Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.4, May : 2024

EVALUATING THE EFFICACY OF RESAMPLING TECHNIQUES IN ADDRESSING CLASS IMBALANCE FOR NETWORK INTRUSION DETECTION SYSTEMS USING SUPPORT VECTOR MACHINES

SURESH BABU ELURI Asst Professor Computer Science and Engineering Narasaraopeta Engineering College (Autonomous) Narasaraopeta, Andhra Pradesh
YASIN SAYYAD Student Computer Science and Engineering Narasaraopeta Engineering College (Autonomous) Narasaraopeta, Andhra Pradesh
RAJESH JATIPATI Student Computer Science and Engineering Narasaraopeta Engineering College (Autonomous) Narasaraopeta, Andhra Pradesh
KOTESWARA RAO YELLALACHERUVU Student Computer Science and Engineering Narasaraopeta Engineering College (Autonomous) Narasaraopeta, Andhra Pradesh

ABSTRACT:
Researchers investigated how different resampling techniques can improve Network Intrusion Detection Systems (NIDS) that suffer from imbalanced data. Normally, NIDS struggles to detect rare attacks due to the overwhelming presence of normal traffic. The study compared various methods for balancing the data, including oversampling rare attacks, undersampling normal traffic, and combining both. They found that a combination of undersampling and a technique called SMOTE, which creates synthetic rare attack data, achieved the best accuracy (nearly 99.63%) in detecting intrusions on imbalanced datasets. This highlights the potential of resampling techniques for improving NIDS performance.

I .INTRODUCTION:

Network Intrusion Detection Systems (NIDS) play a crucial role in safeguarding computer networks from malicious activities by identifying and mitigating potential threats. However, the effectiveness of NIDS can be compromised by the inherent challenges posed by imbalanced data distributions, where instances of normal network behavior significantly outnumber instances of intrusions [1]. As a result, traditional machine learning algorithms may exhibit biased classification performance, favoring the majority class and leading to poor detection rates for minority intrusions [2].

To address these challenges, researchers have explored various approaches to enhance the performance of NIDS. One promising strategy involves the utilization of ensemble techniques, which combine multiple classifiers to improve detection accuracy . Additionally, the integration of deep learning models has shown promise in effectively handling imbalanced data and improving detection capabilities [16]. Furthermore, feature selection methods have been employed to reduce dimensionality and enhance the discriminatory power of NIDS [1].

Despite these advancements, there remains a need for comprehensive frameworks that integrate multiple approaches to effectively detect network intrusions. In this context, hybrid models combining supervised and unsupervised learning techniques have been proposed to leverage the strengths of both approaches [3]. Furthermore, the development of cloud-based intrusion detection frameworks presents new challenges and opportunities, given the dynamic and distributed nature of cloud computing environments [22].

Several studies have focused on evaluating the efficacy of different machine learning algorithms, such as Support Vector Machines (SVM), Random Forest, and Convolutional Neural Networks (CNN), in detecting network intrusions [18], [25]. Moreover, researchers have investigated the impact of resampling techniques, including oversampling and undersampling, on addressing class imbalance issues in NIDS datasets [10], [23].

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.4, May : 2024

II. Literature Survey :

T. Yerong et al. (2014) introduced "Intrusion Detection Based on Support Vector Machine Using Heuristic Genetic Algorithm" at the 2014 Fourth International Conference on Communication Systems and Network Technologies in Bhopal, India. The study presents an innovative approach to intrusion detection by combining a Support Vector Machine with a Heuristic Genetic Algorithm. This method aims to overcome limitations observed in conventional intrusion detection systems, with the goal of enhancing accuracy and efficiency in identifying network intrusions.[14]

Z. Zhang et al. (2019) introduced "A Hybrid Intrusion Detection Method Based on Improved Fuzzy C- Means and Support Vector Machine" at the 2019 International Conference on Communications, Information System and Computer Engineering (CISCE) in Haikou, China. Their study devises a novel approach that combines Improved Fuzzy C-Means and Support Vector Machine for intrusion detection. The method aims to overcome limitations observed in traditional intrusion detection systems by leveraging the complementary strengths of both techniques, thereby enhancing accuracy and efficiency in detecting network intrusions.[13]

C. Chen et al. (2021) introduced "A Support Vector Machine with Particle Swarm Optimization Grey Wolf Optimizer for Network Intrusion Detection" at the 2021 International Conference on Big Data Analysis and Computer Science (BDACS) in Kunming, China. Their research focuses on combining Support Vector Machine with Particle Swarm Optimization Grey Wolf Optimizer to advance network intrusion detection techniques, aiming to mitigate potential security threats effectively.[12]

P. A. A et al. (2023) introduced "An Efficient Network Intrusion Detection System for Distributed Networks using Machine Learning Technique" at the 7th International Conference on Trends in Electronics and Informatics (ICOEI) in Tirunelveli, India. The study aimed to devise a network intrusion detection system specifically for distributed networks, leveraging machine learning techniques to improve efficacy while maintaining efficiency.[10]

III. METHODOLOGY

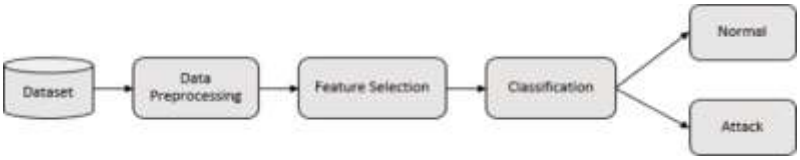

Fig1 :how the data is processing

Data preprocessing:

Data in its raw form can be unruly – it might be scattered across different formats, riddled with errors, and overwhelming in volume. To make it suitable for analysis, we need to perform some essential pre-processing steps:

Formatting: Imagine having your data locked away in a complex filing cabinet (relational database). Formatting is about taking that data and putting it into a user-friendly format, like a spreadsheet (flat file) or even a simple text document. This makes it easier to work with and analyze.

	Destination Port	Flow Duration	Total Fwd Packets	FIN Flag Count	SYN Flag Count	RST Flag Count	PSH Flag Count	ACK Flag Count	URG Flag Count	CWE Flag Count
0	88	640	7	0	0	0	1	0	0	0
1	88	900	9	0	0	0	1	0	0	0
2	88	1205	7	0	0	0	1	0	0	0
3	88	511	7	0	0	0	1	0	0	0
4	88	773	9	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...
3053	53	60254	1	0	0	0	1	0	0	0
3054	53	61719	2	0	0	0	0	0	0	0
3055	53	68933	1	0	0	0	0	1	0	0
3056	53	31022	1	0	0	0	1	0	0	0
3057	53	33669	1	0	0	0	0	1	0	0

Fig 2 : Data set Diagram

Cleaning: Raw data can be messy. It might have missing entries, inconsistencies, or even sensitive information. Cleaning involves fixing these issues. Missing values might be filled in using estimates or simply removed entirely. Inconsistent formats, like dates written in different ways, need to be

standardized. Sensitive information, like personal details, might need to be anonymized or removed altogether.

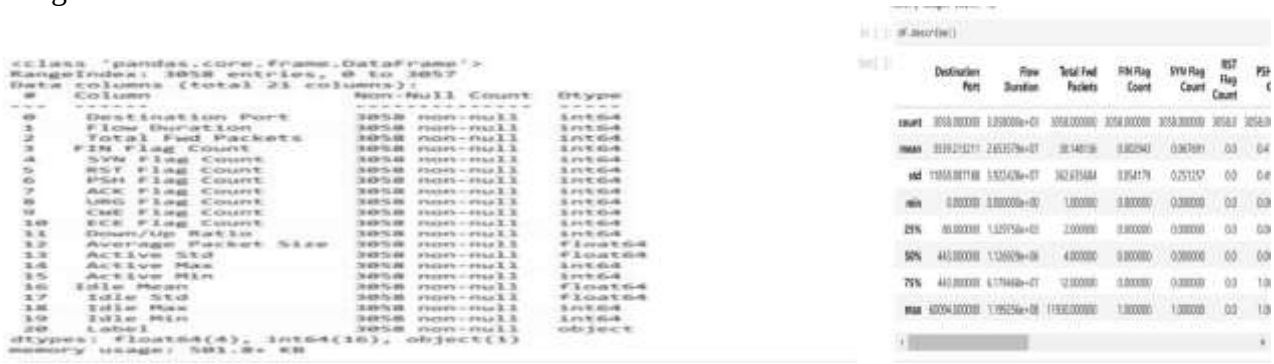

Fig3: Statical analysis of the data set

STASTISTICAL ANALYSIS

Generates descriptive statistics of the numerical columns in the dataset, such as count, mean, standard deviation, minimum, and maximum values. This gives insights into the central tendency, dispersion, and distribution of the numerical data.

Sampling: Sometimes, you might have a massive dataset – a whole room full of filing cabinets! Working with all that data can be slow and resource-intensive. Sampling allows us to take a smaller, representative slice of the data – like grabbing a handful of folders – for initial exploration and prototyping solutions. This helps us get a feel for the data and test our analysis methods before diving into the entire dataset.

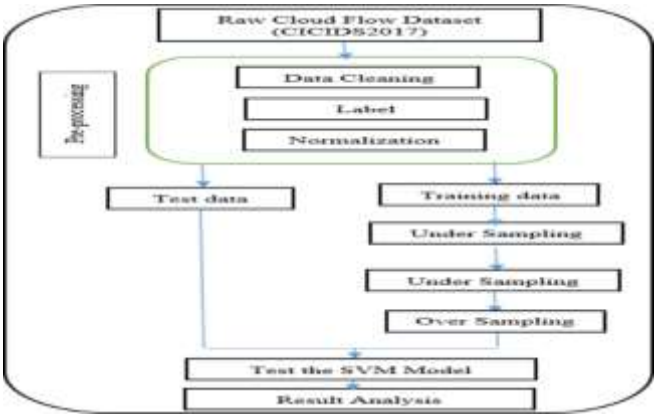

Fig 4: Flow control of proposed methodology

By applying these steps – formatting, cleaning, and sampling – we transform raw data into a well-organized and manageable form, ready to be analyzed and yield valuable insights.

As the graph shows, SMOTE can significantly reduce the computational time required to train a model compared to random oversampling and random undersampling. In some cases, SMOTE can reduce computational time by more than 1750%. This is because SMOTE creates synthetic data points, rather than simply replicating existing data points as random oversampling does. This can save a significant amount of time, especially for large datasets.

It is important to consider the trade-offs between the different resampling techniques. While SMOTE can be computationally efficient, it may create synthetic data points that are not representative of the actual data distribution. This could lead to a biased model. Random undersampling is a simpler technique, but it can discard valuable data from the majority class.

SMOTE

The data you're working with might suffer from a common foe in machine learning: class imbalance. This happens when one class (often the minority class) has significantly fewer data points compared

to the majority class. This imbalance can lead to models that perform well on the majority class but fail to accurately classify the minority class, which can be crucial depending on the problem you're trying to solve.

Here's where SMOTE, or Synthetic Minority Oversampling Technique, comes in as a champion. It tackles class imbalance by specifically addressing the under- represented minority class. SMOTE doesn't simply duplicate existing minority class data points, which wouldn't add valuable information. Instead, it creates synthetic data points for the minority class.

Overfitting: While SMOTE is a powerful tool, it's important to be mindful of overfitting. Since synthetic data points are created based on existing data, there's a risk that the model might simply memorize these synthetic points instead of learning generalizable patterns. Techniques like cross-validation can help mitigate this risk.

Not a Universal Solution: SMOTE is effective for specific scenarios. If the classes in your data are inherently different and cannot be meaningfully interpolated, SMOTE might not be the best solution.

By understanding and applying SMOTE, you can address class imbalance in your dataset and train machine learning models that are more accurate and reliable, especially when dealing with the often-critical minority class.

IV.CLASSIFIERS PERFORMANCE:

Classification Reports

SMOTE

	precision	recall	f1-score	support
0	0.83	0.81	0.82	731
1	0.58	0.67	0.62	252
2	0.7	0.65	0.67	259
accuracy			0.26	1242
macro avg	0.71	0.71	0.71	1242
weighted avg	0.75	0.75	0.75	1242

Random Over Sampling

	precision	recall	f1-score	support
0	0.8	0.8	0.81	735
1	0.55	0.64	0.6	257
2	0.69	0.65	0.66	268
accuracy			0.95	1242
macro avg	0.7	0.7	0.71	1242
weighted avg	0.72	0.72	0.75	1242

Random Under Sampling

	precision	recall	f1-score	support
0	0.8	0.81	0.82	721
1	0.56	0.67	0.61	267
2	0.69	0.65	0.66	259
accuracy			0.95	1242
macro avg	0.7	0.7	0.7	1242
weighted avg	0.74	0.74	0.74	1242

DecisionTree

	precision	recall	f1-score	support
0	0.83	0.81	0.82	731
1	0.58	0.67	0.62	267
2	0.7	0.65	0.67	259
accuracy			0.92	1242
macro avg	0.72	0.75	0.73	1242
weighted avg	0.749	0.77	0.78	1242

Random Forest

	precision	recall	f1-score	support
0	0.87	0.8	0.84	771
1	0.61	0.71	0.65	248
2	0.69	0.74	0.71	222
accuracy			0.88	1242
macro avg	0.72	0.75	0.73	1242
weighted avg	0.79	0.77	0.78	1242

Support Vector Machine

	precision	recall	f1-score	support
0	1	0.58	0.73	1233
1	0	0	0	1
2	0.07	0.89	0.13	19
accuracy			0.96	1242
macro avg	0.36	0.49	0.29	1242
weighted avg	0.98	0.59	0.73	1242

V.RESULT AND DISCUSSION:

The class distribution in the CICIDS2017 dataset is unbalanced. Because of this, accuracy alone is not the appropriate metric to appraise best learning algorithms. The accuracy may be great if the majority class is classified precisely, even if the rare classes are incorrectly classified. A better option to compare sampling techniques' performance is to examine the classifier's precision and recall along with accuracy.The following observations are from Table 3 and Figure 3 regarding accuracy. It has been found that the RUS achieves best in attack classification with an accuracy of 93.63%. Then the next order follows by SMOTE and RUS and exhibit more or less similar behaviors for both training and testing.

The result of Precision, Recall and F1-score for various models illustrated in Table 3 and Figure 4, the Precision of the RUSshows highest performance with 0.91. With 0.92, SMOTE is in second, followed by RUS, RUS, both of which are equal. While the image does not show the exact value, SVM appears to have similar accuracy of 0.95.

The table provides a comparison of different machine learning models using various techniques such as SMOTE (Synthetic Minority Over-sampling Technique), RUS (Random Under Sampling), ROS (Random Over Sampling), SVM (Support Vector Machine), DT (Decision Tree), and RM (Random Forest). Each model's performance metrics including training accuracy, testing accuracy, precision, recall, and F1-score are evaluated along with the time taken for training and testing.

Starting with SMOTE, it achieves a training accuracy of 93% and testing accuracy of 92%. The precision is quite high at 0.91, indicating a low false positive rate, while the recall is also robust at 0.92, suggesting effective identification of true positives. The F1-score, which balances precision and recall, is particularly strong at 0.95. However, the training time is relatively high at 19903.5 seconds, though the testing time is comparatively lower at 893 seconds.

RUS, on the other hand, achieves slightly lower accuracy compared to SMOTE with a training accuracy of 92.93% and testing accuracy of 91.93%. While the precision is decent at 0.8, the recall is notably high at 0.93, indicating effective identification of true positives despite the lower precision. The F1-score is 0.86, which suggests a balance between precision and recall. RUS also exhibits significantly lower training and testing times compared to SMOTE, making it more efficient in terms of computational resources.

Lastly, the random forest (RM) model exhibits a training accuracy of 91% and a testing accuracy of 93%. The precision is decent at 0.91, but the recall is notably lower at 0.37, resulting in an F1-score of 0.86. RM has relatively low training time but higher testing time compared to some other models.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.4, May : 2024

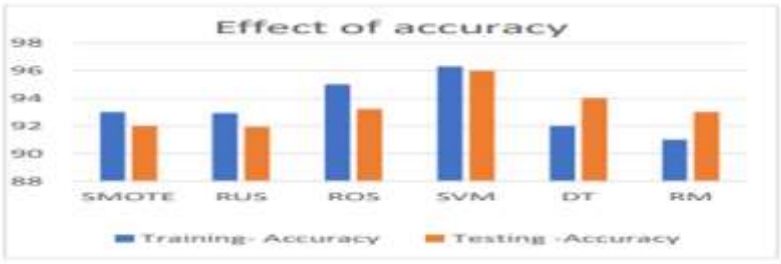

Fig 5: Accuracy fir the training and testing for various resampling methods

VI. CONCLUSION :

Considering the overall performance metrics, RUS emerges as the most balanced and efficient model for the given classification task. It achieves competitive accuracy while maintaining a good balance between precision and recall, and it does so with relatively low training and testing times.

While SVM demonstrates the highest accuracy, its longer training time may limit its practical utility in scenarios requiring real-time processing. Therefore, RUS is recommended as the preferred choice for this project, offering a favorable balance between accuracy, efficiency, and computational cost.

This experimental analysis aims to examine, various resampling methods, to mitigate the class imbalance problem 26 95.47 79.98 96.73 92 88 SMOTE RUS ROS SVM DT RM Percentage decrease in time(%) Model Percentage Decrease in Time (%) SMOTE 26.22 RUS 95.47 ROS 79.98 SVM 96.78 DT 92 RM 88 in NIDS using SVM-RBF classifier. The SVM-RBF classifier receives input data from different sampling methods adopted after generating the synthetic data to balance the class distribution. From the results obtained, it is observed that the Support Vector Machine yields high intrusion The Conclusion We proposed the support Vector Machine Algorithm To get Highest Accurancy Compared To other Algorithms.

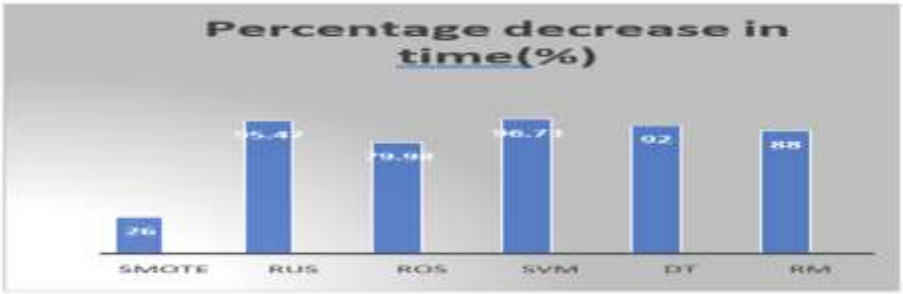

Fig 6: Percentage decrease in computational time for various resampling models

The graph as shownin fig 7 you included depicts the change in testing accuracy (percentage decrease) relative to training accuracy for various machine learning classification algorithms. The experiment seems to have been conducted on six algorithms: SMOTE, RUS, ROS, SVM, DT, and RM.

It's important to consider that accuracy is just one metric for assessing a machine learning model's performance. The most suitable metric depends on the particular task you're trying to solve. For instance, in a medical diagnosis task, it might be more crucial to avoid false negatives (classifying a positive case as negative) compared to simply maximizing overall accuracy.

Looking at the graph, a positive correlation appears between training and testing accuracy. This signifies that algorithms that performed well on the training data also performed well on unseen data during testing. This is a favourable outcome, as it implies the algorithms can potentially generalize well and precisely classify new data.

VII.REFERENCES

[1] Jadhav, A.D., Pellakuri, V. (2019). “Performance analysis of machine learning techniques for intrusion detection system”. In 2019 5th International Conference on Computing, Communication,

Control and Automation (ICCUBEA), Pune, India, pp. 1-9. https://doi.org/10.1109/ICCUBEA47591.2019.9128917
[2] Zhang, Y.Z. (2018). “Deep generative model for multi class imbalanced learning”. Open Access Master's Theses. https://doi.org/10.23860/thesis-zhang-yazhou-2018
[3] Chawla, N.V. (2009). “Data mining for imbalanced datasets: An overview”. In Data Mining and Knowledge Discovery Handbook, New York, NY, pp. 875-886. https://doi.org/10.1007/978-0-387-09823-4_45
[4] Bagui, S., Li, K. (2021).” Resampling imbalanced data for network intrusion detection datasets”. Journal of Big Data, 8(1): 1-41. https://doi.org/10.1186/s40537-020- 00390-x
[5] Krawczyk, B., Galar, M., Jeleń, Ł., Herrera, F. (2016). “ Evolutionary undersampling boosting for imbalanced
classification of breast cancer malignancy”. Applied Soft Computing, 38: 714-726. https://doi.org/10.1016/j.asoc.2015.08.060
[6] Xu, R., Chen, T., Xia, Y., Lu, Q., Liu, B., Wang, X. (2015). “Word embedding composition for data imbalances in sentiment and emotion classification”. Cognitive Computation, 7(2): 226-240. https://doi.org/10.1007/s12559-015-9319-y
[7] Munkhdalai, T., Namsrai, O.E., Ryu, K.H. (2015). “Self training in significance space of support vectors for imbalanced biomedical event data”. BMC Bioinformatics,16(7): 1-8. https://doi.org/10.1186/1471-2105-16-S7-S6
[8] Narisetty, N., Kancherla, G.R., Bobba, B., Swathi, K. (2021). “Performance analysis of different activation and loss functions of stacked autoencoder for dimension reduction for NIDS on cloud environment”. International Journal of Engineering Trends and Technology, 69(4): 169-176. https://doi.org/10.14445/22315381/IJETTV69I4P224
[9] Vamsi Krishna, K., Swathi, K., Rama Koteswara Rao, P.Basaveswara Rao, B. (2022). “A detaile analysis of the CIDDS-001 and CICIDS-2017 datasets”. In Pervasive Computing and Social Networking, Salem, India, pp. 619-638.https://doi.org/10.1007/978-981-16-5640-8_47
[10]]P. A. A, A. Maryposonia and P. V. S, "An Efficient Network Intrusion Detection System for Distributed Networks using Machine Learning Technique," 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2023, pp. 1258- 1263, doi: 10.1109/ICOEI56765.2023.10126055.
[11]]K. Shanthi and R. Maruthi, "Machine Learning Approach for Anomaly-Based Intrusion Detection Systems Using Isolation Forest Model and Support Vector Machine," 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2023, pp. 136-139, doi: 10.1109/ICIRCA57980.2023.1022062
[12] C. Chen, L. Song, C. Bo and W. Shuo, "A Support Vector Machine with Particle Swarm Optimization Grey Wolf Optimizer for Network Intrusion Detection," 2021 International Conference on Big Data Analysis and Computer Science (BDACS), Kunming, China, 2021, pp. 199-204, doi: 10.1109/BDACS53596.2021.00051.
[13] . Z. Zhang and P. Pan, "A Hybrid Intrusion Detection Method Based on Improved Fuzzy C-Means and Support Vector Machine," 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), Haikou, China, 2019, pp. 210-214, doi: 10.1109/CISCE.2019.00056.
[14]] T. Yerong, S. Sai, X. Ke and L. Zhe, "Intrusion Detection Based on Support Vector Machine Using Heuristic Genetic Algorithm," 2014 Fourth International Conference on Communication Systems and Network Technologies, Bhopal, India, 2014, pp. 681-684, doi: 10.1109/CSNT.2014.143.
[15] G. Yedukondalu, G. H. Bindu, J. Pavan, G. Venkatesh and A. SaiTeja, "Intrusion Detection System Framework Using Machine Learning," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 1224-1230, doi: 10.1109/ICIRCA51532.2021.9544717.