A novel deep anomaly detection approach for intrusion detection in futuristic network

Sai Krishna Lakshminarayana, Prabhugoud I. Basarkod
School of Electronics and Communication Engineering, REVA University, Bangalore, India

Article Info

Article history:

Received Nov 3, 2023
Revised Mar 22, 2024
Accepted Apr 17, 2024

Keywords:

Anomaly detection
Futuristic network
Internet of things
Long short-term memory-autoencoder
Security attack

ABSTRACT

In an era where networks are increasingly heterogeneous and multi-domain, establishing robust security models to protect data and network infrastructure is becoming ever more complex. Traditional intrusion detection systems (IDS) often struggle with novel or variant attacks that fall outside predefined rule sets, resulting in significant detection challenges. This paper proposes a methodologically refined approach leveraging data-driven insights and statistically robust feature selection to enhance the training dataset. The study presents a long short-term memory-autoencoder (LSTM-AE) based learning model designed for multi-class anomaly detection. The model's novelty lies in its application of distance metrics to define distinct thresholds for varied attack classifications, a strategy that significantly amplifies detection precision. Experimental results validate the superior performance of the proposed system, achieving 94.82% accuracy rate, outperforming similar existing works. The study also proactively addresses common issues of class imbalance and skewed data representation in benchmark datasets by strategically training the model on normal traffic, enhancing its capability to generalize and identify anomalies effectively.

Corresponding Author:

Sai Krishna Lakshminarayana
School of Electronics and Communication Engineering, REVA University
Bengaluru, India
Email: sklnarayana@gmail.com

1. INTRODUCTION

The evolution of communication technology from 3G to 4G has already revolutionized how humans interact with digital technologies [1]. This transition has also facilitated unprecedented innovations in the actual realization of practical applications such as smart homes, smart cities, smart healthcare, autonomous vehicles, and industrial automation. With the advent of fifth-generation 5G technology, we are on the cusp of another ultra-modern transformation that will dramatically change how humans live and work [2]. The expectation from the 5G is not only limited to offering faster speed and seamless connectivity. But it will ultimately proliferate the current ecosystem of connected devices to the next level, increasing the maturity of current technologies. The integration of 5G technology with the internet of things (IoT), cloud computing, cognitive radio networks (CRN), and software-defined networking (SDN) systems, together will form the foundation of the futuristic network ecosystem [3], [4]. The futuristic network can be characterized by exemplifying the internet of everything (IoE) concept, which will have reliable connectivity, expeditious data transmission, and utmost adaptability. In this context, not only every device but maybe every object will be virtually connected, including home, office, and even our body will generate and share data, thereby transforming human lifestyles with infinite possibilities [5], [6]. However, this transition to this futuristic network ecosystem will also raise significant concerns about security and privacy issues.

As technology develops and advances, it also opens new ways for attackers to exploit vulnerabilities related to network systems. The complex nature of futuristic networks operating on the internet and wireless communication channels will carry inherent vulnerabilities, which makes it open to invite potential security threats [7], [8]. If security requirements are not ensured comprehensively, it will have a significant impact that can turn it from a promising benefit to a potential curse. As IoE evolves, we will face many new security challenges that will significantly exceed the capabilities of traditional security systems. The world has recently seen cases of modern cyber-attacks, which have caused economic loss and damage to reputation and lives [9]. Therefore, it is imperative that we learn from past cyber-attack incidents and put efforts towards developing sophisticated security solutions that meet the security requirements of a futuristic network with a maximum layer of protection. Much research has been done in literature to explore the possible security vulnerabilities associated with the IoT ecosystem, and different countermeasure schemes have been introduced to protect networks. The security solutions based on the cryptographic primitive have shown promising outcomes in ensuring data confidentially, integrity, and user privacy [10]. But with significant technological improvements, cyberattacks have become complex and increasingly intelligent. The existing cryptographic schemes primarily focus on security data rather than monitoring and protecting the network, which may not be sufficient to counter sophisticated and evolving threats effectively [11]. Therefore, to protect the entire network infrastructure a highly responsive, dynamic, and intelligent defence system is required to be design and develop. In this regard, with the advent of machine learning technologies, the development of network intrusion detection system (IDS) is one of the active areas in the current research scenario [12]. In literature, the IDS based security solutions are either based on the signature-based or anomaly-based approaches. However, unlike signature-based IDS, which rely on predefined patterns to identify only knowns attacks, anomaly-based IDS are increasingly gaining traction due to its ability to monitor network and flag any behavior that deviates from the established norm [13]. This IDS system is quick in making response than signature-based IDS and owns ability to detect zero-day attacks, a critically important feature in the face of evolving sophisticated threats in the context of futuristic IoT network.

Alzahrani and Alenazi [14] used XGBoost classifiers to create intrusion detection models for high-order network environments that integrate SDN and wireless sensor networks (WSN). The authors extracted approximately 40 features from the NSL-KDD dataset and trained an XgBoost classifier to detect multiple classes of intrusions. Experimental results demonstrate the effectiveness of the proposed scheme relative to various similar classical learning models such as decision tree (DT), random forest (RF), and artificial neural network (ANN) models. Another research effort using tree-based classifiers can be seen in the study by Sarker *et al.* [15], which addresses the challenge of high computational cost in developing IDS. They proposed a detection method using a DT model that not only performs classification but also prioritizes features based on their importance, thereby effectively reducing profile dimensionality. This strategy can shorten training time and potentially increase functional generality to develop cost-effective and robust IDS. Research efforts to reduce false positives during attack detection are shown [16], where they suggested an intelligent filtering method that uses edge computing devices with threat detection modules to jointly solve the processing burden problem and obtain better detection accuracy.

Alhajjar *et al.* [17] highlighted an exciting area of research exploring susceptibilities in different supervised learning schemes and their weakness to manipulation by attackers seeking to evade intrusion detection. They used different nature-inspired optimization algorithms and generative adversarial networks (GAN) to generate adversarial samples. The results found that many learning models experienced a significant increase in false positive rates when exposed to these adversarial patterns. A research effort to ensure the fairness of IDS on biased datasets was conducted by [18], where they adopted the synthetic minority oversampling technique to solve the class imbalance problem in network intrusion datasets. The presented scheme further utilized the Gini index to select relevant features for training the model. Experimental results on the UNSW-NB15 dataset improve IDS and ensure its suitability for class imbalance problems. Mebawondu *et al.* [19] aimed to improve the efficiency of network IDS by prioritizing accuracy and fast response time. The researchers introduced a neural network model that integrates an uncertainty-aware mechanism to identify the most relevant features in the learning process. The model's performance was validated using the UNSW-NB15 dataset. The results demonstrate the effectiveness of the proposed IDS model in terms of higher detection rate and computational efficiency. However, the inherent challenges associated with the majority of supervised IDSs are progressively giving rise to increasing interest in anomaly-based approaches. These challenges include the need for huge, labelled training data, the need for complex feature engineering pre-processing steps, and the dependence on label accuracy for effective model training. One of the most promising approaches for anomaly-based IDS is autoencoders neural network model which is an unsupervised learning technique. Autoencoders learn compressed representations from a single input, typically reflecting normal system behavior [20].

Any significant deviations from previously identified and learned patterns are then flagged as potential anomalies. In recent years, many researchers have successfully employed various autoencoder models for anomaly-based IDS development [21]. Lopes *et al.* [22] explored the effectiveness of denoising autoencoder for addressing the limited labelled data samples required to train deep learning for building effective IDS for IoT. Here, the denoising autoencoder was trained with an input sample, and obtained compact features were then used to train the deep learning models in a supervised manner. The simulation results claim the usefulness of the presented approach with data reduction to one-tenth of the input training sample size, along with 99% detection accuracy. Research work done towards identifying both known and unknown intrusions using adaptive variational autoencoder is presented [23]. This work uses an extreme value to differentiate between known and unknown intrusions using reconstruction errors obtained from the trained model. The experiments were conducted on NSL-KDD and CICIDS2017 dataset, and the outcome demonstrated that the presented model achieved a minimal false positive rate of approximately 1%. Sun *et al.* [24] presented a sparse feature representation model based on applying the latent space of a variational autoencoder to learn contextual attributes in compact vector representation. The NSL-KDD dataset was used to validate the performance of the presented scheme. The study outcome exhibited a higher detection rate and significant training, storage, and memory cost minimization. Zavrak and Iskefiyeli [25] investigated the use of autoencoders for anomaly detection in network traffic. They employ variational autoencoders on features extracted from normal traffic data. To assess the effectiveness of anomaly detection, they compare the autoencoder models to an SVM classifier. The models were evaluated on several standard datasets, and the results demonstrate that the variational autoencoder achieved lower false-positive rates compared to the SVM and standard autoencoders. Osada *et al.* [26] also see a similar research effort in the same research direction where variational autoencoder is used along with Laplacian regularized least squares method to achieve precise features and reduce training costs. There are other similar approaches, such as Dao and Lee [27], the authors have presented a lightweight IDS model using a stacked autoencoder and network pruning mechanism. An autoencoder leveraging recurrent neuron unit is used [28], [29] to develop real-time IDS for industrial networks and automotive industries, respectively.

However, there are several approaches to counter cyber-attacks in IoT, to achieve the security requirements for networks in the future, a few significant concerns still need to be fixed. Contrary to signature-based IDS, anomaly-based IDS are found be very effective and highly responsive against malicious flow in the network. It has been found that the most of the existing anomaly detection schemes, are not much applicable in futuristic network since enormous amount of data generate due their high-dimensional and heterogeneous nature. Hence, this is where autoencoder models are effective at capturing complex data distributions and reducing data dimensions both essential for anomaly detection in high-dimensional data spaces. Despite many recent works, yet, the application of autoencoders in IDS for futuristic networks has not been matured and is a relatively new area of research. The diverse range of devices and data in these networks, alongside the sophistication of modern cyber threats, necessitates a more refined and nuanced autoencoder model. Furthermore, the real-time nature of these networks demands models that are not just accurate but highly efficient. Therefore, the proposed research work aims to design and develop an effective anomaly-based IDS which is capable of detecting detect zero-day with high response mechanism for rapidly evolving digital landscape.

This paper proposes an advanced long short-term memory-autoencoder (LSTM-AE) based system for detecting anomalies in network traffic, a critical component for securing future network infrastructures. By integrating long short-term memory (LSTM) units in the autoencoder's architecture, the model captures long-term dependencies, making it adept at identifying complex, persistent threats. Utilizing the NSL-KDD dataset and focusing on HTTP traffic features aligns with the prevalent use of internet protocols in IoT networks, thus enhancing the system's practical relevance for real-world applications. The approach is designed to discern and classify intricate malicious traffic patterns effectively, ensuring robust defense mechanisms in the evolving landscape of network security. The main contributions of the proposed study are highlighted as follows:

- The strategic integration of LSTM units within both the encoder and decoder segments to capture long-term dependencies in network traffic.
- Employing statistical techniques and supervised classification for precise feature selection from the NSL-KDD dataset and HTTP traffic.
- Development of a novel approach for computing distinct thresholds for various attack classes, enhancing the model's specificity in anomaly detection.
- Designing of a lightweight and intelligent architecture that is adaptable to the complexities of modern, multi-domain network environments to identify novel threats that have not been previously encountered, addressing a critical need in cybersecurity.

2. METHOD

The proposed study introduces a novel LSTM-AE based anomaly detection system (ADS) to address complexities associated with capturing temporal dependencies in the network traffic, which is a is inherently sequential, with data points occurring at specific timestamps. LSTMs are specifically designed to handle such time series data, capturing the temporal dependencies and relationships between consecutive observations. By learning to reconstruct normal traffic sequences, the LSTM-AE automatically extracts relevant features that characterize normal network behavior. This feature extraction capability is crucial for anomaly detection, as anomalies often deviate from these learned patterns. The proposed LSTM-AE based ADS can identify zero-day attacks, dynamic and novel threats that have not been encountered before, unlike signature-based IDS that depend on predefined patterns. This is crucial in the dynamic and evolving threats in the futuristic network like IoE ecosystem, where new vulnerabilities and attack methods emerge frequently and necessitates advanced detection mechanisms that are both precise and adaptable. The proposed LSTM-AE based ADS is designed to learn and identify normal traffic patterns, thereby detecting deviations that may indicate security breaches. Furthermore, this study enhances the ADS framework by enabling the model to compute distinct thresholds for various attack classes. Therefore, the proposed LSTM-AE based ADS functions similarly to an IDS but with enhanced capabilities. Its primary function is to detect anomalies by recognizing deviations from learned normal patterns. Moreover, it surpasses traditional IDS by not only identifying anomalies but also classifying them based on the specific type of attack, making it a more effective and sophisticated approach for network security. Figure 1 presents the schematic architecture of the proposed LSTM-AE based ADS for futuristic network.

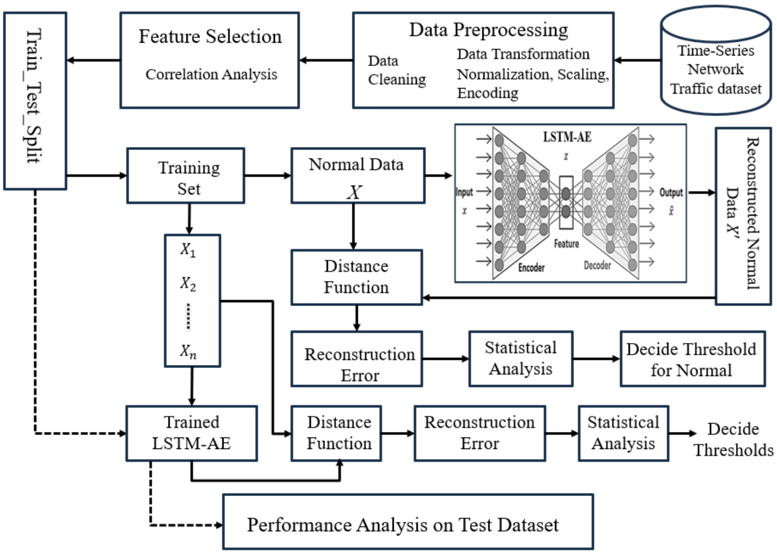


Figure 1. Flow diagram of the proposed LSTM-AE based ADS

The proposed LSTM-AE based framework consists of multiple computational blocks each in progressive and sequential manner as shown in Figure 1. Firstly, the adopted dataset is explored and analyzed to extract significant insights on the dataset samples. Secondly, the preprocessing operation is carried for data normalization and feature vectorization. Then optimal features are selected and fed to train LSTM-AE to learn the normal traffic pattern. The framework uses a distance function-based to computes reconstruction error and using statistical analysis a suitable threshold is determined for anomaly detection. The performance of model is validated in terms of accuracy, precision, recall rate and F1-score.

2.1. Dataset

The proposed study has adopted NSL-KDD dataset, created based on simulation of a variety of intrusions within a military network environment. The statistics of the NSL-KDD dataset is illustrated in Table 1. It inherently provides pre-split training and testing sets, with the classification task cantered around a response variable. A unique characteristic of the NSL-KDD dataset is the inclusion of a more attack samples in the test set compared to the training set, reflecting real-world scenarios where new attack types often emerges after development of the model.

Table 1. Statistics of dataset with respect to number of classes

Traffic classes	Training set Total sample	Testing set Total sample
Normal	67343	9711
DoS	45927	5741
Probe	11656	1106
R2L	995	5949
U2R	52	37

2.2. Preprocessing

Preprocessing is one of the critical steps required in any data-driven task to achieve accurate, and un-biased classification results. The proposed preprocessing operations adopted in the proposed LSTM-AE based ADS framework involves cleaning the dataset to remove noise, outliers, and irrelevant information. Further data transformation techniques such as normalization and encoding are applied to ensure that the LSTM-AE receives input that accurately reflects the underlying patterns of network behavior.

2.2.1. Data cleaning

The initial data cleaning operation focusses on identifying missing instances, substituting it with NaN values and then removal of these NaN to achieve completeness and ensures the integrity of the dataset, such that $D_{clean} = \{d \in D | d \text{ is notNAN}\}$, where D represents the original dataset and D_{clean} represents the dataset after removal of NaN values. The next considered in this data cleaning operation is removal of outliers using median absolute deviation estimator (MADE) method defined as follows:

$$MAD = med(|D_i - med(D)|) \quad (1)$$

$$|D_i - med(D)| > k \times MAD \quad (2)$$

Where in (1), $med(\cdot)$ denotes a function of median operation and in (2), k represents a constant, set to 1.3693 for normally distributed data to achieve approximately a 68% consistency with the standard deviation.

2.2.2. Normalization and scaling

Data normalization is a crucial process in data preprocessing operation to ensure that each feature of the dataset contributes equally to the predictive modelling. The proposed study implements min-max scaling to transform the feature values to a standard range of [0, 1]. The scaled value D'_{ij} is computed using (3):

$$D'_{ij} = \frac{D_{ij} - \min(D_j)}{\max(D_j) - \min(D_j)} \quad (3)$$

Where D_{ij} is the original data, $\min(D_j)$ is the minimum value, and $\max(D_j)$ is the maximum value of the j^{th} feature across entire dataset instances. This scaling technique enhances the ability of the LSTM-AE model to learn and detect anomalies effectively.

2.2.3. Encoding

Categorical variables are converted to a numerical format using one-hot encoding. For a categorical variable C with n unique values, one-hot encoding creates n binary $\{E_1, E_2, \cdots, E_n\}$ variables each representing one of the values in$C$. The encoding is performed as (4):

$$\begin{cases} 1, & \text{if } C_i = \text{j} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

In (4), E_{ij} is the encoded binary variable for the i^{th} instance and j^{th} the category. This process is essential for converting non-numeric data into a machine-readable format and to ensure that the LSTM-AE model can faultlessly process categorical data, preserving the informational content for anomaly detection.

2.3. Feature selection

The feature selection plays an important role in the predictive modelling as it enhances the generalization capability of the learning model by reducing dimensionality and eliminating redundant or irrelevant data. The proposed study employs dual approach of feature selection considering pearson correlation coefficient (PCC) and RF for feature ranking. The PCC is a measure that ranges from -1 to +1, where +1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 signifies no linear correlation, numerically given as (5):

$$PCC\ (X_i, Y) = \frac{\Sigma\left(X_i - \mu_{X_i}\right)(Y - \mu_Y)}{\sqrt{\Sigma\left(X_i - \mu_{X_i}\right)^2 \Sigma(Y - \mu_Y)^2}} \tag{5}$$

Where, X_i (input predictors), Y (target response), and μ is mean value of X_i and Y, respectively features with a PCC close to ±1 is considered highly correlated with the output class. Subsequently, the study implements RF classifier to rank the features based on their importance, which is derived from how well they improve the purity of the node, considering the decrease in impurity across all trees in the forest. Figure 2 illustrates the importance of feature based on their ranking determined by the RF classifier.

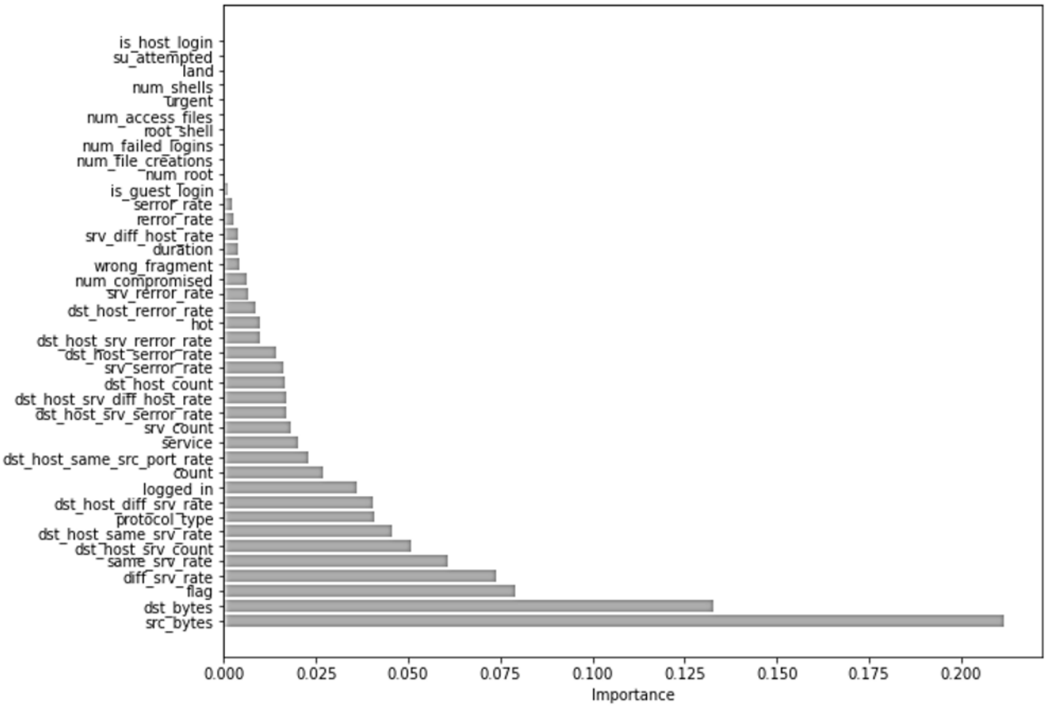


Figure 2. Illustration of top features ranked according to their importance

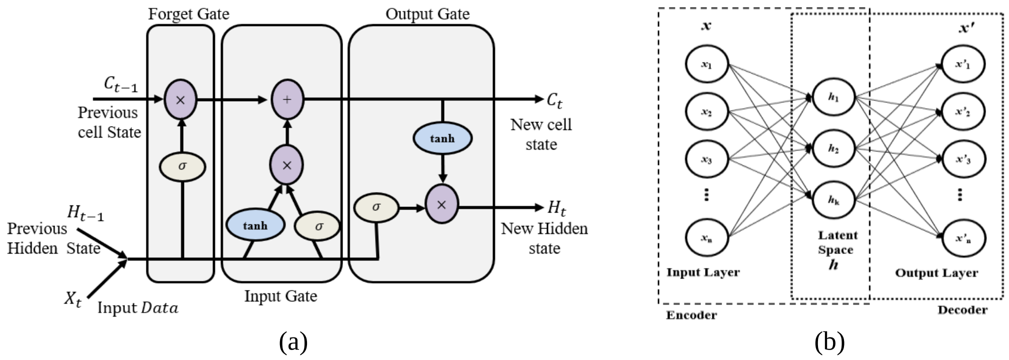
Figure 2 illustrates the feature ranking with their relative importance of each feature in descending order. Based on the insights gained from PCC analysis and feature ranking the study selected the top 30 features focusing on those that consistently show a high correlation or importance across both methods. After feature selection, to prepare the input data for the LSTM-AE, the study performs reshaping the data to reflect the sequential nature required for LSTM processing. The dataset, consisting of a series of time-sequenced observations $[X_1, X_2, X_3, \cdots, X_n]$, is organized into sequences X where each sequence has a fixed length T. Each sequence is a window of T consecutive time steps $[x_1, x_2, x_3 x_t]$, and each time step x_t is a vector in R^m representing m features at time t. The reshaping process involves transforming the data into 2-D arrays where the first dimension is the number of samples (sequences), and the second dimension is the number of time steps T, with each time step containing the m features. This results in an array shape of [samples, time steps×features], which is suitable for input into the LSTM layer of the autoencoder.

2.4. Long short-term memory-autoencoder model for network anomaly detection

The proposed LSTM-AE model integrates the strengths of LSTM networks and autoencoders to analyze high-dimensional time-series data efficiently. In this architecture, the LSTM encoder compresses the input sequence into a low-dimensional latent vector, preserving temporal dependencies through its memory cells. This process transforms the complex input data into a simpler form that captures its essential patterns. Subsequently, the LSTM decoder reconstructs the original sequence from this latent representation, employing reconstruction error to determine anomaly detection thresholds. Figure 3 demonstrates the architecture, highlighting the encoder-decoder mechanism and the critical role of LSTM units in maintaining sequence integrity while facilitating dimensionality reduction.

The LSTM unit is an advanced recurrent neural network (RNN) model that addresses the issues of vanishing gradient by effectively capturing long-term dependencies in sequential data. As represented in Figure 3(a), the LSTM-unit is composed of complex mechanisms involving multiple gates that regulate the flow of information. The first is the forget-gate, which determines what information is rejected from the cell state. It looks at the previous hidden state $H_t - 1$ and the current input $X_t$and applies a sigmoid function to decide which values are allowed to pass, ranging from 0 (completely forgotten) to 1 (completely retained). At the same time, the input gate determines what new information is added to the unit state. It uses a sigmoid layer to update the cell state, and tanh to build a vector of new candidate values to add to the state. The cell-state C_t is updated by multiplying the old state $C_t - 1$ by the output of the forget gate to discard unnecessary information and add the output of the input gate as new information. The final gate in the LSTM unit determines the next hidden state H_t, which contains information about the previous inputs. It is computed based on the cell state passed through a tanh-layer (to normalize the values) and then multiplied by the output of the sigmoid gate, ensuring that only the necessary information is passed on.

As depicted in Figure 3(b), the architecture of an autoencoder consists of two main parts such as the encoder and the decoder. The encoder component of the network compresses the input x into a latent-space representation h. The input layer, consisting of neurons $x_1, x_2, x_3, \cdots, x_n$, is fully connected to the latent space, where each neuron hi represents an encoded feature. The encoder learns to preserve as much relevant information as possible in this reduced representation. The latent space is the central layer of the network represents the compressed knowledge of the inputs and holds the most significant features learned from the input data, which are the critical components for reconstruction by the decoder. In this phase, the network attempts to reconstruct the input data x from the latent representation. The output layer mimics the input layer in size, with neurons $x_1', x_3', x_3', \cdots, x_{\mathrm{n}}'$ representing the reconstructed input data. The goal of the decoder is to output x' that is as close as possible to the original input x. In this manner, the autoencoder learns to capture the most important aspects of the input data within the latent space.



(a) (b)

Figure 3. Neural networks architectures: (a) LSTM and (b) autoencoder

2.5. Long short-term memory-autoencoder model training

The encoder in proposed LSTM-AE model composed of stacked LSTM layers, learns to compress the high-dimensional input sequences into a lower-dimensional latent representation, capturing the essential features essential to recreate the input. The model receives input with a shape corresponding to the chosen window length and number of features (e.g., [samples, 10, 30] for 10 timesteps and 30 features). The first LSTM layer has 128 neurons and returns sequences, passing its output to the second LSTM layer with 64 neurons, also returning sequences to maintain the temporal structure. The third LSTM layer reduces the dimensionality further with 32 neurons and preparing the data for the latent space representation. This layer replicates the 32-dimensional latent representation across the sequence length to facilitate the decoding process. Afterwards a RepeatVector layer is used to duplicate the encoder's output across the timesteps required by the decoder. The decoding process begins with an LSTM layer with 32 neurons that reconstructs the sequence from the latent space representation. Subsequently, a second LSTM layer with 64 neurons continues the reconstruction process, still returning sequences. The third and final LSTM decoder layer has 128 neurons, and reconstructs the sequence at the original feature dimensionality. A TimeDistributed wrapper is applied to a dense layer with a neuralunit equal to the number of selected features (i.e., 30 neural units in proposed context), ensuring the output has the same structure as the input data. The proposed model is compiled with an Adam optimizer and (mean square error) MSE as the loss function. Furthermore, the training process involves feeding the model batches of normal traffic data and using backpropagation through time to update the model weights. The model is trained over 50 epochs until the reconstruction error on the training set stabilizes. Algorithm 1 describes the process of training LSTM-AE for ADS.

```
Algorithm 1: Training LSTM- AE for ADS
Input: $X \in \mathbb{R}^{n \times t \times f}$ a tensor of normal network traffic data, where $n$ is the number of samples, $t$is timesteps per sample, and $f$ is the number of features (30 after feature selection), and E (epochs), BS (batch size)
Output: $\varepsilon$ (Reconstruction error), Th (threshold for normal traffic), $\theta$ (Trained model parameters)
Start
    1.  Initialize the LSTM-Autoencoder model with a specified architecture
    2.  For each layer in the encoder:
        Add LSTM layer with specified number of neurons$h_i$,$i \in \{1,2,\cdots,L\}$
                where $L$ is the number of LSTM layers in the encoder.
    3.  Add RepeatVector layer to replicate the latent representation.
    4.  For each layer in the decoder:
        Add LSTM layer mirroring the encoder's structure in reverse order.
    5.  Add TimeDistributed Dense layer with$f$ neurons to reconstruct the original feature dimension
    6.  Compile the model with the Adam optimizer and MSE loss function.
    7.  Reshape input data X to 3D format suitable for LSTM layers:$X \rightarrow X_{reshape} \in \mathbb{R}^{n \times t \times f}$
    8.  Train the model for the specified number of E and BS.
    9.  After training, use the trained model to reconstruct the input data:$X' = \theta(X_{reshape})$
    10. Calculate the reconstruction error$\varepsilon_i$ for each sample using the Euclidean distance
        $\varepsilon_i = \sqrt{\sum_{j=1}^{f}\left(X_{ij} - X'_{ij}\right)^2}$, for $i = 1$ to $n$
    11. Aggregate the $\varepsilon_i$ to determine a threshold$Th$ for normal traffic using statistical analysis
        $Th = mean(\varepsilon_i) + k \times std(\varepsilon_i)$where k is a scaling factor chosen using empirical analysis.
End
```

Upon the completion of Algorithm 1, a trained LSTM-AE model is capable of reconstructing normal network traffic with minimal error. The reconstruction error ε calculated for each sample in the normal traffic dataset, serves as a baseline for detecting deviations indicative of different potential anomalies. Therefore, the proposed study considers a set of known anomalies, $A = \{A_1, A_2, \dots, A_m\}$, where each A_i represents a different class of attack and m is the total number of attack types, the system first computes the reconstruction error for each anomaly instance using distance function (DF) formula as (6):

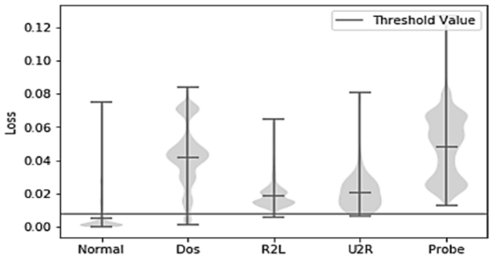
$$\varepsilon_{A_i} = \sqrt{\sum_{j=1}^{f}\left(X_{A_i,j} - X'_{A_i,j}\right)^2} \tag{6}$$

Where, $X_{A_i,j}$ is the original data for attack type A_i at feature j, and $X'_{A_i,j}$ is the reconstructed data. For each attack type A_i the proposed system performs error distribution analysis by considering the distribution of ε_{A_i} to understand how it differs from the normal traffic error distribution. The threshold Th_{A_i} for detecting each attack type is determined based on the statistical analysis of its reconstruction errors, as (7):

$$Th_{A_i} = \mu_{\varepsilon_{A_i} + k \cdot \sigma_{\varepsilon_{A_i}}} \tag{7}$$

Where $\mu_{\varepsilon_{A_i}}$ and $\sigma_{\varepsilon_{A_i}}$ are the mean and standard deviation of the reconstruction errors for attack type A_i and k is a constant that adjusts the sensitivity of the threshold. Figure 4 presents the distribution of identified threshold values for different network traffic classes.

The Figure 4 shows distribution of threshold values for different network traffic classes. Figure 4(a) presents violin plot displaying the median of reconstruction errors for normal traffic and each attack type. A horizontal line represents the set threshold value, which delineates normal behavior from potential attacks. Figure 4(b) tabulates the threshold values Th corresponding to each traffic class A_i, derived from statistical analysis of reconstruction errors serving the criteria for dynamic anomaly classification.



(a)

Th	A_i
0.007	Normal
0.04	DOS
0.03	U2R
0.02	R2L
0.05	Probe

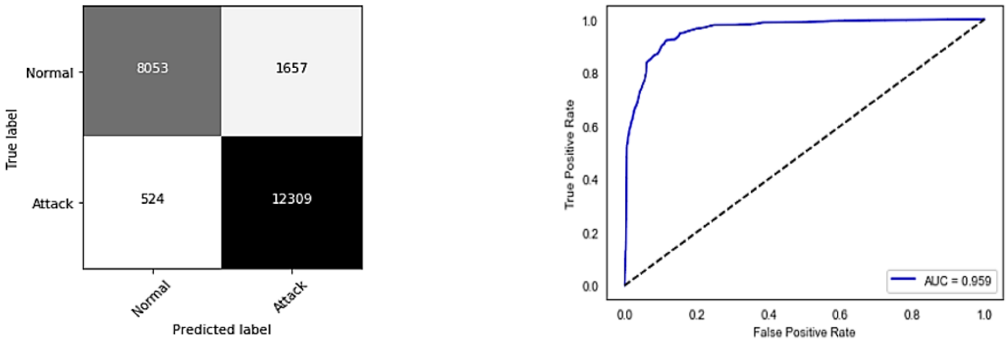
(b)

Figure 4. Threshold values for network traffic classes: (a) visual depiction of identified Th values and (b) numerical depiction of Th values

3. RESULTS AND DISCUSSION

The experiments were conducted on test-set of NSL-KDD datasets, and the results were compared with the similar existing works. Figure 5 provides outcome analysis of the proposed LSTM-AE system with respect to two sub-figures: i) a confusion matrix and ii) a receiver operating characteristic (ROC) curve. In Figure 5(a) the confusion matrix shows the number of correct and incorrect classifications made by the LSTM-AE model. Based on the closer analysis in the case of true negative (normal, normal) with 8053 predicted samples indicating the count of normal instances appropriatelyrecognised as normal, while false positives (normal, attack) show 1657 instances incorrectly identified as attacks. In case of false negatives (attack, normal) the model predicted 524 instances representing the number of attacks that were not detected by the model, while true positive (attack, attack) with 12309 exhibiting the number of attacks correctly identified.

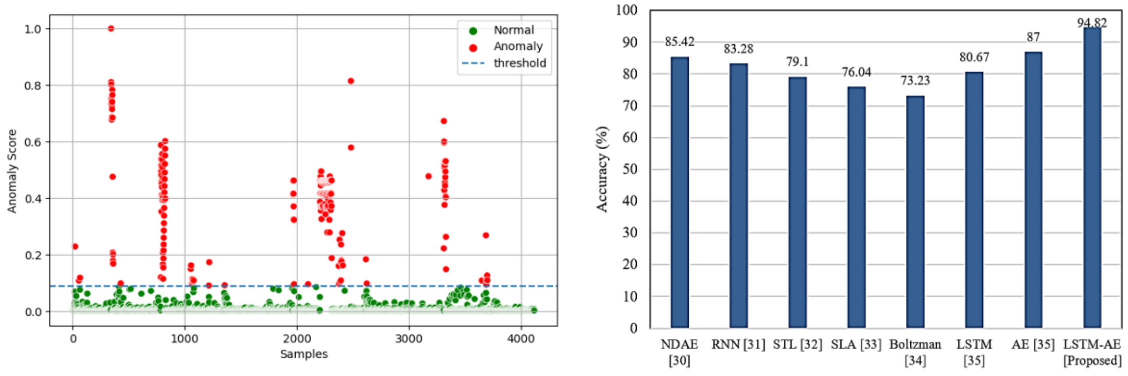
In Figure 5(b) the analysis of ROC curve shows diagnostic ability of the classifier concerning true positive rate (TPR) Vs false positive rate (FPR). The graph trend indicates a high TPR and a low FPR as area under the curve (AUC) with 0.959 suggests excellent model performance, where a value of 1.0 represents a perfect model and 0.5 represents a no-skill classifier. Figure 6 presents the evaluation of the LSTM-AE model concerning anomaly scores of network traffic samples and comparative analysis with similar existing approaches.



(a) (b)

Figure 5. Performance analysis of the LSTM-AE with (a) confusion matrix and (b) ROC curve

Based on the analysis of scatter plot in Figure 6(a) it can be seen that the anomaly scores of network traffic samples where the normal traffic is marked in green, and anomalies are indicated in red. A dashed blue line represents the anomaly detection threshold; and scores above this line are classified as anomalies. The chosen threshold is effective in separating anomalies from normal samples. Figure 6(b) provides a comparative analysis of the proposed LSTM-AE model against several similar kind of the existing research works such as non-symmetric deep auto-encoder (NDAE) proposed in [6], RNN based IDS in [30], self-taught learning (STL) introduced by [31], sequential learning accelerator (SLA) scheme for attack detection in [32], restricted Boltzmann machine for DoS attack by [33], LSTM and AE based IDS presented by [34]. Based on the graph trend, the proposed anomaly-based IDS outperformed existing model with significant improvement in accuracy score. The reason is that the proposed study, employing suitable preprocessing operations and leveraging deep LSTM capabilities within the autoencoder framework, is better able to capture the underlying and complex patterns of the input dataset and reconstructs normal input signal with very less reconstruction error.



(a) (b)

Figure 6. Evaluation of the proposed LSTM-AE with (a) anomaly score for network traffic and (b) comparative analysis

4. CONCLUSION

This paper has presented an advanced and reliable network ADS by integrating LSTM in autoencoder architecture for both uni-class and multi-class anomaly detection for futuristic network. The proposed LSTM-AE model dynamically captures temporal dependencies in the data through its LSTM unit and efficiently compresses and reconstructs the input data, enabling the detection of both known and novel attack patterns. The utilization of a statistically-driven feature selection process ensures that the model is trained on the most relevant features, enhancing its predictive accuracy. By exclusively training on normal traffic data, the LSTM-AE model avoids the common pitfalls associated with class imbalance and unrealistic data representation found in many existing datasets. Moreover, the introduction of distinct thresholds for different classes of network intrusions represents a key advancement in the field of cybersecurity. The future work will explore further enhancements of the proposed system using reinforcement learning technique to maximize the effectiveness of the proposed approach.

ACKNOWLEDGEMENT

The author thanks REVA University Bangalore for the moral support, highlighting that the research study was conducted without external financial support.

REFERENCES

[1] Y. Abuadlla, G. Kvascev, S. Gajin, and Z. Jovanovic, “Flow-based anomaly intrusion detection system using two neural network stages,” *Computer Science and Information Systems*, vol. 11, no. 2, pp. 601–622, 2014, doi: 10.2298/CSIS130415035A.

[2] M. Y. Azar, V. Varadharajan, L. Hamey, and U. Tupakula, “Autoencoder-based feature learning for cyber security applications,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3854–3861, doi: 10.1109/IJCNN.2017.7966342.

[3] B. Zhang, Y. Yu, and J. Li, “Network intrusion detection based on stacked sparse autoencoder and binary tree ensemble method,” in *2018 IEEE International Conference on Communications Workshops*, May 2018, pp. 1–6, doi: 10.1109/ICCW.2018.8403759.

[4] B. Abolhasanzadeh, “Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features,” in *2015 7th Conference on Information and Knowledge Technology (IKT)*, May 2015, pp. 1–5, doi: 10.1109/IKT.2015.7288799.

[5] S. N. Mighan and M. Kahani, “Deep learning based latent feature extraction for intrusion detection,” in *Electrical Engineering (ICEE), Iranian Conference on*, May 2018, pp. 1511–1516, doi: 10.1109/ICEE.2018.8472418.

[6] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[7] U. Çekmez, Z. Erdem, A. G. Yavuz, O. K. Sahingoz, and A. Buldu, “Network anomaly detection with deep learning,” in *2018 26th Signal Processing and Communications Applications Conference*, 2018, pp. 1–4, doi: 10.1109/SIU.2018.8404817.

[8] R. C. Aygun and A. G. Yavuz, “A stochastic data discrimination based autoencoder approach for network anomaly detection,” in *2017 25th Signal Processing and Communications Applications Conference*, 2017, pp. 1–4, doi: 10.1109/SIU.2017.7960410.

[9] S. Naseer *et al.*, “Enhanced network anomaly detection based on deep neural networks,” *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.

[10] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, “Autoencoder-based network anomaly detection,” in *2018 Wireless Telecommunications Symposium (WTS)*, Apr. 2018, pp. 1–5, doi: 10.1109/WTS.2018.8363930.

[11] “KDD cup 1999 data,” *UCI KDD Archive*, 1999. Accessed: Mar. 30, 2019. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[12] F. Farahnakian and J. Heikkonen, “A deep auto-encoder based approach for intrusion detection system,” in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, Feb. 2018, pp. 178–183, doi: 10.23919/ICACT.2018.8323688.

[13] “NSL-KDD dataset,” *University of New Brunswick*. Accessed: Jan. 30, 2019. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html.

[14] A. O. Alzahrani and M. J. F. Alenazi, “Designing a network intrusion detection system based on machine learning for software defined networks,” *Future Internet*, vol. 13, no. 5, Apr. 2021, doi: 10.3390/fi13050111.

[15] I. H. Sarker, Y. B. Abushark, F. Alsolami, and A. I. Khan, "IntruDTree: A machine learning based cyber security intrusion detection model," *Symmetry*, vol. 12, no. 5, May 2020, doi: 10.3390/sym12050754.

[16] Y. Wang, W. Meng, W. Li, Z. Liu, Y. Liu, and H. Xue, "Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 19, Oct. 2019, doi: 10.1002/cpe.5101.

[17] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Systems with Applications*, vol. 186, Dec. 2021, doi: 10.1016/j.eswa.2021.115782.

[18] S. Moualla, K. Khorzom, and A. Jafar, "Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–13, Jun. 2021, doi: 10.1155/2021/5557577.

[19] 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, Sep. 2020, doi: 10.1016/j.sciaf.2020.e00497.

[20] M. Gharib, B. Mohammadi, S. H. Dastgerdi, and M. Sabokrou, "Autoids: Auto-encoder based method for intrusion detection system," *arXiv-Computer Science*, pp. 1-9, 2019, doi: 10.48550/arXiv.1911.03306.

[21] E. Aminanto and K. Kim, "Deep learning in intrusion detection system: An overview," *2016 International Research Conference on Engineering and Technology (2016 IRCET)*, pp. 1-12, 2016.

[22] I. O. Lopes, D. Zou, I. H. Abdulqadder, F. A. Ruambo, B. Yuan, and H. Jin, "Effective network intrusion detection via representation learning: A denoising autoencoder approach," *Computer Communications*, vol. 194, pp. 55–65, Oct. 2022, doi: 10.1016/j.comcom.2022.07.027.

[23] J. Yang, X. Chen, S. Chen, X. Jiang, and X. Tan, "Conditional variational auto-encoder and extreme value theory aided two-stage learning approach for intelligent fine-grained known/unknown intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3538–3553, 2021, doi: 10.1109/TIFS.2021.3083422.

[24] J. Sun, X. Wang, N. Xiong, and J. Shao, "Learning sparse representation with variational auto-encoder for anomaly detection," *IEEE Access*, vol. 6, pp. 33353–33361, 2018, doi: 10.1109/ACCESS.2018.2848210.

[25] S. Zavrak and M. Iskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108346–108358, 2020, doi: 10.1109/ACCESS.2020.3001350.

[26] G. Osada, K. Omote, and T. Nishide, "Network intrusion detection based on semi-supervised variational auto-encoder," *Computer Security–ESORICS 2017*, Oslo, Norway, 2017, pp. 344–361, doi: 10.1007/978-3-319-66399-9_19.

[27] T.-N. Dao and H. Lee, "Stacked autoencoder-based probabilistic feature extraction for on-device network intrusion detection," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14438–14451, Aug. 2022, doi: 10.1109/JIOT.2021.3078292.

[28] S. Longari, D. H. N. Valcarcel, M. Zago, M. Carminati, and S. Zanero, "CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1913–1924, Jun. 2021, doi: 10.1109/TNSM.2020.3038991.

[29] V. K. Kukkala, S. V. Thiruloga, and S. Pasricha, "INDRA: Intrusion detection using recurrent autoencoders in automotive embedded systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3698–3710, Nov. 2020, doi: 10.1109/TCAD.2020.3012749.

[30] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[31] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *9th EAI International Conference on Bio-inspired Information and Communications Technologies*, pp. 21-26, 2016, doi: 10.4108/eai.3-12-2015.2262516.

[32] H. Huang, R. S. Khalid, W. Liu, and H. Yu, "A fast online sequential learning accelerator for IoT network intrusion detection," in *Proceedings of the Twelfth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis Companion*, Oct. 2017, pp. 1–2, doi: 10.1145/3125502.3125532.

[33] Y. Imamverdiyev and F. Abdullayeva, "Deep learning method for denial of service attack detection based on restricted Boltzmann machine," *Big Data*, vol. 6, no. 2, pp. 159–169, Jun. 2018, doi: 10.1089/big.2018.0023.

[34] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, Apr. 2020, doi: 10.1016/j.neucom.2019.11.016.

BIOGRAPHIES OF AUTHORS

Sai Krishna Lakshminarayana working as associate professor in the Department of Electronics and communication at Mother Theresa Institute of Engineering and Technology, Palamaner, Chittoor Dist, Andhra Pradesh, India. He received B.E. degree in Electronics and Telecommunication from AIT, Pune, M.Tech. in Digital Electronics and Communication from AMCE, Bangalore. He is a member of ISTE and pursuing Ph.D. from REVA University, Bangalore, India. He is having more than 17 years of experience in academics. His areas of interest are network security, wireless communication, machine learning, and computer communication network. He can be contacted at email: sklnarayana@gmail.com.

Prabhugoud I. Basarkod received B.E degree in Electronics and Communication from National Institute of Engineering, Mysore, M.E degree in Electronics from the BMS college of Engineering, Bangalore and M S in Software Systems from Birla Institute of Technology and Science, Pillani and completed his Ph.D. in Kuvempu University, Shankaragatta, Shimoga, Karnataka, India. He is currently working as a Professor in Electronics and Communication Department of REVA University, Bangalore. He is having a teaching experience of thirty-three years and his areas of interest are Wireless Communication and Computer Networking. He is a member of ISTE (MISTE, India), member of IEEE (MIEEE, USA). He can be contacted at email: basarkodpi@reva.edu.in.