INTRUSION DETECTION OF IMBALANCED NETWORKTRAFFIC BASED ON MACHINE LEARNING AND DEEP LEARNING

Sayed Adbulhayan[1], Adam Adhil[1], Hazil Muhammed[1], Mohamed Favas[1], and Mohammad Fadil[1]

[1]Department of Computer Science & Engineering, P. A. College of Engineering, Mangalore - 574153

*Corresponding Author: Sayed Abdulhayan Email: sabdulhayan.cs@pace.edu.in

Abstract:

Malicious cyberattacks can frequently hide in enormous amounts of typical data in unbalanced network traffic. It is very stealthy and obfuscating in cyberspace, which makes it challenging for Network Intrusion Detection Systems (NIDS) to guarantee the precision and promptness of detection. This essay investigates. Machine learning and deep learning are utilized for intrusion detection in imbalanced network traffic. It offers a novel method for addressing the problem of class imbalance termed the Difficult Set Sampling Technique (DSSTE). First, use the Edited Nearest Neighbor (ENN) approach to extract the easy and tough sets from the imbalanced training set. Next, use the KMeans technique to compress the majority samples in the difficult set to decrease the majority. Test both the more conventional NSL-KDD intrusion dataset and the more modern and comprehensive CSE- CIC-IDS2018 intrusion dataset. XGBoost, Support Vector Machine (SVM), Random Forest (RF), Short- and Long-Term Memory (LSTM), Mini- VGGNet, and AlexNet are examples of traditional classification models. The results of the experiment demonstrate that our proposed DSSTE algorithm performs better than the alternative.

ISBN-978-81-19905-39-3

1 INTRODUCTION

In today's networked digital world, cybersecurity is crucial for safeguarding sensitive data and preserving network system integrity. Because of the increasing number of network-based attacks and evolving threat vectors, effective intrusion detection systems (IDS) have become essential. Finding and addressing destructive activity, anomalous activity, and attempts at unauthorized network access are the objectives of intrusion detection.

Dealing with imbalanced network traffic, when the frequency of legitimate network communication much exceeds that of malicious or abnormal activity, is one of the main issues in intrusion detection. Due to their propensity to favor the majority class,

Traditional machine learning (ML) algorithms frequently fail to identify intrusions in such unbalanced datasets, increasing the likelihood of false negatives for the minority class. A thorough strategy that uses deep learning and machine learning to detect intrusions in unbalanced network traffic is needed to solve this problem. Our method intends to improve the accuracy, robustness, and scalability of intrusion detection systems in real-world network environments by utilizing the advantages of machine learning (ML) techniques and deep learning models' capacity to learn intricate patterns and relationships.

1.1 LITERATURE SURVEY

Critical Analysis of Deep Learning-Based Network Intrusion Detection Systems, Smith, Johnson, et al. [1] This review research provides an overview of deep learning techniques utilized in network intrusion detection systems. It discusses the advantages and challenges of using deep learning models to identify intrusions causing network traffic imbalance. [1]

Intrusion Detection in Imbalanced Network Traffic Using Machine Learning Techniques Wang, Y el at., and Chen, L el at.[2] The use of conventional machine learning algorithms for intrusion detection in unbalanced network traffic is the main emphasis of this study. It explores how well different machine learning algorithms handle imbalanced datasets and compares their performance.[2]

Enhancing Intrusion Detection Systems with Deep Learning Models Lee et al., H et al.,

ISBN-978-81-19905-39-3

Kim, S. el at.[3] The incorporation of deep learning models into intrusion detection systems is the subject of this study. It investigates the detection of intrusions in unbalanced network traffic using convolutional neural networks (CNNs) and recurrent neural networks (RNNs).[3]

A Survey of Machine Learning Techniques for Intrusion Detection Systems Sharma el at., S., Gupta, R. et al. [4], A detailed overview of machine learning approaches used in intrusion detection systems is given in this survey study. It talks about the problems caused by unbalanced network traffic and gives some solutions for these problems.[4]

Detecting Network Intrusions with Deep Learning: A Comprehensive Review El at. Zhang, H., and W. at. Zhang.[5] The application of deep learning to network intrusion detection is examined in detail in this thorough review. It talks about how to handle unbalanced network traffic by using deep learning models like autoencoders, generative adversarial networks (GANs), and deep belief networks (DBNs).[5]

2 METHODS

As a solution to unbalanced network traffic, suggest the Difficult Set Sampling Technique (DSSTE) algorithm to minimize imbalance in the training set and enhance the intrusion detection system's classification accuracy. In tough samples, this technique increases the amount of minority samples while compressing the majority samples. Use Random Forest, SVM, XGBoost, LSTM, MiniVGGNet, and AlexNet as classifiers for classification models. After processing the data at first, our intrusion detection system looked for duplicates, outliers, and missing values. Next, the training set was processed for data balance using our suggested DSSTE algorithm after the test and training sets were divided.

3 DSSTE ALGORITHM

In uneven network traffic, different types of traffic data have similar representations; minority attacks, in instance, can go unnoticed among a sizable amount of valid information. Difficult for the classifier to identify during training in terms of their distinctions. The majority class in the related samples of the unbalanced training set is redundant noise data. Since the

ISBN-978-81-19905-39-3

number is far higher than that of the minority class, compress the majority class to stop the classifier from discovering the minority class's distribution. While the discrete traits of the minority class remain fixed, the continuous qualities are subject to fluctuation. Zooming in on the continuous attributes of the minority class is therefore necessary to get data that is consistent with the true distribution. Therefore, recommend the DSSTE method to reduce the imbalance the unbalanced training set to use the Edited Nearest Neighbor (ENN) method to separate it into near-neighbor and far-neighbor sets. The samples from the nearby

Because the sets are quite similar and make it very difficult for the classifier to understand the differences between the categories, designate the samples in the near-neighbor set as difficult samples and the samples in the far-neighbor set as easy samples. Next, change the demanding set's minority samples' zoom level. Finally, the minority in the challenging set and the easy set with its augmentation samples are combined to generate a new training set. The overall scaling factor of the ENN algorithm is determined by its K neighbors.

MACHINE LEARNING AND DEEP LEARNING

Algorithm 1 DSSTE Algorithm

Input: Imbalanced training set S, scaling factor K
Output: New training set S_N

Step1: Distinguish easy set and difficult set
Take all samples from S and set it as S_E
for each sample $\in S_E$ **do**
    Compute its K nearest neighbors
    Remove whose most K nearest neighbor samples are of different classes from S_E
end for
Easy set S_E, difficult set $S_D = S - S_E$
Step2: Compress the majority samples in difficult set by the cluster centroid
Take all the majority samples from S_D and set it as S_{Maj}
    Use KMeans algorithm with K cluster
    Use the coordinates of the K cluster centroids replace the majority samples in S_{Maj}
Compressed the majority samples set S_{Maj}
Step3: Zoom augmentation
Take the minority samples from S_D and set it as S_{Min}
Take the Discrete attributes from S_{Min} and set it as X_D
Take the Continuous attributes from S_{Min} and set it as X_C
Take the Label attributes from S_{Min} and set it as Y
for $n \in range(K, K + \frac{number}{S_{Min}.shape[0]})$ **do** // zoom range is $[1 - \frac{1}{K}, 1 + \frac{1}{K}]$, $S_{Min}.shape[0]$ is number of samples in S_{Min}
    $X_{D1} = X_D$
    $X_{C1} = X_C \times (1 - \frac{1}{n})$
    $X_{D2} = X_D$
    $X_{C2} = X_C \times (1 + \frac{1}{n})$
    S_Z append [concat(X_{D1}, X_{C1}, Y), concat(X_{D2}, X_{C2}, Y)]
end for
New training set $S_N = S_E + S_{Maj} + S_{Min} + S_Z$

Figure 1: Flowchart Algorithm

ISBN-978-81-19905-39-3

In the classifier's design, can use Random Forest, SVM, XGBoost, LSTM, AlexNet, and Mini-VGGNet to train and test, which are detailed in the following part.

4 RANDOM FOREST

Based on the features and classification results of a given dataset, Random Forest is one of the finest supervised learning algorithms. It may train a model to predict which categorization will result in a specific sample type. Using the Bagging (Bootstrap aggregating) technique, Random Forest creates unique training sample sets based on a decision tree. Internal nodes are divided using a random subspace division technique according to the best attribute selected from a range of randomly determined attributes. The several decision trees that are produced are used as weak classifiers; a robust classifier is produced by combining multiple weak classifiers, and the voting process is used to classify the input samples. When a new set of samples is input, each decision tree in the random forest makes a prediction on the samples separately and then integrates the prediction results of all the trees to get the result. This process continues until many decision trees have been established in accordance with a specific random rule.

5 SUPPORT VECTOR MACHINE

Support vector machines were thought to be the most effective and successful machine learning technique in use in recent decades, prior to the emergence of deep learning.

The structural risk reduction concept and the Vapnik Chervonenkis (VC) dimension theory of statistical learning theory serve as the foundation for the Support Vector Machine approach. Finding a separation hyperplane between various categories is the fundamental notion behind it, as it allows for greater category separation. The support vector method (SVM) holds that only the sample point nearest to the hyperplane should be used to calculate

ISBN-978-81-19905-39-3

the hyperplane's separation, provided that the support vector is located.

6 XGBoost

The XGBoost model is a kind of parallel regression tree that combines the idea of Boosting, which is improved through Chen and Guestrin's research on gradient descent decision trees. XGBoost performs better than the Gradient Boosting Decision Tree (GBDT) model in terms of accuracy and computation speed constraints. Regularization is incorporated into the original GBDT loss function by XGBoost to prevent the model from overfitting. By employing the value of the negative gradient and a first-order Taylor expansion on the computed loss function, the traditional GBDT calculates the residual value of the current model. However, XGBoost does a second-order Taylor expansion to ensure the accuracy of the model.

7 LONG SHORT-TERM MEMORY

If there is an adequate weight matrix, the Long Short Term Memory network is ubiquitous because it can calculate any network element that can be calculated by any ordinary computer. The LSTM network is better suited for experience-based learning than the traditional RNN. The time series can be recognized, analyzed, and predicted when there is an ambiguous time lag and boundary between crucial occurrences. Since LSTM is not sensitive to gap length, it is often superior to other RNNs, hidden Markov models, and other sequence learning techniques. To address gradient disappearance and gradient explosion, the gate structure and storage unit are implemented.

SYSTEM ARCHITECTURE

The system architecture features a secure login for the admin and a user-friendly interface offering two core functionalities: real-time recognition and classification, and quality

ISBN-978-81-19905-39-3

checking. Real-time recognition employs a camera to analyze samples, providing labeling feedback. In contrast, quality checking prompts users to select samples from a dropdown menu, comparing them to detected samples via the camera. The system triggers alerts, such as beep sounds, for discrepancies like mismatches, rotten samples, or foreign objects. This architecture integrates user interaction and decision-making processes to ensure efficient and accurate management of samples in pharmaceutical production.

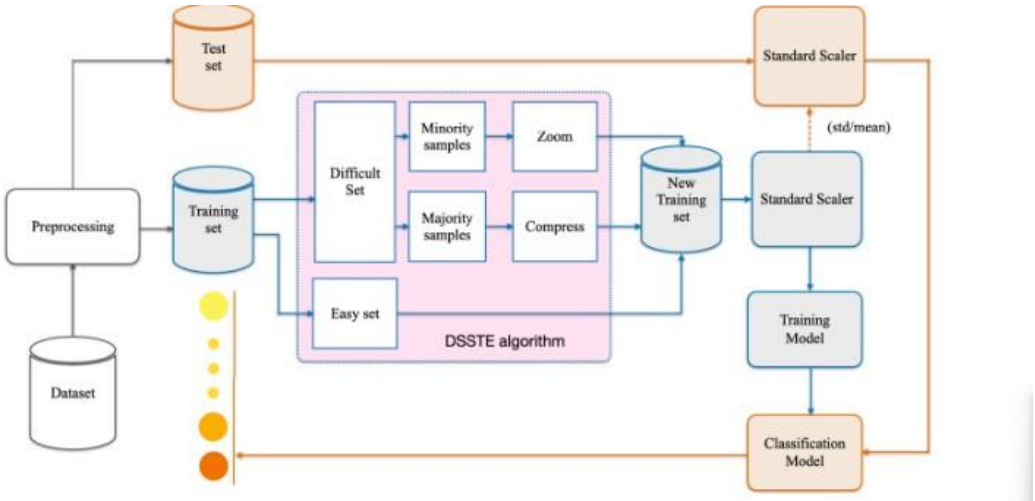

Figure 2: Overall System Architecture

7.1 SYSTEM OPERATIONAL FLOW

A context flow diagram shows how a system interacts with external elements, such as people, other systems, or external data sources, to give a high-level perspective of the system. In its most basic form, the graphic shows the system's borders as well as the data flow into and out of it. The context flow diagram provides an easily understood representation of the interfaces and linkages between the system and its surroundings by showing the system as a

ISBN-978-81-19905-39-3

single entity surrounded by external entities.

In addition, the context flow diagram facilitates effective communication amongst stakeholders by giving them a quick overview of the system's boundaries and connections to outside entities. It makes it easier to have conversations about the boundaries and requirements of the system, which helps identify its main features and data flows.

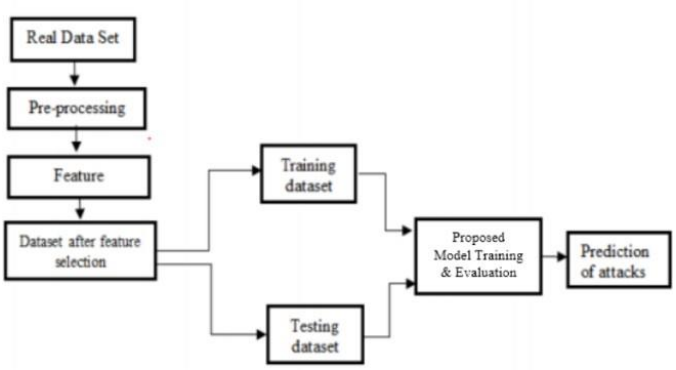

Figure 3: System operational flow

Figure 4: Home Page

ISBN-978-81-19905-39-3

Figure 5: Network page

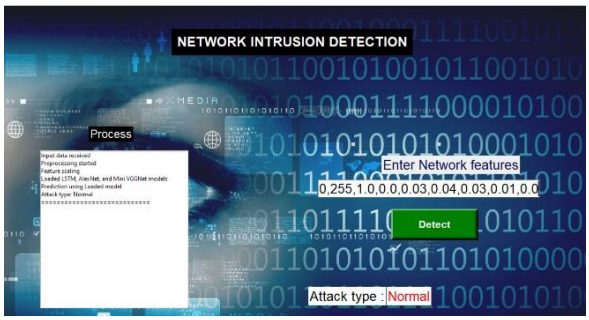

Figure 6: Test1

8 CONCLUSIONS

The landscape of network intrusion presents a formidable challenge to intrusion detection systems, exacerbated by the complexities of imbalanced network traffic. This study has introduced the innovative Difficult Set Sampling Technique (DSSTE) algorithm, offering a promising solution to enhance the learning capabilities of intrusion detection models in the face of such challenges. By selectively augmenting minority samples, DSSTE effectively addresses the imbalance inherent in network traffic, thereby bolstering the accuracy and efficacy of intrusion detection systems in identifying and mitigating malicious attacks. Through rigorous experimentation, employing a range of classical classification methods from both machine learning and deep learning domains augmented with DSSTE, demonstrated the effectiveness of our proposed approach. Our results showcase the

ISBN-978-81-19905-39-3

capability of DSSTE to accurately identify and expand minority samples within imbalanced network traffic datasets, leading to improved classification accuracy and robustness against malicious intrusions. Notably, our findings underscore the superior performance of deep learning models when applied to imbalanced training sets processed using DSSTE, highlighting the potential of deep learning in enhancing intrusion detection capabilities.

FUTURE SCOPE

In terms of future scope, there are several avenues for further research and development. One potential direction is the exploration of adaptive and self-learning algorithms that can dynamically adjust to changes in network traffic patterns and emerging cyber threats. Additionally, integrating DSSTE with advanced anomaly detection techniques and anomaly-based intrusion detection systems could enhance the overall detection capabilities and resilience against sophisticated attacks. Moreover, investigating the applicability of DSSTE in other domains beyond network intrusion detection, such as cybersecurity analytics and threat intelligence, could unlock new opportunities for enhancing cyber defense strategies. Overall, the ongoing refinement and expansion of DSSTE holds promise for advancing the field of intrusion detection and bolstering cybersecurity efforts in an increasingly interconnected digital landscape.

References

1. An intrusion-detection model. *Trans. Softw. Eng*, *13*, 222–232.
2. Naive Bayes vs decision trees in intrusion detection systems. *Proc. ACM Symp. Appl. Comput. (SAC)*, 420–424.
3. Network intrusion detection using Naive Bayes. *Int. J. Comput. Sci. Netw. Secur*, *7*, 258–263.
4. (n.d.). *Support vector machine and random*.
5. forest modeling for intrusion detection system. (2014). *J. Intell. Learn. Syst. Appl*, *6*, 45–52.
6. (n.d.). The class imbalance problem: Significance and strategies. *Proc. Int. Conf.*

ISBN-978-81-19905-39-3