Cost-Sensitive Learning with LightGBM for Class Imbalance in Intrusion Detection Systems

Andien Dwi Novika[1]*, Almuzhidul Mujhidi[2]

[1,2] Computer Science Program, Computer Science Department, School of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia 11480
andien.novika@binus.ac.id; almuzhidul.mujhid@binus.ac.id

*Correspondence: andien.novika@binus.ac.id

Abstract – *Imbalanced data is a common and significant challenge in classification problems, where standard models tend to be biased toward majority classes, leading to poor detection of minority instances. This paper presents a comprehensive comparative study of Light Gradient Boosting Machine (LightGBM) and eXtreme Gradient Boosting (XGBoost) models, enhanced with cost-sensitive learning to address class imbalance at the algorithmic level. The objective is to evaluate the impact of cost-sensitive loss adjustments on overall model performance using various evaluation metrics. Experimental results show that both models achieved high cross-validation and test accuracies, with LightGBM and XGBoost recording over 99.9% accuracy. However, only cost-sensitive LightGBM achieved perfect scores in precision, recall, and F1-score, indicating its superior ability to handle minority class identification effectively. In contrast, XGBoost exhibited noticeably lower recall and F1-score despite similar accuracy, reflecting inherent limitations in sensitivity to minority instances. Models without cost-sensitive learning demonstrated further drops in performance across minority-related metrics. The findings suggest that cost-sensitive LightGBM is a more robust and reliable solution for imbalanced classification tasks, outperforming both its baseline and the cost-sensitive XGBoost variant. This approach is particularly beneficial for critical real-world applications such as fraud detection, cybersecurity, and medical diagnostics, where class imbalance is prevalent and misclassification costs are high.*

Keywords: *Cost-sensitive learning; LightGBM; Imbalanced Data; Minority Class Detection; Cybersecurity*

I. INTRODUCTION

The rapid advancement of machine learning (ML) technologies has revolutionized a wide range of application domains, including healthcare, manufacturing, finance, cybersecurity, and more. These advances have enabled the development of intelligent systems that can make predictions, detect anomalies, and support decision-making processes with high efficiency and precision. However, the success of such systems largely depends on the quality and structure of the data used during training. In practice, real-world datasets often exhibit imperfections such as noise, missing values, and, significantly, class imbalance is a condition where certain classes have significantly fewer instances than others.

Data imbalance poses a serious challenge in supervised learning, especially in classification problems, where the model tends to be biased towards the majority class due to its dominance in the dataset. This often results in poor performance on minority classes, which may represent critical outcomes such as fraudulent transactions, rare diseases, or defective components. The issue is further exacerbated when conventional accuracy metrics are used, which can misrepresent model effectiveness in imbalanced settings by favoring the majority class (Spelmen & Porkodi, 2018).

To address this issue, various strategies have been proposed. At the data level, resampling techniques such as oversampling (e.g., SMOTE) and undersampling aim to balance the class distribution by either duplicating or synthetically generating minority instances, or by removing majority class examples (Altalhan et al., 2025). While these methods can improve class balance, they may introduce overfitting (in the case of oversampling) or lead to information loss (in undersampling), particularly in high-dimensional data. (Zhao et al., 2024) conducted research on addressing data imbalance by combining LightGBM with the SMOTE oversampling technique.

At the algorithmic level, cost-sensitive learning has emerged as a powerful solution. Rather than modifying the data, this approach embeds the imbalance handling directly into the learning algorithm by assigning higher misclassification costs to the minority class. This way, the model is encouraged to pay more attention to underrepresented classes during training, improving recall and overall fairness. Cost-sensitive methods are especially appealing in domains where data integrity must be preserved, or where synthetic generation of data may be impractical or ethically questionable (Araf et al., 2024).

Simultaneously, the rise of ensemble-based algorithms has significantly enhanced predictive modeling capabilities. Among them, the Light Gradient Boosting Machine (LightGBM), developed by Microsoft, has shown remarkable performance in both speed and accuracy, particularly in large-scale, high-dimensional datasets. LightGBM builds upon the Gradient Boosting Decision Tree (GBDT) framework but introduces innovations such as leaf-wise tree growth, Histogram-based splitting, Gradient-based One-Side Sampling (GOSS), and Exclusive Feature Bundling (EFB). These optimizations allow it to handle massive datasets with reduced computational complexity and enhanced accuracy (Ke et al., 2017).

Despite its proven strength, LightGBM does not natively include mechanisms to handle class imbalance, often relying on external preprocessing or parameter tuning. While researchers have attempted to combine it with resampling techniques, the integration of cost-sensitive learning directly into LightGBM’s objective function remains relatively underexplored. Such a combination has the potential to harness the strengths of both approaches: the structural efficiency of LightGBM and the class-awareness of cost-sensitive optimization.

LightGBM is an implementation of the gradient boosting decision tree (GBDT) technique optimised for computational efficiency and scalability. Different from traditional approaches such as XGBoost or Random Forest, LightGBM uses leaf-wise tree growth technique with depth limitation, which enables the formation of more complex yet efficient decision trees. In addition, features such as Histogram-based Decision Tree, Gradient-based One-Side Sampling (GOSS), and Exclusive Feature Bundling (EFB) make LightGBM excel at processing big data and handling sparsity.

These limitations motivate the need for an improved approach that incorporates class imbalance handling directly into the learning process (Sadig et al., 2025). Therefore, this study aims to investigate a cost-sensitive adaptation of LightGBM to address performance degradation caused by class imbalance. We proposes a hybrid framework that integrates cost-sensitive learning with LightGBM to effectively tackle class imbalance in supervised classification tasks.

Handling class imbalance in machine learning has been extensively studied, with various strategies developed to address the issue at different stages of the learning pipeline. According to (Haixiang et al., 2017), data imbalance introduces bias in model training and may lead to underperformance on minority classes, especially in high-stakes domains such as fraud detection and medical diagnosis.

Data-level methods, including oversampling and undersampling, have been widely used due to their simplicity and ease of integration. Oversampling techniques like SMOTE generate synthetic data points to augment the minority class, while undersampling reduces the number of instances in the majority class to balance the dataset. (Jeong et al., 2022) emphasize that while these methods can improve balance, they may also introduce overfitting (in the case of oversampling) or information loss (in undersampling).

On the other hand, cost-sensitive learning offers a model-centric approach that modifies the learning algorithm itself to account for class imbalance. Rather than adjusting the dataset, this method assigns higher misclassification penalties to minority class samples, thus forcing the model to treat them with greater importance during optimization. (Mienye & Sun, 2021) show that cost-sensitive learning improves classification fairness and robustness across several imbalanced datasets. Cost-sensitive learning is a widely recognized and effective approach for handling imbalanced data in cybersecurity applications, as it enables models to focus on minority classes without synthetic data generation or resampling. This approach helps maintain data integrity while improving detection rates of rare but critical intrusion events, a challenge extensively discussed in recent literature (Liu et al., 2021).

In recent years, ensemble models have gained prominence for their ability to produce robust and accurate predictions. Among them, LightGBM stands out due to its efficiency in handling large and high-dimensional data. It adopts techniques such as leaf-wise tree growth with depth constraint, gradient-based one-side sampling (GOSS), and exclusive feature bundling (EFB) to accelerate training while maintaining high accuracy. (Liao et al., 2022; Wang et al., 2022; Zhang & Gong, 2020) highlight LightGBM's advantages in speed, scalability, and its suitability for deployment in real-time systems.

While LightGBM has been explored in various contexts, its application in combination with cost-sensitive learning for class imbalance remains relatively underexplored. Previous studies primarily focus on tuning hyperparameters or integrating with oversampling methods, leaving a gap in research for integrating cost-based modifications directly into LightGBM's training process.

This study contributes to this growing body of work by presenting a novel integration of cost-sensitive learning with LightGBM, aiming to improve model performance on imbalanced datasets without altering the data distribution.

II. METHODS

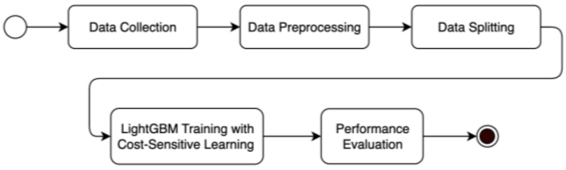

Figure 1 Research Flow

The flow of this research shown in Figure 1. Research starts with data collection

2.1 Dataset

This research used Knowledge Discovery and Data Mining Tools Competition or often called KDD99 dataset. The dataset is pubnlicly accessible through the UCI Machine Learning Repository or Kaggle. The dataset accessed by downloading the official version from *https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html*. KDD99 is a dataset for network intrusion detection. This dataset is highly imbalanced, with 23 labels. Table 1 shows labels and amount of data form each label.

Table 1. KDD99 Label

Label	Value Counts
Smurf	280790
Neptune	107201
Normal	97278
Back	2203
Satan	1589
Ipsweep	1247
Portsweep	1040
Warezclient	1020
Teardrop	979
Pod	264
Nmap	231
Guess_passwd	53
Buffer_overflow	30
Land	21
Warezmaster	20
Imap	12
Rootkit	10
Loadmodule	9
ftp_write	8
Multihop	7
Phf	4
Perl	3
Spy	2

2.2 Data Preprocessing

Preprocessing steps that have been done in this research are one hot encoding, label encoding, and data splitting. Columns that through one hot encoding steps are 'protocol_type', service', and 'flag'.

Protocol_type has 3 types, service has 66 types, and flag has 11 types.

The "label" column, which indicates the class of the data, is the only column that underwent label encoding in this research. Label encoding converts the categorical text labels into numerical values, allowing the machine learning model to process the data. Other columns, such as features representing network attributes, either remain in their original numerical form or undergo techniques like one-hot encoding for categorical data. This ensures that the model can effectively learn from both the target variable and the feature set.

In this research, the dataset was split into three distinct subsets: training, validation, and test sets. The training set contains 345,814 instances and is used to train the model, allowing it to learn the relationships between the features and the target variable. The validation set with 98,804 instances is used to tune hyperparameters and prevent overfitting by evaluating the model during training. Finally, the test set with 49,403 instances is reserved for final evaluation, providing an unbiased assessment of the model's performance after training and hyperparameter optimization.

2.3 LightGBM Training with Cost-Sensitive

Dataset used in this research is highly imbalance that needs to be handled to achieve the best result. This research use cost-sensitive learning in LightGBM to handle the imbalance. Cost-sensitive learning assigns a higher weight to the minority class so that the model has higher sensitivity towards the minor class. This study uses scikit-learn's compute_class_weight function to calculate the weight of each class (Telikani et al., 2022). These weights were passed to the model by specifying them as instance weights during dataset creation via the weight parameter in lgb.Dataset. XGBoost also used in this research as a performance comparator for LightGBM.

Hyperparameter tuning is also done in this phase. This research use grid search for tuning the hyperparameter. Table 2 shows the hyperparameter set for LightGBM and Table 3 shows the hyperparameter set fot XGBoost.

Lambda and alpha are used for making the model more general to prevent overfitting (Chen & Guestrin, 2016). Num leaves and max depth only used in LightGBM since XGBoost does not have those hyperparameter. Num leaves in this research is in charge to control the maximum number of leaves in a tree. Max depth is used for limits the depth of each tree. It has impact to model complexity and training time. Learning rate determines the step size at each boosting iteration.

Table 2. LightGBM Hyperparameter Set

No	Hyperparameter	Value
1.	Lambda	[0, 0.01, 0.1, 1, 10]
2.	Alpha	[0, 0.01, 0.1, 1, 10]
3.	Num Leaves	[31, 63, 127]
4.	Max Depth	[3, 5, 7]
5.	Learning Rate	[0.01, 0.05, 0.1]

Table 3. XGBoost Hyperparameter Set

No	Hyperparameter	Range
1.	Lambda	[0, 0.01, 0.1, 1, 10]
2.	Alpha	[0, 0.01, 0.1, 1, 10]
3.	Max Depth	[3, 5, 7]
4.	Learning Rate	[0.01, 0.05, 0.1]

2.4 Model Evaluation

This research uses four classification metrics, and that are accuracy, precision, recall, and f1-score. Accuracy measures how well it can predict, but because the model is generally biassed to the majority class, accuracy generally isn't meaningful in cases where there is imbalanced data. The precision, recall, and F1-score of the model are also calculated to measure how well it can predict the minority classes. Recall estimates the model's performance in classifying an instance into a class, while precision measures how accurately the model performs in generating correct predictions within the predicted class. The F1-score uses the harmonic mean to merge precision and recall scores.

III. RESULTS AND DISCUSSION

LightGBM achieved a very high cross-validation accuracy of 0.9997 and test set evaluation accuracy of 0.999, revealing the high capability of the model to detect the target variable. The hyperparameters for LightGBM were adjusted to achieve these results, as shown in Table 2. The hyperparameters selected are a Lambda of 0, which will regularize the model by penalizing large coefficients, and an Alpha of 1, which will assist in making the model less prone to overfitting. The model further utilized 63 as the optimal number of leaves and a maximum depth of 7 to permit good tree growth without overfitting. The learning rate of 0.1 was utilized to trade-off convergence speed with model stability in order to allow good learning during training. These hyperparameters have important role in achieving the high accuracy values in both cross-validation and test set tests, confirming the suitability of LightGBM for this specific classification task.

Table 4. LightGBM Best Hyperparameter

No	Hyperparameter	Range
1.	Lambda	0
2.	Alpha	1
3.	Num Leaves	63
4.	Max Depth	7
5.	Learning Rate	0.1

XGBoost demonstrated an excellent cross-validation accuracy of 0.9998 and a test set accuracy of 0.9998, both slightly better than those achieved by LightGBM. The XGBoost optimized hyperparameters, as seen in Table 3, are a Lambda of 0.1 for regularization to prevent overfitting by punishing large coefficients and an Alpha of 0, i.e., no additional regularization on the leaf scores. The model's maximum depth was set to 7, the same as LightGBM's, to have a compromise between model complexity and generalization. A learning rate of 0.1 was chosen to allow for stable and effective training of the model while achieving speed vs. accuracy balance. Comparing with LightGBM, whose accuracy was slightly less (0.9997 for cross-validation and 0.999 for test set), the performance of XGBoost was marginally superior. Both models have very good predictive power, but the slightly better accuracy of XGBoost could make them more efficient in this classification task, though both models perform very well with very good accuracy.

Table 5. XGBoost Best Hyperparameter

No	Hyperparameter	Range
1.	Lambda	0.1
2.	Alpha	0
3.	Max Depth	7
4.	Learning Rate	0.1

Table 4 shows the performance metrics comparison of XGBoost and LightGBM, with their efficiency presented on this imbalanced dataset. LightGBM outperformed XGBoost on all evaluation metrics with best scores of 1.00 in precision, recall, F1-score, and accuracy. This indicates that LightGBM handled the imbalance of the dataset better, with correct identification of minority and majority classes without false negatives or false positives. On the other hand, XGBoost achieved accuracy of 1.00, which means perfect correct classification but its precision (0.85), recall (0.88), and F1-score (0.86) were not as good as LightGBM. XGBoost's low recall and precision show that it performed poorly on the imbalanced distribution of data, having higher false positives and false negatives. Thus, LightGBM proved to be a more stable model in this skewed dataset, as its performance for precision, recall, and F1-score were enhanced.

Table 6. Result Comparison

Model	Precision	Recall	F1-Score	Accuracy
With Cost Sensitive Learning				
XG Boost	0.85	0.88	0.86	1.00
Light GBM	**1.00**	**1.00**	**1.00**	**1.00**
Without Cost-Sensitive Learning				
XG Boost	0.89	0.82	0.84	1.00
Light GBM	0.84	0.80	0.81	1.00

Figure 1 shows feature importance for LightGBM and Figure 2 shows feature importance for XGBoost. The feature importance analysis reveals notable differences between LightGBM and XGBoost in how they utilize the input features. LightGBM assigns the highest importance to features related to connection and service-level statistics such as srv_count, same_srv_rate, and count, which reflect traffic volume and service similarity—key indicators in intrusion detection. In contrast, XGBoost places greater emphasis on features like src_bytes, dst_host_count, and dst_host_srv_count, which are more focused on byte-level traffic data and destination host activities. Although some features like count and dst_host_same_src_port_rate appear among the top in both models, their relative rankings differ significantly. These discrepancies highlight the models' differing learning behaviors: LightGBM's leaf-wise tree growth seems to better capture broader traffic patterns, while XGBoost relies more on detailed host and byte statistics. Additionally, the absolute scales of feature importance differ, with LightGBM's values being orders of magnitude larger, reflecting differences in how each algorithm computes importance metrics. Overall, this divergence in feature prioritization may explain LightGBM's superior performance in detecting minority classes, as it more effectively leverages service-related features that signal anomalous behavior in network traffic.

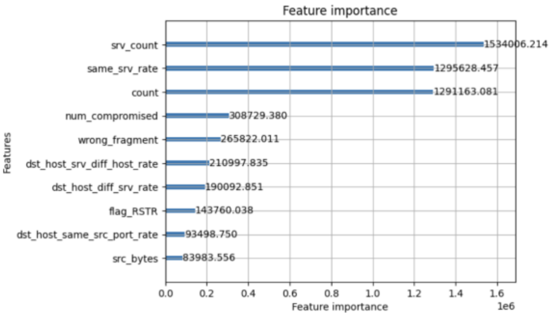

Figure 2. LightGBM Feature Importance

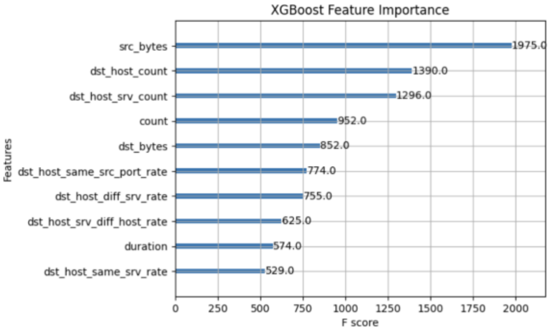

Figure 3. XGBoost Feature Importance

While the proposed cost-sensitive LightGBM shows improved performance on the tested imbalanced dataset, there are several limitations to consider. First, the evaluation was conducted on a limited dataset, which may not fully capture the diversity of class imbalance scenarios found in real-world applications. Additionally, there is a potential risk of overfitting due to the cost adjustments, especially when dealing with small or noisy datasets. Future work should include testing on a broader range of imbalanced datasets from different domains to validate the generalizability of the approach. Furthermore, integrating other imbalance mitigation techniques such as ensemble learning, resampling, or hybrid approaches could provide additional performance gains.

IV. CONCLUSION

This study evaluated the effectiveness of cost-sensitive learning combined with LightGBM and XGBoost in handling imbalanced classification tasks. Although both models achieved very high overall accuracy, further analysis revealed that accuracy alone was not sufficient to assess model performance under class imbalance. LightGBM integrated with cost-sensitive learning outperformed all other configurations, achieving perfect scores (1.00) in precision, recall, and F1-score. This indicates that LightGBM was able to correctly classify both majority and minority classes without misclassifications, making it highly reliable in scenarios with skewed data distributions.

In contrast, XGBoost, even with cost-sensitive tuning, showed lower performance in recall and precision, highlighting its limitations in capturing minority class instances effectively. Models without cost-sensitive learning also underperformed on minority class metrics, reinforcing the importance of algorithm-level strategies over data resampling alone. Overall, the findings support the use of cost-sensitive LightGBM as a robust solution for imbalanced data classification, especially in high-stakes applications such as fraud detection, cybersecurity, and healthcare, where the cost of misclassification can be critical.

REFERENCES

Altalhan, M., Algarni, A., & Turki-Hadj Alouane, M. (2025). Imbalanced Data Problem in Machine Learning: A Review. *IEEE Access*, *13*, 13686–13699. https://doi.org/10.1109/ACCESS.2025.3531662

Araf, I., Idri, A., & Chairi, I. (2024). Cost-sensitive learning for imbalanced medical data: A review. *Artificial Intelligence Review*, *57*(4), 80. https://doi.org/10.1007/s10462-023-10652-8

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220–239. https://doi.org/10.1016/j.eswa.2016.12.035

Jeong, D.-H., Kim, S.-E., Choi, W.-H., & Ahn, S.-H. (2022). A Comparative Study on the Influence of Undersampling and Oversampling Techniques for the Classification of Physical Activities Using an Imbalanced Accelerometer Dataset. *Healthcare*, *10*(7), 1255. https://doi.org/10.3390/healthcare10071255

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*.

Liao, H., Zhang, X., Zhao, C., Chen, Y., Zeng, X., & Li, H. (2022). LightGBM: An efficient and accurate method for predicting pregnancy diseases. *Journal of Obstetrics and Gynaecology*, *42*(4), 620–629. https://doi.org/10.1080/01443615.2021.1945006

Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. *Computers & Security*, *106*, 102289. https://doi.org/10.1016/j.cose.2021.102289

Mienye, I. D., & Sun, Y. (2021). Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Informatics in Medicine Unlocked*, *25*, 100690. https://doi.org/10.1016/j.imu.2021.100690

Sadig, H. E., Kamal, M., Rehman, M. U., Habadi, M. I., Alnagar, D. K., Yusuf, M., Musa Mohammed, M. O., Alqasem, O. A., & Meraou, M. A. (2025). Advanced time complexity analysis for real-time COVID-19 prediction in Saudi Arabia using LightGBM and XGBoost. *Journal of Radiation Research and Applied Sciences*, *18*(2), 101364. https://doi.org/10.1016/j.jrras.2025.101364

Spelmen, V. S., & Porkodi, R. (2018). A Review on Handling Imbalanced Data. *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 1–11. https://doi.org/10.1109/icctct.2018.8551020

Telikani, A., Gandomi, A. H., Choo, K.-K. R., & Shen, J. (2022). A Cost-Sensitive Deep Learning-Based Approach for Network Traffic Classification. *IEEE Transactions on Network and Service Management*, *19*(1), 661–670. https://doi.org/10.1109/TNSM.2021.3112283

Wang, J., Jiang, X., Meng, Q., Saada, M., & Cai, H. (2022). Walking motion real-time detection method based on walking stick, IoT, COPOD and improved LightGBM. *Applied Intelligence*, *52*(14), 16398–16416. https://doi.org/10.1007/s10489-022-03264-2

Zhang, D., & Gong, Y. (2020). The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure. *IEEE Access*, *8*, 220990–221003. https://doi.org/10.1109/ACCESS.2020.3042848

Zhao, C., Yan, Z., Sun, X., & Wu, M. (2024). Enhancing aspect category detection in imbalanced online reviews: An integrated approach using Select-SMOTE and LightGBM. *International Journal of Intelligent Networks*, *5*, 364–372. https://doi.org/10.1016/j.ijin.2024.10.002