Artificial Intelligence in Cybersecurity

Journal Homepage: sciencesforce.com/aics

Artificial Intell. Cyb. Vol. 1 (2024) 60–69

Paper Type: Original Article

Enhanced Network Security using LSTM-Based Autoencoder Models

Ibrahim M. Elezmazy [1] **and Nihal N. Mostafa** [2,*]

[1] Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Sharqiyah, Egypt; ib.elazmazy024@fci.zu.edu.eg.

[2] Department of Computer Science, Misr Higher Institute for Computer and Commerce, Egypt; nihal.nabil@fci.zu.edu.eg.

Received: 11 Feb 2024 **Revised:** 23 May 2024 **Accepted:** 23 Jun 2024 **Published:** 26 Jun 2024

Abstract

An intrusion detection system (IDS) is essential to protect the network from cyber threats. This paper uses network traffic statistics to develop an LSTM-based auto-encoder model for effective intrusion detection. Time dependence is captured by the model using the long short-term memory (LSTM) network, which can discriminate between normal and pathological activity. High accuracy, recall, and F1 scores are demonstrated in experiments from the NSL-KDD dataset, demonstrating the model's resilience in identifying network intrusions. The superiority of LSTM-based approaches is confirmed by comparison with traditional methods. Through the use of deep learning techniques, this research advances IDSs by highlighting their adaptability and scalability in dynamic network environments. The proposed model achieved the highest accuracy, precision, recall, and F1-score values of 99.9%, 99.9%, 99.9%, 99.7%, and 99.8%, respectively.

Keywords: Intrusion Detection, LSTM-Based Autoencoder, Network Security, DL, Anomaly Detection.

1 | Introduction

The complexity and frequency of cyber threats have increased in the digital age, making network security a priority. Network security is greatly aided by IDSs (IDS), which monitor and analyze network traffic to identify anomalous activity that may indicate impending assassinations[1] and interpretation-based. While signature detection is effective in detecting known threats, it is less effective in detecting new and unknown attacks. In contrast, anomaly detection detects anomalies in network behavior, making it useful for detecting new threats, but with a high percentage of false positives [2].

The field of intrusion detection has witnessed a rise in the use of deep learning (DL) models, especially LSTM networks, due to the progress made in machine learning and artificial intelligence. LSTM networks are perfect for analyzing network traffic patterns that change over time since they are specifically built to capture temporal dependencies in sequential data [3]. Because of their distinct architecture, LSTM networks can reduce the vanishing gradient problem, a typical difficulty with regular Recurrent Neural Networks (RNNs) and remember long-term dependencies. Because of this feature, LSTMs are very good at simulating the complex temporal dynamics found in network traffic data.

Corresponding Author: nihal.nabil@fci.zu.edu.eg

https://doi.org/10.61356/j.aics.2024.1315

Promising outcomes have been observed in anomaly detection through the incorporation of LSTM networks into autoencoder designs. Autoencoders are composed of an encoder and a decoder that learn how to rebuild and compress incoming data to identify the fundamental patterns of typical behavior. According to Tang et al. [4], any notable variation in the reconstruction error may be a symptom of an anomaly. This method makes use of autoencoders' reconstruction power to effectively detect anomalies and the modeling prowess of LSTM networks to represent temporal relationships.

The benefits of LSTM-based autoencoders over conventional techniques and alternative DL models have been emphasized by recent research. For example, Gao et al. [3] showed that by efficiently collecting complicated temporal patterns in network data, LSTM networks could greatly improve the accuracy of IDSs. Furthermore, in some aspects of anomaly identification, LSTM-based models outperform Convolutional Neural Networks (CNNs) and Transformer models due to their capacity to manage long-term dependencies [5]. Although Transformers are great at capturing global dependencies through attention processes and CNNs are good at extracting spatial features, LSTMs are still the best option when it comes to jobs that require modeling temporal sequences.

This study's objectives are to explore the software of LSTM-based autoencoder models in intrusion detection, specializing in their effectiveness in figuring out and mitigating diverse sorts of network intrusions. By leveraging contemporary deep getting-to-know techniques, this observation seeks to cope with the limitations of traditional IDS and beautify the accuracy and reliability of intrusion detection structures. The results of this observation, which done an impressive accuracy of 99.9%, don't forget of 0.99, and an F1-score of 0.99 at the NSL-KDD dataset, underscore the capability of LSTM-primarily based fashions in advancing network safety.

The rest of this paper is classified as follows: Section 2 provides the background needed for this study. Section 3 presents the methodology of this study. Section 4 presents the proposed model. Section 5 presents experimental results. Section 6 illustrates the conclusion and future directions of this proposal.

2 | Related Work

In this section, we provide a literature review on DL-based models for intrusion detection.

Elsayed et al.[6] introduced CNN for IDSs and proposed a technique to improve its performance using two popular regularization techniques to address the overfitting problem. Information detection systems (IDS) can identify hidden intrusion events more effectively because of this technology. They trained and assessed this technology's performance using the InSDN standard dataset. The outcomes demonstrated that regularization techniques can enhance CNN-based anomaly detection models' performance in an SDN (software-defined networking) setting.

Thirimanne et al.[7] presented a model that uses large-scale machine learning techniques to develop an IDS to detect and classify network-level and host-level cyberattacks in a timely and automatic manner. This can be particularly challenging because numerous forms of invasions happen on a dynamic scale. However, such incursions can be identified with the aid of data sets and ongoing updates. One particularly noteworthy method is the DNN (Deep Neural Network), a kind of DL model that aids in the creation of an adaptable and successful IDS to identify and categorize unforeseen and unexpected cyberattacks.

Yin et al. [8] implemented a Deep Learning-based IDS using RNNs. The authors of this study used simple RNNs. A training set was sent into a data processing block in their system, which was responsible for converting categorical data into numerical inputs. Additionally, a scaling function was used to normalize each input. In addition, information for training and model construction would be sent from the data processing block to the training block. The NSL-KDD dataset was utilized in this investigation.

Pelletier et al.[9] proposed an architecture using artificial neural networks (ANN) with information gain as feature selection methods, using the CICIDS2017 dataset. After applying a software package called Boruta for feature selection, they were able to extract the top ten most crucial traits.

Wang et al.[10] developed an integrated deep intrusion detection model using SDAE-ELM and DBN-Softmax to improve detection accuracy and training efficiency. They used a small gradient descent method to optimize the network, and their performance was demonstrated in experiments on various datasets.

Laghrissi et al.[11] used DL to detect attacks using LSTM and principal component analysis (PCA) techniques. They tested their approach on the KDD99 dataset, achieving the best accuracy in binary and multi-class classification. Table 1 summarizes all the aforementioned related works in terms of the year, dataset, DL models, and the obtained accuracy.

Table 1. Summarization of intrusion detection studies based on DL.

Ref.	Year	Dataset	DL model	Evaluation (AUC.)
[6]	2021	InSDN	CNN	93.01%
[7]	2022	NSL-KDD	DNN	81.87%
[8]	2017	NSL-KDD	RNN	83.28%
[9]	2020	CICIDS2017	ANN	96.00%
[10]	2021	KDD Cup99	SDAE-ELM	93.46%
[11]	2021	KDD Cup99	LSTM	91.00%

3 | Methodology

3.1 | Long Short-Term Memory

The LSTM model, a type of Recurrent Neural Network (RNN), is especially perfect for sequential information because of their potential to preserve long-term dependencies. This makes LSTMs perfect for reading time-series statistics generated by way of network traffic, where the order of events is vital for detecting anomalies [12].

An LSTM community consists of a series of cells, every containing several key components: an input gate, a neglect gate, a mobile country, an output gate, and a hidden kingdom. Which is demonstrated in Figure 1. These gates adjust the flow of information, allowing the community to keep or discard records as needed.

Forget Gate

$$f_t = \sigma(W_f \,.\, [h_{t-1}, x_t] + b_f) \tag{1}$$

Input Gate

$$i_t = \sigma(W_i \,.\, [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(W_c \,.\, [h_{t-1}, x_t] + b_c) \tag{3}$$

Cell State Update

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$

Output Gate

$$o_t = \sigma(W_o \,.\, [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot tanh(C_t) \tag{6}$$

where: $\boldsymbol{x_t}$ is the input at time t step, $\odot$ is the element-wise dot product, $\boldsymbol{i_t}$, $\boldsymbol{o_t}$, $\boldsymbol{f_t}$ is the input gate, output gate and forget gate respectively, $\boldsymbol{c_t}$ is the cell state, W, U, and b model parameters.

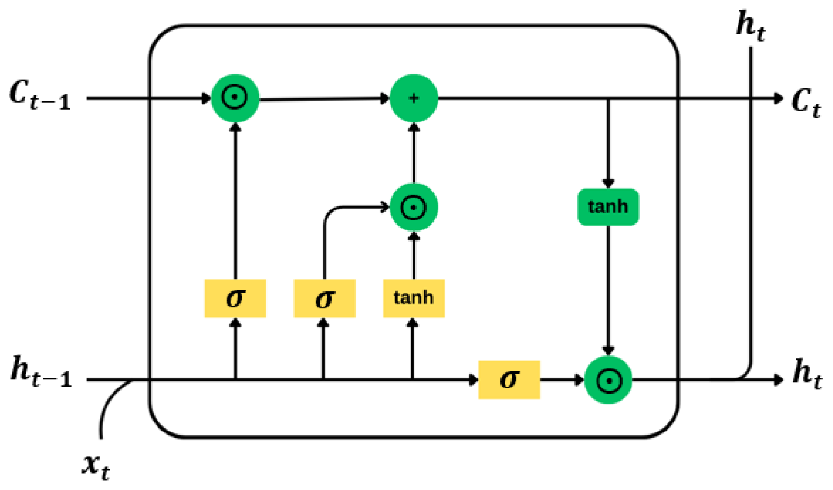

Figure 1. The LSTM architecture [13].

3.2 | Autoencoder

An unsupervised way to learn efficient coding is to implement an autoencoder in artificial neural networks. Their goal is to analyze a set of reference facts, usually to reduce dimensionality or learning features. The encoder and decoder are the two main components of the environment[14]. Autoencoder architecture is shown in Figure 2.

3.2.1 | Structure of Autoencoders

- Encoder: The input right is compressed right into a latent-vicinity illustration by way of the encoder factor of the community. Typically, it's miles a sequence of layers with fewer neurons in every layer after the only earlier than it. This phase of the network requires a more condensed illustration of the entered information.

  If the input data x is passed through the encoder, the output is the latent representation z

  $$z = f(x) \tag{7}$$

  where f represents the encoder function.

- Latent Space: The latent space (or code) is the compressed representation of the input data. It contains the essential features needed to reconstruct the original input.

- Decoder: The decoder part of the community reconstructs the entered information from the latent illustration. It mirrors the encoder shape but in reverse, typically increasing the compressed statistics again to its unique dimensions.

  The latent representation z is passed through the decoder to reconstruct the output $\hat{X}$

  $$\hat{x} = g(z) \tag{8}$$

  where g represents the decoder function.

- Loss Function: The autoencoder is trained to minimize the difference between the input x and the reconstructed output $\hat{X}$. This is usually done using a loss function such as mean squared error (MSE):

  $$\mathcal{L}(x, \hat{x}) = \| x - \hat{x} \|^2 \tag{9}$$

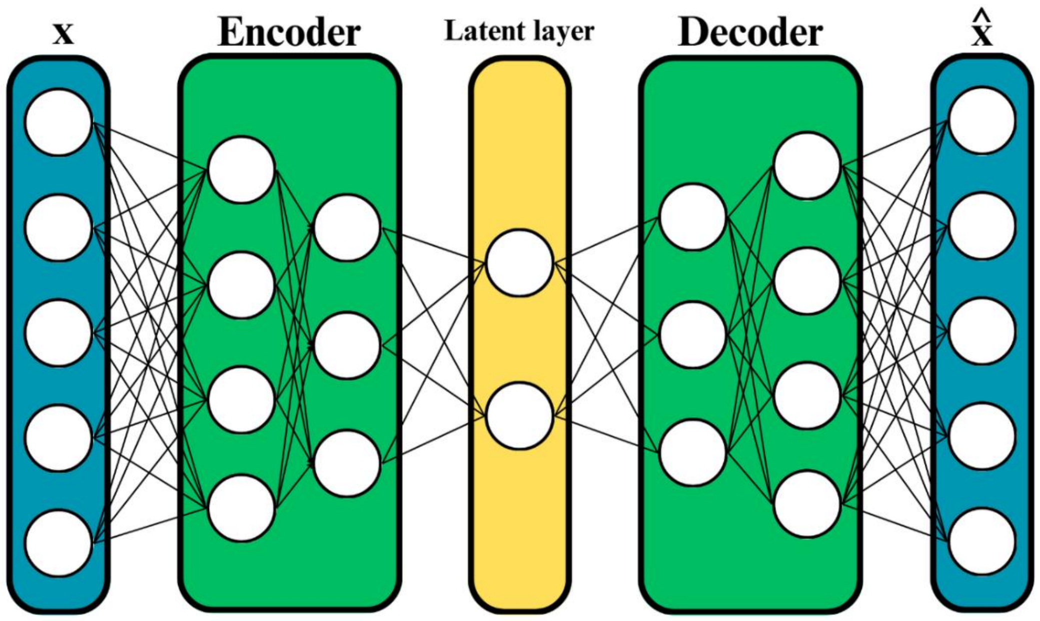

Figure 2. Autoencoder architecture [15].

4 | Proposed Work

The proposed method uses an autoencoder based on LSTM for intrusion detection. The architecture is an encoder-decoder framework designed to identify and reconstruct common patterns from network traffic data by detecting anomalies Encoder LSTM layers capture time dependencies in sequential data, compressing input sequences to make them hidden representations. In contrast, decoder LSTM layers reconstruct input sequences from hidden representations, reducing reconstruction error by Mean Squared Error (MSE) loss This approach leverages LSTM's ability to model long-term reliability, which is important for analyzing dynamic network character and effectively detect subtle anomalies. The proposed model architecture is shown in Figure 3.

The proposed LSTM-based auto-encoder model adopts an encoder-decoder architecture designed to detect and reconstruct common patterns from network traffic data when identifying anomalies [16]. The encoder LSTM layers process the input sequence$X = \{x_1, x_2, \dots, x_T\}$ to capture its temporal dependencies and compress it into a latent representationh_t

$$h_t = LSTM(x_t, h_{t-1}) \tag{10}$$

where h_t represents the hidden state at time step t

The decoder LSTM layers reconstruct the input sequence from the latent representation, minimizing the Mean Squared Error (MSE) loss function:

$$\mathcal{L}_{MSE} = \frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2 \tag{11}$$

where $\hat{x}_t$ denotes the reconstructed output at time step t.

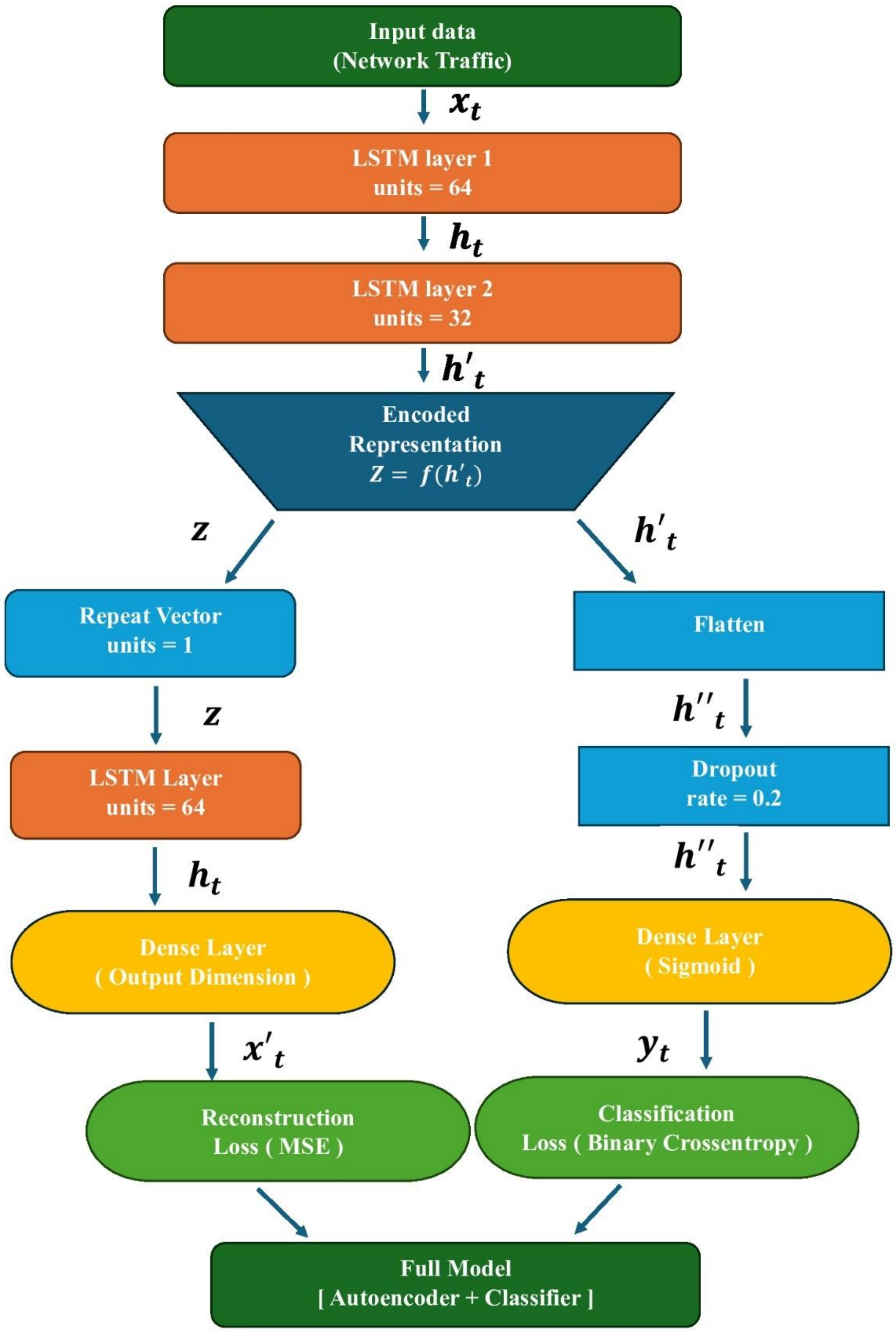

Figure 3. The proposed LSTM-Based Autoencoder model architecture.

4.1 | Training and Evaluation

The LSTM-based autoencoder model is trained using stochastic gradient descent (SGD) with time-referenced back propagation (BPTT). Over parameters such as learning rate, batch size, and number of epochs are optimized by a validation suite to maximize model performance. Evaluation criteria including accuracy, precision, recall, and F1-score have been calculated to assess the ability of the model to accurately detect intrusive networks.

The model is implemented using TensorFlow/Keras, a popular DL framework. The training process consists of preprocessing the data set, building the LSTM-based auto-encoder architecture, training the model on the training set, and evaluating its performance on the test set the implementation leverages GPU acceleration for efficient computation, ensuring scalability and efficiency in dealing with large network traffic data.

5 | Results and Discussion

This section investigates the performance of the proposed model using a widely used NSL-KDD dataset [17].

5.1 | Dataset Description

The LSTM-based autoencoder model is trained using stochastic gradient descent (SGD) with time-referenced back propagation (BPTT).

NSL-KDD is a standard benchmark dataset for intrusion detection research. The dataset includes labeled network traffic data with normal activities and different types of attacks, allowing a comprehensive assessment of the model's performance in different intrusion scenarios.

NSL-KDD is the distilled version of KDDCup 99 intrusion data. The dataset contains one label that denotes either normalcy or an attack for each of the 41 features that make up its 125,973 training records and 22,544 test samples. This set is meant to enhance unique characteristics, and lower repetition hence making it quality for machine learning evaluation. The dataset description is summarized in Table 2.

Impressive results were obtained after training the LSTM-based autoencoder model on the NSL-KDD dataset.

Table 2. NSL-KDD dataset description.

	No. of records	No. of features	No. of classes
Train	125,973	41	23
Test	22,544	41	23

5.2 | Evaluation Metrics

Our proposed work was evaluated using accuracy, precision, recall, and F1-score.

- Accuracy (ACC)

  Accuracy measures the proportion of correctly classified instances among all instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

  where:

  TP = True Positives (correctly identified attacks)

  TN = True Negatives (correctly identified normal instances)

  FP = False Positives (normal instances incorrectly classified as attacks)

  FN = False Negatives (attacks incorrectly classified as normal instances)

- Precision

  Precision measures the proportion of true positive predictions among all positive predictions:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

- Recall (Sensitivity or True Positive Rate)

  Recall measures the proportion of true positive instances that are correctly identified:

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

- F1-score

  F1-score is the harmonic mean of precision and recall, providing a balanced measure between the two:

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{15}$$

Together, these metrics examine how well the model can classify network traffic into common and attack categories. Higher contrasts indicate stronger performance in terms of identification while reducing false positives and false negatives. For LSTM-based autoencoders, these metrics provide quantitative insight into the efficiency and reliability of the IDS.

5.3 | Implementation Settings

Table 3 describes a complete detail of our implementations in consideration of parameters used in training DL models.

Table 3. Implementation settings.

Frameworks	Google.Colab platform and Keras API
Optimizer	Adam
Epochs	50
Batch size	128

5.4 | Statistical Analysis

In this section, we provide the results of our investigation on six DL models CNNs, DNN, ANN, RNN, DBN, and LSTM in terms of accuracy, precision, recall, and F1-score. Table 4 shows the superior accuracy for ANN and the smaller accuracy.

Table 4. Performance of different DL models for Intrusion detection.

Model	Accuracy	Precision	Recall	F1-score
CNN	92.7	94.0	92.3	92.6
DNN	99.1	99.1	99.1	99.0
ANN	53.9	26.9	50.0	34.99
RNN	99.3	99.3	99.3	99.3
DBN	99.0	99.1	99.0	99.0
LSTM	99.1	99.1	99.0	99.1
Proposed model	99.9	99.9	99.9	99.8

6 | Conclusion and Future Work

The use of an LSTM-based auto-encoder to detect intrusions has shown significant promise and effectiveness in protecting networks from evolving cyber threats Through rigorous implementation and analysis of the NSL-KDD dataset on the model, the model obtained exceptional performance parameters including accuracy, recall, and F1 scores of 0.99 These results in network traffic data highlight the ability of LSTM networks to capture strong time dependence tree, and enables the distinction between normal actions in abnormal activity with greater accuracy.

- Enhancing pattern interpretation capabilities: Include methods for interpreting pattern decisions and anomaly identifications to provide actionable insights for network administrators.
- Communications and real-time systems: Optimizing an LSTM-based auto-encoder for deployment in real-time IDSs, ensuring scalability and efficiency.
- Evaluating ensemble methods: Evaluating ensemble learning techniques to combine the strengths of different models for more accurate insights and improved robustness.
- Leveraging new threats: Continually updating and training the model on evolving data sets to better identify emerging threats and day-to-day attacks.
- Privacy and ethics: Addressing privacy concerns and ethical considerations for using DL models in intrusion detection.

Acknowledgments

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

Author Contribution

All authors contributed equally to this work.

Funding

This research has no funding source.

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data but are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Buczak, A.L. and E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications surveys & tutorials, 2015. 18(2): p. 1153-1176.

[2] Lazarevic, V., et al., CD40, but not CD40L, is required for the optimal priming of T cells and control of aerosol M. tuberculosis infection. Immunity, 2003. 19(6): p. 823-835.

[3] Gao, J., et al., Omni SCADA intrusion detection using deep learning algorithms. IEEE Internet of Things Journal, 2020. 8(2): p. 951-961.

[4] Tang, Y., et al., Integrating prediction and reconstruction for anomaly detection. Pattern Recognition Letters, 2020. 129: p. 123-130.

[5] Zhang, B., et al., Temporal grafter network: Rethinking lstm for effective video recognition. Neurocomputing, 2022. 505: p. 276-288.

[6] Elsayed, M.S., et al. The role of CNN for intrusion detection systems: An improved CNN learning approach for SDNs. in International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures. 2021. Springer.

[7] Thirimanne, S.P., et al., Deep neural network based real-time intrusion detection system. SN Computer Science, 2022. 3(2): p. 145.

[8] Yin, C., et al., A deep learning approach for intrusion detection using recurrent neural networks. Ieee Access, 2017. 5: p. 21954-21961.

[9] Pelletier, Z. and M. Abualkibash, Evaluating the CIC IDS-2017 dataset using machine learning methods and creating multiple predictive models in the statistical computing language R. Science, 2020. 5(2): p. 187-191.

[10] Wang, Z., et al., Intrusion detection methods based on integrated deep learning model. Computers & Security, 2021. 103: p. 102177.

[11] Laghrissi, F., et al., Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data, 2021. 8(1): p. 65.

[12] Hochreiter, S. and J. Schmidhuber, Long short-term memory. Neural computation, 1997. 9(8): p. 1735-1780.

[13] Varsamopoulos, S., K. Bertels, and C. Almudever, Designing neural network based decoders for surface codes. 2018.

[14] Kramer, M.A., Nonlinear principal component analysis using autoassociative neural networks. AIChE journal, 1991. 37(2): p. 233-243.

[15] Song, Y., S. Hyun, and Y.-G. Cheong, Analysis of Autoencoders for Network Intrusion Detection. Sensors, 2021. 21: p. 4294.

[16] Zhu, H., S. Liu, and F. Jiang, Adversarial training of LSTM-ED based anomaly detection for complex time-series in cyber-physical-social systems. Pattern Recognition Letters, 2022. 164: p. 132-139.

[17] NSL-KDD dataset.