, Dec. 2021, pp. 1–5, doi: 10.1109/ISAECT53699.2021.9668562.
- [9] Q. A. A. -Haija, C. D. McCurry, and S. Z. -Sabatto, "Intelligent self-reliant cyber-attacks detection and classification system for IoT communication using deep convolutional neural network," *Selected Papers from the 12th International Networking Conference*, 2021, pp. 100–116, doi: 10.1007/978-3-030-64758-2_8.
- [10] K. A. Dhanya, S. Vajipayajula, K. Srinivasan, A. Tibrewal, T. S. Kumar, and T. G. Kumar, "Detection of network attacks using machine learning and deep learning models," *Procedia Computer Science*, vol. 218, pp. 57–66, 2023, doi: 10.1016/j.procs.2022.12.401.
- [11] A. A. Alsulami, Q. A. A. -Haija, A. Tayeb, and A. Alqahtani, "An intrusion detection and classification system for IoT traffic with improved data engineering," *Applied Sciences*, vol. 12, no. 23, Dec. 2022, doi: 10.3390/app122312336.
- [12] Q. A. A. -Haija, "Top-down machine learning-based architecture for cyberattacks identification and classification in IoT communication networks," *Frontiers in Big Data*, vol. 4, Jan. 2022, doi: 10.3389/fdata.2021.782902.
- [13] Q. A. A. -Haija, M. Krichen, and W. A. Elhaija, "Machine-learning-based darknet traffic detection system for IoT applications," *Electronics*, vol. 11, no. 4, Feb. 2022, doi: 10.3390/electronics11040556.
- [14] M. Souhail et al., "Network based intrusion detection using the UNSW-NB15 dataset," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 477–487, Jan. 2019, doi: 10.12785/ijcds/080505.
- [15] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.
- [16] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
- [17] A. R. Sonule, M. Kalla, A. Jain, and D. S. Chouhan, "UNSW-NB15 dataset and machine learning based intrusion detection systems," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 3, pp. 2638–2648, Feb. 2020, doi: 10.35940/ijeat.C5809.029320.
- [18] L. Jin, K. Ding, and Z. Huang, "Incremental learning of LDA model for Chinese writer adaptation," *Neurocomputing*, vol. 73, no. 10–12, pp. 1614–1623, Jun. 2010, doi: 10.1016/j.neucom.2009.11.039.
- [19] G. K. Demir and K. Oz Mehmet, "Online local learning algorithms for linear discriminant analysis," *Pattern Recognition Letters*, vol. 26, no. 4, pp. 421–431, Mar. 2005, doi: 10.1016/j.patrec.2004.08.005.
- [20] H. A. Moghaddam and K. A. Zadeh, "Fast adaptive algorithms and networks for class-separability features," *Pattern Recognition*, vol. 36, no. 8, pp. 1695–1702, Aug. 2003, doi: 10.1016/S0031-3203(03)00006-2.




- [21] Y. A. Ghassabeh, F. Rudzicz, and H. A. Moghaddam, "Fast incremental LDA feature extraction," *Pattern Recognition*, vol. 48, no. 6, pp. 1999–2012, Jun. 2015, doi: 10.1016/j.patcog.2014.12.012.
- [22] C. Chatterjee and V. P. Roychowdhury, "On self-organizing algorithms and networks for class-separability features," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 663–678, May 1997, doi: 10.1109/72.572105.
- [23] X. Meng *et al.*, "Mllib: machine learning in apache spark," *Journal of Machine Learning Research*, vol. 17, no. 34, pp. 1-7, 2016.
- [24] M. Assefi, E. Behraves, G. Liu, and A. P. Tafti, "Big data machine learning using apache spark MLlib," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec. 2017, pp. 3492–3498, doi: 10.1109/BigData.2017.8258338.
- [25] M. Armbrust *et al.*, "Spark SQL: Relational data processing in spark," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, vol. 2015, pp. 1383–1394, 2015, doi: 10.1145/2723372.2742797.

BIOGRAPHIES OF AUTHORS






Abdelwahed Elmoutaoukkil    is a lecturer in electrical engineering at BTS Elkhawarizmi Casablanca, Morocco. He obtained his Master's degree in Big Data and the Internet of Things at Hassan 2 University, Casablanca, Morocco. In 2021, he has been a computer science teacher in high school since 2008. He is currently a writer and researcher on artificial intelligence applied in industry. His research interests include tinyML technologies, continuous learning, industrial electronics, robotics, home automation, predictive maintenance, and STEAM models in the field of education. He can be contacted at email: elmoutaoukkilabd@gmail.com.






Mohamed Hamlich    is the Director of the "Complex Cyber-Physical Systems" Research Laboratory at ENSAM Casablanca (UH2C). He obtained his doctoral thesis in Computer Science from Hassan II University in Casablanca. His areas of research interest include robotics, artificial intelligence, IoT, and big data. He is the president of the Association of Connected Objects and Intelligent Systems and the chair of The International Conference "SADASC". He is the editor of two SPRINGER books and the author of several papers published in indexed journals. He can be contacted at email: moha.hamlich@gmail.com.



Amine Khatib    is a scientific researcher at Littoral Côte d'Opale University, France. He obtained his bachelor of engineering degree in automatic at Hassan 2 University, Casablanca, Morocco. His research interests include robotics and data science. He can be contacted at email: aminekhatib04@gmail.com.



Marouane Chriss    is a teacher of electrical engineering at BTS Elkhawarizmi Casablanca, Morocco. In 2002 he obtained a Diplôme d'Etudes Supérieures Approfondies (DESA) in Electrical Engineering from ENSEM, Hassan 2 University, Casablanca, Morocco. He is currently a Ph.D. student at LCCPS Laboratory, ENSAM, Hassan 2 University. His research interests include artificial intelligence, robotics, and systems control. He can be contacted at email: chriss_marouane@yahoo.fr.