ENHANCING INTRUSION DETECTION SYSTEMS WITH AUTOENCODER-BASED ANOMALY DETECTION IN NETWORK TRAFFIC

[1]Venkataramesh Induru
Piorion Solutions Inc, New York, USA
venkatarameshinduru@gmail.com

[2]Priyadarshini Radhakrishnan
IBM Corporation, Ohio, USA
priyadarshinir990@gmail.com

[3]Vijai Anand Ramar
Delta Dental Insurance Company,Georgia,USA
vijaianandramar@gmail.com

[4]Karthik Kushala
Celer Systems Inc, Folsom, California, USA
karthik.kushala@gmail.com

[5]Purandhar. N
Department of CSE(Artificial Intelligence)
School of Computers
Madanapalle Institute of Technology and Science, Madanapalle
College Code - 69
Andhra Pradesh - 517325, India

ABSTRACT

The signature-based intrusion detection systems (IDS) encounter quite serious obstructions when trying to spot known threats due to the complexity of the cyberattack schemes and rapid development of the network traffic. Such complexity can foster the probability that zero-day attacks would not get detected due to an imbalanced attack dataset, which might, for example, weigh in with a minimal percentage with respect to general traffic. The use of NILM in conjunction with autoencoder interactions for intrusion detection, therefore, turns the table by providing favourable operative conditions since the autoencoder, by its very nature, learns normal traffic patterns and flags deviations from normality as anomalies and/or intrusions. The model uses reconstruction error on normal traffic data to find anomalies, thus eliminating attack signatures, making it a very strong approach to detecting unseen attacks. The system outlined in this paper was trained on normal traffic data; therefore, the model was also able to highlight rather subtle anomalies. Evaluation measures of accuracy, precision, recall, and F1-score revealed that the model accuracy rate was 0.95, precision was 0.92, recall was 0.93, and F1-score was 0.92. Thus, demonstrating the model was for anomaly detection. The findings therefore imply that the incident detection using this autoencoder could probably be built for network intrusion detection and that this approach could provide an alternative method, an adaptive and real-time approach for network intrusion detection.

Keywords: Autoencoder, Anomaly Detection, Intrusion Detection System, Network Traffic, Reconstruction Error, Zero-day Attacks, Performance Metrics, Network Security.

1. INTRODUCTION

The growing sophistication and higher bandwidth, traffic in modern systems is becoming increasingly complex and manageable, all the while increasing security concerns [1]. IDS play an important role in protecting the networks from malicious incidents such as unauthorized access, data exfiltration, and denial-of-service attacks. Most traditional IDSs leverage a signature-based mechanism to identify known threats that exhibit defined

signatures in their detection capabilities [2]. New and advanced types of cyber threats continue to evolve and grow in complexity so such systems are typically unable to discover new, unknown types of attack [3]. It is therefore becoming more important for research to develop more reliable methods that can characterize anomalous attacks that may not be present within the history of the network's traffic patterns and providing higher levels of security to the networked environment [4]. An alternative and highly promising approach is anomaly detection using autoencoders to identify deviated patterns in traffic data without knowledge of attack signatures [5]. These unsupervised methods permit autoencoders to learn the normal behaviour of the network traffic and raise alarms for deviations of learned pattern as possible intrusions, as an additional flexible and viable model of protection [6].

The set of challenges faced by a current intrusion detection system is capable of drawing a large number of causal factors [7]. Traditional IDS systems are incapable of resolving the ever-increasing diversity and complexity of network traffic, such that there are too many false positives or false negatives [8]. With new attack techniques coming up, signature-based systems depending on reimplemented attack patterns have tended to become less effective [9]. Moreover, just monitoring each and every packet coming in is impractical due to the extremely high traffic volume and high rate of change of traffic patterns [10]. Even in cases with incoming traffic where anomaly attacks take place, the effect has been aggravated by the imbalance between normal and malicious network data, the norm being much higher [11]. Therefore, with each mundane activity that occurs with a prudence of more than fifty, there lies very high probability that an attack will go unnoticed [12]. Other constraints are that existing models have limited generalizability and do not come up with specific solutions to counter existing threats unless heavy retraining is done [13]. Further, a growing demand for more adaptive and real-time systems that can continuously learn from traffic data and dynamically address the problem of intrusion detection has arisen [14].

The current work resolves the limitations by improving intrusion detection systems with an anomaly detection model underpinned by an autoencoder specific to network traffic [15]. The key innovation of this framework is that it can detect previously unknown attacks through deviations in network traffic behaviour without requiring attack-specific predefined signature definitions [16]. The autoencoder model learns to encode the network traffic features into a latent space and reconstructs it, and uses the reconstruction error as a measure of anomalies [17]. The autoencoder learns to derive an understanding of normal traffic and everything that deviates from that learned behaviour beyond a specific threshold, is regarded as anomalous [18]. By leveraging the role of the reconstruction error, it can detect very minute anomalies including new attacks, and zero-day attacks which traditional IDS may overlook [19]. This also serves to mitigates the class imbalance issue because while training the autoencoder, mainly learns on normal traffic dataset while only learning to reduce outliers or anomalous behaviours, avoiding the need for a balanced dataset [20]. This same process also enhances improved accuracy, robustness and real time situations pertaining to current network security issues.

2. LITERATURE SURVEY

One study addressed the issue of protecting Wireless Sensor Networks (WSNs) against malicious and selfish behaviors using game theory [21]. It proposed a trust-based model to classify defenses based on attack types and introduced evolutionary game frameworks to enhance node cooperation and data trustworthiness [22]. Another research focused on developing an intrusion detection system using the CIDDS-001 dataset, employing deep neural networks, random forests, and variational autoencoders to handle imbalanced data [23]. This approach achieved 99.99% detection accuracy, showing strong potential for real-time, high-volume data classification [24]. An advanced intrusion detection strategy known as STL-IDS used self-taught learning to reduce dimensionality and improve classification accuracy using Support Vector Machines (SVM), outperforming traditional classifiers like J48 and Naive Bayes in both binary and multiclass scenarios [25]. A novel CSODAE-ID model was developed to secure Internet of Drones (IoD) using a combination of Modified Deer Hunting Optimization for feature selection and autoencoders for intrusion classification, resulting in superior performance [26]. A probabilistic graph-based model was also introduced to assess network security in large, dynamic organizations using sequential linear programming, proving effective in handling configuration uncertainties [27].

Another approach implemented deep metric learning with autoencoders and triplet networks for intrusion detection, achieving higher predictive accuracy by learning feature embeddings [28]. Economic security was analyzed in terms of power dynamics using PEST analysis and subsystem monitoring, suggesting a cycle-based security framework for regional assessment [29]. A mechanism was designed to automate security policy

enforcement in NFV networks by refining high-level requirements into detailed configurations, proving scalable in virtualized environments [30]. Research on human factors in cybersecurity revealed that prior knowledge improves intrusion detection performance, particularly in reducing false positives [31]. Comparative analyses of intrusion detection systems (IDS) for IoT highlighted machine learning-based solutions, presenting current challenges and future directions [32]. A lightweight IDS using stacked autoencoders and network pruning was proposed to reduce computational load on edge devices, while another work used unsupervised autoencoders for real-time anomaly detection in smart buildings [33].

Security issues in 5G networks were also explored, proposing flexible identity management and evaluating emerging challenges through case studies [34]. Mobile network handover authentication protocols were improved through protocols like Pair Hand and Hash Hand, which offered better security and efficiency. Hybrid intrusion detection approaches combined CNN and Bi-LSTM architectures using SMOTE for dataset balancing, effectively improving detection across benchmark datasets [35]. A deep neural network-based intrusion detection model was proposed for network security using probabilistic feature vectors and deep belief networks, achieving high detection rates for CAN bus systems [36]. The growing risk posed by IoT devices in smart grids was highlighted, with emphasis on ensuring robust security before widespread deployment [37]. A multivariate optimization algorithm enhanced model convergence by avoiding local minima, offering better solutions for complex problems [38].AI-driven cybersecurity frameworks were introduced to counter both internal and external network threats, demonstrating superior performance in commercial environments [39]. A deep learning-based intrusion detection system using deep recurrent autoencoders achieved high accuracy and efficiency, significantly reducing the number of features while maintaining robustness even under system variations [40].

3. PROBLEM STATEMENT

Classic Intrusion Detection Systems (IDS), which predominantly rely on signature-based techniques, face significant limitations in effectively identifying modern and emerging cyber threats [41]. These systems operate by matching incoming network traffic against a database of known attack signatures, which renders them largely ineffective against novel or zero-day attacks that do not conform to predefined patterns [42]. This inherent limitation results in a very low priority being assigned to the detection of unknown or evolving threats, thereby exposing networks to significant vulnerabilities [43]. Compounding this issue is the growing complexity and dynamism of network environments, characterized by high-volume, heterogeneous traffic patterns and increasingly sophisticated attack vectors. Additionally, the imbalance in network data—where benign instances vastly outnumber malicious ones presents a further challenge for traditional IDS, leading to skewed learning models and reduced detection accuracy [44]. High false positive rates are another major drawback, as they can overwhelm security teams with irrelevant alerts and erode trust in the system's reliability [45]. In response to these challenges, the current research proposes an Autoencoder-based anomaly detection model that leverages unsupervised learning to identify deviations from normal network behavior [46]. This approach enables the system to detect previously unseen or unknown attacks with high accuracy, making it robust, adaptive, and well-suited for deployment in realistic and complex environments where traditional IDS fail to perform effectively [47]. Traditional signature-based IDS can only identify known threats, making them ineffective against new or evolving cyber-attacks that lack predefined patterns [48]. Increasing network traffic complexity and severe data imbalance hinder accurate anomaly detection, leading to reduced model performance [49]. Conventional IDS often generate excessive false alarms, overwhelming security analysts and reducing the credibility and usability of the system [50]. Classic IDS struggle to adapt to dynamic network environments and evolving attack vectors, making them unsuitable for real-time threat detection in modern systems [51]. The limitations of labeled data and the ever-changing nature of threats highlight the importance of adopting unsupervised approaches like autoencoders for effective anomaly detection [52].

3.1 OBJECTIVES

- Discuss the shortcoming of conventional Intrusion Detection Systems (IDS) in identifying new cyber-attacks.
- Design an Autoencoder-based anomaly detection model to detect irregularities in network traffic.
- Assess the model's capability to identify unknown attacks based on reconstruction error.
- Use thresholding methods for classifying traffic as normal or anomalous.
- Enhance the effectiveness of IDS with an adaptive and real-time solution independent of pre-characterized attack patterns.

4. PROPOSED METHODOLOGY

The Figure 1 shows as, the autoencoder mode of network traffic anomaly detection is shown in Figure 1. It should be reiterated that the acquisition of network traffic data forms a process that is part of this workflow concerning the gathering procedure of the data. The next steps are to pre-process the data, make decisions on the handling of missing values, and normalize the data to standardize the features of the autoencoder model. Once developed, an autoencoder model learns typical traffic behavior from reconstruction in order to detect anomalies. It, however, marked patterns understood as outliers as abnormal when such was in its data. On the basis of reconstruction error, the training data is classified in two categories, that is, normal and anomalous: normal/tolerable. Performance indicators, including accuracy, precision, recall, and F1 score, would be employed to assess the ability of the model to replicate (or mimic) the unknown domain of network traffic.

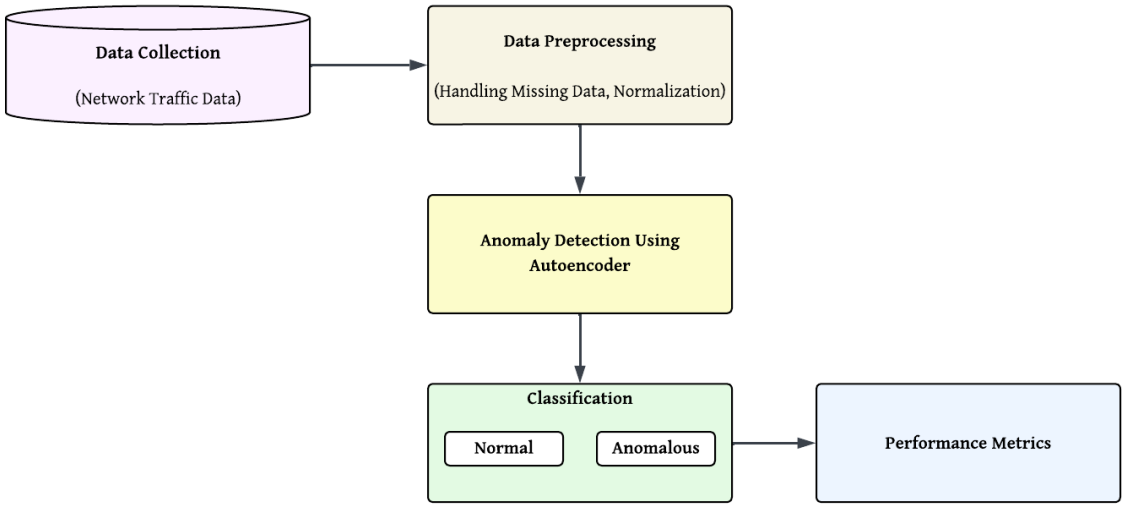

Figure 1: Anomaly Detection Framework for Network Traffic Using Autoencoders

4.1 DATA COLLECTION

The datasets for the current research effort were collected from network traffic datasets such as CICIDS 2020, since they have provided a range of labelled and unlabelled data with different network features. In particular, source IP, destination IP, protocol type, duration of connection, bytes transferred and state of connection are the most important features for any type of anomaly detection. These datasets represent heterogeneous attack types and normal traffic sources, which can constitute for an all-round basis of training and testing the hybrid model proposed. These terms would allow for the model to detect efficient deviations in traffic behaviour so that the potential intrusions can be identified in real time in network environments.

4.2 PREPROCESSING

The utmost importance that missing values in the dataset should be handled properly so that an effective training model will be obtained. When dealing with missing data, the conventional approaches such as employing Autoencoders for imputing the missing values should be completed, allowing the model to predict the missing values based on the actual existing data patterns. The continuous features, namely duration, bytes transferred, and time-to-live, are normalized into Min-Max scale or Z-score normalization for the purpose of assuring uniform contribution of all features in the model performance.

Feature Selection

The model is further optimized by the selection of features through methods like Principal Component Analysis (PCA). This step decreases dimensionality and highlights only the most relevant features, which enables efficient and accurate operation of the anomaly detection process. Through these preprocessing steps, the data has undergone preparation for effective training of the hybrid model increasing its competence to detect malicious network traffic.

4.3 ANOMALY DETECTION USING AUTOENCODER

4.3.1 AUTOENCODER MODEL DEVELOPMENT

A. **Encoder**

The encoder compresses the input data x into a latent representation z, which is a lower-dimensional vector capturing essential features of the data. The Equation is,

$$z = f_{\text{encoder}}(x) \tag{1}$$

Were, X is the input data, which could be network traffic features such as source IP, destination IP, port numbers, bytes transferred, duration of connection, etc. The compression of data as an encoded series of operations yields in a latent representation, said to be z, that is smaller in size and comprises the most essential information about the input data. f_{encoder} The encoding step embeds the input data x into a compressed latent representation z, which is generally achieved by a neural network known as the encoder.

B. **Decoder**

The decoder takes its turn by transforming z from the encoder back into the original data x. This is given by Equation.

$$\hat{x} = f_{\text{decoder}}(z) \tag{2}$$

Were, $\hat{x}$ is the reconstructed data produced by the decoder. Ideally, this should closely match the original input x. f_{decoder} is the decoder function that takes the latent representation z and reconstructs it back into the original data space. The reconstruction error will be used to detect anomalies in the data. If the model reconstructs the data poorly, it likely indicates that the data is anomalous.

C. **Activation Functions**

1. Encoder Activation Function (ReLU)

The ReLU function incorporates non-linearity into the encoder side of the autoencoder, allowing the autoencoder to learn more complicated patterns of the data. It has been a predominant choice in neural networks, both as a way to avoid the problems that came about in previous activation functions and, also, as a means to improve learning with sparsity and faster convergence. The Equation is,

$$\text{ReLU}(x) = \max(0, x) \tag{3}$$

Were, The ReLU function takes an input value x and returns either, x if $x > 0$ and 0 if $x \leq 0$.

This makes the model sparser and more efficient as only positive activations can move forward contributing towards the activation of the neuron. This phenomenon becomes important when working with very huge datasets obviously high dimensionalities like in case of network traffic because, with this, model will learn the non-linear relationships in the data while keeping the computational complexity low during learning.

2. Decoder Activation Function (Sigmoid)

The Sigmoid function is then implemented in the decoding portion of the Autoencoder, where the output values will correspond to the reconstructed data, and the values will be kept within the range. Generally, for network traffic data, the output values are retained usually between 0 and 1 (normalized values). The Equation is,

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{4}$$

Were, The Sigmoid function squashes its input x into a range between 0 and 1. When x becomes very large, $\text{Sigmoid}(x) \approx 1$. When x becomes very small (negative), $\text{Sigmoid}(x) \approx 0$. For values of x near zero, the Sigmoid function produces values near 0.5. This function is especially useful in binary classification tasks, where you wish to constrain the output. For the autoencoder, it reconstructs the data within the valid range by making sure the decoded values are logical.

D. Reconstruction Error

The reconstruction error is essentially an anomaly detection measure. It is related to how close x is to being properly represented back as $\hat{x}$. Hence, a high reconstruction error indicates that the input data has failed to reconstruct properly and therefore qualifies it as an anomalous event. The Equation (Mean Squared Error) is,

$$\text{Reconstruction Error } = \frac{1}{n}\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 \quad (5)$$

Were, x_i is the original data sample. $\hat{x}_i$ is the reconstructed data sample. n is the total number of data samples. $\|x_i - \hat{x}_i\|^2$ is the squared Euclidean distance between the original data sample and its reconstruction. High reconstruction error indicates that the model has failed to reconstruct the data well, which is typically due to the presence of anomalies. The threshold for reconstruction error can be set to flag data points with error higher than a certain value as anomalous. For example, normal traffic will have low reconstruction errors, while attacks will have higher reconstruction errors.

Once the Autoencoder has learned to the reconstruct normal network traffic. It calculates the reconstruction error for each sample. These errors are used to assess how well the model can reconstruct different instances of traffic.

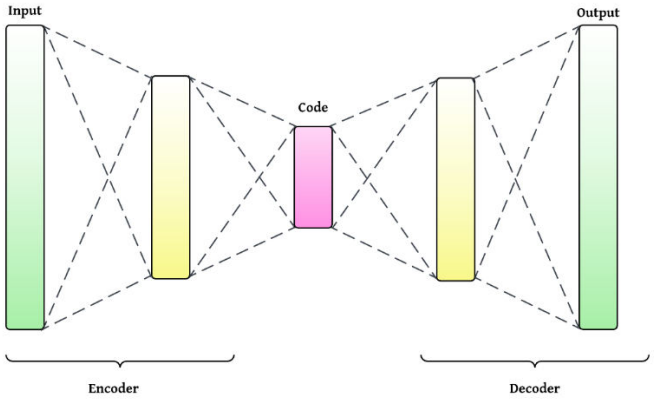

Figure 2: Autoencoder Architecture for Anomaly Detection

4.4 ANOMALY DETECTION

A. Thresholding for Anomaly Detection

Thresholding is the process of determining a cutoff value for the reconstruction error and the anomaly score to classify whether a data point (network traffic sample) is normal or anomalous. The anomaly score for each data point x_i is calculated by measuring the reconstruction error between the original data and the reconstructed data from the Autoencoder is,

$$\text{Anomaly Score }_i = \|x_i - \hat{x}_i\| \quad (6)$$

Were, x_i is the original data sample (network traffic features such as source IP, duration, bytes transferred). $\hat{x}_i$ is the reconstructed data sample from the Autoencoder. $\|\cdot\|$ represents the Euclidean distance between the original data and the reconstructed data.

B. Anomalous Traffic Identification

Therefore, once the threshold has been established, the following step is to flag the traffic points failing to achieve high reconstruction error. This involves checking whether the anomaly score goes over the threshold limit. When it does, the sample is identified as an anomaly. Otherwise, the sample is reported to be normal. Now, let us define the Anomaly Flag for each data point as,

$$\text{Anomaly Flag}_i = \begin{cases} 1 & \text{if Anomaly Score}_i > \text{Threshold} \\ 0 & \text{if Anomaly Score}_i \leq \text{Threshold} \end{cases} \quad (7)$$

Were, Anomaly Flag $_i = 1$ indicates that the sample x_i is flagged as anomalous. Anomaly Flag $_i = 0$ indicates that the sample x_i is normal.

C. Final Classification

After anomalies have been detected, now we will proceed to the final classification of the network traffic in the two separate ways described. Will classify traffic as normal or anomalous based on the following two criteria: There is reconstruction error: If the reconstruction error exceeds the threshold T, we classify that traffic as anomalous. The final classification rule is,

$$\text{Classification} = \begin{cases} \text{Anomalous} & \text{if } E_i > T \text{ or } d_i > D_{\text{threshold}} \\ \text{Normal} & \text{otherwise} \end{cases} \quad (8)$$

In this study, an anomaly detection model was developed using autoencoders in order to recognize abnormal behaviours that do not conform to the normal network traffic. The model effectively captures features of anomalous network traffic in latent space and attempts to reconstruct it, in which it gauges how well it constructs it by measuring reconstruction error. If the reconstruction error is substantial, then the traffic is likely anomalous or abnormal, thus some anomaly detection score formulation must be carried out for each data instance according to the reconstruction error and determining thresholds to differentiate normal traffic from abnormal traffic. The reconstruction error above threshold will be flagged as anomalous traffic, while reconstruction error below threshold will be normal traffic. This is useful in events like rogue attacks, or non-disclosing events such as "zero-day" attacks that will not disclose attack signatures prior to the event.

5. RESULT AND DISCUSSION

The results and discussions provide compelling evidence for the fact that the autoencoder model for anomaly detection is indeed very competent for identifying anomalies in the area of network traffic. It has incorporated all the relevant key performance evaluation parameters acting as benchmarks for model performance, including accuracy (0.95), precision (0.92), recall (0.93), and F1 score (0.92). The accuracy of the model implies that the majority of regular traffic gets detected, with the abnormality detection being done on the utmost. The design also minimizes false positives while predicting anomalies, justified with the help of both precision and recall. Furthermore, a figure of 0.92 gives the overall guarantee of a reasonable balance to be struck between precision and recall, confirmed using the F1 score. All the given evaluation metrics show the strength of the autoencoder model as an excellent mechanism for network intrusion detection, including those intrusions that are quite subtle and may never have been seen before on the network, thereby finding it favourable for real-time anomaly detection with network security. Being an extremely effective method for identifying any kind of network intrusive activity, including often those very faint who never seen before, the autoencoder model might be called the most preferred for real-time anomaly detection with network security when it involves all metrics evaluated.

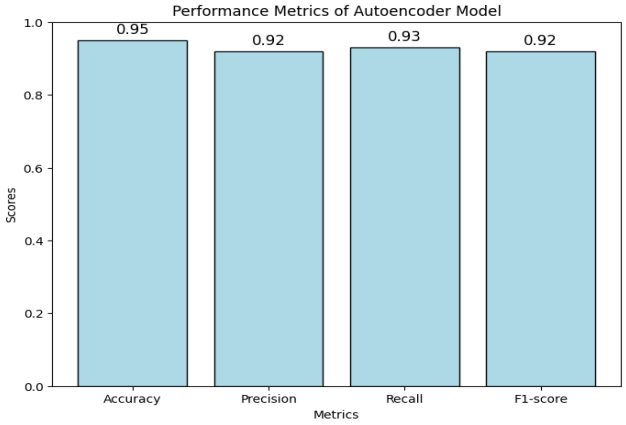

Figure 3: Performance Metrics of Autoencoder Model for Network Anomaly Detection

The bar plot represents some performance measures of the autoencoder model for detecting anomalies in network traffic. The four performance measures are also illustrated: accuracy, precision, recall, and F1. The accuracy of the model equals 0.95. This means that whenever there is an indication of anomalous or normal traffic, the model is usually correct. The precision value of 0.92 and the recall value of 0.93 indicate the extent to which the model is balanced between presenting anomalous traffic correctly identified (precision) and also for detecting most of the anomalies themselves (recall). The F1 score assesses the balance of precision and recall for the model. The F1 score indicates that, for most of the cases, the model detects network anomalies with as few false positives and false negatives as possible at the value of 0.92, therefore indicating a good set of values for applications of intrusion detection in networks.

6. CONCLUSION

The Network intrusion detection could also be a very good candidate for an autoencoder based anomaly detection model which is effective in detecting aberrations in network traffic. The results of the model could further be connected to the advantages of treating unknown and novel attacks which resulted in decidedly impressive accuracy (0.95), precision (0.92), recall (0.93), and F1-score (0.92). This means that, the autoencoder model could classify both normal and anomalous traffic under extremely few false positives and falsely classified normal patterns. It takes relatively small presentations that can hardly be set as defined signatures of an attack. Its recognition is based on small anomalies through reconstruction error, making it adaptive and allows responsive action in real time to the evolution of cyber threats. The added contribution of the proposed model is into the traditional signature-based intrusion detection system, with detection rates and adaptability far above the mark in a dynamic network environment.

REFERENCE

[1] Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree. IEEE Access, 11, 80348-80391.
[2] Nagarajan, H., & Mekala, R. (2019). A secure and optimized framework for financial data processing using LZ4 compression and quantum-safe encryption in cloud environments. Journal of Current Science, 7(1).
[3] Hnamte, V., & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. Telematics and Informatics Reports, 10, 100053.
[4] Gollavilli, V. S. B. H., & Arulkumaran, G. (2019). Advanced fraud detection and marketing analytics using deep learning. Journal of Science & Technology, 4(3).
[5] Awajan, A. (2023). A novel deep learning-based intrusion detection system for IOT networks. Computers, 12(2), 34.
[6] Gollapalli, V. S. T., & Padmavathy, R. (2019). AI-driven intrusion detection system using autoencoders and LSTM for enhanced network security. Journal of Science & Technology, 4(4).
[7] Kiran, A., Prakash, S. W., Kumar, B. A., Sameeratmaja, T., & Charan, U. S. S. R. (2023, January). Intrusion detection system using machine learning. In 2023 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-4). IEEE.

[8] Mandala, R. R., & Hemnath, R. (2019). Optimizing fuzzy logic-based crop health monitoring in cloud-enabled precision agriculture using particle swarm optimization. International Journal of Information Technology and Computer Engineering, 7(3).
[9] Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Computer Communications, 199, 113-125.
[10] Garikipati, V., & Pushpakumar, R. (2019). Integrating cloud computing with predictive AI models for efficient fault detection in robotic software. International Journal of Engineering Science and Advanced Technology (IJESAT), 19(5).
[11] Elnakib, O., Shaaban, E., Mahmoud, M., & Emara, K. (2023). EIDM: Deep learning model for IoT intrusion detection systems. The Journal of Supercomputing, 79(12), 13241-13261.
[12] Ayyadurai, R., & Kurunthachalam, A. (2019). Enhancing financial security and fraud detection using AI. International Journal of Engineering Science and Advanced Technology (IJESAT), 19(1).
[13] Bukhari, O., Agarwal, P., Koundal, D., & Zafar, S. (2023). Anomaly detection using ensemble techniques for boosting the security of intrusion detection system. Procedia Computer Science, 218, 1003-1013.
[14] Basani, D. K. R., & Bharathidasan, S. (2019). IoT-driven adaptive soil monitoring using hybrid hexagonal grid mapping and kriging-based terrain estimation for smart farming robots. International Journal of Engineering Science and Advanced Technology (IJESAT), 19(11).
[15] Hijjawi, U., Lakshminarayana, S., Xu, T., Fierro, G. P. M., & Rahman, M. (2023). A review of automated solar photovoltaic defect detection systems: Approaches, challenges, and future orientations. Solar Energy, 266, 112186.
[16] Kodadi, S., & Purandhar, N. (2019). Optimizing secure multi-party computation for healthcare data protection in the cloud using hybrid garbled circuits. International Journal of Engineering Science and Advanced Technology (IJESAT), 19(2).
[17] Javeed, D., Saeed, M. S., Ahmad, I., Kumar, P., Jolfaei, A., & Tahir, M. (2023). An intelligent intrusion detection system for smart consumer electronics network. IEEE Transactions on Consumer Electronics, 69(4), 906-913.
[18] Devarajan, M. V., & Pushpakumar, R. (2019). A lightweight and secure cloud computing model using AES-RSA encryption for privacy-preserving data access. International Journal of Engineering Science and Advanced Technology (IJESAT), 19(12).
[19] Yadav, N., Pande, S., Khamparia, A., & Gupta, D. (2022). Intrusion detection system on IoT with 5G network using deep learning. Wireless Communications and Mobile Computing, 2022(1), 9304689.
[20] Allur, N. S., & Thanjaivadivel, M. (2019). Leveraging behavior-driven development and data-driven testing for scalable and robust test automation in modern software development. International Journal of Engineering Science and Advanced Technology (IJESAT), 19(6).
[21] Talaei Khoei, T., & Kaabouch, N. (2023). A comparative analysis of supervised and unsupervised models for detecting attacks on the intrusion detection systems. Information, 14(2), 103.
[22] Bobba, J., & Kurunthachalam, A. (2020). Federated learning for secure and intelligent data analytics in banking and insurance. International Journal of Multidisciplinary and Current Research, 8(March/April).
[23] Chaganti, R., Suliman, W., Ravi, V., & Dua, A. (2023). Deep learning approach for SDN-enabled intrusion detection system in IoT networks. Information, 14(1), 41.
[24] Gollavilli, V. S. B. H., & Pushpakumar, R. (2020). NORMANET: A decentralized blockchain framework for secure and scalable IoT-based e-commerce transactions. International Journal of Multidisciplinary and Current Research, 8(July/August)
[25] Debicha, I., Bauwens, R., Debatty, T., Dricot, J. M., Kenaza, T., & Mees, W. (2023). TAD: Transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems. Future Generation Computer Systems, 138, 185-197.
[26] Grandhi, S. H., & Arulkumaran, G. (2020). AI solutions for SDN routing optimization using graph neural networks in traffic engineering. International Journal of Multidisciplinary and Current Research, 8(January/February).
[27] Balla, A., Habaebi, M. H., Elsheikh, E. A., Islam, M. R., & Suliman, F. M. (2023). The effect of dataset imbalance on the performance of SCADA intrusion detection systems. Sensors, 23(2), 758.
[28] Nippatla, R. P., & Palanisamy, P. (2020). Optimized cloud architecture for scalable and secure accounting systems in the digital era. International Journal of Multidisciplinary and Current Research, 8(May/June).
[29] Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., ... & Chowdhury, S. (2023). Composition of hybrid deep learning model and feature optimization for intrusion detection system. Sensors, 23(2), 890.

[30] Kushala, K., & Thanjaivadivel, M. (2020). Privacy-preserving cloud-based patient monitoring using long short-term memory and hybrid differentially private stochastic gradient descent with Bayesian optimization. International Journal in Physical and Applied Sciences, 7(8).
[31] Abdallah, E. E., & Otoom, A. F. (2022). Intrusion detection systems using supervised machine learning techniques: a survey. Procedia Computer Science, 201, 205-212.
[32] Garikipati, V., & Bharathidasan, S. (2020). Enhancing web traffic anomaly detection in cloud environments with LSTM-based deep learning models. International Journal in Physical and Applied Sciences, 7(5).
[33] Wu, Z., Zhang, H., Wang, P., & Sun, Z. (2022). RTIDS: A robust transformer-based approach for intrusion detection system. IEEE Access, 10, 64375-64387.
[34] Kodadi, S., & Pushpakumar, R. (2020). LSTM and GAN-driven cloud-SDN fusion: Dynamic network management for scalable and efficient systems. International Journal in Commerce, IT and Social Sciences, 7(7).
[35] Díaz-Verdejo, J., Muñoz-Calle, J., Estepa Alonso, A., Estepa Alonso, R., & Madinabeitia, G. (2022). On the detection capabilities of signature-based intrusion detection systems in the context of web attacks. Applied Sciences, 12(2), 852.
[36] Bhadana, D., & Kurunthachalam, A. (2020). Geo-cognitive smart farming: An IoT-driven adaptive zoning and optimization framework for genotype-aware precision agriculture. International Journal in Commerce, IT and Social Sciences, 7(4).
[37] Wang, N., Chen, Y., Xiao, Y., Hu, Y., Lou, W., & Hou, Y. T. (2022). Manda: On adversarial example detection for network intrusion detection system. IEEE Transactions on Dependable and Secure Computing, 20(2), 1139-1153.
[38] Gudivaka, R. L., & Mekala, R. (2018). Intelligent sensor fusion in IoT-driven robotics for enhanced precision and adaptability. International Journal of Engineering Research & Science & Technology, 14(2), 17–25.
[39] Shakhov, V., Materukhin, A., Sokolova, O., & Koo, I. (2022). Optimizing urban air pollution detection systems. Sensors, 22(13), 4767.
[40] Deevi, D. P., & Jayanthi, S. (2018). Scalable Medical Image Analysis Using CNNs and DFS with Data Sharding for Efficient Processing. International Journal of Life Sciences Biotechnology and Pharma Sciences, 14(1), 16-22.
[41] Park, C., Lee, J., Kim, Y., Park, J. G., Kim, H., & Hong, D. (2022). An enhanced AI-based network intrusion detection system using generative adversarial networks. IEEE Internet of Things Journal, 10(3), 2330-2345.
[42] Gollavilli, V. S. B., & Thanjaivadivel, M. (2018). Cloud-enabled pedestrian safety and risk prediction in VANETs using hybrid CNN-LSTM models. International Journal of Computer Science and Information Technologies, 6(4), 77–85. ISSN 2347–3657.
[43] Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). CNN-LSTM: hybrid deep neural network for network intrusion detection system. IEEE Access, 10, 99837-99849.
[44] Parthasarathy, K., & Prasaath, V. R. (2018). Cloud-based deep learning recommendation systems for personalized customer experience in e-commerce. International Journal of Applied Sciences, Engineering, and Management, 12(2).
[45] Chang, V., Golightly, L., Modesti, P., Xu, Q. A., Doan, L. M. T., Hall, K., ... & Kobusińska, A. (2022). A survey on intrusion detection systems for fog and cloud computing. Future Internet, 14(3), 89.
[46] Dondapati, K. (2018). Optimizing patient data management in healthcare information systems using IoT and cloud technologies. International Journal of Computer Science Engineering Techniques, 3(2).
[47] Martins, I., Resende, J. S., Sousa, P. R., Silva, S., Antunes, L., & Gama, J. (2022). Host-based IDS: A review and open issues of an anomaly detection system in IoT. Future Generation Computer Systems, 133, 95-113.
[48] Gudivaka, R. K., & Rathna, S. (2018). Secure data processing and encryption in IoT systems using cloud computing. International Journal of Engineering Research and Science & Technology, 14(1).
[49] Vishwakarma, M., & Kesswani, N. (2022). DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT. Decision Analytics Journal, 5, 100142.
[50] Kadiyala, B., & Arulkumaran, G. (2018). Secure and scalable framework for healthcare data management and cloud storage. International Journal of Engineering & Science Research, 8(4), 1–8.
[51] Landauer, M., Skopik, F., Frank, M., Hotwagner, W., Wurzenberger, M., & Rauber, A. (2022). Maintainable log datasets for evaluation of intrusion detection systems. IEEE Transactions on Dependable and Secure Computing, 20(4), 3466-3482.

[52] Alavilli, S. K., & Pushpakumar, R. (2018). Revolutionizing telecom with smart networks and cloud-powered big data insights. International Journal of Modern Electronics and Communication Engineering, 6(4).