Project Work Report on

A novel clustering-based autoencoder driven intelligent intrusion detection approach

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

Alimurtaza Merchant (191IT205)

Suyash Chintawar (191IT109)

Naveen Shenoy (191IT134)

Sohanraj R (191IT149)

under the guidance of

Dr. Nagamma Patil

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575025

April 2023

DECLARATION

We hereby *declare* that the Project Work Report entitled ***"A novel clustering-based autoencoder driven intelligent intrusion detection approach"*** , which is being submitted to the **National Institute of Technology Karnataka, Surathkal**, for the award of the Degree of Bachelor of Technology in Information Technology, is a *bonafide report of the work carried out by us.* The material contained in this Project Work Report has not been submitted to any University or Institution for the award of any degree.

(1) Alimurtaza Merchant (191IT205)

(2) Suyash Chintawar (191IT109)

(3) Naveen Shenoy (191IT134)

(4) Sohanraj R (191IT149)

Department of Information Technology
Place : NITK, Surathkal
Date : 03/04/2023

CERTIFICATE

This is to *certify* that the Project Work Report entitled ***"A novel clustering-based autoencoder driven intelligent intrusion detection approach"*** submitted by

(1) Alimurtaza Merchant (191IT205)

(2) Suyash Chintawar (191IT109)

(3) Naveen Shenoy (191IT134)

(4) Sohanraj R (191IT149)

as the record of the work carried out by them, is *accepted as the B.Tech. Project Work report submission* in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Information Technology in the Department of Information Technology.

Dr. Nagamma Patil
Assistant Professor
Department of Information Technology
NITK Surathkal

ABSTRACT

Network intrusion detection is an essential task in the current age of technology. It is thus necessary to develop robust systems capable of detecting network intrusions by learning their patterns. A wide variety of approaches employing machine learning and deep learning techniques such as SVMs, LSTMs, MLP and auto-encoders have been tested. The NSL-KDD dataset has been used as a benchmark. In this study, we propose a novel clustering-based auto-encoder driven intrusion detection system. Clustering allows to find groups with similar properties in the dataset. Individual models i.e, LSTM and auto-encoder based classifiers are developed for each cluster and combined for final prediction on the NSL-KDD test set. In addition, rigorous data preprocessing is performed in the form of outlier analysis using statistical methods such as Median Absolute Deviation Estimator (MADE) and feature extraction to obtain the final set of features to be used for prediction. We perform both binary classification which classifies a packet as intrusive or safe, as well as multi-class classification where a network packet is classified into either normal or one of the three intrusive categories (Probe, DOS, R2L) respectively. Experimental results show our method outperforms current baselines as well as some of the state-of-the-art scores on the NSL-KDD dataset with accuracy scores of 90.17% and 83.06% in the binary and multi-class classification tasks respectively.

Keywords— Autoencoder, Clustering, Intrusion detection, LSTM, NSL-KDD

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

A security tool called a Network Intrusion Detection System (NIDS) is made to track and examine network traffic for nefarious or suspicious activity. In contrast to host-based intrusion detection systems (HIDS), which monitor individual hosts or devices, NIDS is a form of intrusion detection system (IDS) that is concentrated on analysing network traffic. A NIDS's primary function is to immediately recognise and respond to network-based assaults. NIDS are crucial because they enable organisations to recognise and respond to cyberattacks before they do serious harm. Network traffic can be examined for indications of infiltration or malicious behaviour.NIDS can notify security staff of possible dangers and give them the knowledge they need to take precautions or lessen the effects of an assault..

A crucial component of NIDS is machine learning, which enables the system to automatically recognise and categorise patterns in network traffic that could be signs of an attack. The use of brute-force assaults, port scanning, and attempted SQL injection are just a few examples of harmful activity that may be identified by machine learning algorithms. Modern network security relies heavily on machine learning algorithms because they can analyse vast amounts of network traffic data to spot trends that may be hard for human analysts to see. Moreover, NIDS that make use of machine learning are better able to recognise previously unidentified or zero-day assaults because they can continually learn about and adjust to new and emerging threats. This enables security professionals to keep their organization's network and data secure and to stay one step ahead of thieves. A machine learning approach called clustering is used to put similar objects or data points together according to their traits or qualities.

Clustering can be used in the context of network intrusion detection systems (NIDS) to find patterns or abnormalities in network data that can point to a possible assault. In NIDS, clustering refers to grouping or clustering network traffic according to commonalities in their traffic patterns. It is possible to examine these clusters to find unusual traffic patterns that could point to a security problem. In order to make it simpler to identify and respond to possible threats, clustering may also be used to categorise traffic based on its characteristics or behaviour. By giving a mechanism to categorise network traffic into distinct groups or clusters according to their characteristics, clustering might be helpful in a machine learning-based NIDS. This may be especially helpful in spotting new or emerging risks that haven't yet been noticed. NIDS can immediately spot anomalies or outliers in the traffic and notify security teams of possible risks by clustering network traffic data. Clustering can also aid in lowering the NIDS false-positive rate. False positives occur when an NIDS picks up malicious traffic as an attack or threat. NIDS can use clustering to gather traffic patterns that are similar, and by examining these groupings, it can spot patterns that point to malicious activity. Generally speaking, clustering is a key technique in machine learning-based NIDS because it can increase the precision and efficacy of network intrusion detection by identifying anomalous traffic patterns and classifying them according to their characteristics, making it simpler for security teams to identify and address potential security threats.

The rest of the paper is organised as follows. Chapter 2 discusses the various related works and prior studies done in the field of network intrusion detection using a variety of machine learning and deep learning techniques. In Chapter 3, we propose out novel clustering based intelligent intrusion detection system thoroughly. Chapter 4 shows the various experimental details and results obtained by our model. Finally, we conclude the study in Chapter 5.

Chapter 2

LITERATURE REVIEW

2.1 Background and Related Works

A lot of research has been done to solve the problem of intrusion detection using various data mining techniques like machine learning, statistical or knowledge-based techniques. In Halimaa A. and Sundarakantham (2019) authors have used SVM and Naive Bayes to solve this problem. They used the (Tavallaee et al., 2009a) NSL-KDD dataset and perform comparative analysis on both algorithms and found SVM outperforms Naive Bayes. Desai and Gopalakrishnan (2023) worked on ML-based models on (Homoliak et al., 2020) ASNM dataset and achieved accuracy of 99%.

In the work of Al-Yaseen et al. (2017) have used the popular (Tavallaee et al., 2009b) KDD CUP 1999 dataset and developed a hybrid model of SVM and Extreme Learning Machine(ELM). They used the KMeans clustering algorithm to create a subset of the dataset to reduce the time of training. They have achieved an accuracy of 95.75 on the KDD CUP dataset. They also performed comparative analysis and found that ELM alone outperforms SVM. Dongdong et al. (2022) proposed an optimized algorithm to improve the performance of the intrusion detection system by using the Radial basis kernel function to map the features into high dimensional space. They used the KDDCUP99 dataset to analyze their IDS system. Li et al. (2010) attempts to use a semi-supervised technique.

Deep learning also has been introduced in Intrusion detection systems and works like aut (2020) have done a comparative analysis and found autoencoder outperforms the machine learning algorithms. In this work, the authors also performed a detailed analysis of different preprocessing steps which improved the results significantly. In some works, clustering has been used to create clusters of the dataset and apply classifier models on each cluster. Works like Zhong et al. (2020) have used two kinds of features namely, behavioral features and content features for this task and found this clustering-based method has improved the results significantly. We use these works as our working base to develop a robust model which gives results better than state-of-the-art work, described in detail in the following sections.

Chapter 3

PROPOSED METHODOLOGY

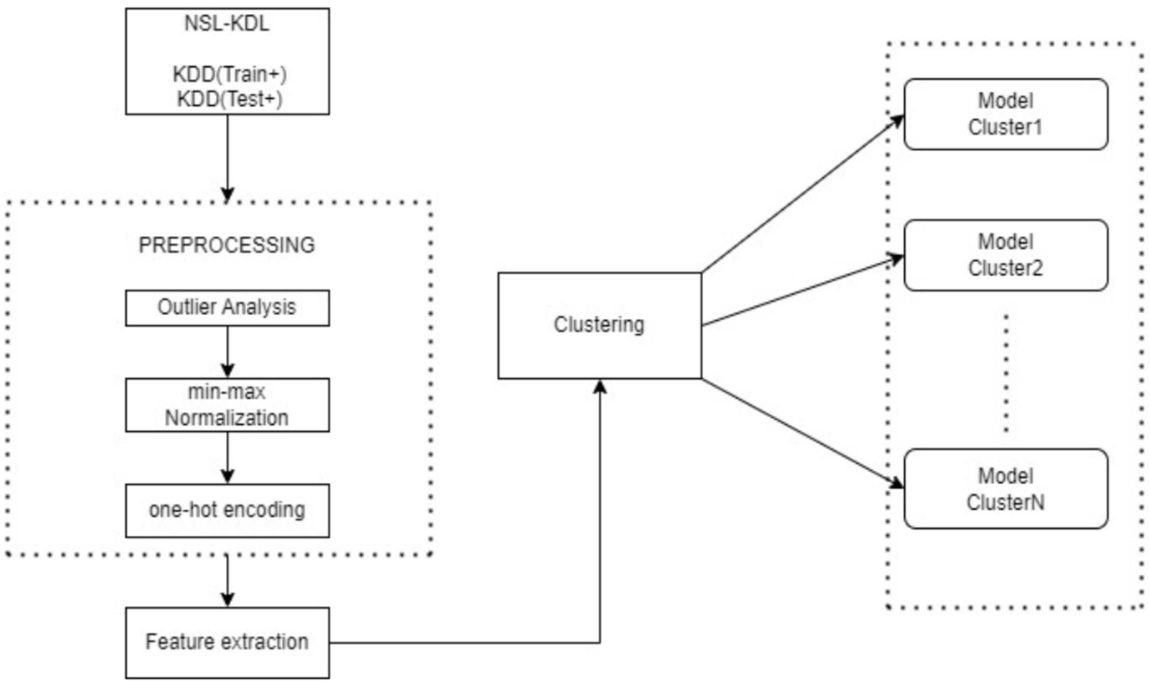

Figure 3.0.1: Proposed Methodlogy

Fig. 3.0.1 depicts the proposed methodology. In this section we discuss the same in detail. Multiple steps are involved starting from basic data preprocessing followed by outlier analysis, data normalization, One hot encoding, feature extraction and finally the classification step for the task of intrusion detection. Each of these steps are explained below.

3.1 Data Preprocessing

Multiple preprocessing techniques have been employed to further clean the original raw data which helps in improving the performance of the deep learning model. The stages of preprocessing are discussed below.

3.1.1 Low-level Preprocessing

The first step involves preprocessing of the raw data. The NSL-KDD dataset has been used in this context. Firstly, as the raw dataset does not contain column names, we assign respective column names as given in the description of the official dataset. Furthermore, the attack labels for each sample are converted to their respective attack classes namely, DoS, U2R, Probe, and R2L. Finally, the dataset is also parsed to check for any missing values or null data.

3.1.2 Outlier Analysis

The next step after preprocessing involves outlier analysis. Here, the data outliers are removed because they affect the model training and decrease the prediction ability in intrusion detection systems. It is to be noted that outlier analysis is only performed on the numeric data of the dataset. The Mean Absolute Deviation technique (MAD estimator a.k.a MADE) is used to detect outliers in the data. The formula of MADE is given as follows in eq 3.1,

$$MADE = P * med(|d_{ij} - |med(d_{ij})||) \tag{3.1}$$

where d_{ij} is the jth feature of the ith data sample. P is a multiplicative constant with value 1.4826. A data sample is called as outlier if it follows eq. 3.2, where p = 10.

$$d_{ij} > p * MADE \tag{3.2}$$

This outlier analysis has been performed on both the train and test sets of the NSL KDD dataset.

3.1.3 Normalization

To improve the performance of the classifier, normalization technique has been used to scale the data values between -1 and 1 bu making the use of min max normalization. The formula for used for scaling a numeric value X in -1 to 1 is given by eq. as follows,

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.3}$$

3.1.4 One-hot encoding

As we use preprocessing techniques like outlier analysis and min max normalization for the numeric data points, similarly, one hot encoding is used to convert the categorical features to their respective numeric forms. In the NSL-KDD dataset there are three categorical features namely, protocol type, service and flag. The domain of values of each of these three features is considered as a new feature independently whose presence marks a 1 and absence marks 0 in the converted dataset. Hence, the dimensionality of the dataset increases after one hot encoding.

3.1.5 Feature Extraction

To reduce the computational complexity required for training and eliminating features which have less impact on the overall performance, the technique of feature extraction is adopted. Here pearson correlation coefficient (PCC) is used to find most similar features. The features which gave correlation value greater than 0.5 were then selected in the final set of features used for model training. The formula for pearson correlation coefficient is gioven by eq. 3.4,

$$PCC_{XY} = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}} \tag{3.4}$$

The selected numerical features were then combined with the one hot encoded categorical features to obtain the final dataset. The same feature extraction technique was used to obtain the datasets for binary as well as multi-class classification.

3.2 Classification using Clustering

To improve overall performance of the intrusion detection model, we use a clustering based classification approach to enhance the prediction accuracy of the model. In the proposed methodology, Fig. 3.0.1, it can be seen that after feature extraction,

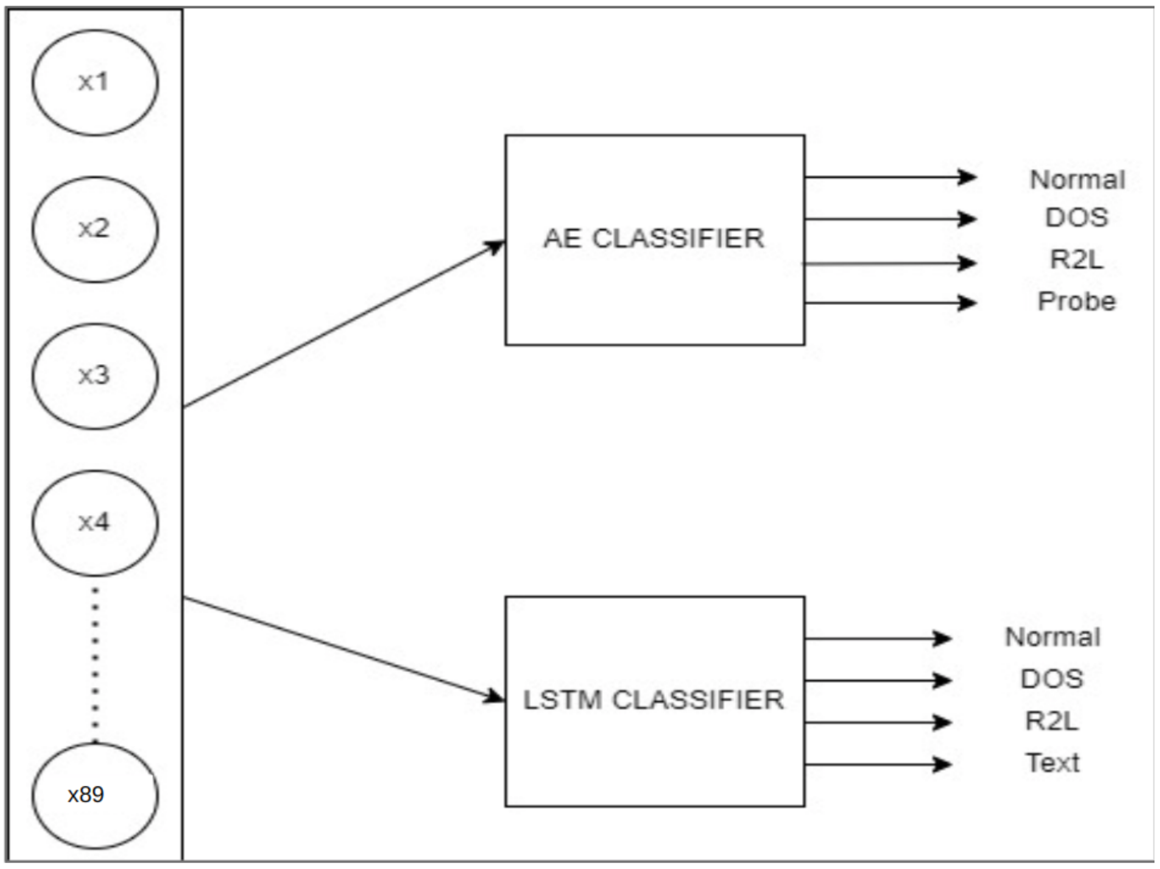

Figure 3.2.1: Classification using LSTMs or Autoencoders

clustering of the data samples is performed to identify similar data points. K-means clustering has been used for this purpose. The LSTM or autoencoder classifiers are then trained independently on each of the formed clusters. To obtain predictions on the test data, each test data point is assigned to one of the clusters formed by the training data. The predictions of the test data points are then obtained using the predictions of the respective cluster.

Fig. 3.2.1 shows the data points represented as feature vectors which are classified using LSTM or autoencoder classifiers. For binary classification, each data point is classified as either normal or attack packet. For multi-class classification, each data point is classified as either normal or one the three attack types (DoS, R2L, Probe). In this research we also develop and train an autoencoder classifier. It is used to encode the original feature vector into a compressed form. The autoencoder usually consist of three parts, namely encoder, code and decoder. They are feed forward neural networks who have the same input as well as output. The aim of the autoencoder is to reconstruct the input by minimizing the loss. The weights of the compressed form of the input feature vector, usually known as *code*, is then used in the classifier model. It is important to note that the number of nodes in the code layer is usually a hyperparameter. Also, the encoder and decoder have the same number of neural network layers arranged in laterally inverted fashion.

Chapter 4

EXPERIMENTAL SETUP & RESULTS

4.1 Dataset

The NSL-KDD dataset is used as the benchmark for this study. The Canadian Institute of Cybersecurity has made NSL-KDD available for free. The NSL-KDD improves some of the prior KDD99 benchmark's drawbacks, such as redundant and duplicate entries in sets for training and testing, which skew classifiers towards more regular samples. Fig. 4.1.1 represents the data distribution of data points in the training data of NSL-KDD dataset.

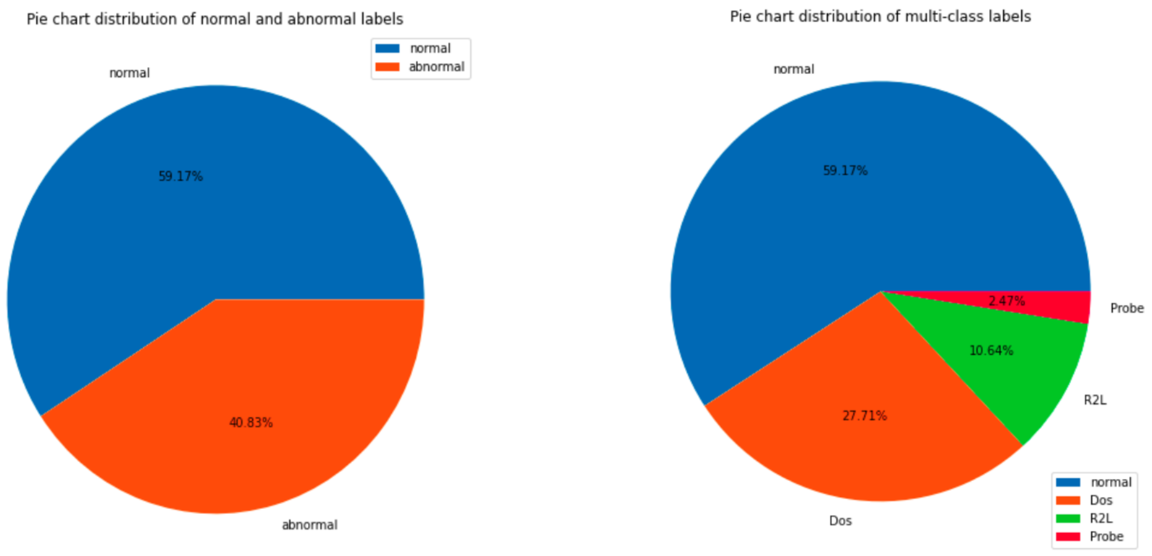

Figure 4.1.1: Pie distribution of binary and multiclass data

Each data point contains a set of 41 features out of which 38 are numeric and 3 are categorical. Source bytes, destination bytes, number of failed logins, wrong fragment are some examples of numeric features. The three symbolic features are protocol type, service and flag. A total of four types of attacks are covered in the dataset i.e, DoS, R2L, U2R and Probe. While these are the four attack classes, the packets are classified on their attack profiles. Each attack class contains multiple attack profiles. The dataset contains a total of 21 attack profiles which are converted into their respective attack classes for classification. The NSL-KDD dataset consists of a train set (NSL-KDD train) and test sets (NSL-KDD test) which contain 125,973 and 18,793 instances respectively.

4.2 Evaluation Metrics

The evaluation metrics used to evaluate the performance of the developed models are accuracy, precision and recall.

- **Accuracy**: Accuracy is the percentage of packets that are correctly classified i.e intrusive/safe in the case of binary classification or into one of normal, DOS, R2L, Probe in the case of multi-class classification to that of total packets evaluated.

- **Precision**: Precision is the total percentage of packets that are actually intrusive out of the the ones that are classified as intrusive in the case of binary classification. In other words, it is the number of true positives divided by sum of true positives and false positives.

- **Recall**: Recall is the percentage of packets that are classified as intrusive out of the total number of intrusive packets in the case of binary classification. In other words, it is the number of true positives divided by sum of true positives and false negatives.

4.3 Results and Discussion

Experiments involved performing the multi-class classification and binary classification task for the LSTM and Auto-Encoder based models individually, using LSTM and Auto-Encoder models for all clusters individually and finally using a combination of Aito-Encoder and LSTM models for different clusters and performing final prediction on the NSL-KDD test dataset.

Table 4.3.1: Results: Multi-class classification on NSL-KDD dataset

Model	Accuracy	Precision	Recall
LSTM	77.73	78.01	77.43
Auto-Encoder	80.22	79.23	79.21
Clustered LSTM	76.29	79.68	76.29
Clustered Auto-Encoder	82.40	80.55	82.40
Clustered Combined (proposed)	**83.06**	**82.04**	**83.06**

4.3.1 Multi-class classification

A regular LSTM based model obtained accuracy of 77.73% in multi-class classification. The auto-encoder based model without clustering obtained an accuracy of 80.22%. After performing clustering of the data points of the NSL-KDD train dataset, we perform evaluation traning either LSTM or Auto-encoder models on all the clusters. The results obtained with clustered LSTM were accuracy of 76.29%, precision of 79.68% and recall of 76.29%. Clustering with only auto-encoders resulted in accuracy of 82.4%, precision of 80.55% and recall of 82.40%. The final combined model involves including the model with the best performance on each cluster as the classifier for that cluster in the combined model. Thus, finally, when data points were clustered into four clusters, we obtain LSTM predictors and Auto-encoder based predictors for 2 clusters each respectively. The final performance obtained was accuuracy of 83.06%, precision of 82.04% and recall of 83.06%.

4.3.2 Binary classification

Table 4.3.2: Results: Binary classification on NSL-KDD dataset

Model	Accuracy	Precision	Recall
LSTM	80.68	71.07	87.64
Auto-Encoder	82.75	72.88	87.39
Clustered LSTM	81.25	71.76	**88.03**
Clustered Auto-Encoder	89.32	86.03	87.68
Clustered Combined (proposed)	**90.17**	**90.07**	84.93

A regular LSTM based model obtained accuracy of 80.68% in binary classification. The auto-encoder based model without clustering obtained an accuracy of 82.75%.

The results obtained with clustered LSTM were accuracy of 81.25%, precision of 72.88% and recall of 87.39%. Clustering with only auto-encoders resulted in accuracy of 89.32%, precision of 86.03% and recall of 87.68%. Thus, finally, when data points were clustered into four clusters, we obtain LSTM predictors and Auto-encoder based predictors for 2 clusters each respectively as obtained in the multi-class classification. The final performance obtained was accuracy of 90.17%, precision of 90.07% and recall of 84.93%.

Chapter 5

CONCLUSION

This research attempts to tackle the problem of intrusion detection by using a novel clustering based classification approach. The NSL-KDD dataset was used for this purpose. K-means clustering was used to cluster the training samples into multiple clusters and deep learning models like Long Short Term Memory (LSTM) and Autoencoders were employed for the task of classification. The use of autoencoders boosted the performance of the models significantly as it helped in compressing the original feature vectors efficiently. This in turn helped in fine-tuning some of the layers in the autoencoder classifiers by sharing the trained weights of the code layer in the autoencoder. Deep learning models were built for the attack types or class labels as DoS, R2L and Probe along with the normal class labels in case of multi-class classification and just the presence or absence of an attack in packet for binary classification. It was observed that the proposed approach performed comparable to the state-of-the-art techniques, rather, outperformed in some of the scenarios. The autoencoders performed better than the LSTM classifiers for this particular task. Also, the combined model of LSTM and autoencoders outperformed each of the models when used independently.

REFERENCES

(2020). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing*, 387:51–62.

Al-Yaseen, W. L., Othman, Z. A., and Nazri, M. Z. A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. *Expert Systems with Applications*, 67:296–303.

Desai, R. and Gopalakrishnan, V. T. (2023). Network intrusion detection through machine learning with efficient feature selection. In *2023 15th International Conference on COMmunication Systems NETworkS (COMSNETS)*, pages 797–801.

Dongdong, L., Hongtao, D., Bo, H., and Lei, N. (2022). An optimized network intrusion detection model. In *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, volume 5, pages 227–232.

Halimaa A., A. and Sundarakantham, K. (2019). Machine learning based intrusion detection system. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 916–920.

Homoliak, I., Malinka, K., and Hanacek, P. (2020). Asnm datasets: A collection of network attacks for testing of adversarial classifiers and intrusion detectors. *IEEE Access*, 8:112427–112453.

Li, K., Zhang, Z., and Liu, M. (2010). One data preprocessing method in high-speed network intrusion detection. In *IET 3rd International Conference on Wireless, Mobile and Multimedia Networks (ICWMNN 2010)*, pages 60–63.

Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (2009a). A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6.

Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (2009b). A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6.

Zhong, W., Yu, N., and Ai, C. (2020). Applying big data based deep learning system to intrusion detection. *Big Data Mining and Analytics*, 3(3):181–195.