International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 3, June 2024, pp. 2908~2917
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i3.pp2908-2917

Intelligent intrusion detection through deep autoencoder and stacked long short-term memory

Mehdi Moukhafi[1], Mouad Tantaoui[1], Idriss Chana[2], Aziz Bouazi[3]
[1]IEVIA Team, IMAGE Laboratory, Department of Sciences, Ecole Normale Supérieure, Moulay Ismail University of Meknes, Meknes, Morocco
[2]SCIAM Team, IMAGE Laboratory, ESTM, Moulay Ismail University of Meknes, Meknes, Morocco
[3]IEVIA Team, IMAGE Laboratory, ESTM, Moulay Ismail University of Meknes, Meknes, Morocco

Article Info

Article history:

Received Sep 5, 2023
Revised Jan 3, 2024
Accepted Jan 9, 2024

Keywords:

Deep autoencoder
Deep learning
Features extraction
Intrusion detection system
Stacked long short-term memory
UNSW-NB15

ABSTRACT

In the realm of network intrusion detection, the escalating complexity and diversity of cyber threats necessitate innovative approaches to enhance detection accuracy. This study introduces an integrated solution leveraging deep learning techniques for improved intrusion detection. The proposed framework consists on a deep autoencoder for feature extraction, and a stacked long short-term memory (LSTM) network ensemble for classification. The deep autoencoder compresses raw network data, extracting salient features and mitigating noise. Subsequently, the stacked LSTM ensemble captures intricate temporal dependencies, correcting anomaly detection precision. Experiments conducted on the UNSW-NB15 dataset, and a benchmark in intrusion detection validate the effectiveness of the approach. The solution achieves an accuracy of 90.59%, with precision, recall, and F1-Score metrics reaching 90.65, 90.59, and 90.57, respectively. Notably, the framework outperforms standalone models and demonstrates the advantage of synergizing deep autoencoder-driven feature extraction with the stacked LSTM ensemble. Furthermore, a binary classification experiment attains an accuracy of about 90.59%, surpassing the multiclass classification and affirming the model's potential for binary threat identification. Comparative analyses highlight the pivotal role of feature extraction, while experimentation illustrates the enhancement achieved by incorporating the synergistic deep autoencoder-Stacked LSTM approach.

Corresponding Author:

Mehdi Moukhafi
IEVIA Team, IMAGE Laboratory, Department of Sciences, Ecole Normale Supérieure, Moulay Ismail University of Meknes
Meknes, Morocco
Email: Mehdi.moukhafi@gmail.com

1. INTRODUCTION

The burgeoning increase in cyberattacks over the past few years has accentuated the exigency for pioneering and effective methodologies to fortify our computer systems and preserve classified data [1], [2]. Artificial intelligence (AI) has surfaced as a promising implement in the crusade against cybercrime, with its unparalleled ability to scrutinize colossal volumes of data, spot anomalies, and respond to looming threats promptly [3]. By utilizing machine learning algorithms, AI-based security systems can incessantly oversee network activities, recognize latent susceptibilities, and take preemptive measures to prevent malicious intrusions [4]. The growing interest in machine learning for intrusion detection arises from its superiority over traditional signature-based systems [5]. Machine learning algorithms can identify malicious activity by

analyzing network traffic patterns, reducing false positives and enhancing accuracy. However, their effectiveness relies on data quality and quantity. The author Kasongo [6] introduces an Intrusion Detection System framework utilizing machine learning and recurrent neural networks to enhance network security. Leveraging an extreme gradient boosting-based feature selection algorithm, the framework demonstrates superior performance, achieving optimal accuracy and efficiency in binary and multiclass classification tasks on benchmark datasets. Another study done by Wei *et al*. [7] employed a neural network with improved feature selection and multi-objective immune algorithm but faced challenges in detecting various attacks within the University of New South Wales-Network based-15 (UNSW-NB15) dataset despite success in NSL-KDD (network-based intrusion detection system (IDS) evaluation dataset – KDD).

In the work by Mebawondu *et al*. [8], the effectiveness of gain ratio (GR) was explored as a means of feature selection in conjunction with a multi-layer neural network for training the model. This approach was tested on the UNSW-NB15 dataset, and through the selection process, a total of 30 features were identified. Notably, the proposed system yielded an accuracy rate of 76.96%, which is indicative of the complexity of the dataset and its applicability within network intrusion detection systems (NIDS). However, it should be emphasized that the achieved accuracy rate is still considered low and requires further refinement and improvement. Alzaqebah *et al*. [9] have introduced a new bio-inspired meta-heuristic algorithm, which demonstrates a high degree of effectiveness in detecting and classifying multi-stage attacks. The proposed algorithm utilizes a one-versus-all sub-model based approach that addresses the multi-class classification challenge. Each sub-model in the proposed hierarchy employs an enhanced Harris Hawk optimization method with extreme learning machine (ELM) serving as the fundamental classifier. The work by Acharya and Singh [10] presented a new strategy to improve the performance of IDS that uses the intelligence water drops (IWD) algorithm. This algorithm is based on natural phenomena and starts by creating a graph that includes a collection of nodes and edges representing the search space. Then, the algorithm initializes a series of paths across the graph and employs these paths to form the feature subset, afterward, each subset is assessed using the support vector machine (SVM) classifier.

Alzubi *et al*. [11] proposes modifications to the binary grey wolf optimizer (GWO) for feature selection in multi-attack classification using SVM, yielding promising outcomes on the NSL-KDD dataset. While traditional machine learning has achieved significant progress in automating anomaly detection [12], [13], deep learning pushes boundaries by offering a unique capacity to extract intricate features from raw data [14]. Unlike conventional methods that require manual feature engineering, deep learning can automatically uncover and represent subtle patterns and relationships present within the data [15], [16]. This end-to-end feature extraction capability proves particularly advantageous in an environment where threats and attacks evolve swiftly. In this work, we present a groundbreaking approach aimed at fortifying intrusion detection capabilities by seamlessly integrating state-of-the-art deep learning techniques. The core concept behind our methodology is to initiate the process by leveraging the potential of deep autoencoders. These autoencoders play a pivotal role in extracting both salient and latent features from raw network data. By employing this strategy, our approach seeks to enhance the sophistication of intrusion detection systems, paving the way for more robust and effective cybersecurity measures. Employing deep autoencoders as the initial phase of our approach lets us address a critical preprocessing challenge. Indeed Raw network data is voluminous and noisy, presenting a significant hurdle for effective intrusion detection; fortunately, deep autoencoders tackle this issue by learning to represent the data in a compressed format, focusing on the essential elements while discarding redundant or noisy information. This process inherently identifies the latent features that are most relevant for distinguishing normal network behavior from potentially malicious activities.

Furthermore, the deep autoencoder phase establishes a foundation for the subsequent classification step. Therefore, by presenting the long short-term memory-based classifier with distilled and enriched features, we enhance the model's ability to discern intricate patterns and anomalies in network activity sequences. This strategic approach boosts detection accuracy and contributes to a more efficient utilization of computational resources, as the classifier now operates on a refined feature space. The rest of this paper is organized as follows: section 2 presents the proposed method, where we delve into the details of our approach that combines deep autoencoders and stacked long short-term memory (LSTM) networks for intrusion detection. We outline the architecture, explain the process of feature extraction, and elaborate on the network's configuration. In section 3, results and analysis, we present the outcomes of our experimentation and discuss the performance metrics achieved by our model.

2. PROPOSED METHOD

This section outlines our proposed malware detection framework and details the methodologies employed. As shown in Figure 1, the framework comprises three key steps. The initial step involves preprocessing of the UNSW-NB15 dataset, which includes essential data cleaning, label encoding, and standard scaling procedures to ensure data readiness and alignment with the ensuing methodologies.

Subsequently, in the second step, we leverage a deep autoencoder to distill essential attributes from the preprocessed raw network data. Trained to compactly represent input data, the deep autoencoder accentuates vital patterns while diminishing noise, ensuring subsequent analysis benefits from a focused feature set. The third step involves employing a stacked LSTM network ensemble for intrusion classification.

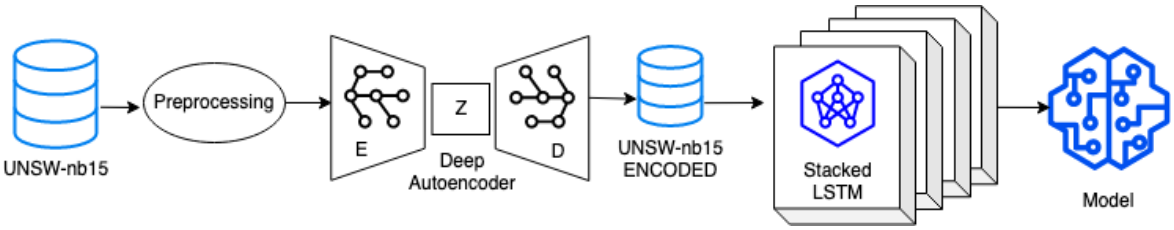


Figure 1. Proposed system block diagram

2.1. Dataset description

In this study, the analyses are conducted on the well-established UNSW-NB15 dataset [17], a widely recognized as a benchmark in the field of intrusion detection. The dataset was generated using the NIDS to simulate a real-world network environment. It encompasses various network activities, both benign and malicious, and provides a comprehensive representation of the complex behaviors encountered in actual networks. The UNSW-NB15 dataset is composed of a diverse range of attributes, totaling 49 features organized into seven major categories. These categories encapsulate attributes associated with basic flow features, content features, and time-based features. Each category contributes distinct information to the characterization of network traffic.

This comprehensive dataset provides a rich source of information, enabling a nuanced analysis of network behavior. The inclusion of various categories ensures a holistic understanding of network traffic patterns, encompassing both fundamental flow characteristics and more intricate content and time-related attributes. Such diversity within the dataset enhances its utility in training and evaluating intrusion detection systems. The training subset, UNSW-NB15_train, consists of 175,341 instances, while the testing subset, UNSW-NB15_test, contains 82,332 instances. This division ensures a realistic assessment of our approach's generalization capabilities on unseen data. This distribution underscores the complexity of the dataset, encompassing a diverse range of attack scenarios that enables a thorough evaluation of intrusion detection techniques.

2.2. Preprocessing

In preparation for subsequent analysis, we executed essential preprocessing steps to enhance the dataset's suitability. Two critical procedures, namely label encoding and standard scaling, were employed to ensure data readiness and alignment with the forthcoming methodologies. Label encoding was applied to transform categorical variables into numerical representations, a crucial step for machine learning algorithms that require numerical input. This conversion allows for a seamless integration of categorical information into our analytical framework.

In the context of the UNSW-NB15 dataset, the challenge lies in effectively integrating both numerical and categorical attributes into data mining algorithms, particularly neural networks. This amalgamation can hinder cohesive analysis due to inherent differences between data types. Algorithms requiring numerical inputs encounter difficulties when processing nominal attributes. To address this, a vital preprocessing step transforms categorical attributes like "proto", "service", "state", and "label" into numeric values using *LabelEncoder* [18]. This conversion enables algorithms to navigate these attributes effectively. By employing *LabelEncoder* on specific categorical features within the UNSW-NB15 dataset, a seamless interface is established, facilitating compatibility with algorithms designed for numeric inputs.

Within our exploration of the UNSW-NB15 dataset, a key preprocessing phase focused on normalization. This critical step facilitated the transformation of feature values, aligning them with a standardized scale. Employing the widely recognized *StandardScaler* technique [19], lets us to join the assumption of a standard normal distribution with a mean of 0 and a deviation of 1. This strategic approach contributes significantly to model convergence and the acceleration of the training process. Mathematically, the *StandardScaler* normalization process for a feature (X) is given by (1):

$$X_{\text{normalized}} = \frac{X-\mu}{\sigma} \quad (1)$$

where $\mu$is the mean $\frac{1}{N}\sum_{i=1}^{N} X_i$ and σ is the standard deviation $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i-\mu)^2}$ of the feature values and N represents the number of data points in the feature.

2.3. Features extraction with autoencoder

One of the key reasons for utilizing the autoencoder for feature extraction in the UNSW-NB15 dataset is its capacity to reduce data dimensionality [20]. By learning to represent data succinctly, the autoencoder enables the compression of information while retaining crucial features for intrusion detection. This helps to improve efficiency of the detection process, diminishing resource requirements, and expediting computations, while maintaining a high detection performance. Furthermore, the autoencoder is adept at handling noisy and missing data present in the UNSW-NB15 dataset. By learning robust data representation, the autoencoder can reconstruct network connections despite imperfections in the data, enhancing system reliability and resilience by reducing the risk of false positives and false negatives.

The architecture of the deep autoencoder [21] comprises 10 encoding layers and 10 decoding layers. Each encoding layer takes as input the output from the previous layer, enabling the learning of increasingly intricate features as information progresses through the network. Similarly, the decoding layers progressively reconstruct the data from the latent code, yielding high-quality outputs. The input data is represented by the vector $x \in R^{m_1}$, where m_1 is the number of dimensions (features) of the input. For each encoding layer ($k \in [1, 10]$), the computations are defined by (2) and (3):

$$e^{(k)} = W^{(k)}h^{(k-1)} + b^{(k)} \quad (2)$$

$$h^{(k)} = \sigma_k\left(e^{(k)}\right) \quad (3)$$

where $h^{(k)}$ is the output vector of the hidden layer (k); $W^{(k)} \in R^{m_k \times m_{k-1}}$ is the weight matrix of hidden layer k, where m_k is the dimension of layer k; $b^{(k)} \in R^{m_k}$ is the bias vector of hidden layer k; σ_k is the activation function of hidden layer k.

Within the deep autoencoder architecture, the output of the final hidden layer, denoted as $h^{(10)}$, serves as the latent code. This layer is alternatively referred to as the latent code output layer or the bottleneck layer. The latent code plays a pivotal role in encapsulating a compressed representation of the input data, effectively capturing abstract and essential features. By utilizing the output of the final hidden layer as the latent code, our deep autoencoder acts as a feature extractor, distilling intricate patterns from the raw input. This compressed representation is characterized by its ability to retain critical information while discarding less relevant details, thereby creating a focused and efficient encoding of the input data. The 10 decoding layers perform the inverse operations of the encoding layers to reconstruct the data from the latent code. For each decoding layer $k \in [11, 20]$, the computations are defined by (4).

$$d^{(k)} = W^{(k)}h^{(k-1)} + b^{(k)} \quad (4)$$

The final output layer $\hat{y}$ produces the ultimate reconstruction of the input data from the output of the last decoding layer $h^{(20)}$. For the training of the deep autoencoder, a loss function is employed to evaluate the disparity between the input data x and the reconstructed output data $\hat{y}$. In this article, we utilize the mean squared error (MSE) as the chosen loss function. The objective of the training is to minimize this loss function by adjusting the weights and biases of the deep autoencoder, thereby enabling the generation of optimal and compressed representations of the input data.

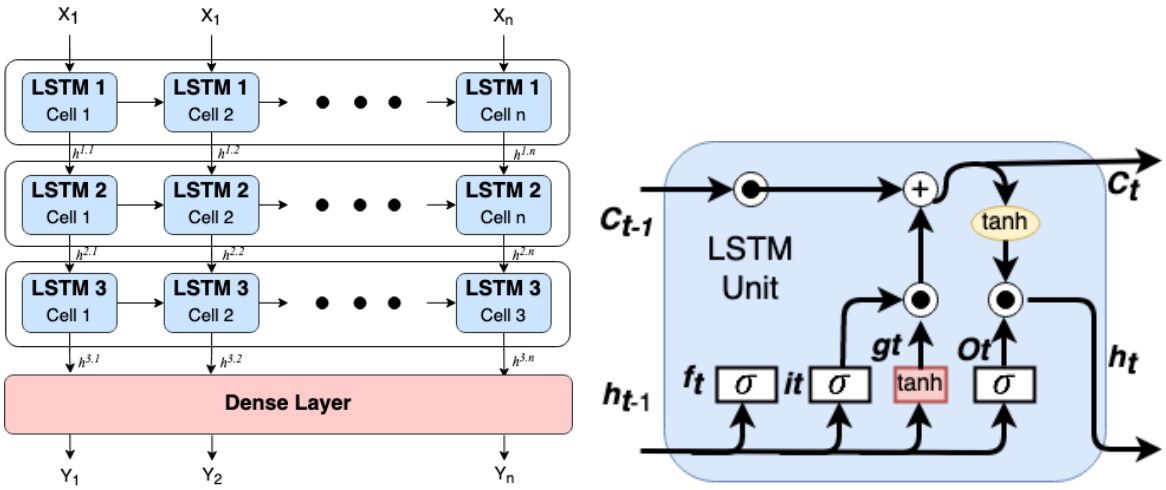
2.4. Architecture of stacked LSTM networks

Classification is a pivotal aspect in intrusion detection, and stacked LSTM networks provide a relevant approach. This architecture extends classical LSTM neural networks [22] by stacking multiple LSTM layers [23], enabling the learning of intricate hierarchical representations of sequential data. Each LSTM layer in this stacked network in Figure 2 takes the outputs of the preceding layer as input [24], producing its own hidden outputs and thus facilitating the gradual learning of more sophisticated abstractions.

Classification plays a pivotal role in the realm of intrusion detection, and one promising approach is the utilization of stacked LSTM networks. This architectural paradigm represents an extension of the conventional LSTM neural networks [22], achieved through the stacking of multiple LSTM layers [23]. This

innovation empowers the model to grasp intricate hierarchical representations within sequential data. The architectural configuration, depicted in Figure 2(a), demonstrates how each stratum of LSTM operates. Notably, the outputs from the previous layer serve as the input for each subsequent LSTM layer [24]. Consequently, every layer generates its distinctive hidden outputs, thereby facilitating the incremental acquisition of more sophisticated abstractions.

Furthermore, it is crucial to delve into the internal structure of each LSTM cell, as illustrated in Figure 2(b). The core of each LSTM cell encompasses a set of interconnected components that bestow the network with its unique capabilities. Within this intricate structure, key elements such as the input gate, forget gate, and output gate operate in tandem to regulate the flow of information. The input gate governs the incorporation of new information, the forget gate manages the retention or removal of existing information, and the output gate oversees the information to be transmitted to the next time step. This orchestrated interplay allows LSTM cells to effectively capture and utilize sequential dependencies in the input data.



(a) (b)

Figure 2. Integrated architecture of (a) stacked LSTM network and (b) internal cell structure

The proposed architecture commences with an input LSTM layer. This layer receives data encoded by the deep autoencoder, enabling the capture of relevant features and dimensionality reduction of the data. Following this, two hidden LSTM layers are stacked, facilitating the learning of intricate and hierarchical dependencies within temporal sequences. Each LSTM layer processes the outputs of the preceding layer and generates its own hidden outputs. The equations above describe the operation of stacked LSTM units in a recurrent network. Each unit (l, u) at time 't' calculates different gates and states to handle information based on the layer index l and the unit index u. The forget gate (5) $f_t^{(l,u)}$ determines how much previous information to forget, the input gate (6) $i_t^{(l,u)}$ decides on new information to add, and the candidate update (7) $g_t^{(l,u)}$ represents potential new information using the hyperbolic tangent function (tanh). The output gate (8) $o_t^{(l,u)}$controls the amount of information to output using the sigmoid function σ. By using these gates, the cell state (9) $c_t^{(l,u)}$ is updated, combining prior information with new information. Finally, the hidden output (10) $h_t^{(l,u)}$ is calculated, utilizing the output gate to control the amount of information to be output. These equations illustrate how stacked LSTM units interact to capture long-term dependencies in sequential data, considering the specific indices of the layer l and the unit u. For the uppermost layer $l = N, x_t$ typically corresponds to the actual data sequence, while for intermediate layers $1 < l < N, x_t$ represents the hidden output of the preceding layer $h_t^{(l-1)}$:

$$f_t^{(l,u)} = \sigma\left(W_f^{(l,u)} \cdot \left[h_{t-1}^{(l,u)}, x_t\right] + b_f^{(l,u)}\right) \quad (5)$$

$$i_t^{(l,u)} = \sigma\left(W_i^{(l,u)} \cdot \left[h_{t-1}^{(l,u)}, x_t\right] + b_i^{(l,u)}\right) \quad (6)$$

$$g_t^{(l,u)} = tanh\left(W_g^{(l,u)} \cdot \left[h_{t-1}^{(l,u)}, x_t\right] + b_g^{(l,u)}\right) \quad (7)$$

$$o_t^{(l,u)} = \sigma\left(W_o^{(l,u)} \cdot \left[h_{t-1}^{(l,u)}, x_t\right] + b_o^{(l,u)}\right) \quad (8)$$

$$c_t^{(l,u)} = f_t^{(l,u)} \cdot c_{t-1}^{(l,u)} + i_t^{(l,u)} \cdot g_t^{(l,u)} \quad (9)$$

$$h_t^{(l,u)} = o_t^{(l,u)} \cdot tanh\left(c_t^{(l,u)}\right) \quad (10)$$

Stacked LSTM networks offer enhanced modeling capabilities by virtue of their capacity to learn hierarchical representations. The key strength of stacked LSTMs lies in their ability to capture abstract features and temporal dependencies at different scales. This is achieved by stacking multiple LSTM layers on top of each other, creating a hierarchical structure that allows the network to learn intricate patterns within sequential data. By introducing multiple layers, each subsequent layer can build upon the representations learned by the preceding layers, enabling the network to discern increasingly complex features. This hierarchical learning approach makes stacked LSTMs well-suited for addressing the challenges of complex sequence processing tasks, where understanding long-range dependencies and extracting high-level abstractions are crucial. Following the LSTM layers, a dense layer with a softmax activation function [25] is added to generate. probabilities for each intrusion class, enabling multi-class classification. The softmax activation ensures that the probabilities lie within the range [0, 1], and their sum is equal to 1, facilitating the interpretation of classification outcomes. Mathematically, the output of the dense layer can be computed as (11):

$$P(y_i|h) = \frac{e^{h_i}}{\sum_{j=1}^{C} e^{h_j}} \quad (11)$$

where $P(y_i|h)$ is the probability of class i given the input h, e^{h_i} is the exponential of the i-th element of h, and C is the total number of classes. In summary, our stacked LSTM architecture leverages the advantages of LSTMs to capture intricate temporal patterns, integrating data previously encoded by the deep autoencoder. This approach enhances intrusion detection accuracy and presents a promising method to tackle challenges in computer security systems.

3. RESULTS AND DISCUSSION

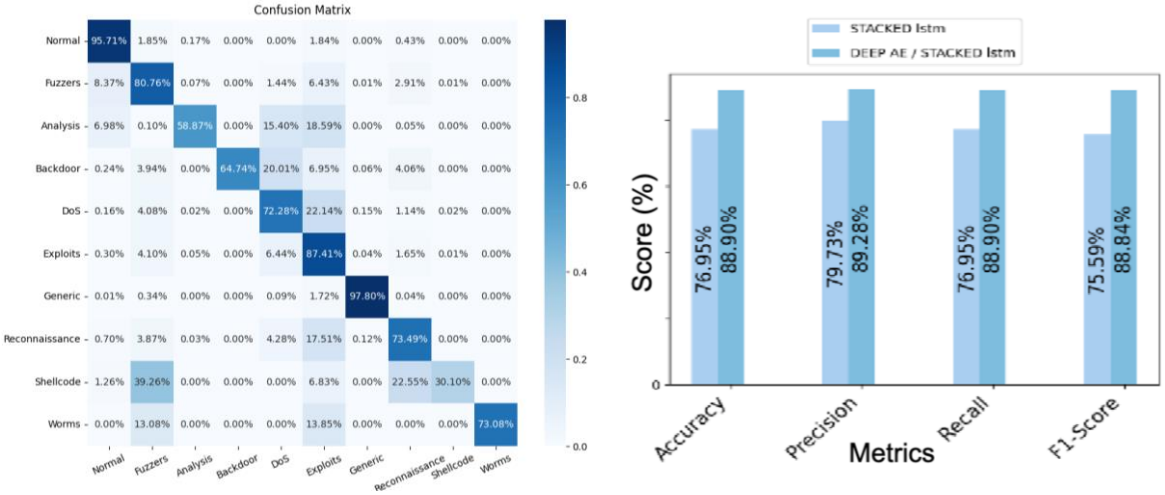
In assessing the performance of machine learning and deep learning models for network intrusion detection, the current study focuses on employing a set of metrics to evaluate their effectiveness. These metrics provide an objective measure of each model's performance, enabling a rigorous comparison among different approaches. Table 1 summarizes these metrics, providing a clear overview of the criteria used to evaluate the models' performance.

Table 1. Performance metrics of the IDS machine learning experiment

Metric	Formula		Explanation
Accuracy	$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$	(12)	Proportion of correctly classified instances out of the total instances.
Precision	$\text{Precision} = \frac{TP}{TP+FP}$	(13)	Proportion of true positive predictions among all positive predictions made by the model.
Recall (Sensitivity)	$\text{Recall} = \frac{TP}{TP+FN}$	(14)	Proportion of true positive predictions among all actual positive instances.
F1-Score	$\text{F1-Score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$	(15)	Harmonic mean of precision and recall, providing a balance between the two metrics.

Where TP is true positives, TN is true negatives, FP is false positives, FN is false negatives. The subsequent phase of our investigation delves into a comprehensive comparison between two pivotal models, thereby further illuminating the efficacy of our approach. As previously demonstrated, the integrated methodology proposed, which combines deep autoencoders for feature extraction and stacked LSTM networks for classification, has showcased remarkable success in the domain of network intrusion detection. Our approach has yielded an impressive accuracy of 88.90%, marking a substantial leap in distinguishing between normal and malicious network activities. Moreover, the precision, recall, and F1-Scores attained by

the proposed model stand at 89.28%, 88.90%, and 88.84%, respectively, underscoring the model's capability in accurately classifying network behaviors. To provide a more holistic perspective on our model's performance, we invite readers to refer to Figure 3 to view the performance of the proposed solution's multiclass classification, Figure 3(a) presents the multiclass confusion matrix derived from the assessment of the UNSW-NB15 dataset. In a parallel analysis, Figure 3(b) portrays a bar graph, shedding light on a head-to-head comparison between the deep autoencoder/stacked LSTM model and the standalone stacked LSTM model. This visual representation juxtaposes the two models' performance across essential metrics, including accuracy, precision, recall, and F1-Score. The graph vividly illustrates the tangible enhancements realized by our deep autoencoder-enhanced approach, showcasing its superior performance in multiple dimensions.



(a) (b)

Figure 3. Evaluation of multiclass performance (a) confusion matrix and (b) displays the comparison between a stacked LSTM model and a deep AE stacked LSTM model

Overall, these comparative analyses serve to emphasize the transformative impact of integrating deep autoencoders for feature extraction within the context of stacked LSTM networks. The profound improvements evidenced across multiple metrics collectively reinforce the significance of strategic preprocessing, while also validating the symbiotic synergy of advanced techniques. This synergy culminates in an intrusion detection model with heightened accuracy and precision, firmly establishing its potential to proactively safeguard against evolving cyber threats. The results are visually reinforced by Figure 4, illustrating the recall, precision, and F1-Score metrics for each class. Notably, the proposed approach excels in accurately classifying normal network traffic and generic attacks, as evidenced by high recall and precision values. However, challenges arise in distinguishing specific attack types such as "Analysis" and "Shellcode", reflected in comparatively lower recall and precision metrics.

The difficulty in identifying these attack categories emphasizes the intricacies of differentiating nuanced and sophisticated attack patterns. Despite advancements in intrusion detection systems, the evolving landscape of cyber threats poses challenges in accurately discerning between benign and malicious activities. This intricacy is particularly pronounced when faced with sophisticated attacks that employ deceptive tactics to evade detection.

Additionally, a binary classification task was conducted using the same model to differentiate between normal and attack instances. Remarkably, this approach achieved an accuracy of 90.59%, surpassing the performance of the multiclass classification. The recall, precision, and F1-Score metrics for this binary classification were measured at 90.75, 90.35, and 90.5, respectively. This substantial improvement underscores the model's proficiency in correctly identifying instances as either normal or attack. This trend is vividly depicted in Figure 5. In Figure 5(a) the binary confusion matrix, where the diagonal elements represent accurate predictions, further validating the model's effectiveness. Moreover, Figure 5(b) provides an insightful graphical representation of the recall, precision, and F1-Score metrics for both normal and attack instances, elucidating the model's robust capability in the binary context. These results underscore the model's capacity to excel in specific classification tasks, presenting avenues for tailored intrusion detection strategies.

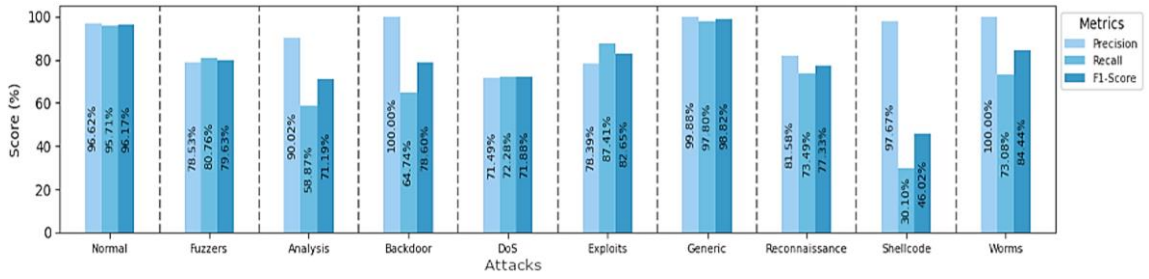


Figure 4. Performance analysis across different attack classes

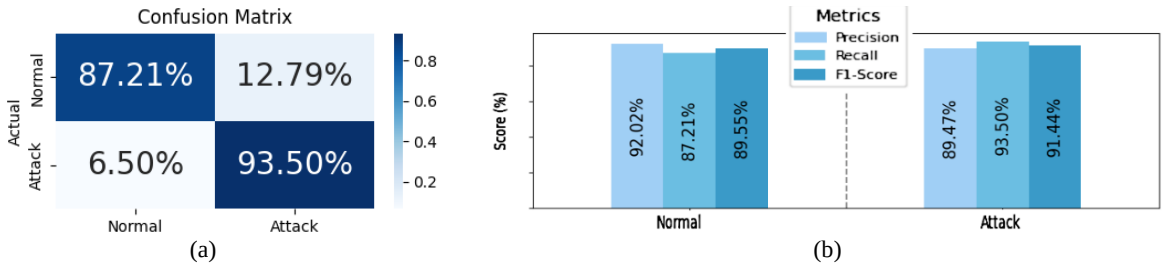


Figure 5. Evaluation of binary class performance (a) confusion matrix and (b) displays the comparison between a stacked LSTM model and a deep AE stacked LSTM model

Moreover, a visual representation in Figure 3(b) has been generated to illustrate the comparative efficacy of integrating feature extraction via a deep autoencoder in conjunction with the stacked LSTM model. This comparative analysis unveils a substantial amelioration across a spectrum of assessment metrics, thereby underscoring the paramount significance of the feature extraction phase. The amalgamation of the stacked LSTM model with the deep autoencoder manifests a discernible augmentation in accuracy, recall, precision, and F1-Score metrics, in contrast to the solitary stacked LSTM approach. This discernible enhancement corroborates the underlying hypothesis that the extraction of discerning features synergistically equips the model to capture intricate patterns, culminating in a more efficacious and resilient intrusion detection system. The graphical exposition vividly portrays the tangible benefits ensuing from this synergetic fusion, thereby accentuating the pivotal role of feature extraction in elevating the performance paradigm of the stacked LSTM model.

4. CONCLUSION

In this study, we presented an innovative approach to enhance network intrusion detection through the integration of deep learning techniques. The proposed framework, combining deep autoencoder-driven feature extraction and a stacked LSTM ensemble for classification, proved its effectiveness in improving detection accuracy on the challenging UNSW-NB15 dataset. The achieved accuracy of 90.59% underscores the potential of the approach to accurately differentiate between normal network activities and malicious intrusions. The synergy of deep autoencoder and stacked LSTM architecture has proved to be instrumental in addressing the complexities of network data. Our model's ability to capture nuanced temporal dependencies and hierarchically learn features contributes to its enhanced performance. Comparative analyses have highlighted the advantages of our approach, showing improvements in key metrics such as precision, recall, and F1-Score. Notably, the binary classification experiment further emphasizes the model's potential in binary threat identification, showcasing its adaptability to real-world scenarios.

While the proposed approach has shown promising results, further research can focus on refining the framework's performance in recognizing specific attack types, such as "attack analysis" and "shell code", where challenges persist. Despite the advancements, accurately distinguishing these intricate attack patterns remains a persistent challenge due to their evolving nature and deceptive techniques. Additionally, investigations into optimizing the architecture's parameters and leveraging additional domain-specific features could lead to even better results. Fine-tuning the model's hyperparameters and incorporating features that capture nuanced aspects of specific attacks may enhance the overall performance and adaptability of the proposed framework. This iterative process of refinement is essential for developing intrusion detection systems that can effectively cope with the evolving tactics employed by cyber adversaries.

In summary, while the current approach has demonstrated promise, there is ongoing room for improvement in recognizing specific attack types. By delving deeper into the nuances of attack analysis and shell code detection and refining the model's architecture and features, we aim to advance the framework's capabilities and contribute to the ongoing efforts in enhancing cybersecurity defenses. The integration of deep autoencoder and stacked LSTM architecture presents a viable solution for enhancing network intrusion detection. As cyber threats continue to evolve, our approach provides a foundation for more robust and accurate intrusion detection systems, bolstering network security in the face of ever-growing challenges.

REFERENCES

[1] Microsoft Threat Intelligence, "Microsoft digital defense report 2023," *Microsoft Security Insider*. https://www.microsoft.com/en-us/security/security-insider/microsoft-digital-defense-report-2023 (accessed Feb. 20, 2024).

[2] E. Kost, "How to write the executive summary of a cybersecurity report," *UpGuard*, 2023. https://www.upguard.com/blog/writing-a-cybersecurity-executive-summary (accessed Feb. 15, 2024).

[3] L. Chongrui *et al.*, "Artificial intelligence in cyber security," *Journal of Physics: Conference Series*, vol. 1964, no. 4, Art. no. 42072, Jul. 2021, doi: 10.1088/1742-6596/1964/4/042072.

[4] J. Singh and J. Singh, "A survey on machine learning-based malware detection in executable files," *Journal of Systems Architecture*, vol. 112, Jan. 2021, doi: 10.1016/j.sysarc.2020.101861.

[5] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure," *Sensors*, vol. 23, no. 5, Art. no. 2415, Feb. 2023, doi: 10.3390/s23052415.

[6] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Computer Communications*, vol. 199, pp. 113–125, Feb. 2023, doi: 10.1016/j.comcom.2022.12.010.

[7] W. Wei, S. Chen, Q. Lin, J. Ji, and J. Chen, "A multi-objective immune algorithm for intrusion feature selection," *Applied Soft Computing*, vol. 95, p. 106522, Oct. 2020, doi: 10.1016/j.asoc.2020.106522.

[8] J. O. Mebawondu, O. D. Alowolodu, J. O. Mebawondu, and A. O. Adetunmbi, "Network intrusion detection system using supervised learning paradigm," *Scientific African*, vol. 9, p. e00497, Sep. 2020, doi: 10.1016/j.sciaf.2020.e00497.

[9] A. Alzaqebah, I. Aljarah, and O. Al-Kadi, "A hierarchical intrusion detection system based on extreme learning machine and nature-inspired optimization," *Computers & Security*, vol. 124, p. 102957, Jan. 2023, doi: 10.1016/j.cose.2022.102957.

[10] N. Acharya and S. Singh, "An IWD-based feature selection method for intrusion detection system," *Soft Computing*, vol. 22, no. 13, pp. 4407–4416, May 2017, doi: 10.1007/s00500-017-2635-2.

[11] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimisation," *Neural Computing and Applications*, vol. 32, no. 10, pp. 6125–6137, Feb. 2019, doi: 10.1007/s00521-019-04103-1.

[12] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," Sep. 2017, doi: 10.1109/sisy.2017.8080566.

[13] R. Tahri, Y. Balouki, A. Jarrar, and A. Lasbahani, "Intrusion detection system using machine learning algorithms," *ITM Web of Conferences*, vol. 46, p. 2003, 2022, doi: 10.1051/itmconf/20224602003.

[14] J. Lansky *et al.*, "Deep learning-based intrusion detection systems: A systematic review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021, doi: 10.1109/access.2021.3097247.

[15] N. Thapa, Z. Liu, D. B. KC, B. Gokaraju, and K. Roy, "Comparison of machine learning and deep learning models for network intrusion detection systems," *Future Internet*, vol. 12, no. 10, p. 167, Sep. 2020, doi: 10.3390/fi12100167.

[16] Z. El Mrabet, M. Ezzari, H. Elghazi, and B. A. El Majd, "Deep learning-based intrusion detection system for advanced metering infrastructure," in *Proceedings of the 2nd International Conference on Networking, Information Systems & SecurityMarch 2019, NISS '19*, 2019, Art. no. 58, pp. 1–7, doi: 10.1145/3320326.3320391.

[17] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, Jun. 2019, doi: 10.1109/JIOT.2018.2871719.

[18] M. Srikanth Yadav. and R. Kalpana., "Data preprocessing for intrusion detection system using encoding and normalization approaches," in *2019 11th International Conference on Advanced Computing (ICoAC)*, Chennai, India, 2019, pp. 265-269, doi: 10.1109/icoac48765.2019.246851.

[19] L. B. V de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, "The choice of scaling technique matters for classification performance," *Applied Soft Computing*, vol. 133, Art. no. 109924, Jan. 2023, doi: 10.1016/j.asoc.2022.109924.

[20] E.-R. Ardelean, A. Coporîie, A.-M. Ichim, M. Dînşoreanu, and R. C. Mureşan, "A study of autoencoders as a feature extraction technique for spike sorting," *PLOS ONE*, vol. 18, no. 3, p. e0282810, Mar. 2023, doi: 10.1371/journal.pone.0282810.

[21] S. Chen and W. Guo, "Auto-encoders in deep learning—a review with new perspectives," *Mathematics*, vol. 11, no. 8, Art. no. 1777, Apr. 2023, doi: 10.3390/math11081777.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

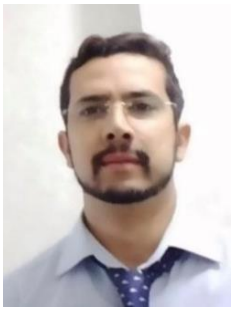
[23] M. F. Rabby, Y. Tu, M. I. Hossen, I. Lee, A. S. Maida, and X. Hei, "Stacked LSTM based deep recurrent neural network with Kalman smoothing for blood glucose prediction," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, Mar. 2021, doi: 10.1186/s12911-021-01462-5.

[24] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting," *Procedia Computer Science*, vol. 135, pp. 89–98, 2018, doi: 10.1016/j.procs.2018.08.153.

[25] H. M. Bui and A. Liu, "Density-softmax: Scalable and calibrated uncertainty estimation under distribution shifts," *Arxiv.org/abs/2302.06495*, Feb. 2023.

BIOGRAPHIES OF AUTHORS

Mehdi Moukhafi with a master's degree in quality from the Faculty of Sciences Dhar El Mehraz, Fez in 2014, they further advanced their education by obtaining a doctoral degree in computer science from Moulay Ismail University, Meknes. Presently, they hold the position of professor at the same university. Their expertise flourishes within the research team "Electrical Engineering, Vision, and Artificial Intelligence (IEVIA)" of the IMAGE laboratory. Their research endeavors are concentrated in the realm of computer security, approached through an artificial intelligence-based perspective. He can be contacted at email: mehdi.moukhafi@gmail.com.



Mouad Tantaoui is a computer science engineer from National School of Computer Science and Systems Analysis (ENSIAS) in 2013 and obtained a doctoral degree in computer science from Faculty of Science and Technology, Hassan 2 University, Mohammedia, now he holds the position of a professor at Moulay Ismail University, Meknes. He is part of the research team Electrical Engineering, Vision, and Artificial Intelligence (IEVIA) of the Computer Science, Applied Mathematics, and Electrical Engineering (IMAGE) Laboratory. Their research endeavors are concentrated in the realm of computer security, approached through an artificial intelligence-based perspective. He can be contacted at email: tantaoui.mouad@gmail.com.

Idriss Chana received the Ph.D. degree from Mohamed V University of Rabat, Morocco in 2013. He is currently an associate professor at Moulay Ismail University of Meknès, Morocco. His research interests include information and communication technologies and artificial intelligence. A large part of his research projects is related to error correcting codes and Turbo codes. Idriss Chana has published more than 40 papers in major journals and conferences in information theory and artificial intelligence. He can be contacted at email: idrisschana@gmail.com.

Aziz Bouazi obtained his Ph.D. from University of Le Havre (France) specializing in electrical engineering and computer vision. He is currently a research professor in the Electrical Engineering Department of E.S.T Meknes. His research works mainly focus on artificial vision and intelligence as well as on the control of electrical machines, and he is at the same time responsible for the IEVIA team of the Image Laboratory at My Ismail University. He can be contacted at email: a.bouazi@gmx.fr.