LSTM-RNN method for Anomaly-Based Intrusion Detection Systems

Alexander Alexandrov[1,*]

[1]Institute of Robotics – Bulgarian Academy of Sciences, Acad. Georgi Bonchev Str., Bl. 2, Sofia, 1113, Bulgaria

Abstract

Intrusion Detection Systems (IDS) play a key role in protecting networks and systems from malicious activities and unauthorized access. With the increasing complexity of cyber threats, traditional methods for detecting intrusions often fail to meet the demands of modern network security. This paper proposes a method based on version of Recurrent Neural Networks (RNNs) called Long Short-Term Memory (LSTM) to improve the efficiency of Anomaly-Based Intrusion Detection Systems (AIDS). LSTM-RNNs approach are particularly well-suited for analyzing time-based network traffic and identifying deviations from normal behavior.

The paper proposes a new method based on LSTM-RNNs of AIDS, to improve anomaly detection capabilities and the system's performance. The research also addresses the benefits and limitations of using LSTM-RNNs for intrusion detection, as well as potential future developments in this area.

Keywords

IDS, AIDS, ML, LSTM-RNNs

1. Introduction

The proliferation of networked devices and the rise of digital platforms bring significant benefits to modern society. At the same time, this also leads to a wide range of cyber threats. Intrusion Detection Systems (IDS) are software tools that monitor network or system activities for malicious actions or policy violations [1]. Once detected, these activities are either reported to a system administrator or handled autonomously. The main goal of IDS is to identify unauthorized use, misuse, and abuse of computer systems by both internal and external parties. Traditionally, IDS are classified into two main categories: signature-based detection and anomaly-based detection. Signature-based detection methods rely on predefined models or signatures of known attacks. This type of IDS relies on a database of known attack signatures or patterns [2].

When network traffic matches a pattern in the database, the system alerts the control center and/or performs procedures to protect against the attack. Although the Signature-Based Detection (SBD) approach is effective against known threats, it fails to detect new, unknown attacks or "zero-day" exploits, as there is no predefined signature for these intrusions. Anomaly-based intrusion detection systems monitor the normal behavior of a network or system and raise an alert when deviations from this baseline are observed. This approach is especially useful for detecting unknown attacks, such as "zero-day attacks" and other emerging threats, as it does not require prior knowledge of the attack's signature [3, 4]. With the increase in the volume of Big Data and the growing complexity of cyber-attacks, machine learning (ML) techniques have become a promising approach to enhancing the capabilities of IDS.

Machine learning (ML) is emerging as a powerful tool for improving IDS performance [5]. By learning from historical data, machine learning models can detect patterns and identify deviations that may indicate an intrusion. Moreover, machine learning algorithms can adapt to evolving network behavior, improving the detection of new and unknown attacks [6]. The development of algorithms based on Machine Deep Learning, such as Recurrent Neural Networks (RNNs), shows significant potential in

BISEC'2024: 15th International Conference on Business Information Security, November 28-29, 2024, Niš, Serbia

*Corresponding author.

akalexandrov@ir.bas.bg (A. Alexandrov)

0000-0002-8787-9235 (A. Alexandrov)

IDS due to their ability to process sequential and temporal data, making them particularly suitable for detecting anomalies in network traffic, which often follows temporal patterns [7].

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to recognize patterns in sequences of data, such as time series. Unlike feedforward neural networks, RNNs have loops that allow information to persist as a form of memory. This memory enables RNNs to capture temporal dependencies and process sequences of inputs in a more context-aware manner [8, 9]. RNNs are particularly useful for tasks related to sequential data, such as network traffic analysis, language analysis, etc., where the order of inputs significantly affects the output [10].

2. Related Works

The implementation of reliable IDS is crucial for the network security of systems handling data, as it can detect attempts by hackers and bots to hack the network, steal sensitive data, or initiate DOS or DDOS attacks. The present study focuses on the development of a new method and software algorithm based on Machine Deep Learning, which can be implemented in Anomaly-based Intrusion Detection Systems to improve efficiency by reducing False Positives. The authors in [11] propose an IDS using a classification algorithm SVM with SGD technology, DT, and LR. The chi-square criterion is applied for feature selection. The results show that the proposed method, with SVM and SGD, significantly improves intrusion detection accuracy. The authors in [12] propose an anomaly IDS based on a combination of the Support Vector Machine (SVM) algorithm and the Information Gain Ratio (IGR) method for feature selection.

The authors in [13] propose an IDS using the Support Vector Machine (SVM) for classification and the multiple learning automata (MLA) method for identifying optimal and significant features, removing redundant features, and fully accounting for the redundancy of one function and multiple embedded functions. The authors in [14] demonstrate the use of ML-based technologies such as the restricted Boltzmann machine (RBM) in combination with Persistent Contrastive Divergence (PCD) and Contrastive Divergence (CD) for tuning intrusion detection parameters. The authors in [15] use LSTM-RNN as the algorithm for their proposed Anomaly IDS for the Internet of Drones (IoD) network, but the focus in this study is on implementation rather than performance and accuracy of the proposed algorithm.

3. Architecture of LSTM-RNNs

The basic architecture of an RNN consists of a series of nodes (neurons) arranged in layers, similar to a traditional feedforward neural network [16]. At the same time, the type of RNNs called LSTM (Long Short-Term Memory) differ in that each node in a hidden layer not only receives input from the previous layer but also from the previous step of the same layer [17] as is shown in Fig. 1.

This recurrent connection allows the network to retain information from previous inputs when processing new ones. Mathematically, the hidden state h_t at time t in an RNN is defined as:

$$h_t = \sigma\left(W_{hh}h_t - 1 + W_{xh}x_t\right) \tag{1}$$

where h_t is the hidden state at time t, W_{hh} and W_{xh} are weight matrices,x_t is the input at time t, and σ is the activation function.

The key to the ability of LSTM-RNNs to process sequential data lies in their hidden state, which acts as memory, capturing relevant information from previous time steps.

4. Theoretical background about Long Short-Term Memory (LSTM) RNNs

One of the main limitations of standard LSTM-RNNs is the vanishing gradient problem, which makes it difficult for the network to retain information over long sequences. To address this, Long Short-Term

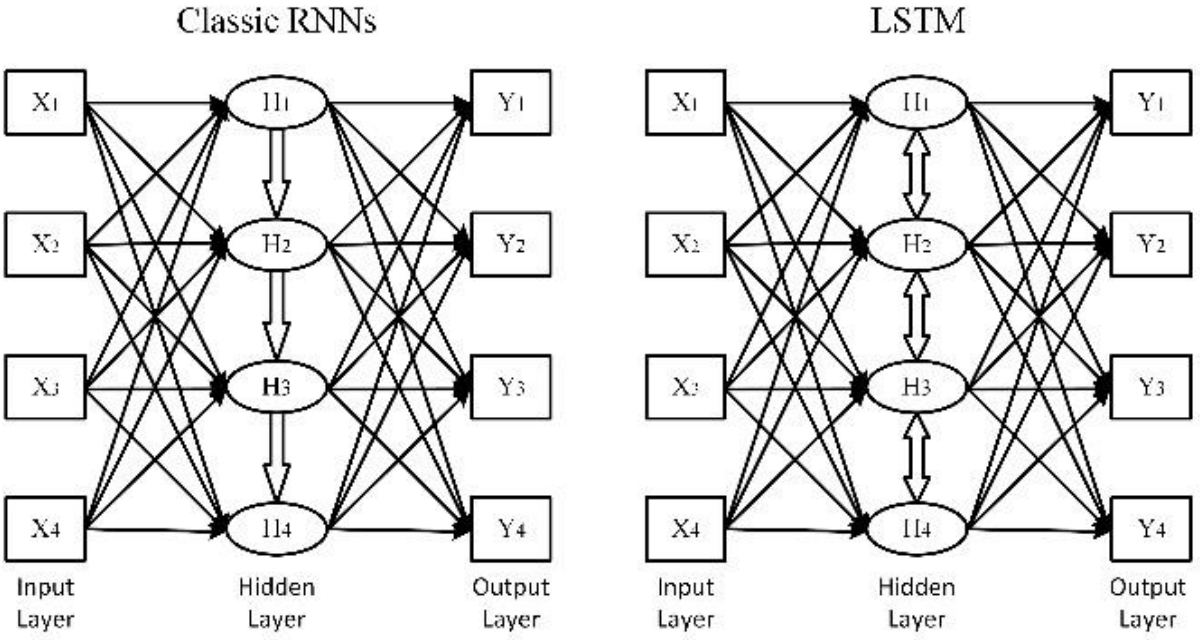

Figure 1: Differences between RNNs and LSTM.

Memory (LSTM) networks were introduced.

An LSTM cell contains several components that regulate the flow of information through the network, allowing it to decide what information to keep, update, or discard. These components include:

Forget Gate: Decides how much of the previous memory to retain.

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \tag{2}$$

Input Gate: Controls how much of the current input should be stored in the cell.

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \tag{3}$$

$$\tilde{c_t} = \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{4}$$

Cell State: The memory of the LSTM cell that carries relevant information across time steps. It is updated as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{5}$$

Output Gate: Determines how much of the memory should be used to compute the current output.

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_0\right) \tag{6}$$

The memory cell c_t allows LSTM networks to capture long-term dependencies, while the gating mechanisms control the flow of information, ensuring that irrelevant or outdated information is discarded.

5. LSTM-RNNs approach in AIDS

Network Traffic as Sequential Data

In an anomaly-based intrusion detection system, network traffic can be modeled as a sequence of feature vectors over time. These feature vectors are typically derived from network packets and can include attributes such as packet size, protocol type, connection duration, and more.

Since network traffic naturally occurs in a temporal order, LSTM-RNNs, particularly LSTMs, are well-suited for learning and modeling normal network behavior over time. An LSTM network processes the sequence of network features and captures the underlying patterns.

By training the LSTM on normal traffic data, the network learns to recognize the typical behavior of the system. Any significant deviations from this learned behavior are flagged as potential anomalies, signaling the presence of an intrusion or malicious activity.

5.1. Anomaly Detection Process

The typical process for using an LSTM-RNN in an Anomaly-based IDS can be broken down into the following steps.

Data Preprocessing.

Network traffic data is collected and preprocessed to generate feature vectors that describe each packet or flow. Common preprocessing steps include:

- **Feature Extraction**: Relevant features, such as packet size, flow duration, protocol, and flags, are extracted from the raw network traffic data.
- **Normalization**: Feature values are often normalized to ensure that all features are on the same scale, which helps improve the training efficiency of the LSTM model.
- **Sequence Creation**: The data is divided into overlapping sequences of fixed length. Each sequence represents a window of consecutive feature vectors from the network traffic.

Model Training

The LSTM model is trained on sequences of normal network traffic. During training, the model learns to predict the next data point in the sequence or reconstruct the input sequence itself. The goal is to minimize the error between the predicted and actual values, effectively teaching the model the normal patterns of network traffic.

The loss function commonly used in this setting is Mean Squared Error (MSE):

$$\mathcal{L} = \frac{1}{T}\sum_{t-1}^{T} \|x_t - \hat{x}_t\|^2 \tag{7}$$

where x_t is the actual input at time t, $\hat{x}_t$ is the predicted value, T is the total number of time steps in the sequence.

Anomaly Detection

Once the LSTM model is trained, it can be deployed to monitor real-time network traffic. For each sequence of input data, the model predicts the expected next data point or reconstructs the sequence.

If the prediction error (or reconstruction error) exceeds a predefined threshold, the system flags the current sequence as anomalous. The anomaly score s_t at time t can be defined as the magnitude of the prediction error:

$$s_t = \|x_t - \hat{x}_t\| \tag{8}$$

If s_t exceeds a certain threshold θ, the system raises an alert, indicating a potential intrusion.

5.2. Advantages of LSTM-RNNs for AIDS

LSTM-RNNs are well-suited for detecting anomalies in network traffic due to several key advantages:

- **Temporal Context**: LSTM-RNNs excel at capturing temporal dependencies, which are critical for identifying patterns in network traffic over time.
- **Adaptive Learning**: LSTM-RNNs can adapt to changing network behavior and detect deviations from normal patterns, even as normal behavior evolves.
- **Sequential Data Processing**: Network traffic is inherently sequential, and LSTM-RNNs are designed to efficiently process sequential data, making them an ideal choice for IDS.

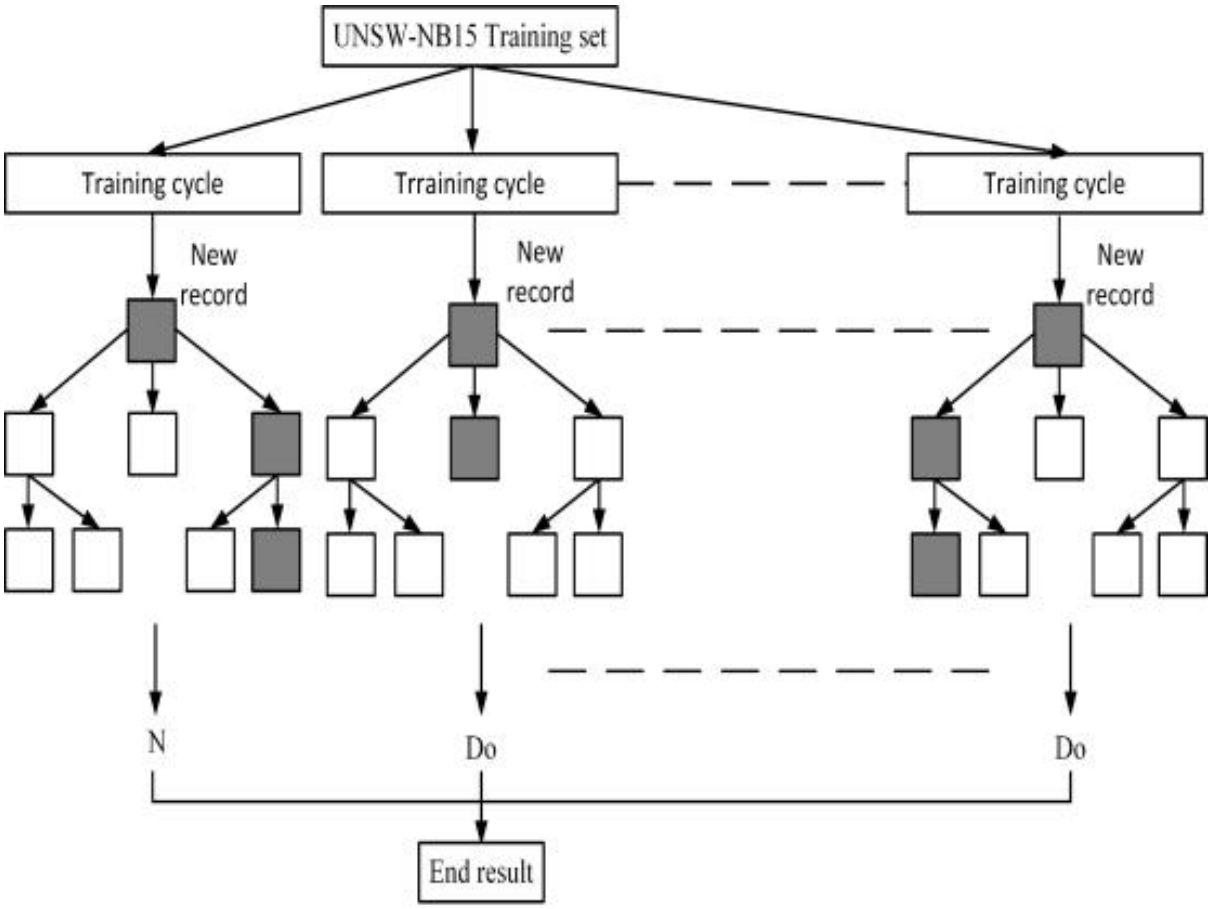

Figure 2: LSTM based training process.

6. Proposed LSTM-RNNs Based Method for AIDS

To build a reliable anomaly-based IDS using LSTM-RNNs, the following methodology is proposed.

Preprocessing of network traffic data

This includes the following tasks:

- **Feature Extraction**: Relevant features from network traffic, such as packet size, flow duration, and protocol type, are extracted. These features serve as inputs to the RNN model.
- **Normalization**: The extracted features are often normalized to ensure that the RNN can process the data efficiently.
- **Labeling**: If labeled data is available, attacks and normal traffic are labeled to create a training dataset. Unlabeled data can also be used in an unsupervised learning approach.

Model training

The LSTM model is then trained on the preprocessed data. During training, the RNN learns the normal behavior of the network by analyzing temporal patterns in network traffic. The goal of the training process is to minimize the error between the predicted output and the actual output (e.g., normal or anomalous). Supervised learning techniques can be used if labeled data is available, where the RNN is trained to classify network traffic as either normal or anomalous as is shown on Fig. 2.

For this study, the training dataset UNSW-NB15 was used. This dataset was created at the Cyber Range Lab of UNSW Canberra for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors.

src_ip	dst_ip	src_port	dst_port	proto	state	dur	sbytes	dbytes	service	sttl	dttl	rate	label
192.168.1.1	10.0.0.1	443	80	TCP	FIN	0.15	1500	3000	HTTP	64	60	3.5	0
172.16.0.2	192.168.1.4	53	8080	UDP	CON	0.25	512	1024	DNS	128	110	2.4	0
10.0.0.3	192.168.1.3	22	22	TCP	EST	0.50	2048	4096	SSH	52	56	1.7	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
192.168.1.14	10.0.0.14	80	80	TCP	FIN	0.45	3072	6144	HTTP	60	60	2.8	1
172.16.0.18	192.168.0.11	443	8080	TCP	CON	0.22	1024	2048	HTTPS	63	62	2.2	0
192.168.1.13	172.16.0.14	110	25	TCP	EST	0.50	2048	4096	POP3	54	56	1.9	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 3: UNSW-NB15 dataset example.

The dataset contains a mix of normal and malicious traffic, covering nine different attack categories. It is widely used in research for machine learning and anomaly detection in network security, and it has become one of the benchmarks for evaluating the performance of intrusion detection models.

The UNSW-NB15 dataset consists of over 2 million records, with each record representing a network connection or flow. Each connection in the dataset is characterized by a set of features extracted from the network traffic. The dataset provides 49 features for each connection, plus a class label indicating whether the traffic is normal or associated with an attack illustrated on Fig. 3.

The used in the research data set has the following features:

- `Source IP, Destination IP`: The IP addresses of the source and destination hosts.
- `Source Port, Destination Port`: The port numbers used for the connection.
- `Protocol`: The network protocol used in the connection (e.g., TCP, UDP, ICMP).
- `Service`: The type of network service involved in the connection (e.g., HTTP, FTP, DNS).
- `Packet Size`: The size of the packets transmitted during the connection.
- `Duration`: The length of the connection or session.
- `Flow Duration`: The duration of traffic flow between the source and destination.
- `Bytes Sent and Received`: The total number of bytes transmitted from the source to the destination and vice versa.
- `Label`: Indicates whether the traffic is normal or belongs to one of the nine attack categories.

The features in UNSW-NB15 include both continuous and categorical attributes, which are handled differently by machine learning models:

Feature explanations:

- `rc_ip`: Source IP address.
- `dst_ip`: Destination IP address.

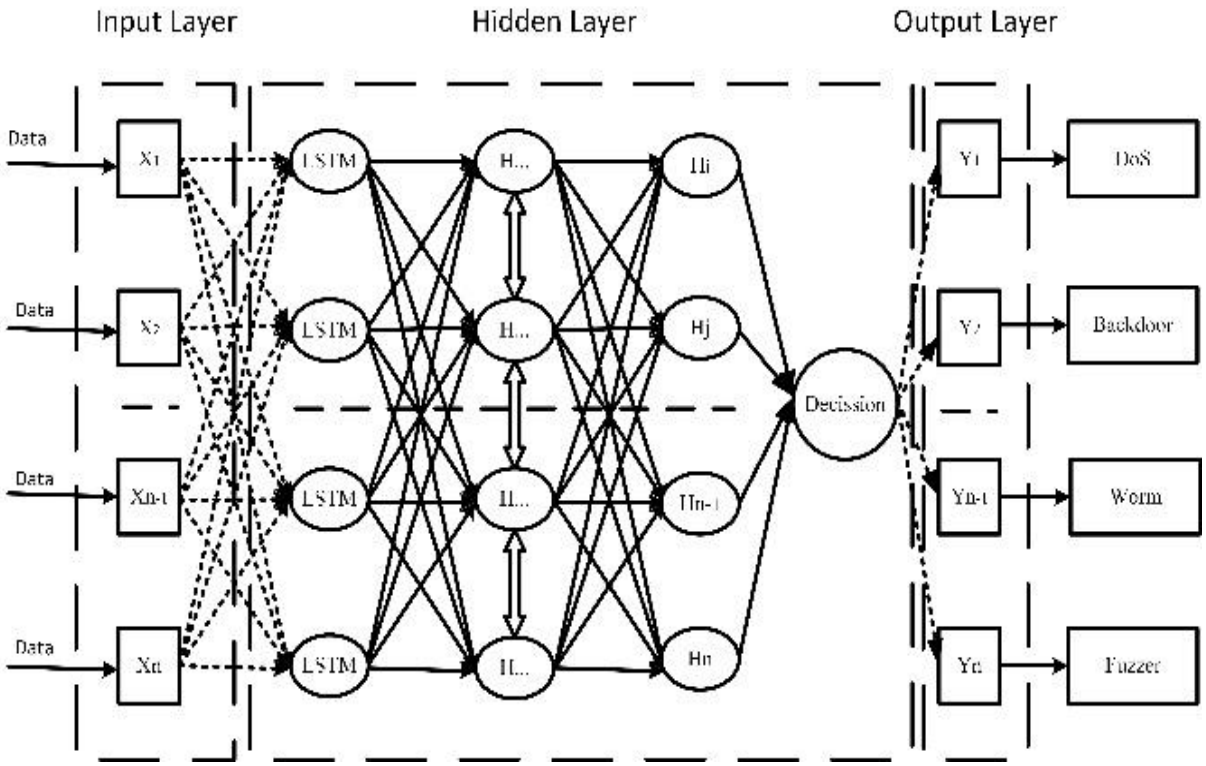

Figure 4: LSTM based Anomaly detection process.

- `src_port`: Source port number.
- `dst_port`: Destination port number.
- `proto`: Protocol used for the connection (e.g., TCP, UDP).
- `state`: The state of the connection (e.g., FIN: finished, EST: established, CON: connected).
- `dur`: Duration of the connection (in seconds).
- `sbytes`: Number of bytes sent by the source.
- `dbytes`: Number of bytes received by the destination.
- `service`: Service or application type (e.g., HTTP, DNS, FTP).
- `label`: Class label indicating whether the traffic is normal (0) or anomalous (1).

Continuous Features: These include numeric features such as packet size, duration, and the number of bytes transmitted.
Categorical Features: These include attributes like protocol type and service type, which must be converted to numerical representations (e.g., using one-hot encoding) for machine learning algorithms.

The dataset has nine types of attacks, namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

Alternatively, learning techniques can also be applied where the RNN algorithm learns normal behavior without any labeled data and marks deviations as potential anomalies, which are manually processed during the training phase.

Anomaly Detection

Once trained, the LSTM model is deployed to monitor network traffic in real-time as is shown on Fig. 4.

The model analyzes incoming traffic and compares it to the learned patterns of normal behavior. If the network traffic deviates significantly from the expected behavior, the system raises an alert, indicating a potential intrusion.

7. Experimental results

To test the proposed LSTM-RNNs based method and algorithm was developed a test environment including data server with installed UNSW-NB15 dataset, traffic generator, router, and computer with installed Wireshark tool and software IDS with implemented the LSTM based algorithm written on Python code with installed pandas, numpy, torch, and scikit-learn libraries.

The main steps in the proposed algorithm include the following steps:

- Building the LSTM model
- Train the LSTM model
- Evaluate the LSTM model
- Run the trained model with mixed with real traffic UBSW-NB15 dataset.

Scale the features UNSW-NB15 data set features.

An example Python code related to the process of building the LSTM model is shown below:

```
#Build the LSTM model
model = Sequential()

# Add an LSTM layer with 1000 units
model.add(LSTM(units=1000, return_sequences=True,
 input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))

# Add another LSTM layer
model.add(LSTM(units=1000))
model.add(Dropout(0.2))

# Add the output layer (binary classification:
# normal or anomaly)
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(optimizer='test',
 loss='binary_crossentropy',metrics=['accuracy'])
```

The part of the 1000 records UNSW-NB15 dataset with extracted and specified anomalies is shown bellow on Fig. 5.

In the lab environment the experimental results on Fig. 6. illustrate that the proposed LSTM-RNN model implemented in Anomaly-Based Intrussion Detection systems achieves 98.7% accuracy

8. Challenges and Limitations

While LSTM-RNNs offer significant advantages for anomaly detection, several challenges must be addressed:

- Data Imbalance: Network traffic data often suffers from class imbalance, where normal traffic vastly outnumbers anomalous traffic. This can lead to models that are biased toward normal traffic.
- Computational Complexity: Training RNNs, especially LSTMs, can be computationally intensive and may require significant resources, particularly for large-scale networks.
- Interpretability: Deep learning models like LSTM-RNNs are often considered "black boxes," making it difficult to interpret the model's decisions. This lack of transparency can be a concern in security applications where explainability is crucial.

src_ip	dst_ip	src_port	dst_port	proto	state	dur	sbytes	dbytes	service	sttl	dttl	rate	label	attack_cat
192.168.1.1	10.0.0.1	443	80	TCP	FIN	0.12	2000	4000	HTTP	64	63	3.2	0	Normal
172.16.0.2	192.168.1.4	53	443	UDP	CON	0.18	512	1024	DNS	128	110	2.1	0	Normal
10.0.0.3	192.168.1.3	80	80	TCP	EST	0.15	2048	4096	HTTP	52	56	3.0	1	DoS
172.16.1.5	10.0.0.5	21	21	TCP	FIN	0.25	1024	2048	FTP	59	61	2.4	0	Normal
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
192.168.1.9	10.0.0.9	443	443	TCP	FIN	0.40	4096	8192	HTTPS	65	63	2.9	1	Exploits
172.16.0.16	192.168.0.12	53	53	UDP	CON	0.12	512	512	DNS	57	56	2.1	0	Normal
192.168.1.14	10.0.0.14	80	80	TCP	FIN	0.50	3072	6144	HTTP	60	60	2.8	1	DoS
172.16.0.18	192.168.0.11	443	8080	TCP	CON	0.22	1024	2048	HTTPS	64	63	2.2	0	Normal
192.168.1.13	172.16.0.14	110	25	TCP	EST	0.30	2048	4096	POP3	54	56	1.9	1	Worms
172.16.0.19	192.168.0.6	80	443	TCP	FIN	0.15	1024	2048	HTTP	60	60	2.7	0	Normal
192.168.0.15	10.0.0.15	53	443	UDP	CON	0.14	1024	1024	DNS	61	60	2.0	1	Fuzzers
172.16.0.17	192.168.1.14	443	80	TCP	EST	0.38	3072	4096	HTTPS	64	60	2.8	0	Normal

Figure 5: Extracted and detected attacks sample.

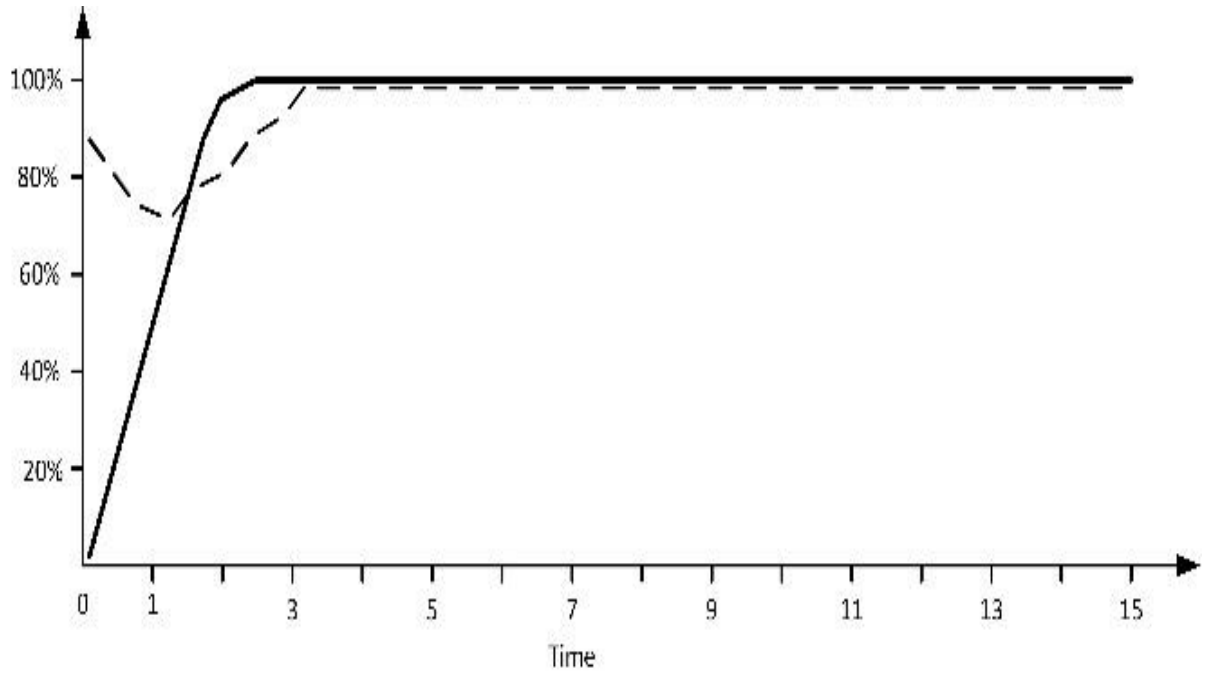

Figure 6: Output results about the accuracy of the proposed LSTM-RNN method and algorithm.

LSTM-RNN based IDS for anomaly detection has been applied in various network security scenarios, such as:

- DDoS Attack Detection: LSTM-RNNs can analyze network traffic patterns to identify abnormal spikes in traffic that may indicate a DDoS attack.
- Insider Threat Detection: LSTM-RNNs can monitor user behavior within a network and detect deviations from normal patterns, which may indicate insider threats.

- Botnet Detection: LSTM-RNNs can detect botnet activity by identifying anomalous communication patterns between infected devices.

9. Conclusion

The proposed LSTM-RNNs based method offer a powerful solution for intrusion detection in Anomaly-based Intrusion Detection Systems (AIDS). Their ability to capture temporal dependencies in network traffic makes them particularly suited for detecting anomalous behavior that may indicate cyber-attacks. While RNN-based IDS face challenges, such as data imbalance and computational complexity, they have shown significant potential in improving the accuracy and efficiency of IDS. As cyber threats evolve, the development and application of LSTM-RNNs in AIDS will play a key role in protecting modern networks.

Future research directions include combining LSTM-RNNs with other machine learning models, such as Convolutional Neural Networks (CNNs) or auto encoders, to explore the accuracy and robustness of AIDS. Further research should also investigate the reliability of RNN-based IDS deployed at the network periphery to explore latency and improve real-time detection capabilities in distributed environments.

Declaration on Generative AI

The author have not employed any Generative AI tools.

References

[1] N. Dimitrijević, A. Mesterovic, M. Bogdanović, N. Zdravković, Fraud detection and malicious code injection analysis in autograding systems, in: Proceedings of the Twelfth International Conference on Business Information Security, Belgrade, 3rd December 2021., 2021, pp. 81–85.

[2] J. M. Kizza, System intrusion detection and prevention, in: Guide to computer network security, Springer, 2024, pp. 295–323.

[3] M. Swarnkar, S. S. Rajput, Artificial Intelligence for Intrusion Detection Systems, CRC Press, 2023.

[4] J.-x. Zhou, J.-h. Yan, Secure and efficient identity-based batch verification signature scheme for ads-b system, KSII Transactions on Internet and Information Systems (TIIS) 13 (2019) 6243–6259.

[5] A. Pinto, L.-C. Herrera, Y. Donoso, J. A. Gutierrez, Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure, Sensors 23 (2023) 2415.

[6] E. Gyamfi, A. Jurcut, Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets, Sensors 22 (2022) 3744.

[7] A. Imanbayev, S. Tynymbayev, R. Odarchenko, S. Gnatyuk, R. Berdibayev, A. Baikenov, N. Kaniyeva, Research of machine learning algorithms for the development of intrusion detection systems in 5g mobile networks and beyond, Sensors 22 (2022) 9957.

[8] I. Essop, J. C. Ribeiro, M. Papaioannou, G. Zachos, G. Mantas, J. Rodriguez, Generating datasets for anomaly-based intrusion detection systems in iot and industrial iot networks, Sensors 21 (2021) 1528.

[9] W. Ma, Y. Hou, M. Jin, P. Jian, Anomaly based multi-stage attack detection method, Plos one 19 (2024) e0300821.

[10] Y.-C. Wang, Y.-C. Houng, H.-X. Chen, S.-M. Tseng, Network anomaly intrusion detection based on deep learning approach, Sensors 23 (2023) 2171.

[11] S. Saravanan, et al., Performance evaluation of classification algorithms in the design of apache spark based intrusion detection system, in: 2020 5th International Conference on Communication and Electronics Systems (ICCES), IEEE, 2020, pp. 443–447.

[12] S. Krishnaveni, P. Vigneshwar, S. Kishore, B. Jothi, S. Sivamohan, Anomaly-based intrusion detection system using support vector machine, in: Artificial intelligence and evolutionary computations in engineering systems, Springer, 2020, pp. 723–731.
[13] Y. Su, K. Qi, C. Di, Y. Ma, S. Li, Learning automata based feature selection for network traffic intrusion detection, in: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), IEEE, 2018, pp. 622–627.
[14] T. Aldwairi, D. Perera, M. A. Novotny, An evaluation of the performance of restricted boltzmann machines as a model for anomaly network intrusion detection, Computer Networks 144 (2018) 111–119.
[15] R. A. Ramadan, A.-H. Emara, M. Al-Sarem, M. Elhamahmy, Internet of drones intrusion detection using deep learning, Electronics 10 (2021) 2633.
[16] G. Ciaburro, Machine fault detection methods based on machine learning algorithms: A review, Mathematical Biosciences and Engineering 19 (2022) 11453–11490.
[17] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306.