COMPARATIVE ANALYSIS OF FEATURE SELECTION TECHNIQUES FOR LSTM BASED NETWORK INTRUSION DETECTION MODELS

Shahir Kottilingal

Department of Artificial Intelligence, Wakeb Data, Riyadh, KSA

ABSTRACT

Network intrusion systems are inevitable in protecting enterprise assets and improving cybersecurity. The research community is always on the lookout for new approaches to improve network intrusion detection. Network intrusion systems with deep learning models are the major advancement in this field. There are many varieties of network intrusion detection systems with deep learning models that are used nowadays. Even though researchers are investing heavily in their efforts on developing better network intrusion detection systems, rapid advancement in network intrusion attempts warrant further studies in this area. In this study, I am trying to explore the impact of the different most common feature selection methods on the performance of the LSTM-based network intrusion detection system. Benchmark network intrusion dataset UNSW-NB15 was explored for this study. This study explored 8 types of LSTM models in combination with different feature selection methods. The outcome of the study was very interesting as the LSTM model without applying any feature selection method, outperformed other combinations of models.

KEYWORDS

Network Intrusion Detection, Deep Learning, LSTM, Feature Selection & UNSW-NB15

1. INTRODUCTION

IDS systems play a pivotal part in guarding networks from cyber-attacks by continuously monitoring network and system logs [1]. Protecting data is of paramount importance and data is life blood of contemprory enterprises. It is imperative to protect data from ever-evolving intrusions [2]. An intrusion detection system (IDS) is primarily designed as either a software application or a hardware device that is specifically intended to keep track of network and system activities, and promptly notify system administrators when necessary. IDS detect an intrusion by monitoring the incoming and outgoing network traffic. There are generally two main types of intrusion detection systems (IDS), which are network intrusion detection systems (NIDS) and host intrusion detection systems (HIDS). NIDS can monitor network traffic from all the devices that enters into and leaves from the network. If, for instance, any traffic matches known library or attack patterns or if any unusual behaviour is detected, a single alert is triggered, and notifications are sent to the administrator. NIDS is capable to compare similar packets from its signatures and can detect malicious data packets matching the signatures on stored in the database [3]. All the activities in the network are tracked by IDS [4].

Intrusion attempts to break in to network infrastructure from remote location is an act of exploitation and it causes to compromise confidentiality, accessibility and integrity of the network [5]. Most regular cyber-attacks or intrusion attempts are access control breach, contamination of data, jamming, DDOS attacks, Dos attacks , exploitation and backdoor attacks [6].Even though increased number of signature based techniques available for NIDS solutions,

DOI: 10.5121/ijnsa.2024.16501

their drawback force a change to anomaly detection techniques. Signature based detection having low false positive rate but can detect only known intrusion. There is a long delay in entering newly discovered intrusions into the database. Signature [1]. However, there are many obstacles that IDS systems must overcome to quickly detect malicious intrusions due to the surge in network traffic and the associated security risks. Deep learning methods provide better results than standard machine learning methods when working with large amounts of data. Deep learning techniques for intrusion detection systems (IDS) are currently being explored and there is more scope for the development of this technology in IDS [7]. Deep learning uses artificial neural networks to capture complex relationships between inputs and outputs. In network intrusion detection systems (NIDS), deep learning has an edge over traditional machine learning methods [1]. In intrusion detection systems built using deep learning techniques, advanced and new data sets are better suited to perform multiple input detections. Intrusion detection systems based on deep learning are trained using newly created datasets for intrusion detection. In addition, there are many types of attacks and distributions in the updated data set [6]. Many machine-learning network detection methods have been developed for IoT networks, usually based on feature extraction or feature selection techniques to reduce the amount of input data. before feeding it to machine learning models. The goal of this work is to reduce the complexity of the detection for real-time operations, which is very important in intrusion detection systems [8].

Summary of major types of intrusion detection systems are used nowadays is shown in the figure 1.

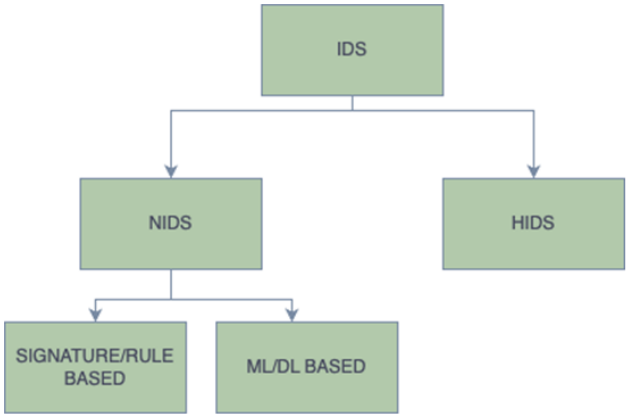

Figure 1: Major Types of Intrusion Detection Systems

2. LITERATURE REVIEW

Elsayed et.al (2024) conducted a study to compare different deep learning models for network intrusion detection. Comparison of deep learning models such as DNN, CNN, RNN, LSTM, GRU and hybrid CNN-LSTM architecture. They used the NSLKDD dataset in their study. They identified GRU as the optimal model and adjusted all meta-parameters with careful optimization to achieve optimal performance [1]. Isiaka (2024) proposed intrusion preventive system using a window-based convolutional neural network (CNN), autoencoders (AutoE) and an integrated recurrent neural network (RNN) to identify and test the performance of the intrusion detection system. Network data packets were converted to images and the pixels were used as input [3].

Shahir (2024) proposed a network intrusion detection system using Deep Abstract Networks model. This model is designed to utilize the spatial and temporal characteristics of the UNSW-

NB15 benchmark dataset. This model improves accuracy in attack detection in network as a binary classifier [5]. Three types of deep learning models were proposed by Altunay and Albayrak (2023) to identify IIoT network intrusions by using Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and CNN + LSTM generated from a hybrid combination of these. In the study conducted by using the UNSW-NB15 and X-IIoTID datasets, common and deviant data were determined and compared to other studies. The combined CNN + LSTM model achieved the most accurate interference detection value in both binary and multiclass datasets among the considered models. The proposed CNN + LSTM architecture achieved an accuracy of 93.21% for binary classification and achieved 92.9% for multi-class classification in the UNSW-NB15 dataset [6]. Ngo et al. (2023) provides a comprehensive comparison between feature Selection and feature extraction for machine learning model. They studied about these two feature reduction methods of intrusion detection by using various evaluation metrics like precision, recall, accuracy and runtime complexity. They used UNSW-NB15 dataset for both binary and multiclass classification [8].

Alkanhel et al. (2023) suggested a hybrid optimization algorithm for feature selection in intrusion detection systems. The algorithm they proposed was GWDTO and it was based on grey wolf and dipper throated optimization algorithms. The suggested algorithm achieves an improved equilibrium between the exploration and the parameters of the optimization and can achieve higher performance on the employed IoT-IDS dataset [9]. Binary classification model proposed by Alshariah et al. (2024) by leveraging the advantages of LSTM and attention mechanisms demonstrated superior accuracy compared to traditional machine learning techniques like support vector machines and k-nearest neighbours [10].Different machine learning and deep learning models for network intrusion system was compared by khan et al.(2024) in their study. The performance of the CNN and LSTM algorithm is impressive in their study. But it was unclear which dataset they used to test their model [11]. Saheed et al. (2023) presented a hybrid feature selection approach combining the Bat metaheuristic algorithm with the Residue Number System (RNS). First, the Bat algorithm is used to segment the training data and remove outliers. Realizing the slow training and testing times of the Bat algorithm, RNS has been incorporated to increase the processing speed. Additionally, principal component analysis (PCA) is used for feature extraction. They achieved high accuracy and F-score by this methods [12]. Kimanzi et al. (2024) presented a study on deep learning techniques like Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Deep Belief Networks (DBN), Deep Neural Networks (DNN), Long Short-Term Memory (LSTM), autoencoders (AE), Multi-Layer Perceptron (MLP), Self-Normalizing Networks (SNN) and hybrid models, within network intrusion detection systems. Their observation also points towards a need for extra study and stronger protection against new cyberthreats.

From the literature review, we could understand that deep learning models are performing well in network intrusion detection systems. There is numerous research in deep learning for network intrusion detection already published and many are under experimentation. But still there is a need for better deep learning model for network intrusion detection. In this study I am trying to explore some approaches with LSTM model and feature selection methods for better detection and accuracy.

3. MODEL ARCHITECTURE

This paper concentrates on evaluating the impact of feature selection methods on LSTM model for network intrusin detection system. This study experimented with different major types of feature selection methods and PCA feature extraction methods on network intrusion dataset. Dataset with selected features are trained with Long short-term memory (LSTM) model and evaluated the outcome.

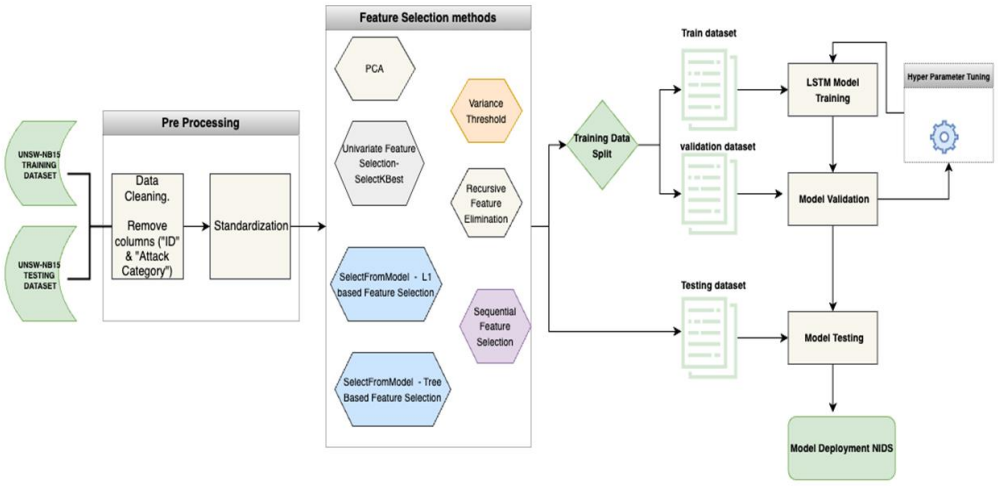

Figure 2: Model Development Architecture

3.1. Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) networks are a variant of RNNs specifically designed to overcome the vanishing gradient problem, which can limit the RNN's ability to learn long-term dependencies. LSTMs contain gating techniques that allow them to selectively store and update information over long periods of time, making them good fit for detecting anomalies that occur over longer periods of time [1].

3.2. Feature Selection and Feature Extraction

Feature selection and feature reduction on sample sets is a technique used in machine learning and deep learning modelling, either to improve estimators' accuracy scores or to boost their performance on very high-dimensional datasets [13].

3.2.1. Feature Extraction

Autoencoder and Principal Component Analysis are the important two feature extraction methods used in netowork intrusion detection systems. Fetaure extraction techniques compress the high dimensional data X to a low dimensional scale using the majorly to techniques, namely projection matrix and auto encoded based neural network.Auto encoder approach normally suffers with high computational complexity of deep neural network that causes higher latency than PCA[8]. Therefore in this reearch, concentrated on the PCA-based feature extraction approach in order to overcome latency issues of the NIDS for realtime detection. While feature selection preserves a subset of the original features in NIDS, feature extraction attempts to compress as many original features into a low-dimensional vector as keep most of the information [8].

3.2.2. Feature Selection

Major methods in feature selection are explained below,

3.2.2.1. Removing Features with Low Variance

Variance Threshold is a baseline technique for feature selection. It removes the features bases on some threshold if the feature doesn't meet it. All zero variance features are removed by default.

$$Var[X] = p(1-p) \quad [13]$$

3.2.2.2. Univariate Feature Selection:

Best features are selected in this method based on the univariate statistical analysis. This is pre-processing step to an estimator. SelectKBest is the univariate feature selection method that removes all but the highest scoring features.

3.2.2.3. Recursive Feature Elimination

Features are selected recursively by smaller set of features. An external estimator assigns weights to the features. Initially, the estimator is trained on the initial set of features, and the importance of each feature is determined by analysing any specific attribute. The least significant features are removed from the current set of features. This process is repeated recursively on the pruned set of features until the target number of features is finally reached. RFECV performs RFE in a cross-validation loop to find the optimal number of features [13].

3.2.2.4. Feature Selection using SelectFromModel

SelectFromModel is a versatile meta-transformer that can be integrated with any estimator that assigns importance to each feature based on a specific attribute like feature_importances. The features are considered unimportant and removed if the corresponding importance of the feature values are below the provided threshold parameter [13]. Below two models are used in this study that are used with SelectFromModel.

3.2.2.4.1. L1-Based Feature Selection

Linear Models are penalized with the L1 norm to have sparse solutions. The estimated coefficients are zero in this model. If target is to achieve the dimensionality reduction of the data to use along with a classifier model, they could use with SelectFromModel to retrieve the non-zero coefficients.

3.2.2.4.2. Tree-Based Feature Selection

Tree-based estimators can be used to compute impurity-based feature importances, which in turn can be used to discard irrelevant features.

3.2.2.5. Sequential Feature Selection

Sequential Feature Selection are either forward or backward. Forward-SFS is a greedy approach that iteratively discovers the optimal new feature to add to the collection of selected features. The process begins with no features, and the feature that maximizes a cross-validated score when an estimator is trained on it alone is identified. The procedure is repeated by including the new feature in the set of selected features, and it continues until the desired number of selected features is attained. This is controlled by the n_features_to_select parameter. Backward-SFS is a same as Forward-SFS but in opposite direction. Instead of commencing with no features and then incrementally adding them, it initiates with all the available features and subsequently eliminates them in a greedy manner. The direction parameter governs whether the forward or backward SFS algorithm is employed [13].

4. METHODOLOGY

As described in above sections, this paper evaluate the impact of different feature selection combining with LSTM model for network intrusion detection on UNSW-NB15 dataset. We utilized jupyter notebook for the data exploration and training the model. NVIDIA RTX 4090 GPU supported hardware was used for model training. This hardware was having configuration 128 GB RAM and 11th Gen Intel Core i9-1900K @ 3.50GHZ x16 processor. Each model with differenet features selection method, trained with the same workflow.

4.1. Data Collection

Dataset used in this study was collected from UNSW-NB15 repository. This repository contains train and test dataset curated by the authors of this repository. Training dataset contains 45 features and 175341 records. Testing dataset contains 45 features and 82332 records. UNSW repository from the below link.

https://unsw-my.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?ga=1&id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FUNSW%2DNB15%20dataset%2FCSV%20Files%2FTraining%20and%20Testing%20Sets

4.2. Pre-Processing

Pre-processing steps includes data cleaning, encoding of categorical variables, standardization, etc. First, dropped “id” and ‘attack_cat” columns as these two columns are not required for modelling. Since we are building binary model, we will consider the “label” columns as target column. The target columns contain 0 and 1 values. 1 for malicious values and 0 for benign values.

4.3. Feature Selection

There are 42 total number of features and label column in the dataset after the pre-processing step. Each models selected different number of features based on the feature selection methods used.

4.4. Train and Test split

Training dataset split into train and validation set. For all models used 75% training and 25% validation data for training.

4.5. Model Training and Validation

Model training was accomplished using pytorch library [14]. 8 LSTM models are trained with 8 types of feature selection methods. Each model trained on jupyter notebook on-premises hardware. Each model trained on 75% of the UNSW-NB15 training dataset and evaluated using other 25% of the UNSW-NB15 training dataset. Each model achieved more than 90% training accuracy and F1 Score on the evaluation dataset.

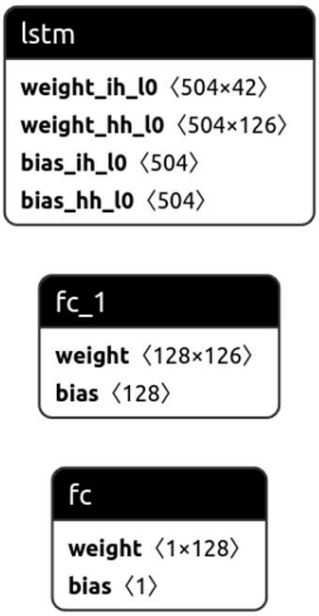

Figure 3: Architecture of selected LSTM model

Hidden layer selected for each model based on the criteria that input is multiplied with 3. For example, number of hidden layers is 126 if number of input features are 42.

4.6. Model Testing and Deployment

Each of these 8 LTSM models are tested on testing dataset from UNSW-NB15 dataset repository. The accuracy of each model on testing dataset was slightly lesser than the training and evaluation dataset. The models are selected based on accuracy score and F1 Score on the test dataset and can be deployed as a Rest API for the use of network intrusion detection system.

5. RESULT

The result of the 8 LSTM models with different feature selection methods, PCA feature extraction and with full set of 42 features without any selection on test dataset evaluated by using model evaluation metrics like accuracy and F1 Score. Also used confusion matrix for evaluating the different models. Evaluation score on testing data from UNSW-NB15 dataset repository was slightly lesser than training and evaluation dataset used for training. However, it was ensured that model is not overfit on training data by using the evaluation dataset before applying to testing dataset.

Accuracy refers to the proportion of network activity that is correctly classified, including malicious and normal traffic. A high accuracy value indicates that NIDS can distinguish between bad and good network activity and reduce the occurrence of false alarms [1].

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad [1]$$

The F1 score is *harmonic mean* of precision and recall that provides a balanced measure of the effectiveness of NIDS in detecting and classifying intrusions. A high F1 score indicates that the

NIDS achieves a good balance between accuracy and recall, ensuring that it can detect and classify intrusions without generating many false alarms or missing genuine attacks [1].

F1 = 2 * (Precision * Recall) / (Precision + Recall) [1]

Confusion matrix was another evaluation tool used evaluate the performance of different models. The confusion matrix gives valuable information about the count of true positives. It shows the samples that are correctly classified into the appropriate class, and it truly belong to that class. Also, it takes into account true negatives, which belong to a different class but are properly classified[10].The confusion matrix provides insights into NIDS performance by grouping TP, FP, TN and FN values. This breakdown can help identify areas where NIDS goes wrong and guide the development of better rules and algorithms for intrusion detection [1].

Confusion matrix graph for each LSTM model combined with different features selection methods are shown in Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, Figure 10 and Figure 11.

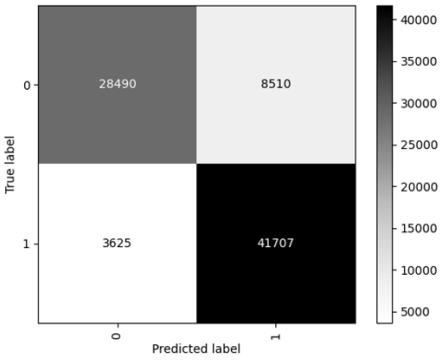

Figure 4: Without Feature Selection

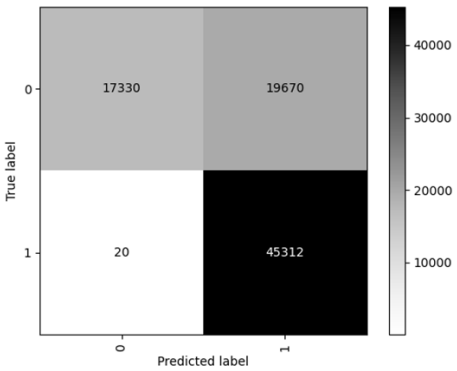

Figure 5: PCA

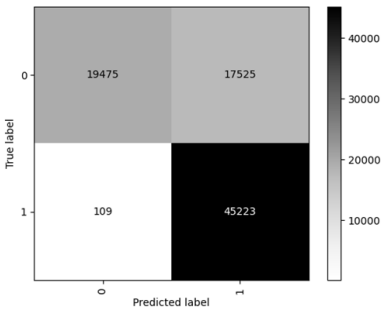

Figure 6: Univariate Feature Selection

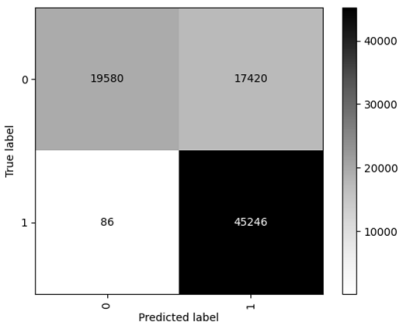

Figure 7: Recursive Feature Elimination

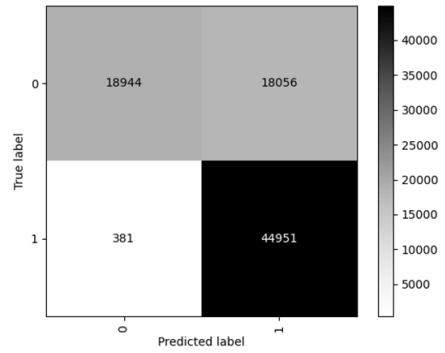

Figure 8: L1- Based Feature Selection

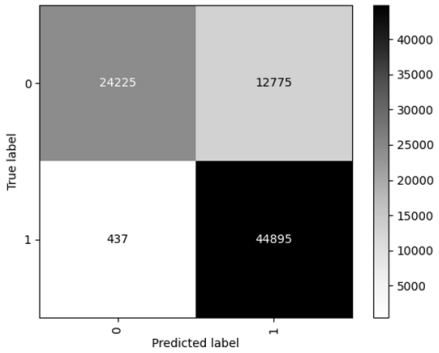

Figure 9: Select Best Feature Selection

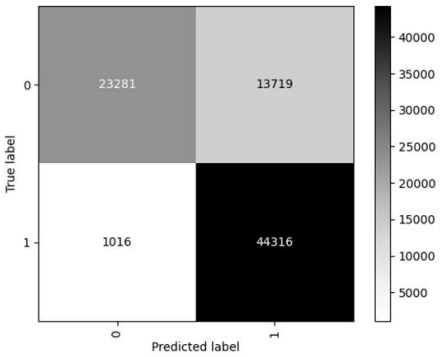

Figure 10: Tree Based Feature Selection

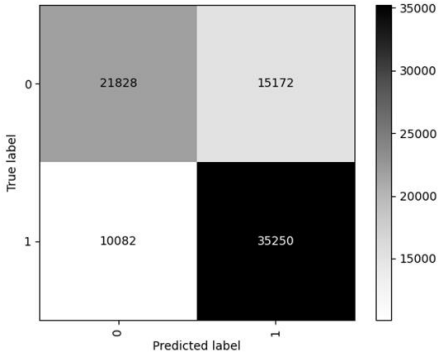

Figure 11: Sequential Feature Selection

Table 1. Outcome of the study.

Model Configuration	**Accuracy**	**F1 Score**
LSTM + Without Feature selection	85%	0.872
LSTM + PCA	76%	0.821
LSTM + Univariate Feature Selection	78%	0.836
LSTM + Recursive Feature Elimination	78%	0.837
LSTM + L1- Based Feature Selection	77%	0.829
LSTM + Select Best Feature Selection	83%	0.871
LSTM + Tree Based Feature Selection	82%	0.857
LSTM + Sequential Feature Selection	69%	0.736

Table 2. Comparison of research out come with a benchmark study on UNSW-NB15 dataset.

Research Study	**Deep Learning Model**	**Accuracy**
Alsharaiah et al. (2022)	AT-LSTM model	78.86%
Selected Model from this study	LSTM model with all features of UNSW-NB15 dataset	85%

6. CONCLUSIONS

This study conducted to investigate about the LSTM model performance for NIDS system by combining the LSTM model with different feature selection methods. Different LSTM models

with different feature selection methods developed and evaluated by using training and test dataset from the well-known UNSW-NB15 dataset repository, a benchmark dataset for network intrusion detection. Detailed literature review was conducted on NIDS system with deep learning models developed by using the benchmark network intrusion datasets. The outcome of this study demonstrated that LSTM model with full features from UNSW-NB15 dataset without applying any feature selection methods was outperforming other combination of models. Also, this model outperformed other benchmark study using LSTM methods on UNSW-NB15 dataset. However, NIDS system can further be improved by combining LSTM with other deep learning models. Also, can be explore the class imbalance of the dataset for better performance. Developing LSTM model for the multiclass label classifier is recommended in future work. In summary, this research underscores the importance of conducting further investigations on deep learning models like LSTM in NIDS.

REFERENCES

[1] S. Elsayed, K. Mohamed, and M. A. Madkour, "A Comparative Study of Using Deep Learning Algorithms in Network Intrusion Detection," *IEEE Access*, vol. 12, pp. 58851–58870, 2024, doi: 10.1109/ACCESS.2024.3389096.

[2] Department - Information Technology Greater Noida Institute of Technology (Engineering Institute) Gautam Buddh Nagar, India *et al.*, "Beyond the Firewall: Understanding and Mitigating Cloud Security Challenges," *Int. Res. J. Comput. Sci.*, vol. 11, no. 01, pp. 28–34, Jan. 2024, doi: 10.26562/irjcs.2024.v1101.06.

[3] Nasarawa State Univerisity, Nigeria and F. Isiaka, "Performance Metrics of an Intrusion Detection System Through Window-Based Deep Learning Models," *J. Data Sci. Intell. Syst.*, Oct. 2023, doi: 10.47852/bonviewJDSIS32021485.

[4] J. K. Kiruki, G. M. Muketha, and G. Kamau, "Metrics for Evaluating Alerts in Intrusion Detection Systems," *Int. J. Netw. Secur. Its Appl.*, vol. 15, no. 01, pp. 15–37, Jan. 2023, doi: 10.5121/ijnsa.2023.15102.

[5] Department of Artificial Intelligence: Wakeb Data Co., KSA and S. Kottilingal, "Deep Learning Based Network Intrusion Detection System: A Deep Abstract Networks (DANets) Model Approach," *Int. Res. J. Comput. Sci.*, vol. 11, no. 07, pp. 539–544, Jul. 2024, doi: 10.26562/irjcs.2024.v1107.01.

[6] H. C. Altunay and Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Eng. Sci. Technol. Int. J.*, vol. 38, p. 101322, Feb. 2023, doi: 10.1016/j.jestch.2022.101322.

[7] R. Kimanzi, P. Kimanga, D. Cherori, and P. K. Gikunda, "Deep Learning Algorithms Used in Intrusion Detection Systems -- A Review," Feb. 26, 2024, *arXiv*: arXiv:2402.17020. Accessed: Aug. 03, 2024. [Online]. Available: http://arxiv.org/abs/2402.17020

[8] V.-D. Ngo, T.-C. Vuong, T. Van Luong, and H. Tran, "Machine Learning-Based Intrusion Detection: Feature Selection versus Feature Extraction," Jul. 04, 2023, *arXiv*: arXiv:2307.01570. Accessed: Aug. 16, 2024. [Online]. Available: http://arxiv.org/abs/2307.01570

[9] R. Alkanhel *et al.*, "Network Intrusion Detection Based on Feature Selection and Hybrid Metaheuristic Optimization," 2023.

[10] M. A. Alsharaiah, M. Abualhaj, L. H. Baniata, A. Al-saaidah, Q. M. Kharma, and M. M. Al-Zyoud, "An innovative network intrusion detection system (NIDS): Hierarchical deep learning model based on Unsw-Nb15 dataset," *Int. J. Data Netw. Sci.*, vol. 8, no. 2, pp. 709–722, 2024, doi: 10.5267/j.ijdns.2024.1.007.

[11] I. Khan, J. Khan, S. H. Bangash, W. Ahmad, A. I. Khan, and K. Hameed, "Intrusion Detection Using Machine Learning and Deep Learning Models on Cyber Security Attacks," *VFAST Trans. Softw. Eng.*, vol. 12, no. 2, Art. no. 2, Jun. 2024, doi: 10.21015/vtse.v12i2.1817.

[12] Y. K. Saheed, T. O. Kehinde, M. Ayobami Raji, and U. A. Baba, "Feature selection in intrusion detection systems: a new hybrid fusion of Bat algorithm and Residue Number System," *J. Inf. Telecommun.*, vol. 8, no. 2, pp. 189–207, Apr. 2024, doi: 10.1080/24751839.2023.2272484.

[13] "Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011."

[14] J. Ansel *et al.*, "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, La Jolla CA USA: ACM, Apr. 2024, pp. 929–947. doi: 10.1145/3620665.3640366.

AUTHOR

Shahir Kottilingal is currently working as Principal AI Scientist at Wakeb Data in KSA. He is currently PhD scholar at Suresh Gyan Vihar University, Jaipur. His research interests include AI, Machine learning, Deep learning, Bioinformatics, Generative AI, cyber-Security etc.