Wasit Journal for Pure Science

Journal Homepage: https://wjps.uowasit.edu.iq/index.php/wjps/index
e-ISSN: 2790-5241 p-ISSN: 2790-5233

Enhancing Intrusion Detection with Autoencoder Based Classifier and Statistical Feature Selection

Abbas Fadhil Hamzah Alharan[1]*

[1]Computer science department, Faculty of education for girls, University of Kufa, Najaf, IRAQ

*Corresponding Author: Abbas Fadhil Hamzah Alharan

DOI: https://doi.org/10.31185/wjps.257

Received 13 October 2023; Accepted 04 December 2023; Available online 30 December 2023

ABSTRACT: In today's digital landscape, the rapid expansion of computer networks and the increasing reliance on information technology have made network security a paramount concern. With the growing sophistication of cyber threats, traditional intrusion detection systems (IDS) face significant challenges in effectively identifying and mitigating security breaches. To address these evolving threats, novel approaches that combine cutting-edge technologies are required. This paper explores the fusion of autoencoder based classifier to training and classifying the attacks of IDS. This approach is applied on the most meaningful feature that selected based on the pearson correlation (for continues vales) and chi-square test (for binary values). The benchmark NSL-KDD database is utilized to assess the validity of the suggested IDS. The experimental outcomes demonstrate that the designed Intrusion Detection System (IDS) attains superior classification accuracy at 92.27%, outperforming both prior research efforts and alternative classifier methods.

Keywords: Intrusion detection system, Autoencoder, classification, Feature selection, Chi-square, Pearson correlation.

1. INTRODUCTION

In today's interconnected world, the use of internet services and communication has become pervasive, offering unprecedented convenience and opportunities. However, it also comes with its share of risks that users need to be aware of [1]. The threats such as hacking and malware, pose considerable risks to internet services and communication platforms. The significant increase in the number of intrusions over the years is indicating a continuous upward trend[2]. Solving the risks associated with digital communication is not only about individual protection but also about creating a secure, trustworthy, and resilient digital ecosystem that underpins modern society. It involves a collective effort from specialized researchers, businesses, governments, and technology providers to implement effective security measures, policies, and practices. Intrusion detection is a critical component of cybersecurity, serving as the first line of defense against unauthorized access, data breaches, and malicious activities within computer networks[3]. Traditional IDSs typically rely on rule-based or signature-based methods to detect known threats, making them vulnerable to zero-day exploits and highly adaptive attackers[4]. Moreover, these systems often generate a high rate of false positives, leading to alert fatigue and making it challenging for security teams to identify genuine threats amidst the noise.

In recent IDS, researchers have ventured into the realm of machine learning (ML) and deep learning (DL) techniques. Within the broader field of artificial intelligence (AI), ML and DL are complementary approaches that share the goal of extracting meaningful insights from large datasets. [5]. These approaches have gained significant traction in the field of network security, particularly in the past decade, largely owing to the advent of highly potent graphics processing units (GPUs). ML-based IDS relies heavily on feature engineering to glean insights from network data, while DL-based IDS stands out for its capacity to automatically extract intricate patterns from raw data due to its deep architecture[6]. In IDS, Striking a balance between capturing genuine patterns and avoiding over-fitting can be challenging. This challenge cannot be addressed by combining ML techniques with conventional data analysis methods. Conventional ML techniques may struggle when faced with large and complex dataset. As well as, it is not being well-suited to capture and understand intricate, non-linear patterns or relationships within the data. To tackle the identified limitations and enhance the IDS performance, we propose a strategy that integrates deep learning with statistical data analysis. Deep learning is crucial for extracting intricate patterns and representations from complex data, enabling advanced solutions in tasks such as image recognition [7], [8], natural language processing[9], [10], and in recent times, it has been utilized in cybersecurity[11], [12].

In this work, we incorporate an autoencoder-based classifier to improve intrusion detection. Autoencoder technology combined with a classification layer accomplishes two things: it learns from the data at first to improve the input representation, and second, it classifies instances into two different classes (normal and abnormal). The model is applied to features statistically selected from the NSL-KDD dataset.

2. RELATED WORKS

Intrusion detection systems (IDSs) are pivotal for securing network environments by identifying and mitigating potential threats. While several studies have delved into the development and evaluation of IDSs, there exists a need for a more comprehensive analysis and comparison of existing methodologies. In particular, the use of the NSL-KDD dataset has been prevalent as a benchmark in this domain.

In the realm of ensemble methods, [13] introduced an ensemble IDS for machine learning, employing PCA for feature extraction. The presented ensemble method, combining Decision Tree, Random Forest, KNN, DNN, and MultiTree, demonstrated an accuracy of 85.2%, surpassing the performance of individual algorithms. Similarly, Kumar proposed a novel approach in [14] that integrates a multi-objective genetic algorithm with neural networks to form an ensemble of solutions for efficient network intrusion detection. Their evaluation, conducted on benchmark datasets such as NSL_KDD and ISCX-2012, yielded a detection accuracy of 97% for ISCX-2012 and 88% for NSL_KDD datasets. Building on fuzzy and ensemble learning theories, [15] developed a semi-supervised learning technique, achieving an accuracy of 84.54% using the NSL-KDD dataset. In a dual ensemble approach, [16] fused bagging and gradient boosting decision tree (GBDT) algorithms to address network anomaly detection challenges. Experimental findings from various intrusion detection datasets showcased the effectiveness of combining bagging with finely-tuned gradient boosting.

Instead, [17] applied an efficient Deep Neural Network (DNN) architecture on six datasets, including NSL-KDD, demonstrating the superiority of their suggested DNN over conventional classifiers. [18] Introduced a 5-layer autoencoder (AE) model tailored for improved detection of anomalous network traffic, achieving a peak accuracy of 90.61% on the NSL-KDD dataset. This innovative model enhances the landscape of anomaly detection. Moving beyond traditional datasets, [19] employed eight machine learning algorithms to identify DDoS attacks in the context of IoT. Their experimental findings favored the BiLSTM architecture, exhibiting the highest accuracy and suitability for diagnosing DDoS attacks in IoT. Yu et al. [20] leveraged Few-Shot Learning (FSL) in their IDS model, incorporating CNN and DNN for feature extraction. Impressively, the model achieved accuracy rates of 92.34% and 92% on the UNSW-NB15 and NSL-KDD datasets, respectively, demonstrating its effectiveness with limited data. Kasongo's study [21] showcased the use of various RNN algorithms alongside the XGBoost feature selection method to construct an IDS framework. The XGBoost-LSTM excelled in binary classification, achieving an accuracy of 88.13% on the NSL-KDD dataset, while the XGBoost-Simple-RNN achieved 87.07% on the UNSW-NB15 dataset.

This is how the remainder of the article is organized: The methods and materials are provided in section 3. Section 4 provides a full overview of the proposed system outcomes and discussions. At the end, Section 5 provides a final summary of the work and a roadmap for the future.

3. METHODOLOGY OF THE PROPOSED APPROACH

We start this section by providing an overview of the NSL-KDD dataset. Following that, we provide a comprehensive outline of the suggested methodology, which consists of four fundamental steps: data preprocessing, feature selection, classification, and model evaluation. To enhance clarity, these steps are visually depicted in Fig. 1, offering an illustrative representation of the entire workflow.

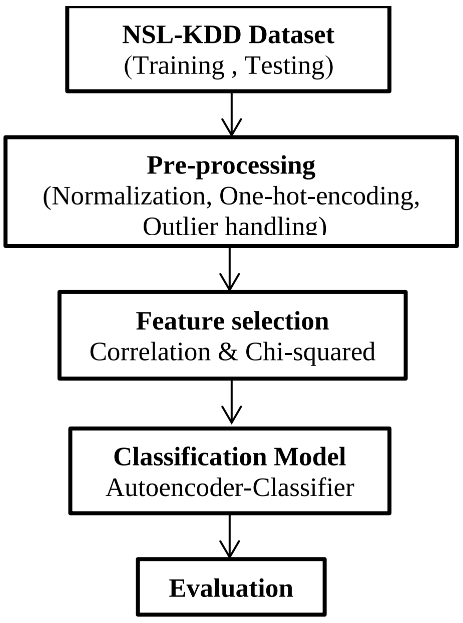


Figure1. Workflow of the proposed approach

3.1 Description of dataset

The NSL-KDD represents an enhanced edition of the widely used KDD Cup 1999 dataset, which is often used for intrusion detection and network security research. It is available in [22]. The dataset include training and testing sets, with 125973 and 22544 instances respectively. In the testing set, there are seventeen attack types that do not appear in the training set, while in the training set, two attack types are absent from the testing set. To ensure fair classification, these particular attack types were removed from consideration. Moreover, the dataset comprises 43 features, with two features ('difficulty_level', 'num_outbound_cmds') excluded from consideration as they do not have a significant impact on the classification process. All types of attack (DoS, R2L, U2R, and Probe) are converted to abnormal (0) output. In the end, the dataset comprises 41 features (1 output attribute, 37 continues attributes, and 3 symbolic attributes). Table 1 displays the dataset details prior to the preprocessing stage.

Table 1. – NSL-KDD dataset details

NSL-KDD dataset	Total	Normal(1)	Abnormal(0)
Training set	125081	67343	57738
Testing set	18794	9711	9083
Total	143875	77054	66821

3.2 Data pre-processing

The initial preprocessing phase serves to prepare the data for seamless processing by subsequent modules. This step encompasses three key stages: an examination of outliers, data normalization, and the application of one-hot encoding.

- **Examination of outliers**: z-score method is utilized in this stage to detect the outliers. It involves identifying data points that deviate significantly from the mean of the dataset, often indicating the presence of extreme values. For every data point within the dataset, determine its z-score using the formula provided in reference [23]. If the absolute value of the z-score for a data point surpasses a specified threshold, that data point is identified as an outlier. These outliers can then be managed by excluding them from the dataset. After removing the outliers, the number of instances in training and testing sets will be 102676 and 13168 respectively.

- **Data normalization**: Here, the numeric values of data are rescaled using *StandardScaler* method. It involves changing data so that its mean is equal to zero and its standard deviation is equal to one. See the equation(1), where the $x_{i,j}$ value of *i*th feature in data transformed to $z_{i,j}$. μ_i and σ_i represent to the mean and standard deviation of the feature respectively.

$$z_{i,j} = \frac{x_{i,j} - \mu_i}{\sigma_i} \quad (1)$$

- **One-hot encoding**: In the realm of data preprocessing, This encoding is employed to depict categorical variables as binary vectors [24]. This process entails converting categorical data into a format that can be effortlessly comprehended and processed by learning models. There are three categorical attributes (protocol type, service, and flag) in dataset are converted to binary values using one-hot encoding. As an illustration, let's consider the "protocol type" attribute with three distinct values: *tcp*, *udp*, and *icmp*. By employing the one-hot encoding, the values were transformed into binary vectors (1, 0, 0), (0, 1, 0), and (0, 0, 1) respectively. Likewise, the "service" and "flag" attributes underwent a similar transformation into one-hot encoding vectors. Consequently, the initial three attributes have been expanded to a total of 84 attributes. When combined with the existing 37 continuous attributes, the dataset's overall attribute count now amounts to 121.

3.3 Feature selection

Feature selection is the procedure of picking relevant features to include in a dataset from the original set of features. To enhance the efficacy of machine learning models, feature selection aims to reduce dimensionality while focusing on the most valuable characteristics [25]. In this research, we utilize two distinct statistical feature selection methods: one based on Pearson correlation for selecting the top 20 continuous features from the processed data, and the other based on the chi-square test for selecting 20 features with binary values from the preprocessed dataset.

- Pearson's correlation coefficient is utilized to assess the linear connection between the distance-related variables. In this study, we employ the Pearson's correlation for the purpose of feature selection. The values produced by Pearson's correlation range from -1 to 1. There is a perfect positive correlation when the value is +1, 0 indicates that there is no linear link between the two variables, and the value -1 refers to a perfect negative correlation [26]. The formula for calculating Pearson's correlation coefficient between the variables x and y is described in [27].

- If two categorical variables are significantly associated, it can be ascertained using a statistical test called the chi-square (χ^2) test. In contingency tables, where the variables are tabulated versus one another, it is frequently used to analyze data. With the assumption that the variables are independent, the test determines whether the observed data distribution significantly deviates from the expected distribution.. In [28], the Chi-square statistic is defined.

3.4 The proposed AE based classifier

In our research, we crafted an Autoencoder (AE) that leverages the capabilities of classifier. The features that were chosen in the previous stage (section 3.3) will be employed as the AE-classifier's input. The AE architecture involves two core operations: an encoder, which reduces the dimensionality of the input data, followed by a decoder, which strives to reconstruct the initial input from the compressed representation [29]. This autoencoder is trained using unsupervised learning, making it proficient at extracting meaningful features from unlabeled data. The main structure of AE is described in the Fig2. Where x is the input data, y is the encoded data, and z is the decoded data. Table 2 provides information on the dataset utilized as input data for the model.

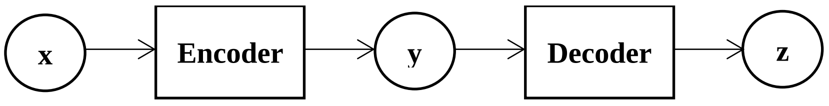


Figure2. The main structure of AE

Table 2. – NSL-KDD dataset splitting details

NSL-KDD dataset	Total	Normal(1)	Abnormal(0)
Training set	102676	57146	45530
Testing set	13168	8253	4915
Total	115844	65399	50445

Figure 3(a) illustrates the proposed autoencoder framework (AE[40:30:40]). Specifically, this autoencoder takes a 40-dimensional feature representation (x) and encodes it into a 30-dimensional vector (y), subsequently decoding it back to the original input feature space (z). During this study, the AE[40:30:40] is trained in an unsupervised manner using the Root Mean Square Propagation (RMSprop) optimization algorithm, with 30 epochs and a batch size of 200. The quality of the encoding and decoding process is measured by the mean squared error (MSE). Following the training of AE[40:30:40], the reconstructed features are used as input for the classification layer based on the sigmoid activation function (Classification layer, Figure 3(b)). At this stage, the sigmoid layer is adapted for binary classification tasks.

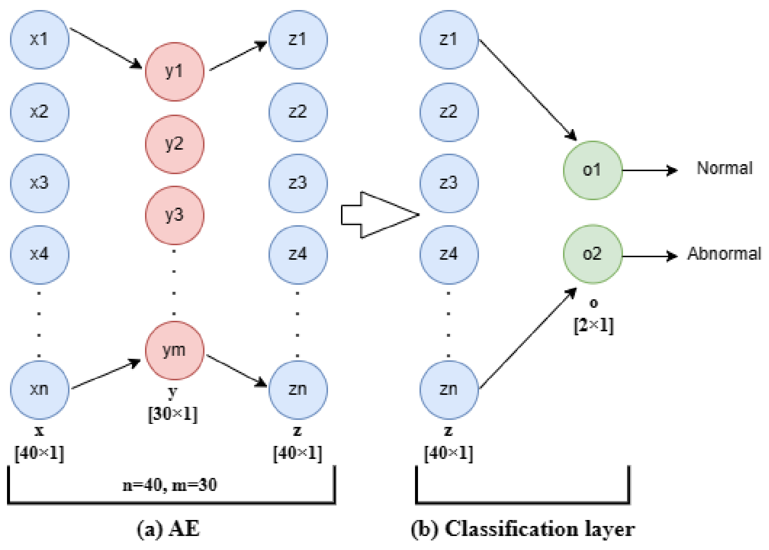


Figure 3. The proposed AE based classification, (a) The AE architecture. (b) The combined classification layer with AE

3.5 Evaluation metric

The proposed IDS's effectiveness is evaluated using four commonly employed metrics: accuracy, recall, precision, and F1-score. These metrics' equations can be discribed as follows:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

(3)

$$precision = \frac{TP}{TP + FP} \quad (4)$$

To $$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (5)$$ clarify, True Positives (TP) in an intrusion detection system represent accurate intrusion detections. On the other hand, True Negatives (TN) indicates correct identifications of non-intrusive events. False Positives (FP) occur when the system wrongly classifies non-intrusive events as intrusions. Conversely, False Negatives (FN) happens when the system fails to detect real intrusions and incorrectly labels them as non-intrusive. These terms are integral to the understanding of the evaluation metrics presented in Equations (2) to (5).

4. Results and discussion

To assess the classifier's capability to accurately discern between abnormal and normal attacks, we examined the performance of the proposed AE-classifier architectures using a binary classification approach, where instances are categorized as either "Normal" or "Abnormal. The experiments were conducted using a computer equipped with an Intel Core i5 processor from the 3rd generation running Windows 10, and the proposed model was implemented using Python 3. In the context of accuracy (as shown in Table 3), the proposed classifier demonstrated superior performance, achieving a rate as high as 92.27%. This performance surpasses that of the LDA, random forest, KNN, MLP, and SVM classifiers, which attained accuracies of 92%, 91.86%, 91.29%, 91%, and 90.94%, respectively.

Table 3. – The accuracy results of the propose classifier and other different classifiers

Classifier	Accuracy
AE-classifier	92.27%
LDA	92%
Random forest	91.86
KNN	91.29%
MLP	91%
SVM	90.94%

Table 4 show the result of common performance's measures(precision, recall and F1-score) of the proposed classifier as well as LDA, random forest, KNN, MLP, SVM.

Table 4. – The classification performance (precision, recall and F1-score) of AE based classifier, LDA, Random forest, KNN, MLP, SVM

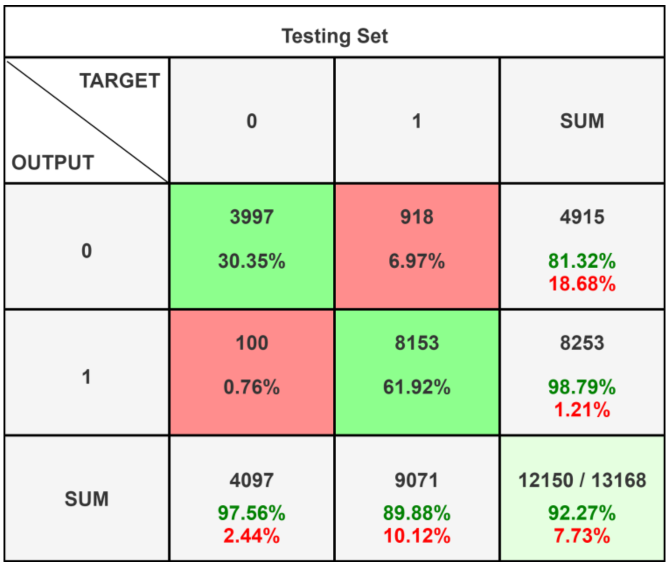
Precision						
Class of attack	AE-classifier	LDA	Random forest	KNN	MLP	SVM
Normal (1)	89.88	89.54	89.43	88.99	89.07	89.01
Abnormal (0)	97.56	97.59	97.29	96.45	95.25	95.17
avg	93.72	93.56	93.36	92.72	92.16	92.09
Recall						
Class of attack	AE-classifier	LDA	Random forest	KNN	MLP	SVM
Normal (1)	98.79	98.81	98.67	98.26	97.63	97.59
Abnormal (0)	81.32	80.61	80.43	79.59	79.88	79.78
avg	90.06	89.71	89.55	88.92	88.75	88.68
F1-score						
Class of attack	AE-classifier	LDA	Random forest	KNN	MLP	SVM
Normal (1)	94.12	93.95	93.82	93.39	93.15	93.10
Abnormal (0)	88.70	88.29	88.06	87.21	86.89	86.80
avg	91.41	91.12	90.94	90.30	90.02	89.95

Concerning precision, the AE-classifier exhibited superior performance in identifying the normal class with a score of 89.88%. On the other hand, for detecting the abnormal class, LDA performed better with a score of 97.59%, with only a marginal difference of 0.03% compared to the AE-classifier. Notably, KNN obtained the lowest precision score in detecting the normal class at 88.99%, while SVM had the lowest precision score in detecting the abnormal class at 95.17%.

Moving to recall, the AE-classifier outperformed all other classifiers in detecting the abnormal class, achieving a score of 81.32%. Conversely, for detecting the normal class, LDA had the higher recall score at 81.81%, with a slight difference of 0.02% compared to the AE-classifier. In this case, SVM obtained the lowest recall score in detecting the normal class at 97.59%, while KNN had the lowest recall score in detecting the abnormal class at 79.59%.

As for the F1 measure, the AE-classifier surpassed all other classifiers in identifying classes that are abnormal and normal, achieving scores of 94.12% and 88.70%, respectively. In contrast, SVM had the lowest F1 scores identifying classes that are abnormal and normal, scoring 93.10% and 86.80%, respectively.

Figure 4 elucidates the confusion matrix of the suggested classifier, while Figure 5 provides clarity on the confusion matrices for each of the following models: LDA, Random Forest, KNN, MLP, and SVM.



Testing Set			
TARGET / OUTPUT	0	1	SUM
0	3997 30.35%	918 6.97%	4915 81.32% 18.68%
1	100 0.76%	8153 61.92%	8253 98.79% 1.21%
SUM	4097 97.56% 2.44%	9071 89.88% 10.12%	12150 / 13168 92.27% 7.73%

Figure 4. The confusion matric of the proposed classifier (AE—classifier)

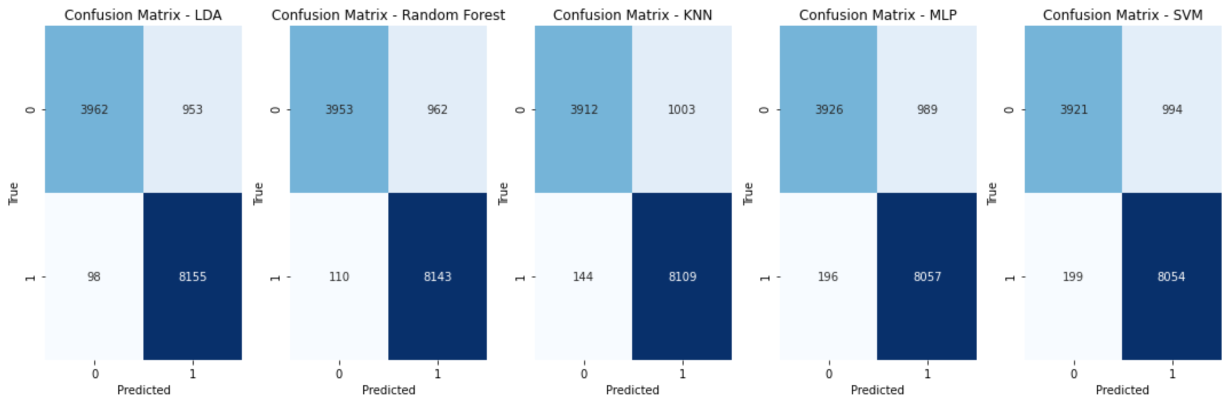


Figure 5. The confusion matrices of the existing classifiers (LDA, random forest, KNN, MLP, SVM)

Additionally, we conducted a comparative analysis of the proposed classifier against contemporary methodologies found in the literature that have employed the NSL-KDD dataset, as detailed in Table 5. Given that a significant portion of prior research has centered on distinguishing between different types of NSL-KDD attacks, our evaluation specifically focused on assessing the performance of the AE-classifier in a binary classification context.

In the early stages, researchers in [18] introduced an innovative 5-layer Autoencoder (AE)-based model designed specifically for intrusion detection. This architectural model excelled with an accuracy rate of 90.61%. Notably, this accomplishment was attributed to the utilization of a novel two-sigma (95th percentile) outlier disposal technique in conjunction with Mean Absolute Error (MAE) as the reconstruction loss metric. Similarly, in reference [30], authors proposed a novel unsupervised approach for intrusion detection, capitalizing on the advantages of deep learning and an OC (One-Class) classifier. This approach achieved an impressive accuracy score of 91.58%. Reference [31] presented a different approach by employing Extreme Learning Machine (ElM) classifiers on features selected using the Differential Evolution (DE) technique, yielding an accuracy of 87.53%. Furthermore, reference [32] introduced an architectural framework that combined Autoencoders (AE) and Long Short-Term Memory (LSTM) components. The resultant system achieved an accuracy level of 89%. Lastly, the study described in reference [33] integrated Convolutional Neural Networks (CNN) and LSTM models on data augmented through the ADASYN resampling method, achieving an impressive accuracy rate of 92%. Our analysis of the results indicated that the proposed AE-classifier surpasses the previously mentioned works regarding accuracy.

Table 5. – The accuracy result of the proposed classifier and the prior techniques

Method	Accuracy
AE by Xu et al.[18]	90.61%
1D CAE & OCSVM by Binbusayyis et al, [30]	91.58%
DE & ELM by Almasoudy et al. [31]	87.53%
AE-LSTM by Mushtaq et al. [32]	89%
CNN+LSTM+ ADASYN by Zakariah et al. [33]	92%
The proposed model	92.27%

5. CONCLUSION AND FUTURE DIRECTIONS

In this study, we present a classifier based on an Autoencoder (AE) architecture, employed as an approach for building an intrusion detection system (IDS). The NSL-KDD dataset was used as a bechmark for testing the suggested IDS. We identified the most relevant features through Pearson correlation and Chi-square tests, and these were subsequently used as input for our AE architecture, specifically designed with a single hidden layer containing 30 units. Our extensive comparative analysis involved benchmarking the AE-classifier against both traditional classifiers (Table 3 and Table 4) and contemporary techniques (Table 5). The results of this comparison clearly demonstrate that the AE30 classifier outperformed all other methods, achieving an impressive accuracy rate of 92.27%.

In our forthcoming research, we plan to delve into the creation of more complex Autoencoder (AE)-based architectures. Our aim is to investigate how adjusting the number of hidden layers, units, and activation functions within the AE architecture might offer potential enhancements in intrusion detection performance. Furthermore, we aspire to explore methods for the seamless adaptation of the intrusion detection system to real-time monitoring and response. We also seek to establish mechanisms that ensure the timely notification and effective mitigation of security threats.

REFERENCES

[1] 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, 2020.
[2] “http://www.cert.org/stats.”
[3] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.
[4] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, 2021.
[5] R. Prasad, V. Rohokale, R. Prasad, and V. Rohokale, “Artificial intelligence and machine learning in cyber security,” *Cyber Secur. Lifeline Inf. Commun. Technol.*, pp. 231–247, 2020.
[6] B. Dong and X. Wang, “Comparison deep learning method to traditional methods using for network intrusion detection,” in *2016 8th IEEE international conference on communication software and networks (ICCSN)*, 2016, pp. 581–585.
[7] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, “Facial expression recognition via learning deep sparse autoencoders,” *Neurocomputing*, vol. 273, pp. 643–649, 2018.
[8] X. Sun and M. Lv, “Facial expression recognition based on a hybrid model combining deep and shallow features,” *Cognit. Comput.*, vol. 11, no. 4, pp. 587–597, 2019.
[9] I. B. R. Salman and G. Varaprasad, “Product Recommendation System Using Deep Learning Techniques: CNN and NLP,” in *International Conference on Data Management, Analytics & Innovation*, 2023, pp. 331–343.
[10] Z. Z. Lim, “Sentiment analysis based on NLP and deep learning,” 2023.
[11] R. Brindha, S. Nandagopal, H. Azath, V. Sathana, G. P. Joshi, and S. W. Kim, “Intelligent Deep Learning Based Cybersecurity Phishing Email Detection and Classification.,” *Comput. Mater. Contin.*, vol. 74, no. 3, 2023.
[12] F. S. Alrayes *et al.*, “Enhanced Gorilla Troops Optimizer with Deep Learning Enabled Cybersecurity Threat

Detection," *Comput. Syst. Sci. Eng.*, vol. 45, no. 3, 2023.

[13] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *Ieee Access*, vol. 7, pp. 82512–82521, 2019.

[14] G. Kumar, "An improved ensemble approach for effective intrusion detection," *J. Supercomput.*, vol. 76, no. 1, pp. 275–291, 2020.

[15] Y. Gao, Y. Liu, Y. Jin, J. Chen, and H. Wu, "A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system," *IEEE Access*, vol. 6, pp. 50927–50938, 2018.

[16] M. H. L. Louk and B. A. Tama, "Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system," *Expert Syst. Appl.*, vol. 213, p. 119030, 2023.

[17] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *Ieee Access*, vol. 7, pp. 41525–41550, 2019.

[18] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, "Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset," *IEEE Access*, vol. 9, pp. 140136–140146, 2021.

[19] M. Esmaeili, S. H. Goki, B. H. K. Masjidi, M. Sameh, H. Gharagozlou, and A. S. Mohammed, "Ml-ddosnet: Iot intrusion detection based on denial-of-service attacks using machine learning methods and nsl-kdd," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022.

[20] Y. Yu and N. Bian, "An intrusion detection method using few-shot learning," *IEEE Access*, vol. 8, pp. 49730–49740, 2020.

[21] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Comput. Commun.*, vol. 199, pp. 113–125, 2023.

[22] "https://www.unb.ca/cic/datasets/nsl.html."

[23] N. B. Chikodili, M. D. Abdulmalik, O. A. Abisoye, and S. A. Bashir, "Outlier detection in multivariate time series data using a fusion of K-medoid, standardized euclidean distance and Z-score," in *International Conference on Information and Communication Technology and Applications*, 2020, pp. 259–271.

[24] A. Y. Hussein, P. Falcarin, and A. T. Sadiq, "Enhancement performance of random forest algorithm via one hot encoding for IoT IDS," *Period. Eng. Nat. Sci.*, vol. 9, no. 3, pp. 579–591, 2021.

[25] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[26] Y. Bae and H. Lee, "Sentiment analysis of twitter audiences: Measuring the positive or negative influence of popular twitterers," *J. Am. Soc. Inf. Sci. Technol.*, vol. 63, no. 12, pp. 2521–2535, 2012.

[27] F. R. S. Rangkuti, M. A. Fauzi, Y. A. Sari, and E. D. L. Sari, "Sentiment analysis on movie reviews using ensemble features and pearson correlation based feature selection," in *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2018, pp. 88–91.

[28] I. S. Thaseen, C. A. Kumar, and A. Ahmad, "Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers," *Arab. J. Sci. Eng.*, vol. 44, pp. 3357–3368, 2019.

[29] J. Zhai, S. Zhang, J. Chen, and Q. He, "Autoencoder and its various variants," in *2018 IEEE international conference on systems, man, and cybernetics (SMC)*, 2018, pp. 415–419.

[30] A. Binbusayyis and T. Vaiyapuri, "Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM," *Appl. Intell.*, vol. 51, no. 10, pp. 7094–7108, 2021.

[31] F. H. Almasoudy, W. L. Al-Yaseen, and A. K. Idrees, "Differential evolution wrapper feature selection for intrusion detection system," *Procedia Comput. Sci.*, vol. 167, pp. 1230–1239, 2020.

[32] E. Mushtaq, A. Zameer, M. Umer, and A. A. Abbasi, "A two-stage intrusion detection system with auto-encoder and LSTMs," *Appl. Soft Comput.*, vol. 121, p. 108768, 2022.

[33] M. Zakariah, S. A. AlQahtani, and M. S. Al-Rakhami, "Machine Learning-Based Adaptive Synthetic Sampling Technique for Intrusion Detection," *Appl. Sci.*, vol. 13, no. 11, p. 6504, 2023.