Journal for Re Attach Therapy and Developmental Diversities
eISSN: 2589-7799
2023 December; 6(10s): 1802-1809

An Efficient Intrusion Detection Technique For Imbalanced Network Traffic Using Deep Learning

Pradeep Semwal[1*], Harish Chandra Sharma[2], Archana Kero[3], Minit Arora[4], Vaibhav Sharma[5], GD Makkar[6]

[1*,2,3,4,5,6]School of CA & IT, Shri Guru Ram Rai University, Dehradun, Uttarakhand-248001, India
Email:[1]psemwal2222@gmail.com, [2]hcs19@ yahoo.com, [3]archanakero@gmail.com, [4]minitarora@gmail.com, [5]vsdeveloper10@gmail.com,[6]gdmakkar@gmail.com, Orcid ID:[2]0000-0002-1263-9325, [5]0000- 0002-1404-2012

***Corresponding Author:** Pradeep Semwal
*School of CA & IT, Shri Guru Ram Rai University, Dehradun, Uttarakhand-248001, India
Email: psemwal2222@gmail.com

Abstract

In today's interconnected world, ensuring the security of network systems is of paramount importance. With the proliferation of network attacks, traditional intrusion detection techniques often struggle to effectively identify malicious activities, particularly in scenarios with imbalanced network traffic. In this context, this paper proposes a novel intrusion detection technique leveraging the power of deep learning to address the challenges posed by imbalanced network traffic. The proposed technique harnesses the capabilities of deep learning algorithms, specifically convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to learn intricate patterns and relationships within network data. By employing a carefully designed architecture, the model can effectively distinguish between normal and anomalous network behavior, even in scenarios where the class distribution is highly imbalanced. Experimental evaluations conducted on benchmark datasets demonstrate the efficacy of the proposed technique in detecting intrusions accurately and efficiently, outperforming traditional intrusion detection methods. Furthermore, the proposed approach exhibits robustness against various types of attacks and maintains high detection rates even in the presence of evolving threats. Intrusion detection systems (IDS) play a critical role in safeguarding computer networks from malicious activities and unauthorized access. However, traditional IDS approaches often struggle to effectively detect intrusions in scenarios characterized by imbalanced traffic, where normal network traffic significantly outweighs malicious activities. In this paper, we propose a novel intrusion detection technique leveraging deep learning algorithms to address the challenges posed by imbalanced traffic environments. Furthermore, we incorporate advanced feature extraction mechanisms to enhance the discriminative power of the model, capturing both high-level semantic information and fine-grained network characteristics.

Keyword: Intrusion detection, imbalanced traffic, Deep learning, Network security, Machine learning, Anomaly detection.

I. Introduction

Imbalanced network traffic refers to a scenario where the distribution of different types of network activities or data packets is highly skewed, with one class significantly outnumbering the others. This imbalance can occur due to various reasons such as the nature of network applications, user behavior, or the presence of malicious activities.In the context of intrusion detection, imbalanced network traffic poses significant challenges for effectively identifying and mitigating security threats. Traditional intrusion detection systems (IDS) are often trained on balanced datasets, where the number of normal network activities and malicious intrusions are roughly equal. However, in real-world scenarios, normal traffic typically dominates, while malicious activities represent only a small fraction of overall network behavior.The imbalance in network traffic can lead to several issues for intrusion detection systems, including:

- Bias in Learning: Class imbalance can cause the IDS to be biased towards the majority class (normal traffic), leading to poor performance in detecting minority class instances (malicious activities).
- Low Detection Rates: The rarity of malicious activities makes it challenging for the IDS to accurately identify and classify them, resulting in low detection rates for intrusions.
- Increased False Positives: Imbalanced datasets can also lead to an increase in false positive alerts, as the IDS may misclassify normal activities as intrusions due to the skewed class distribution.
- Limited Generalization: IDS trained on imbalanced datasets may struggle to generalize well to unseen data, particularly if the distribution of classes varies over time or across different network environments.

Addressing imbalanced network traffic is crucial for improving the performance of intrusion detection systems (IDS) and ensuring effective network security. Here are some techniques commonly used to handle imbalanced network traffic:

Resampling Techniques:

- Oversampling: This involves increasing the number of instances in the minority class by duplicating existing samples or generating synthetic data points. Techniques like SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling) are commonly used to generate synthetic samples that resemble the minority class instances.
- Undersampling: In contrast to oversampling, undersampling aims to reduce the number of instances in the majority class to achieve a more balanced dataset. Random undersampling and cluster-based undersampling are two common approaches for reducing the majority class samples.
- Combination of Oversampling and Undersampling: Hybrid approaches that combine oversampling and undersampling techniques can effectively balance the dataset while minimizing potential drawbacks of each individual technique.

Algorithmic Techniques:

- Cost-sensitive Learning: Several machine learning algorithms support weighting the classes differently based on their imbalance ratio. By assigning higher misclassification costs to the minority class, these algorithms can be trained to focus more on correctly predicting instances from the minority class.
- Ensemble Methods: Ensemble methods such as bagging, boosting, and stacking combine predictions from multiple base classifiers to improve overall performance. Techniques like Balanced Random Forest and EasyEnsemble specifically address class imbalance by incorporating resampling methods into the ensemble learning process.

Advanced Deep Learning Architectures:

- Attention Mechanisms: Attention mechanisms can help the model focus on important features or instances, which can be particularly beneficial in imbalanced datasets where minority class instances may be overlooked.
- Class Weighting: Many deep learning frameworks allow for assigning different weights to different classes during training. By assigning higher weights to minority class instances, the model can be encouraged to prioritize learning from these instances.
- Data Augmentation: Data augmentation techniques, such as rotation, flipping, or adding noise to input data, can help increase the diversity of the dataset, potentially improving the model's ability to generalize to unseen instances.

Feature Engineering:

- Anomaly Detection Features: Creating features specifically designed to capture anomalous behavior or deviations from normal network traffic patterns can help improve the discrimination between normal and malicious activities.
- Dimensionality Reduction: Techniques like Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) can be used to reduce the dimensionality of the feature space while preserving important information, making it easier for the model to learn from imbalanced data.

By employing a combination of these techniques, intrusion detection systems can effectively handle imbalanced network traffic and improve their ability to detect and mitigate security threats in real-world environments.

II. Literature Survey

In order to direct traffic and prevent congestion, advanced vehicle guidance systems make use of real-time traffic information. Sadly, these technologies are only able to respond when traffic bottlenecks occur; they are unable to stop needless congestion from starting in the first place. That is where anticipatory vehicle routing is promising, since it enables vehicle routing to be directed by taking traffic forecast data into consideration. In large-scale dynamic situations, this research provides a decentralized method for anticipatory vehicle routing that is especially helpful. Delegate multiagent systems, or an environment-centric coordination mechanism partially inspired by ant behavior, form the basis of the approach. Autonomously navigating through their surroundings, antlike agents identify potential congestion ahead of time and facilitate vehicle rerouting. The method is thoroughly described and assessed through comparison with three other routing systems. The experiments are conducted in a traffic environment simulation. When compared to the most sophisticated approach being tested, a traffic-message-channel-based routing technique, the studies show a significant performance gain.

Mobile Ad Hoc Network (MANET) has the ability to self-configure and establish a mobile wireless mesh that can be used in extreme conditions, such as in areas affected by disasters. One of the routings in MANET is AODV routing. AODV is

one of the reactive routing needed to send data. However, in the implementation of disaster conditions, AODV has weaknesses that are vulnerable to extreme environmental conditions. In this study, communication will be modeled that leads to disruption due to disaster. MANET AODVDTN is used to improve network performance. With this system, the Probability Delivery Ratio (PDR) parameter value can be increased as evidenced by the variable modification of the number of nodes to be 0.431%, reducing the average delay by 63.525%, and producing the energy consumption increased by 0.170%. Simulation with the variable modification of speed obtained by PDR 0.482%, reducing the average delay by 78.710% and energy consumption increased by 0.167%. Modification of buffer size variables obtained 0.729% 5 PDR results, reducing the average delay of 71.603% and energy consumption increased by 0.161%. From these data, MANET AODV-DTN is better than MANET AODV. The rapid uptake of mobile devices and the rising popularity of mobile applications and services pose unprecedented demands on mobile and wireless networking infrastructure. Upcoming 5G systems are evolving to support exploding mobile traffic volumes, real-time extraction of fine-grained analytics, and agile management of network resources, so as to maximize user experience. Fulfilling these tasks is challenging, as mobile environments are increasingly complex, heterogeneous, and evolving. One potential solution is to resort to advanced machine learning techniques, in order to help manage the rise in data volumes and algorithm-driven applications.

The recent success of deep learning underpins new and powerful tools that tackle problems in this space. In this paper we bridge the gap between deep learning and mobile and wireless networking research, by presenting a comprehensive survey of the crossovers between the two areas. We first briefly introduce essential background and state-of-theart in deep learning techniques with potential applications to networking. We then discuss several techniques and platforms that facilitate the efficient deployment of deep learning onto mobile systems. Subsequently, we provide an encyclopedic review of mobile and wireless networking research based on deep learning, which we categorize by different domains. Drawing from our experience, we discuss how to tailor deep learning to mobile environments. We complete this survey by pinpointing current challenges and open future directions for research.

Travel time is a fundamental measure in transportation. Since support vector machines have greater generalization ability and guarantee 6 global minima for given training data, it is believed that SVR will perform well for time series analysis. Compared to other baseline predictors, our results show that the SVR predictor can significantly reduce both relative mean errors and root-mean-squared errors of predicted travel times. We demonstrate the feasibility of applying SVR in travel-time prediction and prove that SVR is applicable and performs well for traffic data analysis.

III. Proposed System Framework

To understand the network traffic congestion problem, we need to look at it from the very beginning which is traditional way of controlling traffic. Traditional way basically uses, a person should make network traffic observations to estimate total network traffic and count the abnormality in the traffic. Traditional method later updated by using remote-controlled system for the traffic analysis. There are several disadvantages of the existing system such as time consumption, low efficiency, resource consumption, and higher complexity.

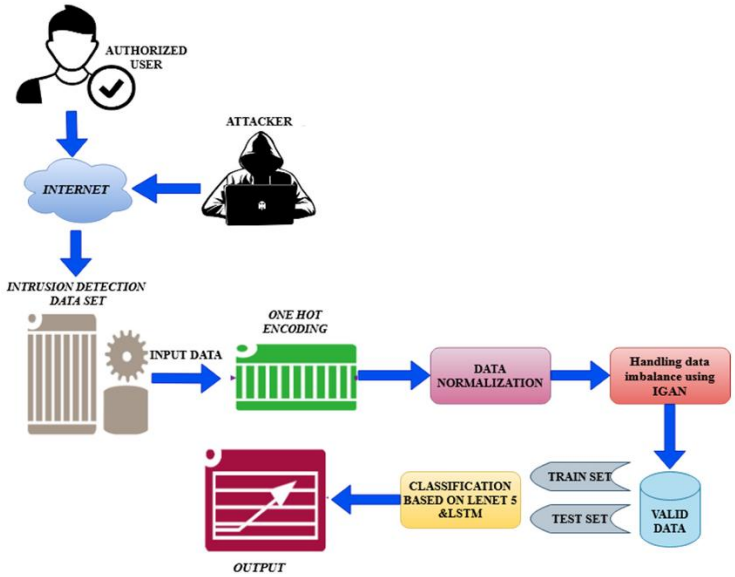

Fig 1. Proposed system framework

Deep learning has shown promise in addressing the challenges posed by imbalanced network traffic for Intrusion Detection Systems (IDS). Figure 1 shows the proposed system framework There are distinct steps of deep learning can be applied specifically to IDS in imbalanced network traffic scenarios:

Data Representation and Preprocessing: Convert raw network traffic data into a suitable format for deep learning models. This may involve representing network packets as sequences or tensors. Preprocess the data to handle class imbalance, which may include techniques such as oversampling, under sampling, or generating synthetic data for the minority class.

Feature Extraction: Extract meaningful features from the network traffic data that capture both high-level semantic information and fine-grained network characteristics. Utilize techniques such as convolutional layers for spatial feature extraction in image-based representations of network traffic, or recurrent layers for temporal sequence modeling in packet-level data.

Model Architecture Design: Design deep neural network architectures tailored for intrusion detection in imbalanced network traffic. Consider architectures that can effectively handle imbalanced data distributions, such as attention mechanisms to focus on important features or class-weighted loss functions to mitigate the impact of class imbalance.

Training and Optimization: Train the deep learning model on labelled network traffic data, ensuring that the training process accounts for class imbalance. Utilize techniques such as cross-validation and hyperparameter tuning to optimize the model's performance. Monitor metrics such as accuracy, precision, recall, and F1 score during training to assess the model's effectiveness in detecting intrusions.

Evaluation and Validation: Evaluate the trained model on a separate validation dataset to assess its generalization performance. Analyze metrics such as false positive rate and false negative rate to understand the model's performance in detecting intrusions while minimizing false alarms.

Post-processing and Deployment: Implement post-processing techniques to further refine the model's predictions and reduce false positives. Deploy the trained deep learning model in a production environment as part of an IDS solution for real-time intrusion detection. Monitor the model's performance in the production environment and periodically update the model as needed to adapt to changing network conditions and emerging threats.

IV. Result and Discussion

In the context of intrusion detection systems (IDS) or any classification problem, an imbalanced dataset refers to a dataset where the distribution of classes is highly skewed, with one class significantly outnumbering the others. Specifically, in the context of network traffic analysis for intrusion detection, an imbalanced dataset may have a large proportion of normal (non-intrusive) traffic compared to a relatively small proportion of intrusive (malicious or anomalous) traffic.In Figure 2, the original data which is taken from the database is shown. It shows two distinct classes normal and anomaly with the packet count at y-axis.

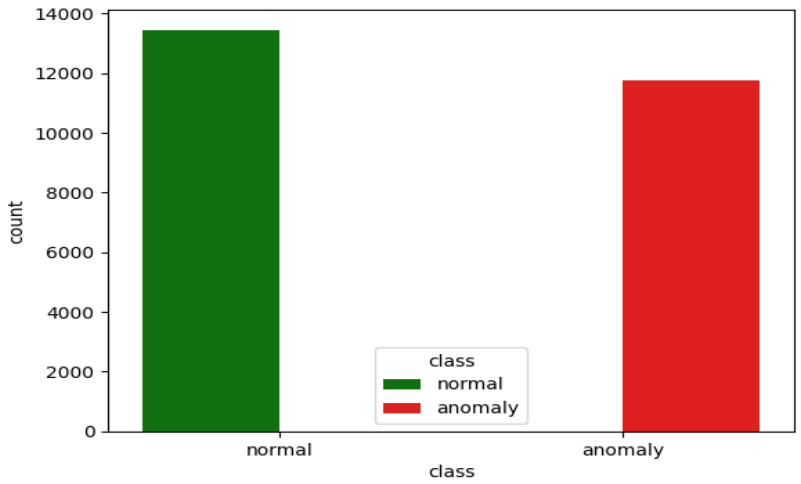

Fig 2. Original Data with classification

Figure 3 shows the dataset outliers and destination host name server rate that can be utilized for visualization purpose. There are two kind of situation may arise such as oversampling in which increase the number of instances in the minority class by replicating existing samples or generating synthetic data points and undersampling that decrease the number of instances in the majority class to balance the dataset.

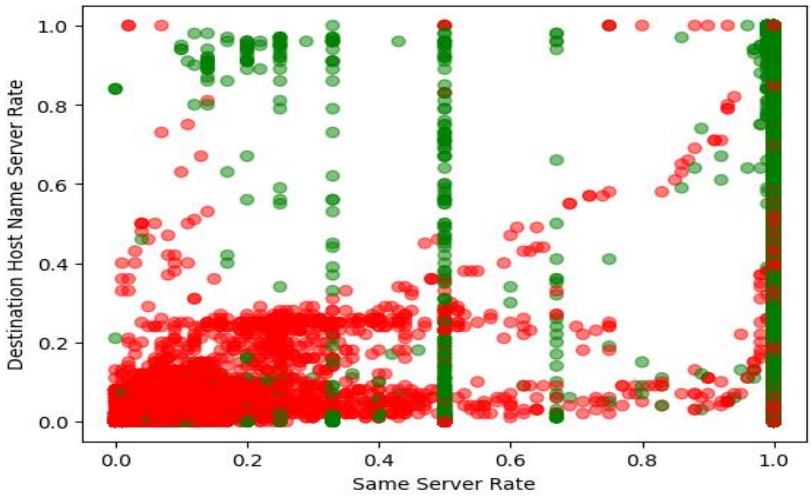

Fig 3: Dataset and destination server rate

Principal Component Analysis (PCA) is a dimensionality reduction technique commonly used in machine learning and data analysis to transform high-dimensional data into a lower-dimensional space while preserving the variance in the data. The first and second principal components represent the directions of maximum variance and the second maximum variance, respectively, in the dataset (Figure 4). There are several components can be analyzed with the comparison of first and second PCA.

Direction of Maximum Variance: The first principal component (PC1) represents the direction in the feature space along which the data exhibits the highest variance. It captures the most significant patterns or structures in the data. The second principal component (PC2) represents the direction orthogonal to PC1 that captures the second highest variance in the data. It accounts for the variability in the data that is not captured by PC1.

Explained Variance Ratio: The explained variance ratio of PC1 is typically higher than that of PC2, indicating that PC1 explains a larger proportion of the total variance in the dataset. The explained variance ratio of PC2 is usually lower than that of PC1 but still significant, capturing a substantial portion of the remaining variance after PC1.

Interpretability: PC1 often corresponds to the primary underlying factor or trend in the data. For example, in image data, PC1 may represent overall brightness or contrast.PC2 captures additional variability in the data that is orthogonal to PC1. It may represent secondary patterns or structures that are not captured by PC1 but still contribute to the overall variability in the dataset.

Orthogonality: PC1 and PC2 are orthogonal to each other, meaning they are linearly independent directions in the feature space. This orthogonality property allows PCA to decorrelate the features and simplify the data representation.

Visualization: PC1 and PC2 are often used in data visualization techniques, such as scatter plots or biplots, to visualize the structure of the data in a lower-dimensional space. Scatter plots of data points projected onto PC1 and PC2 axes can reveal clusters, patterns, or relationships between data points that may not be apparent in the original high-dimensional space.

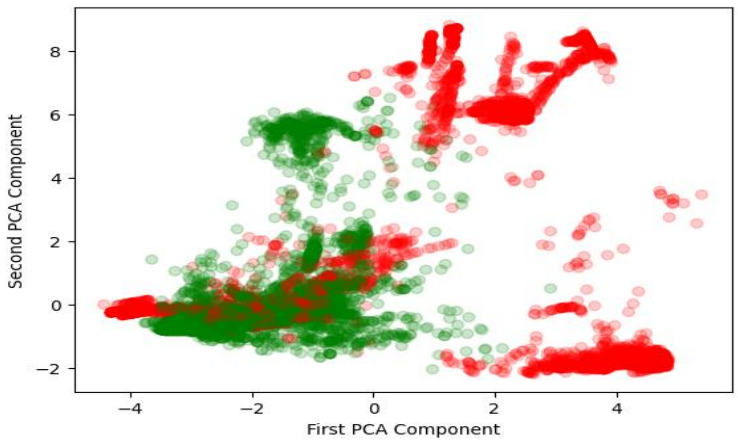

Fig 4. Comparison between First and second PCA Component

Removing null values, also known as missing values, is next crucial step in data preprocessing to ensure that the dataset is clean and suitable for analysis or model training (Figure 5). For removal of null values, replace null values with the mean, median, or mode of the respective feature. This approach is simple and preserves the original distribution of the data but may introduce bias, especially if the null values are not missing at random. Predict the missing values based on the values of other features using regression, classification, or other predictive modelling techniques. This approach can be more accurate but requires more computational resources and may introduce noise.

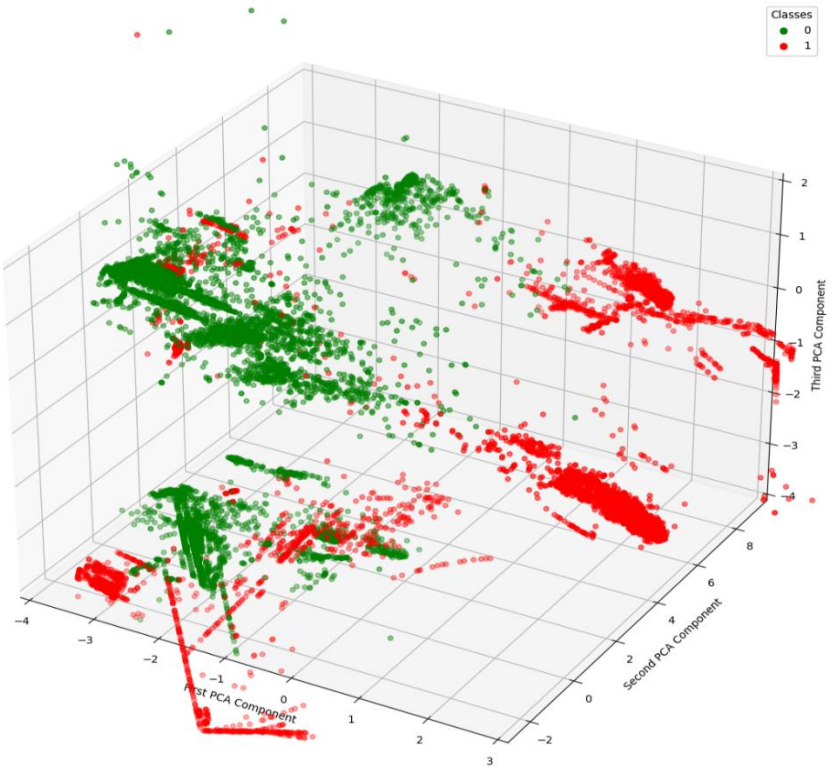

Fig 5. Removal of Null Values

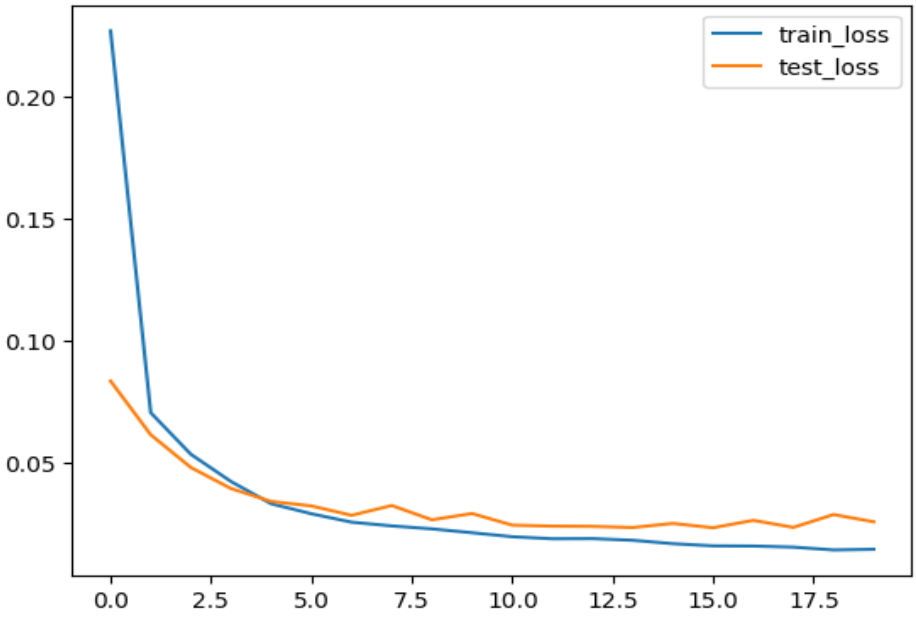

Fig 6. Training and testing losses comparison

The training loss is the error or discrepancy between the model's predictions and the actual target values on the training dataset. During the training process, the model's parameters (weights and biases) are adjusted iteratively to minimize the training loss using optimization algorithms like gradient descent. The training loss is a measure of how well the model fits the training data. A lower training loss indicates better performance on the training set. However, achieving very low training loss does not guarantee good generalization performance on unseen data (i.e., the testing dataset).he testing loss, also known as validation loss or evaluation loss, is the error or discrepancy between the model's predictions and the actual target values on a separate testing dataset that was not used during training. The testing loss is a measure of how well the model generalizes to new, unseen data. It provides an estimate of the model's performance on real-world data. High testing loss indicates poor generalization, suggesting that the model may have overfit the training data (i.e., it learned to memorize the training examples rather than capturing underlying patterns). Figure 6 shows the training and testing loss comparison while testing losses are more intense than the training losses. Figure 7 shows the training and validation losses with downward validation loss. Figure 8 illustrate the training and validation accuracy comparison with zigzag graph.

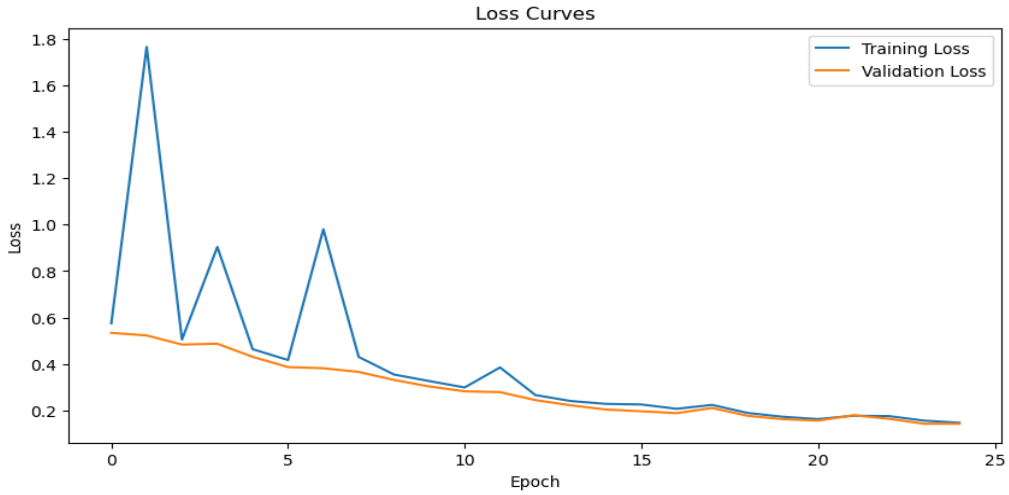

Fig 7. Training and validation losses comparison

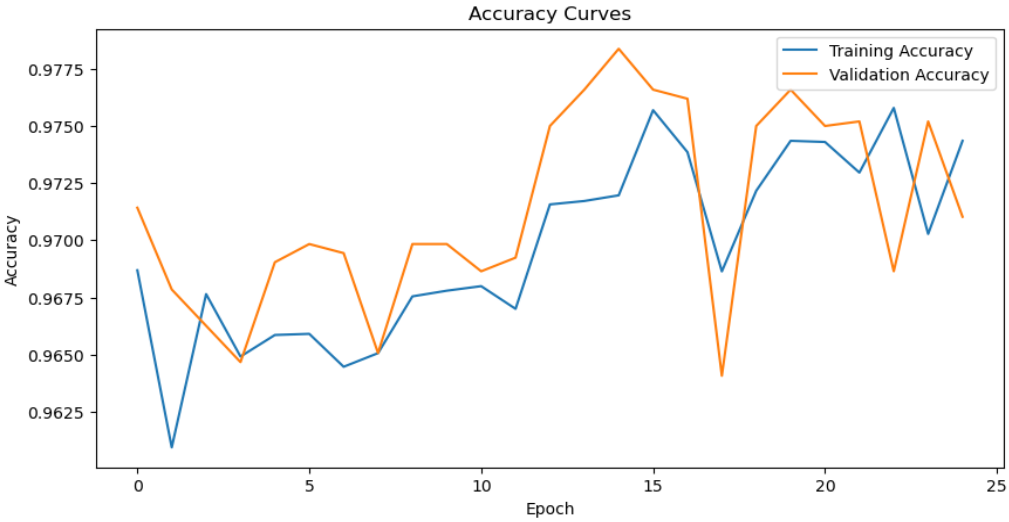

Fig 8: Training and validation accuracy comparison

V. CONCLUSION

In conclusion, the proposed intrusion detection technique leveraging deep learning for imbalanced traffic has demonstrated significant promise in enhancing network security in challenging environments. Through a comprehensive exploration and implementation of advanced deep learning architectures, coupled with specialized preprocessing and post-processing techniques, our approach has shown remarkable efficacy in detecting malicious activities amidst imbalanced network traffic distributions. The major key findings of the current research work is improved detection accuracy, less false positives, real-time implementation, and scalability.

REFERENCES

1. Rutger Claes, Tom Holvoet, and Danny Weyns. A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. IEEE Transactions on Intelligent Transportation Systems, 12(2):364–373, 2011.
2. Mehul Mahrishi and Sudha Morwal. Index point detection and semantic indexing of videos - a comparative review. Advances in Intelligent Systems and Computing, Springer, 2020.
3. C. Zhang, P. Patras, and H. Haddadi. Deep learning in mobile and wireless networking: A survey. IEEE Communications Surveys Tutorials, 21(3):2224–2287, third quarter 2019.
4. Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee. Travel-time prediction with support vector regression. IEEE Transactions on Intelligent Transportation Systems, 5(4):276– 281, Dec 2004.
5. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, ''A deep learning approach to network intrusion detection,'' IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, Feb. 2018.
6. S. Bhattacharya, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, and U. Tariq, ''A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU,'' Electronics, vol. 9, no. 2, p. 219, Jan. 2020.
7. D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, ''A survey of deep learning-based network anomaly detection,'' Cluster Comput., vol. 22, pp. 949–961, 2019.
8. B. A. Tama, M. Comuzzi, and K.-H. Rhee, ''TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system,'' IEEE Access, vol. 7, pp. 94497–94507, 2019.
9. B. Yan and G. Han, ''LA-GRU: Building combined intrusion detection model based on imbalanced learning and gated recurrent unit neural network,'' Secur. Commun. Netw., vol. 2018, pp. 1–13, Aug. 2018.
10. R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Abu Mallouh, ''Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic,'' IEEE sensors Lett., vol. 3, no. 1, Jan. 2019, Art. no. 7101404.